# Neural Reflectance Fields for Appearance Acquisition

SAI BI*, UC San Diego
ZEXIANG XU*, UC San Diego, Adobe Research
PRATUL SRINIVASAN and BEN MILDENHALL, UC Berkeley
KALYAN SUNKAVALLI, MILOŠ HAŠAN, and YANNICK HOLD-GEOFFROY, Adobe Research
DAVID KRIEGMAN and RAVI RAMAMOORTHI, UC San Diego

Fig. 1. Renderings from our novel neural reflectance field representation that is reconstructed from images of these scenes captured with a simple collocated camera-light setup. Our neural reflectance fields can model complex scene geometry and reflectance and render images from new viewpoints (odd images) and non-collocated lighting (even images) that were never captured.

We present *neural reflectance fields*, a novel deep scene representation that encodes volume density, normal and reflectance properties at any 3D point in a scene using a fully-connected neural network. We combine this representation with a physically-based differentiable ray marching framework that can render images from a neural reflectance field under any viewpoint and light. We demonstrate that neural reflectance fields can be estimated from images captured with a simple collocated camera-light setup, and accurately model the appearance of real-world scenes with complex geometry and reflectance. Once estimated, they can be used to render photo-realistic images under novel viewpoint and (non-collocated) lighting conditions and accurately reproduce challenging effects like specularities, shadows and occlusions. This allows us to perform high-quality view synthesis and relighting that is significantly better than previous methods. We also demonstrate that we can compose the estimated neural reflectance field of a real scene with traditional scene models and render them using standard Monte Carlo rendering engines. Our work thus enables a complete pipeline from high-quality and practical appearance acquisition to 3D scene composition and rendering.

Additional Key Words and Phrases: View synthesis, relighting, appearance acquisition, neural rendering.

*Both authors contributed equally to this research.

Authors' addresses: Sai Bi, UC San Diego, bisai@cs.ucsd.edu; Zexiang Xu, UC San Diego, Adobe Research, zexu@adobe.com; Pratul Srinivasan, pratul@berkeley.edu; Ben Mildenhall, bmild@cs.berkeley.edu, UC Berkeley; Kalyan Sunkavalli, sunkaval@adobe.com; Miloš Hašan, mihasan@adobe.com; Yannick Hold-Geoffroy, holdgeof@adobe.com, Adobe Research; David Kriegman, kriegman@cs.ucsd.edu; Ravi Ramamoorthi, ravir@cs.ucsd.edu, UC San Diego.


XXXX-XXXX/2020/8-ART
https://doi.org/nnnnnnn.nnnnnn

**ACM Reference Format:**
Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020. Neural Reflectance Fields for Appearance Acquisition. 1, 1 (August 2020), 11 pages. https://doi.org/nnnnnnn.nnnnnn

## 1 INTRODUCTION

Modeling a real scene from captured images and reproducing its appearance under novel conditions is a central problem in computer graphics and vision. This has traditionally been accomplished by using 3D reconstruction and inverse rendering methods to recover scene geometry and reflectance [Nam et al. 2018; Zhou et al. 2013]. However, this is an extremely challenging task, and even state-of-the-art methods generate inaccurate reconstructions that produce images with significant artifacts when rendered.

More recently, many approaches have been proposed that circumvent the problem of explicit reconstruction, and instead estimate a "neural" scene representations that can be combined with an appropriate differentiable rendering method to generate novel images (see [Tewari et al. 2020] for a recent survey). One line of work in this space combines neural scene representations with classical ray marching—a volume rendering approach that is naturally differentiable—to achieve realistic rendering without requiring any pre-acquired 3D geometry [Lombardi et al. 2019; Mildenhall et al. 2020; Sitzmann et al. 2019b]. However, these methods are mostly designed for view synthesis and do not model scene appearance as a function of reflectance or lighting. As a result, they do not allow for tasks such as relighting or scene editing. While ray marching can be used with discrete volumes with explicit per-voxel BRDFs [Bi et al. 2020a] to enable both relighting and view synthesis, an explicit discretized volume representation is highly restricted by its fixed resolution, and cannot reproduce high-frequency appearance details like fine textures and sharp boundaries.

In this work, we propose a novel scene representation that we refer to as *Neural Reflectance Fields*. Unlike previous work that models scene color [Lombardi et al. 2019] or radiance [Mildenhall et al. 2020], neural reflectance fields account for both scene geometry and *reflectance*. This allows us to combine neural reflectance fields with a ray-marching framework (see Fig. 2) to render images under arbitrary view and lighting. Moreover, the whole pipeline is differentiable allowing us to pose the problem of appearance acquisition as one of *optimizing* for a neural reflectance field that, when rendered, will match the captured scene images. Based on this, we capture multiple images around the scene with a cellphone camera and its built-in flash, similar to the acquisition in recent work on material acquisition [Deschaintre et al. 2018; Li et al. 2018a], relighting [Xu et al. 2018] and inverse rendering [Nam et al. 2018]. This practical setup yields unstructured multi-view images under collocated point illumination and captures complex high-frequency scene appearance. As we illustrate in Fig. 1, neural reflectance fields can be reconstructed from such "simple" inputs and allow for the photo-realistic rendering of complex real scenes under novel viewpoints and lighting conditions (that are arbitrarily different from the captured collocated lighting).

Neural Reflectance Fields are a continuous function neural representation that implicitly models both scene geometry and reflectance. We represent them using a deep multi-layer perceptron (MLP) that can regress reflectance properties, normals, and volume density at a given 3D scene point $(x, y, z)$. This representation can be combined with a differentiable ray marching framework—based on classical physically based volume rendering [Kniss et al. 2003; Max 1995; Novák et al. 2018]. In particular, we march rays from the viewpoint through each pixel, and along each marching ray sample points where we compute shading using the regressed normal and reflectance properties at sampled shading points. This shading is modulated with transmittance (computed from regressed volume density), and accumulated along the ray to compute radiance.

We utilize the transmittance not only along the camera ray but also along the light ray to model light effects like shadows for complex real scenes (see Fig. 1). Naively computing the light transmittance requires marching a rays from all the points sampled along the camera ray to the light, making it intractable both for reconstruction and rendering. Instead, we note that the collocated nature of our input data simplifies this for us because it only requires us to evaluate transmittance along the identical camera-light ray, thus allowing us to efficiently fit neural reflectance fields to image data. To further speed up re-rendering under arbitrary light and view positions, we pre-compute a light transmittance volume at adaptively sampled points, enabling efficient shadow rendering.

Our entire rendering pipeline is general and can support any network that can map a 3D point to rendering parameters and any differentiable reflectance model. For example, in Fig. 1, 5, 7, we demonstrate that we can accurately model the appearance of a diverse set of real scenes, including scenes with intricate geometry, highly specular reflectance, furry objects, and human portraits. These results are significantly better than the state-of-the-art mesh-based reconstruction method [Nam et al. 2018] and discrete volume-based representations [Bi et al. 2020a] (see Fig. 4).

Moreover, because our representation is designed to work with a physically-based volume renderer, it can in fact be naturally incorporated into modern rendering engines, like Mitsuba [Jakob 2010]. This allows us to compose neural reflectance fields with traditional 3D models (with explicit meshes and BRDFs) and capture light transport interactions between these disparate scene elements (see Fig. 9). This is something that has not been demonstrated by previous neural reconstruction methods, and in our opinion, represents an important step towards building neural capture and rendering approaches that can be incorporated into traditional 3D design workflows.

In summary, our main contributions are:

- A novel neural reflectance field representation that models both scene geometry and reflectance,
- A physically-based ray marching scheme that can render neural reflectance fields under any view and lighting,
- A method to reconstruct neural reflectance fields from unstructured flash images, and
- Applications of this representation to tasks like view synthesis, relighting, and scene composition.

