Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Decomposing a scene into its shape, reflectance and illumination is a fundamental problem in computer vision and graphics. Neural approaches such as NeRF have achieved remarkable success in view synthesis, but do not explicitly perform decomposition and instead operate exclusively on radiance (the product of reflectance and illumination). Extensions to NeRF, such as NeRD, can perform decomposition but struggle to accurately recover detailed illumination, thereby significantly limiting realism. We propose a novel reflectance decomposition network that can estimate shape, BRDF and per-image illumination given a set of object images captured under varying illumination. We propose a novel illumination integration network called Neural-PIL that replaces a costly illumination integral operation in the rendering with a simple network query result in efficient differentiable rendering. In addition, we also learn deep low-dimensional priors on BRDF and illumination representations using novel smooth manifold auto-encoders. Our decompositions can result in considerably better BRDF and light estimates enabling more accurate novel view-synthesis and relighting compared to prior art. Decompositions produced by our technique are compatible with any conventional 3D game engine or rendering engine and thus can be used for photorealistic real-time view synthesis and relighting. We will release the code upon publication.

1 Introduction

Inverse rendering is the task of decomposing a scene into its underlying physical properties, such as geometry and materials. Recovering these properties is useful for several vision and graphics applications such as view synthesis [10, 11, 51, 57], relighting [4, 10, 11, 23, 24, 38, 51, 58], and object insertion [7, 21, 38]. In this work, we aim to recover the 3D shape and spatially-varying bidirectional reflectance distribution function (SVBRDF) of an object imaged under different illumination conditions, as shown in Fig. 1. Estimating shape, illumination, and SVBRDF from 2D images is a highly ill-posed problem, as an observed pixel may appear dark either due to a dark surface material, or due to the incident light at that surface being reduced or absent.

Our approach follows the recent success of coordinate-based scene representation networks [14, 40, 43, 44, 46, 49] in representing 3D scenes for high-quality view-synthesis [44, 49]. These models decompose the scene into models of shape and radiance (emitted light), thereby enabling view synthesis. However, performing complete *inverse rendering* requires that radiance is decomposed further, into illumination and material appearances (SVBRDF) [6, 11, 51, 63]. A key component in learning these neural SVBRDF decomposition networks is the differentiable rendering [10, 11, 63] that generates images and gradients for the estimated lighting and SVBRDF parameters. These methods leverage traditional rendering techniques within modern deep learning frameworks to enable backpropagation. This is often expensive, as rendering requires computing integrals over the incoming light at each 3D location in the scene. As a remedy, recent works [11, 63] approximate the incident

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

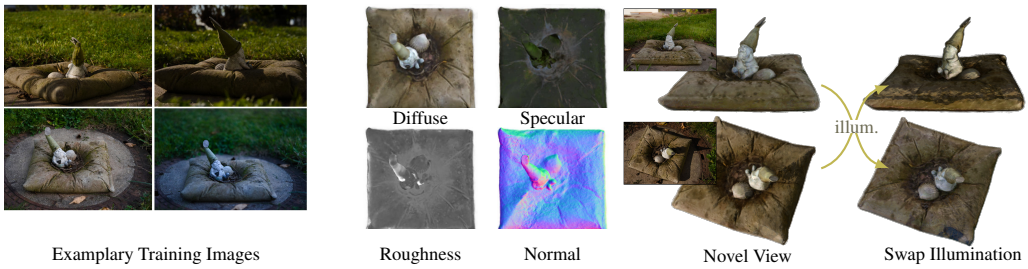


Figure 1: **Problem setting and example results.** Our neural-PIL based technique decomposes images observed under unknown illumination into high-quality BRDF, shape and illumination estimates. This allows us to then synthesize novel views (targets shown in insets) and perform relighting or illumination transfer.

light by spherical Gaussians (SG), thereby accelerating the illumination integration. However, these SG representations lack the capacity required to model or recover the shape and material properties of highly reflective objects or images in complex natural environments.