## 2 RELATED WORK

*Neural scene representations.* Previous work has applied deep neural networks to many 3D tasks with scene geometry modeled by various representations, such as volumes [Ji et al. 2017; Richter and Roth 2018], point clouds [Qi et al. 2017], implicit functions [Mescheder et al. 2018; Sitzmann et al. 2019b], etc. Reflectance modeling has also been explored with neural networks [Kuznetsov et al. 2019; Rainer et al. 2019; Vicini et al. 2019]. We present the novel neural reflectance field that models both geometry and reflectance in a real scene.

Thies et al. [2019] apply neural textures for realistic image synthesis, but require a pre-acquired mesh as input. Many previous works aim to do view synthesis without any known geometry. Multiplane images have been used in small-baseline view synthesis [Srinivasan et al. 2019; Zhou et al. 2018]; however, such a view-dependent representation only supports limited viewing range and requires special fusion techniques to extend the range [Mildenhall et al. 2019]. Recent works leverage view-independent volumes, which are able to handle complex view-dependent effects [Lombardi et al. 2019; Sitzmann et al. 2019a]. Our neural reflectance field models complete scene appearance; in addition to view synthesis, ours can also be used for other applications such as relighting.

Recently, ray marching has been used to train many neural scene representations for view synthesis without any ground-truth 3D representations [Lombardi et al. 2019; Mildenhall et al. 2020; Sitzmann et al. 2019b]. Lombardi et al. [2019] apply ray marching in a discrete volume with a warping field for view synthesis. To make it generalizable to relighting, Bi et al. [2020a] reconstruct discrete reflectance volumes with explicit per-voxel BRDFs; however, the fixed resolution of the discrete volume limits the appearance details in the rendering. In contrast, we leverage a *continuous* functional neural representation and achieve much better results (see Fig. 4). Mildenhal et al. [2020] present a neural radiance field, which also represents a scene as a continuous function with a MLP. However,

their representation only supports view synthesis by directly rendering radiance from a new viewpoint under fixed illumination. We leverage a novel reflectance-aware ray marching framework and learn to regress multiple decomposed shading components, which enables relighting and many other images synthesis applications.

*Geometry and reflectance capture.* Classically, modeling and rendering a real scene requires full reconstruction of its geometry and reflectance. From captured images, scene geometry is usually reconstructed by structure-from-motion and multi-view stereo (MVS) [Esteban and Schmitt 2004; Furukawa and Ponce 2009; Kutulakos and Seitz 2000; Schönberger and Frahm 2016; Schönberger et al. 2016], which have recently been extended using deep learning techniques [Chen et al. 2019; Cheng et al. 2019; Yao et al. 2018, 2019].

Reflectance acquisition traditionally requires sophisticated devices to sample the light-view space [Foo 1997; Kang et al. 2018, 2019; Matusik et al. 2003; Nielsen et al. 2015; Xu et al. 2016]. Recently, many works use a practical device – a modern cellphone that has a camera and a built-in flash light – and capture flash images to acquire spatially varying BRDFs [Aittala et al. 2016, 2015; Hui et al. 2017; Nam et al. 2018]. While such a device only acquires reflectance samples under collocated light and view, with enough samples, it is still sufficient to reconstruct many standard analytic reflectance models that are governed by the half-angle vector [Hui et al. 2017; Nam et al. 2018]. More recently, deep learning methods have made BRDF acquisition with a single flash image possible [Deschaintre et al. 2018; Li et al. 2018a,b]. Bi et al. [2020b] extend the single-view case to a structured multi-view configuration, and reconstruct meshes with per-vertex BRDFs from only six images.

We aim to model geometry and appearance of complex real scenes from multi-view unstructured flash images. From such inputs, Nam et al. [2018] leverage an initial mesh from MVS and reconstruct per-vertex BRDFs via traditional optimization. However, it is very difficult for traditional mesh-based methods to recover challenging thin structures and sharp specularities of complex real scenes. In this work, we address these issues by proposing a novel neural reflectance field to implicitly model the scene's geometry and reflectance, bypassing explicit mesh reconstruction. Our approach achieves photo-realistic rendering results with high-frequency appearance details that are significantly better than previous works.

*Relighting and view synthesis.* Scene acquisition and rendering can be also achieved using image-based techniques without explicit reconstruction [Debevec et al. 2000; Levoy and Hanrahan 1996]. Recently, many learning based view synthesis methods have been presented [Hedman et al. 2018; Mildenhall et al. 2020; Srinivasan et al. 2017; Xu et al. 2019; Zhou et al. 2018]. We extend the ray marching in the view synthesis works to a more general reflectance-aware ray marching framework, which can also be used to do relighting. Learning-based relighting methods have also been presented [Ren et al. 2015; Xu et al. 2018], which are able to reproduce challenging appearance effects. Many techniques regress images under novel lighting from sparse inputs without any explicit geometry reasoning [Sun et al. 2019; Xu et al. 2018; Zhou et al. 2019], but are unable to recover accurate hard shadows. Philip et al. [2019] require a mesh from MVS for shadowing computation. Our network learns to regress volume density to model detailed scene geometry. Our ray marching considers light transmittance in ray integration, which recovers challenging hard shadows.

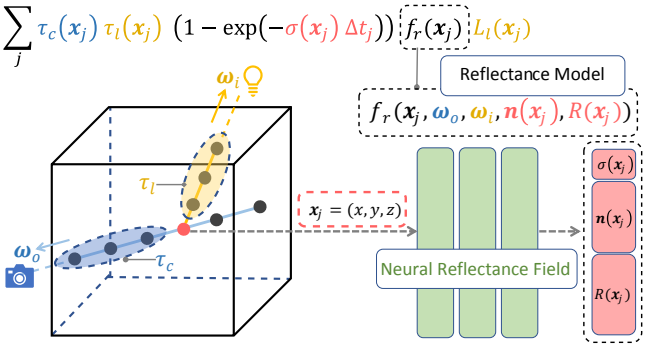


Fig. 2. Overview of the neural reflectance field and ray marching. We march a ray (blue) through each pixel from the camera and sample a sequence of shading points on the ray. The ray radiance under a point light source is computed from the volume density $\sigma$, normal $\boldsymbol{n}$ and reflectance properties $\boldsymbol{R}$ via a differentiable ray marching equation (Eqn. 7) shown on top. At each point $\boldsymbol{x}_j$, shading is computed from the normal $\boldsymbol{n}(\boldsymbol{x}_j)$, reflectance properties $\boldsymbol{R}(\boldsymbol{x}_j)$ and corresponding light and view directions ($\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$) using a specified reflectance model $f_r$. Volume density $\sigma(\boldsymbol{x}_j)$ acts like opacity ($\alpha = 1-\exp(-\sigma\Delta t)$) to attenuate the shading; it is also used to compute the view transmittance $\tau_{\boldsymbol{c}}$ (Eqn. 8) along the camera ray (blue ellipsoid) and the light transmittance $\tau_{\boldsymbol{l}}$ (Eqn. 9) along an additional ray toward the light (yellow ellipsoid). We propose to use a novel neural reflectance field, represented by an MLP, to regress the required rendering properties ($\sigma$, $\boldsymbol{n}$, $\boldsymbol{R}$) from the 3D location $\boldsymbol{x}_j = (x, y, z)$ for ray marching.

Note that, previous image-based techniques often send viewing [Lombardi et al. 2019; Mildenhall et al. 2020] or lighting [Sun et al. 2019; Xu et al. 2018] information as additional inputs to the network, and compute challenging view- or light- dependent shading effects through the network processing. In contrast, we leverage classical reflectance models to regularize the learning process; our neural reflectance field is independent of the viewing and lighting directions, and we use the regressed reflectance and normal to compute shading under any lighting and viewpoint. Our approach can properly model scene appearance and reproduce challenging view-dependent and light-dependent shading effects.

## 3 REFLECTANCE-AWARE RAY MARCHING

While differentiable ray marching has been used in recent works [Lombardi et al. 2019; Mildenhall et al. 2020], these methods focus on view synthesis and only consider view-dependent effects. We utilize classical reflectance models in a more general ray marching framework (see Fig. 2) that also models lighting and enables relighting and other re-rendering applications. Our reflectance-aware rendering framework is differentiable and can be easily combined with deep learning to learn scene appearance. In this section, we first discuss the underlying rendering equation that governs our volume rendering (Sec. 3.1), and then introduce our ray marching framework that numerically computes the equation in a differentiable way (Sec. 3.2).

### 3.1 Rendering equation

In general, for non-emissive and non-absorptive volumes, physically-based volume rendering is governed by the volume rendering equation [Novák et al. 2018] that computes the radiance $L(\boldsymbol{c}, \boldsymbol{\omega_o})$ at point $\boldsymbol{c}$ in direction $\boldsymbol{\omega_o}$:

$$L(\boldsymbol{c}, \boldsymbol{\omega_o}) = \int_0^\infty \tau_{\boldsymbol{c}}(\boldsymbol{x})\sigma(\boldsymbol{x})L_s(\boldsymbol{x}, \boldsymbol{\omega_o})dt, \tag{1}$$

$$\text{where } \tau_{\boldsymbol{c}}(\boldsymbol{x}) = e^{-\int_0^t \sigma(\boldsymbol{c}-u\boldsymbol{\omega_o})du}. \tag{2}$$

Here, $t$ represents the 1D location on a ray traced in the volume, $\boldsymbol{x} = \boldsymbol{c} - t\boldsymbol{\omega_o}$ represents the 3D point at $t$, the point $\boldsymbol{c}$ typically represents the camera location, and $L_s(\boldsymbol{x}, \boldsymbol{\omega_o})$ represents the scattered light at $\boldsymbol{x}$ along $\boldsymbol{\omega_o}$. $\sigma$ is the extinction coefficient that indicates the probability density of medium particles; we refer to $\sigma$ as volume density in this paper. $\tau_{\boldsymbol{c}}(\boldsymbol{x})$ represents the transmittance factor which determines the loss of light along the ray from $\boldsymbol{c}$ to $\boldsymbol{x}$.

Eqn. 1 computes the radiance that arrives at $\boldsymbol{c}$ by integrating the modulated in-scattered light $L_s$ along the ray,

$$L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) = \int_{\mathcal{S}} f_p(\boldsymbol{x}, \boldsymbol{\omega_o}, \boldsymbol{\omega_i})L_i(\boldsymbol{x}, \boldsymbol{\omega_i})d\boldsymbol{\omega_i}, \tag{3}$$

where $\mathcal{S}$ is a unit sphere, $f_p$ is a phase function that governs light scattering, and $L_i(\boldsymbol{x}, \boldsymbol{\omega_i})$ is the incident radiance arriving at $\boldsymbol{x}$ from direction $\boldsymbol{\omega_i}$.

Note that previous work [Mildenhall et al. 2020] directly encodes $L_s$ without considering any form of Eqn. 3; this assumes fixed illumination and only works for view synthesis. In contrast, we consider single-bounce direct illumination under a single point light source to approximate $L_s$. Inspired by [Max 1995], we compute $L_s$ with an explicit reflectance term that assumes the role of a phase function:

$$L_s(\boldsymbol{x}, \boldsymbol{\omega_o}) = f_r(\boldsymbol{x}, \boldsymbol{\omega_o}, \boldsymbol{\omega_i}, \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{R}(\boldsymbol{x}))L_i(\boldsymbol{x}, \boldsymbol{\omega_i}), \tag{4}$$

where $f_r$ represents a differentiable reflectance model with parameters $\boldsymbol{R}$, $\boldsymbol{n}$ is the local surface shading normal, and $L_i$ represents the incident radiance as in Eqn. 3. When only considering direct illumination from a point light source, $L_i$ is determined by the intensity of the light source and the loss of light due to extinction through the volume:

$$L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) = \tau_{\boldsymbol{l}}(\boldsymbol{x})L_{\boldsymbol{l}}(\boldsymbol{x}), \tag{5}$$

where $\tau_{\boldsymbol{l}}$ is the transmittance from the light to the shading point, and $L_{\boldsymbol{l}}$ represents the light intensity with the consideration of distance attenuation. Here, $\boldsymbol{l}$ denotes the position of the point light source, and thus $\boldsymbol{\omega_i}$ corresponds to the direction of the vector $\boldsymbol{l} - \boldsymbol{x}$.

Therefore, by combining Eqn. 1, 4, 5, our volume rendering is governed by a *reflectance-aware* rendering equation [Kniss et al. 2003; Max 1995]:

$$L(\boldsymbol{c}, \boldsymbol{\omega_o}) = \int \tau_{\boldsymbol{c}}(\boldsymbol{x})\tau_{\boldsymbol{l}}(\boldsymbol{x})\sigma(t)f_r(\boldsymbol{x}, \boldsymbol{\omega_o}, \boldsymbol{\omega_i}, \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{R}(\boldsymbol{x}))L_{\boldsymbol{l}}(\boldsymbol{x})dt. \tag{6}$$

Our equation considers complete one-bounce camera-volume-light paths in the light transport. Unlike previous work that only considers the view transmittance or opacity between the shading point and the camera [Lombardi et al. 2019; Mildenhall et al. 2020], we also explicitly express the light transmittance ($\tau_{\boldsymbol{l}}$) from the point to the light, which allows us to render realistic shadows under different point light sources. Essentially, instead of modeling scene *radiance* $L_s$ as is in previous view synthesis work [Mildenhall et al. 2020], we decouple the multiple factors ($\tau_{\boldsymbol{l}}$, $f_r$, $L_{\boldsymbol{l}}$) that are embedded in $L_s$, and explicitly model the scene *reflectance* parameters in $f_r$, thus allowing for reflectance-aware rendering for both view synthesis and relighting with realistic shading and shadowing effects.

### 3.2 Ray marching

We use ray marching to numerically estimate the continuous integral in Eqn. 6 similarly to prior work on volume rendering [Kniss et al. 2003; Max 1995]; this is illustrated in Fig. 2. Specifically, we march rays from the camera center through each pixel on the image plane and sample a sequence of $N$ shading points $\boldsymbol{x}_j$ on each ray. The rendering equation can be estimated by:

$$L(\boldsymbol{c}, \boldsymbol{\omega_o}) = \sum_{j=0}^{N} \tau_{\boldsymbol{c}}(\boldsymbol{x}_j)\tau_{\boldsymbol{l}}(\boldsymbol{x}_j)(1 - \exp(-\sigma(\boldsymbol{x}_j)\Delta t_j))f_r(\boldsymbol{x}_j)L_{\boldsymbol{l}}(\boldsymbol{x}_j), \tag{7}$$

where $\Delta t_j$ represents the ray step size at point $\boldsymbol{x}_j$. Here, we omit the other parameters ($\boldsymbol{\omega_o}$, $\boldsymbol{\omega_i}$, $\boldsymbol{n}$, $\boldsymbol{R}$) in $f_r$ for brevity. Here, $\tau_{\boldsymbol{c}}(\boldsymbol{x}_j)$ is also an integral (Eqn. 2) and can be numerically evaluated by:

$$\tau_{\boldsymbol{c}}(\boldsymbol{x}_j) = \exp\left(-\sum_{k=0}^{j} \sigma(\boldsymbol{x}_k)\Delta t_k\right). \tag{8}$$

The transmittance $\tau_{\boldsymbol{l}}(\boldsymbol{x}_j)$ can be similarly evaluated, but it requires sampling another sequence of points $\boldsymbol{x}'_p$ on an additional ray marched from the light source to the shading point $\boldsymbol{x}_j$:

$$\tau_{\boldsymbol{l}}(\boldsymbol{x}_j) = \exp\left(-\sum_{p} \sigma(\boldsymbol{x}'_p)\Delta t'_p\right). \tag{9}$$

Naively computing Eqn. 9 for Eqn. 7 would require marching a large number of light rays for all shading points on all camera rays. Instead, we leverage a collocated light source and camera setup (where the camera and light rays are the same) to avoid this during training; this is described in Sec. 4.2. At inference time we precompute an adaptive transmittance volume to efficiently approximate Eqn. 9 under any point light; this is described in Sec. 4.3.