In this work, we aim to replace the costly illumination integration step within these rendering approaches with a learned network. Inspired by the real-time graphics literature on image-based lighting [28], we propose a novel pre-integrated lighting (PIL) network that converts the illumination integration process used in rendering into a simple network query. Our neural-PIL takes as input a latent vector representation for the environment map, the surface roughness, and an incident ray direction, and from them predicts an integrated illumination estimate. This query-based approach for light integration results in efficient rendering and thereby simplifies and accelerates rendering and optimization. This neural light representation is also significantly more expressive than the more commonly used SG representation, thereby enabling higher-fidelity renderings. The architecture of our neural-PIL uses conditional multi-layer perceptrons (MLP) with FiLM layers [12]. Fig. 2 illustrates this illumination pre-integration for different surface roughness levels.

In addition, we also present a smooth manifold auto-encoder (SMAE), based on interpolating auto-encoders [5], that can learn effective low-dimensional representations of light and BRDFs. This learned low-dimensional space serves as a strong regularizer or prior for constraining the solution space of BRDFs and illumination. These constraints are critical, due to the ill-posedness of our problem setting. The smoothness of this manifold enables stable and effective gradient-based optimization of BRDF and light parameters. The neural-PIL, light-SMAE, and BRDF-SMAE networks are pre-trained individually on a dataset with high-quality environment maps (illumination) and materials (BRDFs). We integrate these component networks into our overall reflectance decomposition framework, in which we optimize a 3D neural volume with shape and SVBRDF while also optimizing per-image illumination estimates.

We perform an empirical analysis on synthetic datasets, along with qualitative visual results on real-world datasets. We demonstrate that our decomposition network using our neural-PIL can estimate more accurate shape and material properties compared to prior art. The 3D assets with material properties produced by our model can be used to generate high-quality relighting and view-synthesis results with finer details that existing approaches struggle to reproduce.

2 Related Work

Coordinate-based MLPs allow spatial information to be stored within the weights of a neural network, thereby allowing the retrieval of information solely by querying coordinates [14, 43, 46, 52]. These methods have been combined with neural volume rendering [40] to enable photorealistic results on novel view synthesis, well-exemplified by NeRF [44]. In NeRF, a coordinate-based model is used to model a field of volumetric density and color, and renderings are produced by raymarching through that neural volume. Though NeRF is capable of photorealistic renderings, it has many limitations that have been explored by recent work, such as: no relighting capabilities [6, 11, 42, 51, 63], long training times [39, 54], long inference times [11, 25, 31, 39, 60], extraction of 3D geometry and materials [11], and generalization [12, 54, 64]. Our model addresses many of these ongoing research

efforts, as we enable relighting, extract a conventional 3D geometry and material estimate, and enable real-time rendering (as our 3D assets are compatible with existing real-time rendering engines). The concurrent works of NeRD [11], Srinivasan *et al*. [51] and Zhang *et al*. [63] are most clostly related to ours. These methods decomposes the scene into shape and analytical SVBRDF parameters. However, Srinivasan *et al*. require known illumination, and Boss *et al*. and Zhang *et al*. employ a spherical Gaussian (SG) model, which is not capable of modeling detailed illumination patterns.

BRDF estimation is a challenging research topic that aims to estimate the appearance of a physical material. For the highest accuracy results, measurements are performed under controlled laboratory conditions with known view and light positions [3, 9, 32, 33, 34], but this does not allow for the on-site capture of materials. Casual capture methods aim to solve this constraint by only requiring a camera, and sometimes a light source. Often, machine learning techniques are leveraged to reduce the ambiguity through the use of data-drive priors and large datasets of BRDFs. Additional constraints from planar surfaces viewed under camera flash illumination are considered for single-shot [1, 16, 26, 36, 47], few-shot [1] or multi-shot [2, 9, 17, 18, 20] estimation. This casual setup can be extended to estimating the BRDF and shape of objects [6, 7, 8, 10, 30, 45, 47, 62] or scenes [38, 48]. Most of these methods are based on known active illumination. A limited number of light sources — most often a single one — are assumed to be responsible for the majority of illumination in a scene. Relying on only natural, uncontrolled illumination adds several additional challenges due to the drastically increased ambiguity in the scene. Often these challenges are reduced by keeping the specular albedo non-spatially-varying, or by removing it entirely [35, 59, 63]. Other approaches require temporal traces and limit the casual capture setup [19, 56].