Equations 7, 8, 9 express our reflectance-aware ray marching framework. Given a camera $\boldsymbol{c}$ and a point light $\boldsymbol{l}$, the framework computes the radiance of any marching ray through a scene from the volume density $\sigma$, normal $\boldsymbol{n}$, and reflectance properties $\boldsymbol{R}$ of the points in the scene. [Bi et al. 2020a] utilizes a rendering equation similar to ours (Eqn. 6), but they leverage classical opacity accumulation – where the opacity is given by $\alpha = 1 - \exp(-\sigma\Delta t)$ – to numerically evaluate the integral, which only supports a fixed step size for ray marching. In contrast, we utilize volume density $\sigma$ for numerical estimation, which is more general and allows the step sizes to vary across the shading points. This enables applying better adaptive sampling strategies to distribute the shading points along both camera and light rays (See 4.2 and Sec. 4.3). In addition, since volume density is standard in Monte Carlo based volume rendering, the model learned from our ray marching framework can be also used in standard rendering engines for general graphics applications (See. Fig. 9).

Our ray marching framework supports any differentiable reflectance model $f_r$, which makes the full rendering process trivially

differentiable. In this work, we demonstrate most results using a classical analytic BRDF [Karis and Games 2013] for $f_r$, which models the reflectance of opaque surfaces with a diffuse albedo and a specular roughness. We also show results with hair/fur reflectance models [Kajiya and Kay 1989] that model the appearance of furry objects, demonstrating the generality of this formulation. Our reflectance-aware ray marching framework can potentially be combined with any module that is able to provide the rendering properties ($\sigma$, $\boldsymbol{n}$, $\boldsymbol{R}$) of an arbitrary point in the scene. In this work, we use a neural network to regress the necessary rendering properties.

## 4 NEURAL REFLECTANCE FIELDS

We now present our neural reflectance field representation that uses deep fully connected networks to model scene geometry and reflectance (Sec. 4.1). As shown in Fig. 2, this network can be used in conjunction with the reflectance-aware ray marching scheme described previously. We show how it can effectively trained from cellphone flash images (Sec. 4.2). We also present an adaptive transmittance volume for light transmittance precomputation, enabling efficient rendering under any novel light and view positions with realistic shadows (Sec. 4.3).

### 4.1 Network

Given a reflectance model $f_r$ with $m$ parameters, a neural reflectance field outputs a $(4 + m)$-dimensional vector—comprising volume density $\sigma$ (1-D), normal $\boldsymbol{n}$ (3-D) and reflectance properties $\boldsymbol{R}$ ($m$-D)—at any 3-D position $\boldsymbol{x} = (x, y, z)$ in a scene. In practice, we use a microfacet BRDF model [Walter et al. 2007] where $\boldsymbol{R}$ comprises diffuse albedo and specular roughness, though we also demonstrate an extension using a fur reflectance model [Kajiya and Kay 1989]. We parameterize neural reflectance fields using an MLP with 14 fully connected layers and ReLU activation layers. Please refer to the supplementary material for the detailed architecture of our network.

Inspired by [Mildenhall et al. 2020; Rahaman et al. 2018; Vaswani et al. 2017], we use a frequency-based positional encoding of a given 3D location $\boldsymbol{x} = (x, y, z)$. In particular, given each dimension of the 3D $(x, y, z)$ point, we map the scalar value $v$ to

$$\gamma(v) = (\sin(2^0\pi v), \cos(2^0\pi v), \ldots, \sin(2^{W-1}\pi v), \cos(2^{W-1}\pi v)), \tag{10}$$

where $W$ represents the highest frequency level ($W = 10$ in our experiments). These are input to the MLP to regress the scene properties at the encoding $\gamma(x)$, $\gamma(y)$ and $\gamma(z)$.

Unlike [Mildenhall et al. 2020], which also use a positional encoding of the viewing direction, we only use the 3D location as input, inferring view- (and light-) independent scene appearance properties. This is possible because we separate out these factors and compute shading, viewing, and lighting information in our ray marching framework (Eqn. 6, 7); this allows neural reflectance fields to be directly plugged into it for high-quality rendering.

### 4.2 Learning neural reflectance fields from flash images

We now describe how we can use neural reflectance fields to reconstruct the appearance of a real-world scene from images. Each neural reflectance field is fit to a specific scene via a training process. Since our whole rendering process (the representation and the ray marching) is differentiable, we train the neural reflectance field network to minimize the error between rendered images and captured images of the scene.

*Collocated light and view.* In particular, we capture flash images with collocated light and view to train our networks. Such images can be easily captured by a cellphone with a camera and a flash. The collocated setting leads to $\boldsymbol{c} = \boldsymbol{l}$ in our training. One key benefit of using collocated light and view is that the view transmittance $\tau_{\boldsymbol{c}}$ and the light transmittance $\tau_{\boldsymbol{l}}$ become equal in Eqn. 7. This avoids marching an additional ray towards the light at every shading point, which would make training intractable. This capture setup thus has the advantage of making both *acquisition* and *training* practical. However, this also means that our input images represent an extremely sparse sampling of scene appearance across the view-light space. In fact, we have no samples of the scene for lighting from any non-zero angle with respect to the camera. In spite of this, we show that we can reconstruct high-quality scene appearance and render images under arbitrary view and even non-collocated lighting.

*Adaptive sampling for camera rays.* To optimize the point sampling in ray marching, for each scene, we train two networks—a coarse and a fine neural reflectance field—and render using a coarse-to-fine adaptive sampling procedure. Inspired by [Mildenhall et al. 2020], we first sample a sparse set of points on each marching ray with stratified sampling to compute a distribution function using the coarse network, then sample a dense set of points from the distribution function to compute the final radiance value using the fine network.

In particular, we divide each full ray segment into $N_1$ bins and randomly sample a point from each bin to get stratified samples. From these points, we can compute the radiance from the coarse-level network for the ray using Eqn. 7. As a side product, we can also produce corresponding per-point contribution weights

$$a(\boldsymbol{x}_j) = \tau_{\boldsymbol{c}}(\boldsymbol{x}_j)(1 - \exp(-\sigma(\boldsymbol{x}_j)\Delta t_j)). \tag{11}$$

The weight $a(\boldsymbol{x})$ essentially describes how visible the point at $\boldsymbol{x}$ is to the camera. We construct a piece-wise constant probability distribution by normalizing the per-point weights $a(\boldsymbol{x}_j)$ and then sample $N_2$ points from this distribution, which adaptively selects new samples according to the visibility information gathered from the coarse neural reflectance field. We then use all $N = N_1 + N_2$ sampled points to compute the final radiance with Eqn. 7 using the rendering parameters from the fine reflectance field. This coarse-to-fine adaptive sampling effectively distributes more sampled points in the regions that contribute most to the rendering integral, allowing for accurate shading computation with high-frequency details.

### 4.3 Efficient rendering under novel light and view

While our neural reflectance field is learned from flash images with collocated light and view, the learned representation can be directly used to render the scene with single-scattering effects using any light and view positions with Eqn. 7. However, accurately computing $\tau_{\boldsymbol{l}}$ at inference time under novel non-collocated light and view (unlike training) is extremely computationally expensive. Therefore,

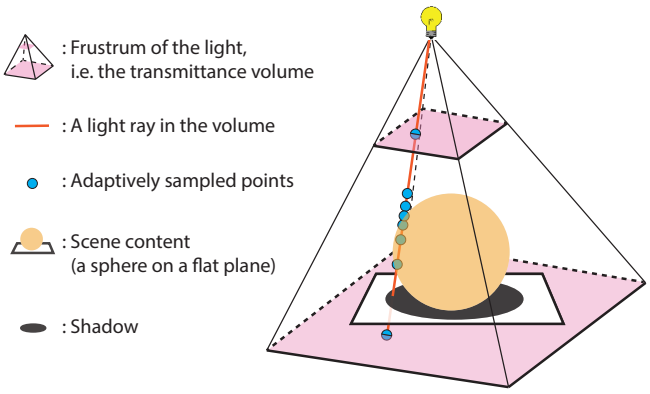


Fig. 3. Adaptive transmittance volume. We pre-compute a light transmittance volume within the frustrum of a virtual view at the point light source. We leverage a coarse-to-fine strategy to adaptively distribute the sampled points (blue) around the visible scene structures along each light ray (red), enabling an efficient transmittance representation.

we propose to pre-compute an adaptive transmittance volume to effectively approximate $\tau_{\boldsymbol{l}}$.

*Adaptive sampling for light transmittance volume.* Inspired by the classical shadow map technique [Stamminger and Drettakis 2002; Williams 1978] in rasterization, we use our learned neural reflectance field to compute a transmittance volume similar to [Lokovic and Veach 2000] for fast light transmittance computation. Specifically, we place a virtual image plane in front of the point light source towards the scene and march a ray through each pixel, analogously to what is classically done for ray marching from the camera. Similar to the *adaptive sampling for camera rays* described in Sec. 4.2, we use the two trained networks (a coarse and a fine network) to perform adaptive sampling. We first utilize the coarse representation to compute a visibility-aware distribution function using sparse points sampled from stratified bins; we then sample dense points from the distribution. We combine the samples from both passes and compute their light transmittance, resulting in a transmittance volume that adapts to the visibility information inferred from the coarse network. This adaptive transmittance volume is illustrated in Fig. 3.

*Final rendering.* We do ray marching from the viewpoint to render an image under any point light source from any viewpoint using the learned network and the pre-computed adaptive transmittance volume. At any given shading point, we locate the nearest sampled points and then linearly interpolate the transmittance volume to get the required light transmittance, similar to [Lokovic and Veach 2000]. This allows for realistic shadowing effects to be well recovered in our results when doing relighting.

We also apply coarse-to-fine sampling on the camera rays for the rendering at inference, as we described in Sec. 4.2 at training. Basically, at inference, we apply coarse-to-fine adaptive sampling in ray marching from both the light and the camera, which achieves efficient light transmittance computation and effective final image synthesis.

As noted before, during training, our network only sees images that are captured under collocated light and view and do not have any shadows. Yet, our method is able to learn a volume density that meaningfully expresses scene geometry. This allows us to synthesize high-quality relighting and view synthesis results with realistic shadows, specularities and other appearance effects under novel, non-collocated light and view, as illustrated in Figs. 1, 4, 5.

## 5 IMPLEMENTATION

*Data acquisition.* As discussed in Sec. 4.2, we reconstruct neural reflectance fields from images captured under collocated view and lighting. Such data can be practically acquired by shooting a video using a handheld cellphone with flash. We show acquisition and rendering results of one human portrait using this handheld setup in Fig. 7; in this case, we selected 150 frames from the video as input. To facilitate the data acquisition, for other results, we use a robotic arm holding a cellphone to automatically capture scenes that are composed of different static objects. We capture about 400 images using this automatic setup. We use a Samsung Galaxy Note 8 to capture all our real scenes. The camera parameters are calibrated using structure from motion in COLMAP [Schönberger and Frahm 2016]. Our method does not require accurate background masks for the input images to train the network. We simply crop the central regions around the objects in the captured images to avoid training on too many background pixels. Each network is trained in a scene-dependent way, using the input images for that single scene.

*Reflectance model.* Our representation works with any differentiable reflectance model. In practice, we use a microfacet BRDF model that combines a diffuse Lambertian term with a specular term that uses the GGX distribution [Walter et al. 2007]. The parameters of this model include a diffuse albedo and a specular roughness. With this model, the neural reflectance field MLP thus outputs a 8-D vector at every scene point, corresponding to the 3-D diffuse albedo, 1-D specular roughness, 3-D surface normal and 1-D transmittance. We use this BRDF model for every result in the paper, except for Fig. 6 where we capture a furry object. Here we use the classical fur reflectance model [Kajiya and Kay 1989] and replace the surface normal $\boldsymbol{n}$ with a fiber tangent vector.

*Training parameters and loss function.* We implement our neural reflectance field and ray marching in PyTorch. During training, we randomly sample $50 \times 50$ pixel rays as a batch to train our network under collocated light as described in Sec. 4.2. We use Adam optimizer with an initial learning rate of 0.0001. We use $N_1 = 64$ coarse samples and $N_2 = 128$ fine samples to adaptively sample light rays when building the adaptive transmittance volume and camera rays when computing the final radiance.

We supervise the regressed radiance values from both the coarse and the fine network with the ground truth radiance $\tilde{L}$ from the captured images using the $L_2$ loss. Since we consider opaque objects, we also regularize the ray transmittance (from the fine network), forcing it to be close to 0 or 1, which is helpful to get a clean background. Our total loss function is given by:

$$\sum_{q} \|L^q_{\text{coarse}} - \tilde{L}^q\|^2 + \|L^q_{\text{fine}} - \tilde{L}^q\|^2 + \beta[\log(\tau^q_{\boldsymbol{c}}) + \log(1 - \tau^q_{\boldsymbol{c}})], \quad (12)$$

where $q$ denotes a pixel ray and $\beta = 0.0001$ is a hyper-parameter that controls the strength of the regularization term.

*Run time.* We use 4 NVIDIA RTX 2080Ti GPUs to train each reflectance field network for about 2 days. At inference time, the network takes about 30 seconds to render a $512 \times 512$ image using our adaptive transmittance volume.

## 6 RESULTS

We now demonstrate our results in this section. We first evaluate our method by comparing our view synthesis and relighting results with other methods. We then show more results and applications of our method. Please refer to the supplementary video for more video results.

*Comparisons with previous methods.* Most previous learning-based works focus only on the sub-problems of relighting [Ren et al. 2015; Xu et al. 2018] or view synthesis [Lombardi et al. 2019; Mildenhall et al. 2019, 2020; Xu et al. 2019], and capture images with a fixed camera or fixed illumination, respectively. Instead, our input light and view are collocated and vary across all input images, allowing us to build a holistic scene representation that allows for both view synthesis and relighting. We are aware of only a few methods that address this problem and we compare against two of them. The first is a state-of-the-art mesh-based appearance acquisition method [Nam et al. 2018] that reconstructs a 3D mesh and per-vertex BRDFs from collocated flash images. The reconstructed geometry and reflectance can then be used to achieve relighting and view synthesis. We also compare with a learning-based method [Bi et al. 2020a], that predicts a discrete volume with explicit per-voxel reflectance properties. This technique supports relighting and view synthesis via opacity accumulation-based ray marching. In Fig. 4, we show qualitative comparisons of images renderer from the respective reconstructions under novel collocated and non-collocated light-view settings. Results for all methods were generated from the same inputs by their respective authors. Please refer to the supplementary video for video comparisons.