Illumination estimation from an image is an inherently challenging problem. The task is inherently linked to BRDF estimation, as illumination affects appearance and is only indirectly observable from its interactions with surface materials. These two tasks are therefore often solved in conjunction, sometimes by decomposing a single object into shape, reflectance, and a global set of spherical Gaussians (SGs) [11, 63]. Chen *et al*. [13] leverage a deep prior of environment maps with homogeneous materials, using an invertible neural BRDF model. Li *et al*. [38] decompose an entire scene into a simplified BRDF model with hemispherical SGs per point in the scene. The image of the environment in the background may be incorporated into prediction, shifting the problem to completion of the HDR environment map from sparse observations [21, 50, 55]. In our method, we not only learn a deep prior but a rendering aware network which is capable of integrating the environment illumination for a full reflectance lobe with a single sample. This reduces the number of required samples for rendering an image.

3 Method

Given an image collection of an object captured under varying illumination conditions and camera viewpoints, we aim to jointly estimate the object's 3D shape and spatially-varying BRDF, as well as the illumination conditions of each image. Our input consists of a set of q images with s pixels each: $C_j \in \mathbb{R}^{s\times 3}; j \in \{1, \dots, q\}$ along with per-pixel masks $M_j \in \{0, 1\}$ indicating which pixels belong to the object. Our goal is to learn a neural 3D volume $\mathcal{V}$ where, at each point $\boldsymbol{x} \in \mathbb{R}^3$, we estimate the BRDF parameters for the Cook-Torrance model [15] $\boldsymbol{b} \in \mathbb{R}^7$ (diffuse $\boldsymbol{b}_d \in \mathbb{R}^3$, specular $\boldsymbol{b}_s \in \mathbb{R}^3$, roughness $b_r \in \mathbb{R}$), unit-length surface normal $\boldsymbol{n} \in \mathbb{R}^3$ and optical density $\sigma \in \mathbb{R}$. In addition, we also estimate latent vectors representing per-image illumination $\boldsymbol{z}^l \in \mathbb{R}^{128}$. This problem statement corresponds to the practical application of recovering a 3D model of some real-world object (such as a statue or landmark) that has been photographed by different people at different times.

3.1 Preliminaries

Brief overview of neural radiance fields (NeRF). Our method is based on NeRF [44] which creates a neural volume for novel view synthesis using two Multi-Layer-Perceptrons (MLPs). The first MLP learns a coarse representation, which samples the 3D volume in a fixed sampling pattern, and the second MLP uses this knowledge to sample the volume in high-density areas at a finer resolution. The output of the MLP is a view-dependent color $\boldsymbol{c} \in \mathbb{R}^3$ and optical density $\sigma \in \mathbb{R}$ for each given 3D location $\boldsymbol{x} \in \mathbb{R}^3$ and view direction $\boldsymbol{d} \in \mathbb{R}^3$. In order to render the output color $\hat{\boldsymbol{c}} \in \mathbb{R}^3$ for a camera ray $r(t) = \boldsymbol{o} + t\boldsymbol{d}$, with ray origin $\boldsymbol{o} \in \mathbb{R}^3$ and view direction $\boldsymbol{d}$, we approximate (via numerical quadrature) the integral $\hat{\boldsymbol{c}}(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\boldsymbol{c}(t)\, dt$ with $T(t) = \exp(-\int_{t_n}^{t} \sigma(s)\, ds)$, using the near and far bounds of the ray t_n and t_f respectively [44].

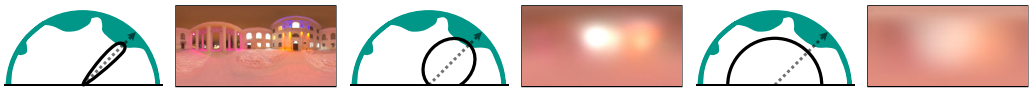


Figure 2: **Pre-integrated lighting.** As the roughness of the material increases, the reflected radiance depends on a larger region of the environment map. Brute-force integrals over the environment map are expensive, hence our use of a coordinate-based MLP that is trained to directly output the integrated valued conditioned on the requested roughness.