Fig. 4 shows that our method achieves significantly better rendering results than [Nam et al. 2018]. They leverage a classical multi-view stereo (MVS) method to reconstruct an initial mesh, and then recover a refined mesh and per-vertex BRDFs via traditional optimization. However, for challenging real scenes, MVS often fails to recover reasonable initial geometry in regions that with little texture, high specularity, or thin structures. This leads to highly distorted and even missing geometry in their results. In addition, since specular effects typically influence very few pixels, their optimization-based reflectance estimation step is unable to recover them, leading to a mostly diffuse appearance, . In contrast, our neural reflectance field bypasses mesh reconstruction and is able to accurately resolve fine geometric structure with volume densities. This leads to high-quality rendering results with realistic geometric details, high specularities and hard shadows.

Our method also outperforms the previous deep volume rendering method [Bi et al. 2020a]. While that method also avoids the geometric reconstruction issues arising from Nam et al. [2018], it fails to recover high-frequency details in the results, as reflected in many of the insets shown in Fig. 4. This is because they regress a discrete volume with per-voxel BRDFs; the rendering quality is limited by the resolution of the volume, which is strictly constrained by system memory. Instead, by leveraging a continuous functional representation, our network can properly recover high-frequency appearance. Our neural reflectance field is also extremely compact, with weights consuming only 5 MB of memory. In contrast, [Bi et al. 2020a] uses a network that requires 400 MB to predict a volume that consumes several gigabytes of memory during rendering. Our approach is more efficient in terms of memory usage and has more potential to be extended to capture of large-scale real scenes.

*Additional results on diverse real scenes.* We now demonstrate additional view synthesis and relighting results from our method on diverse real scenes in Figs. 5, 6, and 7. Fig. 5 shows results on complex objects. Our method successfully recovers various challenging high-frequency appearance effects, such as detailed geometry, complex textures, specularities, and hard shadows. Note that the detailed thin geometry of the grass in Plane and the complex normal variation on the surfaces in Dragon and Superhero are all well reproduced realistically. Our method can also handle challenging scenes that consist of multiple objects, like Shop. These lead to complex cast shadows between objects, that our method accurately reproduces in spite of never having observed them in the input images. This can be attributed to the ability of our method to infer reliable geometry (in the form of a volume density) from just collocated image samples.

In Fig. 6, we acquire the appearance of a furry object. Here, we plug in the classical fur reflectance model [Kajiya and Kay 1989] into our representation, demonstrating the ability of neural reflectance fields to work with a wide range of reflectance models. While the results here are slightly blurrier than the other scenes (Fig. 5), they still look very realistic with the desired furry appearance. Our method can also be used to capture facial appearance, as shown in Fig. 7. Here, we use a handheld cellphone and simply capture a video (with flash) walking around the person. From this video, we sample 150 images and train a neural reflectance field that allows for re-rendering under varying viewpoint and lighting. Acquiring facial appearance is an extensively studied problem and recent deep learning-based approaches have demonstrated portrait relighting from sparse inputs. However, these either require calibrated illumination [Meka et al. 2019; Xu et al. 2018] or focus on low-frequency illumination [Sun et al. 2019; Zhou et al. 2019]. In contrast, our images are captured with a practical setup, and are of high quality with realistic specularities and hard shadows, in spite of not making any face-specific assumptions in our method.

*Synthetic results.* Since our setup only captures images under collocated view and light, we do not have ground truth captured images to evaluate renderings under non-collocated camera and light. We thus compare using a synthetic scene in Fig. 8, where we can render the ground truth under any lighting and viewpoint. As shown in Fig. 8, our method is able to accurately reproduce the high-frequency textures, specularities, and hard shadows in the rendered images, which are very close to the ground truth.

*Integrating with Monte-Carlo renderers.* While neural rendering approaches have made remarkable progress in the recent past, one

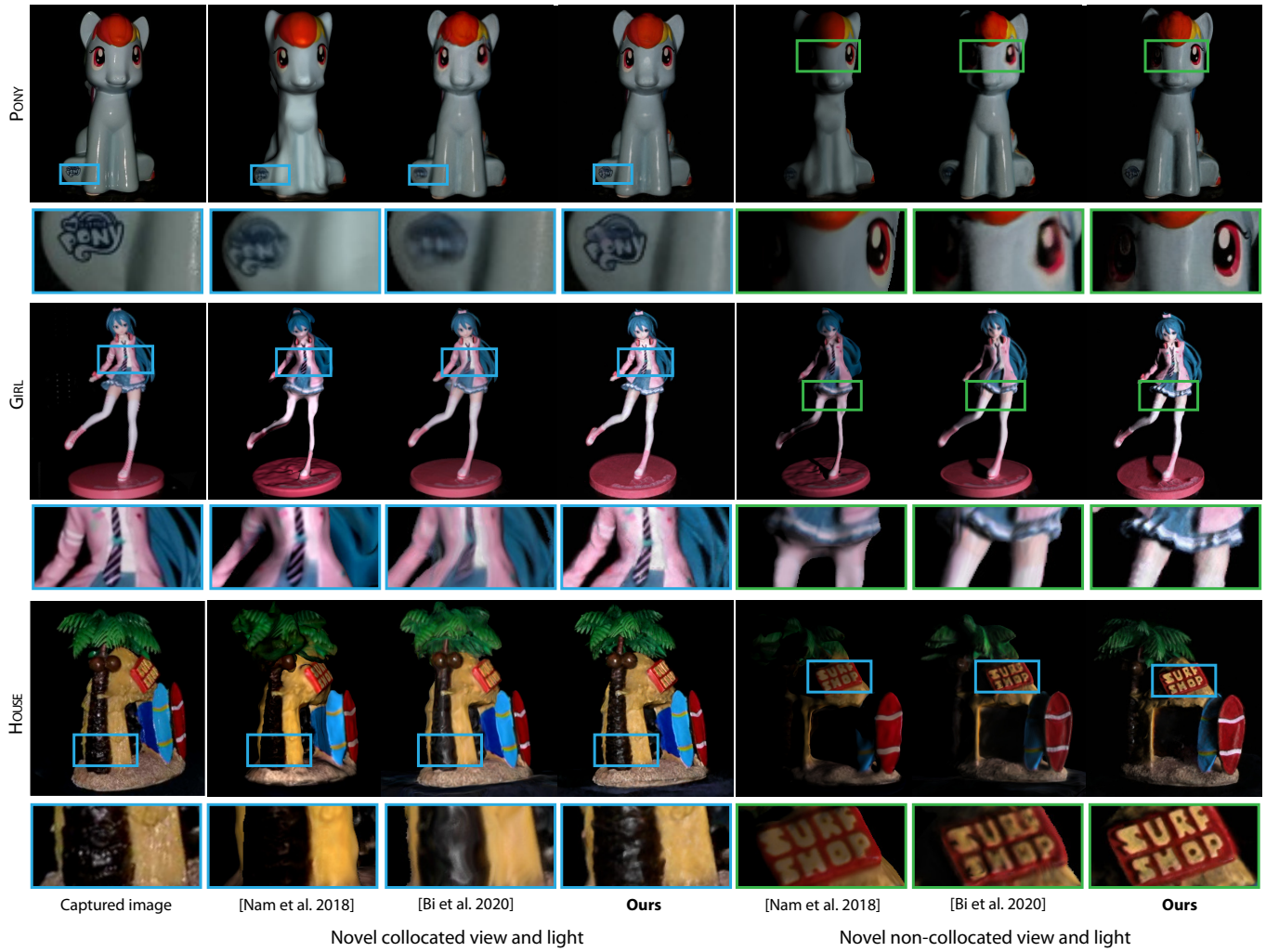


Fig. 4. Comparisons with previous work. We compare our view synthesis and relighting results with a state-of-the-art mesh-based method [Nam et al. 2018] and a previous learning-based method [Bi et al. 2020a] on complex real scenes. We show one captured image (not used for training) on the left. We compare re-renderings under novel collocated camera and light (middle) and novel *non-collocated* camera and light (right). As can be seen here, even a state-of-the-art mesh reconstruction method fails to accurately reconstruct complex real-world scenes. While the learning-based approach improves on this result, it produces blurry results. In contrast, our method produces realistic results with high-frequency textures, specular highlights, and complex shadowing.

challenge with them is that they still require custom components that may not be consistent with standard scene representations and rendering engines. In addition, most current methods focus on the view synthesis task [Lombardi et al. 2019; Mildenhall et al. 2020] and do not model the interaction of lighting with the captured scene. While Bi et al. [2020a] do model lighting, it is based on opacity accumulation and only supports a fixed step size, which is not valid for Monte Carlo rendering. In contrast, our neural reflectance field representation models *all camera-light interactions with the scene*. In addition, it is trained in conjunction with a physically-based ray marching framework. As a result, it can be easily integrated using standard graphics rendering engines, by simply implementing the reflectance function as a special phase function.