Image formation and image-based lighting. NeRF directly models view-dependent color $\hat{\boldsymbol{c}}$ at each 3D location. Thus, a simple image formation process that integrates the color information along camera rays is sufficient to render images. In contrast, we want to explicitly estimate an object material decomposition at each 3D location. We therefore must use a more explicit rendering formulation that relates image formation to BRDFs and illumination. The rendering equation [27] estimates the radiance $L_o \in \mathbb{R}^3$ at $\boldsymbol{x}$ along the outgoing view direction $\boldsymbol{\omega}_o \in \mathbb{R}^3$ ($\boldsymbol{\omega}_o = -\boldsymbol{d}$): $L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \boldsymbol{b}) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \boldsymbol{n})\, d\boldsymbol{\omega}_i$, where f_r is the BRDF evaluation, $L_i \in \mathbb{R}^3$ is incoming light, and $\boldsymbol{\omega}_i \in \mathbb{R}^3$ is the incoming light direction. Using this single bounce rendering formulation, and ignoring exposure variation and tone-mapping, L_o is equivalent to NeRF's color $\hat{\boldsymbol{c}}$. The formulation can be split into diffuse and specular components:

$$L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \underbrace{\frac{\boldsymbol{b}_d}{\pi} \int_\Omega L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \boldsymbol{n})\, d\boldsymbol{\omega}_i}_{\text{diffuse}} + \underbrace{\int_\Omega f_s(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \boldsymbol{b}_s, b_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \boldsymbol{n})\, d\boldsymbol{\omega}_i}_{\text{specular}} \tag{1}$$

where f_s now only represents the specular portion of the BRDF evaluation. Following Karis *et al*. [28], several parts of this integration can be pre-computed [29]:

$$L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) \approx \underbrace{(\boldsymbol{b}_d/\pi)\tilde{L}_i(\boldsymbol{n}, 1)}_{\text{diffuse}} + \underbrace{\boldsymbol{b}_s(F_0(\boldsymbol{\omega}_o, \boldsymbol{n})B_0(\boldsymbol{\omega}_o \cdot \boldsymbol{n}, b_r) + B_1(\boldsymbol{\omega}_o \cdot \boldsymbol{n}, b_r))\tilde{L}_i(\boldsymbol{\omega}_r, b_r)}_{\text{specular}} \tag{2}$$

The illumination is now pre-integrated as: $\tilde{L}_i(\boldsymbol{\omega}_r, b_r) = \int_\Omega D(b_r, \boldsymbol{\omega}_i, \boldsymbol{\omega}_r) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i$ which only depends on the mirrored view direction $\boldsymbol{\omega}_r$ (which subsumes the surface normal $\boldsymbol{n}$) and the roughness b_r, where D describes the microfacet distribution [53]. This light pre-integration is illustrated in Fig. 2. Note that the same pre-integrated $\tilde{L}_i(\boldsymbol{\omega}_r, b_r)$ is queried twice: the diffuse part captures the entire hemisphere and therefore is parameterized by the surface normal $\boldsymbol{n} \in \mathbb{R}^3$ and a diffuse roughness of 1. The specular part looks up $\tilde{L}_i(\boldsymbol{\omega}_r, b_r)$ for the reflected view direction $\boldsymbol{\omega}_r \in \mathbb{R}^3$ and the specular roughness b_r. While the pre-integration already considers the microfacet distribution D, one must also account for shadowing, masking, and the Fresnel term. As shown in Karis *et al*. [28], these remaining parts can be pre-computed into two 2D lookup textures (LUT) B_0 and B_1 indexed by $(\boldsymbol{\omega}_o \cdot \boldsymbol{n})$ and the roughness b_r. These are combined with the Fresnel term at normal incidence $F_0(\boldsymbol{\omega}_o, \boldsymbol{n}) = (1 - \boldsymbol{\omega}_o \cdot \boldsymbol{h})^5$ with $\boldsymbol{h} = \|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\|$.

This pre-integration approach replaces the complex integration during shading with a set of simple additions and multiplications. We have integrated the core idea of this approach into an efficient differentiable neural rendering framework, which allows for the optimization of geometry, BRDF, and illumination simultaneously via standard backpropagation. We further reduce the computational complexity by mimicking the pre-integration of $\tilde{L}_i(\boldsymbol{\omega}_r, b_r)$ with a simple query through our Neural-PIL that operates directly on a neural representation of the illumination, as we will now explain.

3.2 Decomposition with Neural-PIL

Fig. 3 shows the neural decomposition architecture which closely follows the architectures of NeRF [44] and NeRD [11], but with some key differences. The coarse network learns a view and illumination-dependent color whereas the fine network decomposes the scene into BRDF parameters.