In particular, we use Mitsuba [Jakob 2010] to render one of our captured neural reflectance fields under complex environment illumination, and show these results in Fig. 9. We simply compute discrete $512 \times 512 \times 512$ volumes from our reflectance fields and use the volume to do Monte Carlo rendering. While simple, this leads to very realistic rendering results in Fig. 9. Also note that this allows us to compose a scene that is made up of our captured object and traditional 3D models represented by meshes with BRDFs, and *simulate the light transport between these different representations* including complex shadows and inter-reflections. While these results contain fewer details compared to our other results, this is caused by the limited volume resolution and can be addressed by potentially implementing our network in Mitsuba.

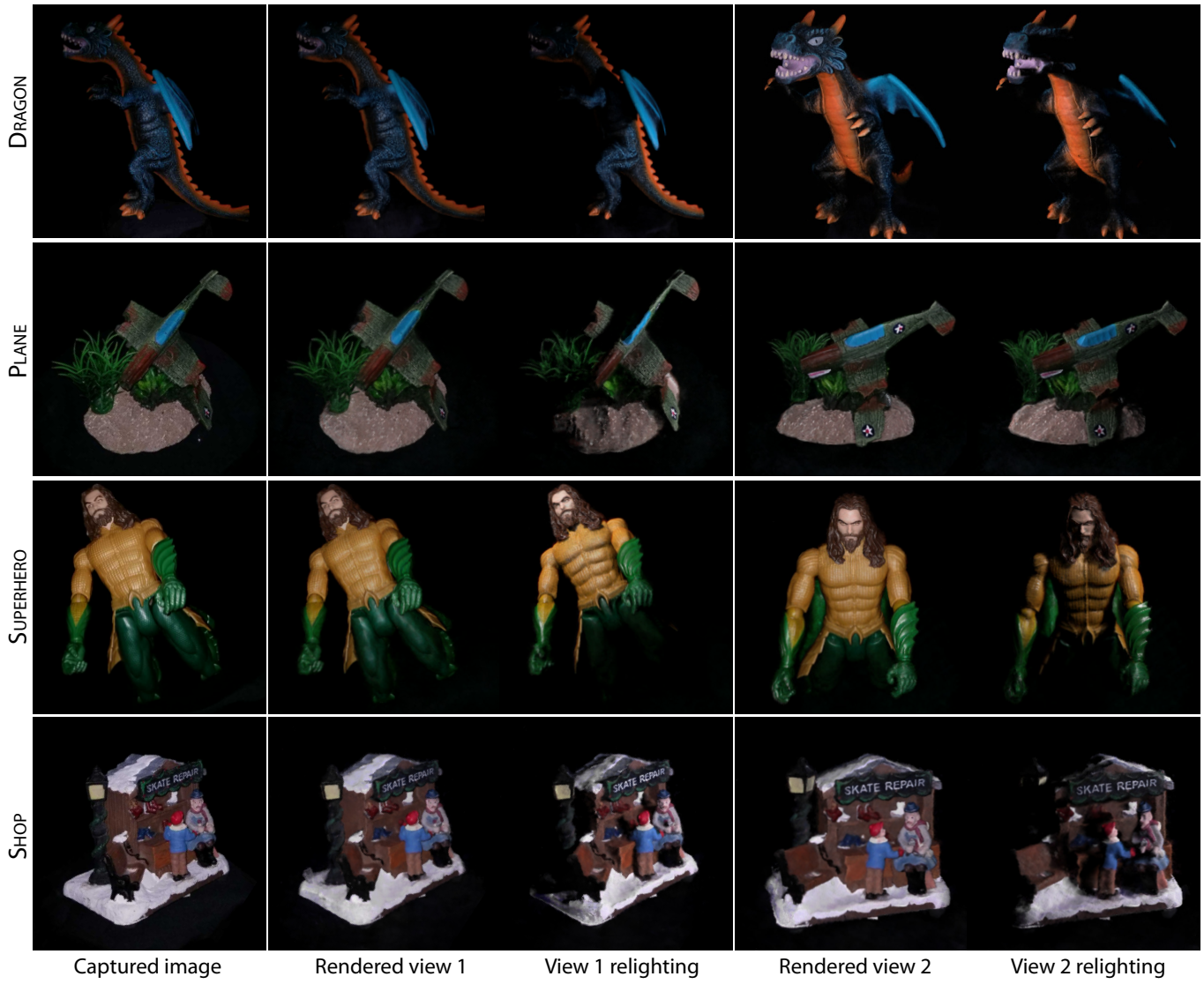


Fig. 5. Additional view synthesis and relighting results of our method. We show our rendered images under collocated light-view settings from two different views (columns 2 and 4). We also show captured image (not used for training) from view 1 in the left most column to demonstrate that our renderings closely reproduce the ground truth appearance of these scenes. We also demonstrate relighting results (columns 3 and 5) from each view, in which the light and view are no longer collocated, leading to challenging cast shadows.

*Limitations.* Our method is able to produce high-frequency appearance effects with fine details in most cases. However, it may still result in slightly blurry results when there are too many details (like the results in Fig. 6 and 7). Increasing the network capacity could potentially alleviate this. While our method generally generates a clean background without requiring any masks, some minor dark floaters occasionally appear, mainly coming from background regions that are not dark enough and are seen by several views. This usually can be addressed by masking the volume density in 3D with a bounding box. Our adaptive transmittance is efficient, but it may introduce some minor flickering in videos when doing relighting, due to inconsistent adaptive samples across frames. Increasing the number of samples in the volume usually resolves this. Some of these issues are visible in the supplementary video.

## 7 CONCLUSION

We present a deep learning based approach for appearance acquisition using a simple mobile phone setup. We present a novel neural reflectance field representation, which encodes volume rendering properties to model the geometry and reflectance of real scenes. We leverage a differentiable physically based ray marching framework


Fig. 6. Results on a furry object. We incorporate a fur reflectance model [Kajiya and Kay 1989] in our representation to capture this object. We show examples of ground truth captured images (that were not used for training) on the left, and renderings from our method on the right.

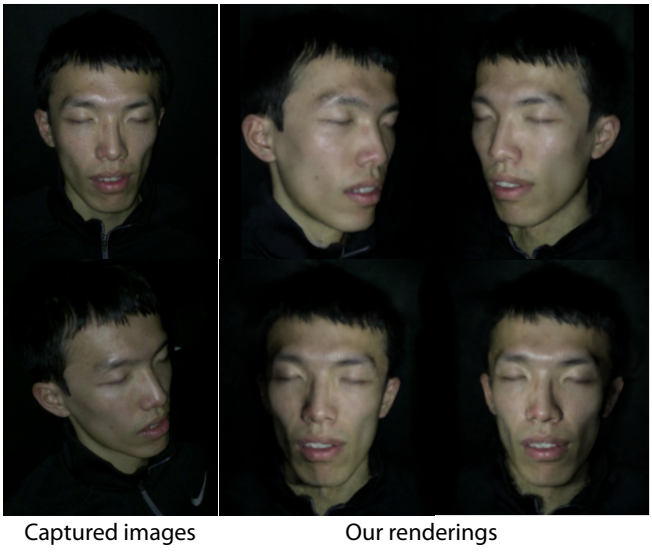


Fig. 7. Results on a human face. We show examples of ground truth captured images (that were not used for training) on the left, and on the right show renderings from our method for novel view synthesis with a collocated light (top) and relighting with a non-collocated light (bottom).

to learn the neural reflectance field in a scene-dependent deep training process. We demonstrate that our neural reflectance field can be effectively estimated from cellphone flash images under collocated camera and light, allowing us to render photo-realistic images under arbitrary camera and (non-collocated) light positions. Our method is able to generate high-quality relighting and view synthesis results,

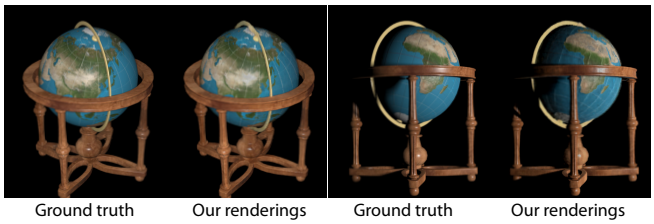


Fig. 8. Synthetic results. We compare the rendered images of our method under novel lighting and viewpoint with the ground truth images.

Fig. 9. We show examples of combining the neural reflectance field of a real scene (House) with a traditional synthetic scene (teapot). We render the composed scene using a standard rendering engine (Mitsuba [Jakob 2010]) under complex illumination.

reproducing challenging appearance effects, such as specularities, shadows, occlusions, and fine textures, which are significantly better than results from previous mesh-based and volume-based methods. Moreover, since our neural reflectance field are learned in a physically based rendering framework, they can be also rendered in standard graphics rendering engines, enabling scene modeling applications. Our approach takes a step towards making neural capture and rendering more practical and compatible with standard graphics pipelines.

## ACKNOWLEDGEMENTS

This work was supported in part by ONR grants N000141712687, N000141912293, N000142012529, NSF grant 1617234, Adobe, the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing.

## REFERENCES

Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2016. Reflectance Modeling by Neural Texture Synthesis. *ACM Trans. Graph.* 35, 4, Article 65 (July 2016), 13 pages. https://doi.org/10.1145/2897824.2925917

Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2015. Two-shot SVBRDF Capture for Stationary Materials. *ACM Transactions on Graphics* 34, 4, Article 110 (July 2015), 13 pages. https://doi.org/10.1145/2766967

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020a. Deep Reflectance Volumes: Relightable Reconstructionsfrom Multi-View Photometric Images. *ECCV* (2020).

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. 2020b. Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images. *arXiv preprint arXiv:2003.12642* (2020).

Rui Chen, Songfang Han, Jing Xu, and Hao Su. 2019. Point-based multi-view Stereo Network. In *ICCV*.

Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. 2019. Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness. *arXiv preprint arXiv:1911.12012* (2019).

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co., 145–156.

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* 37, 4 (2018), 128.

Carlos Hernández Esteban and Francis Schmitt. 2004. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding* 96, 3 (2004), 367–392.

Sing Choong Foo. 1997. *A gonioreflectometer for measuring the bidirectional reflectance of material for use in illumination computation.* Ph.D. Dissertation. Citeseer.

Yasutaka Furukawa and Jean Ponce. 2009. Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI* 32, 8 (2009), 1362–1376.

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

Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C Sankaranarayanan. 2017. Reflectance capture using univariate sampling of BRDFs. In *ICCV*. 5362–5370.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. 2017. SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In *ICCV*. 2307–2315.

James T Kajiya and Timothy L Kay. 1989. Rendering fur with three dimensional textures. *ACM Siggraph Computer Graphics* 23, 3 (1989), 271–280.

Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. 2018. Efficient reflectance capture using an autoencoder. *SIGGRAPH* 37, 4 (2018), 127–1.

Kaizhang Kang, Cihui Xie, Chengan He, Mingqi Yi, Minyi Gu, Zimin Chen, Kun Zhou, and Hongzhi Wu. 2019. Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–12.

Brian Karis and Epic Games. 2013. Real shading in unreal engine 4. *SIGGRAPH 2013 Course* (2013).

Joe Kniss, Simon Premoze, Charles Hansen, Peter Shirley, and Allen McPherson. 2003. A model for volume lighting and modeling. *IEEE transactions on visualization and computer graphics* 9, 2 (2003), 150–162.

Kiriakos N Kutulakos and Steven M Seitz. 2000. A theory of shape by space carving. *International journal of computer vision* 38, 3 (2000), 199–218.

Alexandr Kuznetsov, Miloš Hašan, Zexiang Xu, Ling-Qi Yan, Bruce Walter, Nima Khademi Kalantari, Steve Marschner, and Ravi Ramamoorthi. 2019. Learning generative models for rendering specular microgeometry. *ACM Transactions on Graphics (SIGGRAPH Asia 2019)* 38, 6 (2019), 225.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 31–42.

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018a. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *ECCV*. 72–87.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018b. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018*. ACM, 269.

Tom Lokovic and Eric Veach. 2000. Deep shadow maps. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 385–392.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 65.

Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. 2003. A Data-Driven Reflectance Model. *SIGGRAPH* 22, 3 (July 2003), 759–769.

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

Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, et al. 2019. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2018. Occupancy Networks: Learning 3D Reconstruction in Function Space. *arXiv preprint arXiv:1812.03828* (2018).

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *arXiv preprint arXiv:2003.08934* (2020).

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. 2018. Practical SVBRDF acquisition of 3D objects with unstructured flash photography. In *SIGGRAPH Asia 2018*. ACM, 267.

Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. 2015. On optimal, minimal BRDF sampling for reflectance acquisition. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–11.

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo methods for volumetric light transport simulation. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 551–576.

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. 2019. Multi-view relighting using a geometry-aware network. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. 2018. On the spectral bias of neural networks. *arXiv preprint arXiv:1806.08734* (2018).

Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural btf compression and interpolation. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 235–244.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image based relighting using neural networks. *ACM Transactions on Graphics* 34, 4 (2015), 1–12.

Stephan R Richter and Stefan Roth. 2018. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*. 1936–1944.

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*.

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. 2019a. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*. 2437–2446.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019b. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*. 1119–1130.

Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 175–184.

Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. 2017. Learning to synthesize a 4d rgbd light field from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*. 2243–2251.

Marc Stamminger and George Drettakis. 2002. Perspective shadow maps. In *SIGGRAPH*. 557–562.

Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. 2019. Single image portrait relighting. *SIGGRAPH* (2019).

A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. 2020. State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)* (2020).

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

Delio Vicini, Vladlen Koltun, and Wenzel Jakob. 2019. A learned shape-adaptive subsurface scattering model. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. *EGSR 07* (2007).

Lance Williams. 1978. Casting curved shadows on curved surfaces. In *SIGGRAPH*, Vol. 12. ACM, 270–274.

Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. 2019. Deep view synthesis from sparse photometric images. *SIGGRAPH* 38, 4 (2019), 76.

Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. 2016. Minimal BRDF sampling for two-shot near-field reflectance acquisition. *ACM Transactions on Graphics* 35, 6 (2016), 188.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep image-based relighting from optimal sparse samples. *SIGGRAPH* 37, 4 (2018), 126.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 2018. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*. 767–783.

Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. 2019. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*. 5525–5534.

Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. 2019. Deep Single-Image Portrait Relighting. In *CVPR*. 7194–7202.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Zhenglong Zhou, Zhe Wu, and Ping Tan. 2013. Multi-view photometric stereo with spatially varying isotropic materials. In *CVPR*. 1482–1489.