Coarse network. Like in NeRF [44], the aim of the coarse network is to obtain rough point density that helps in finer sampling for the following decomposition network. As illustrated in Figure 3a, the coarse network takes 3D location $\boldsymbol{x}$, view direction $\boldsymbol{\omega}_o$ and illumination embedding $\boldsymbol{z}^l$ as input and predicts point density σ and color $\boldsymbol{c}$ at $\boldsymbol{x}$. In contrast to NeRF, which estimates view-conditioned

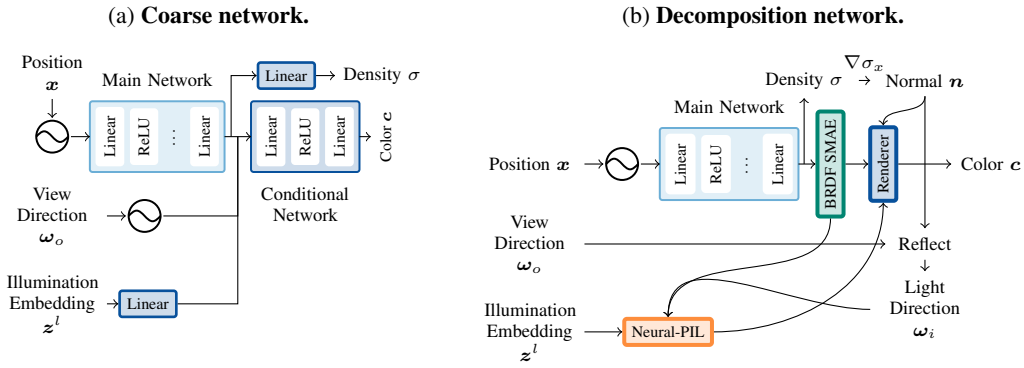


Figure 3: **Decomposition with Neural-PIL architecture.** (a) Similar to NeRF-W[42] our coarse network uses a latent illumination estimate to predict a view-dependent color and density. (b) The actual decomposition is performed in the fine network. Pre-trained networks restrict the possible BRDF representation (BRDF-SMAE) and the incident lighting (PIL) to lower-dimensional spaces. A single evaluation of Neural-PIL returns a pre-filtered illumination cone according to surface roughness. Using that, the BRDF estimate, and a surface normal (the unit-norm gradient of our density estimate), we render the shaded color $\boldsymbol{c}$.

colors, we estimate both view and illumination conditioned colors, as our input images can be captured under varying illumination. Refer to the supplement for architecture details.

Decomposition network. The decomposition network estimates density σ and BRDF embedding $\boldsymbol{z}^b \in \mathbb{R}^4$ at each 3D location $\boldsymbol{x}$ in the implicit volume. As illustrated in Figure 3b, the conditional network in the coarse network is replaced by explicit rendering in the decomposition network. There are two key innovations in the decomposition network: 1) Use a novel pre-integrated light (PIL) network that results in efficient rendering while also representing the illumination with high fidelity. 2) We learn smooth low-dimensional manifolds to represent illumination and BRDF parameters, which serve as strong priors. We will now explain our rendering process, the Neural-PIL, and the smooth manifold auto-encoders (SMAE).

Rendering process. For rendering, we use the rendering formulation in Equation 2. We estimate normal $\boldsymbol{n}$ at $\boldsymbol{x}$ by computing the gradient $\nabla\sigma_x$ of the density w.r.t. the input position (by passing gradients through the decomposition network). We also convert the BRDF embedding $\boldsymbol{z}^b$ into BRDF parameters $\boldsymbol{b}$ with our BRDF-SMAE. Unlike NeRF [44], which integrates sample colors along the camera ray, we first compute the expected termination of each ray (similar to depth map) with the sample densities along the camera ray, and then do the rendering only at that point along each camera ray. At the ray termination positions, we compute the integrated illumination $\tilde{L}_i(\boldsymbol{\omega}_r, b_r)$ and use Equation 2 for rendering with the estimated BRDF $\boldsymbol{b}$ and normal $\boldsymbol{n}$.

Neural-PIL. The integration of the incoming light is traditionally approximated by Monte Carlo sampling, in which illumination contributions from many directions are numerically integrating. The computation of the pre-integrated $\tilde{L}_i(\boldsymbol{\omega}_r, b_r)$ also involves either this costly numerical accumulation, or a convolution performed on the complete environment map — though neither approach is practical within a differential rendering engine. We therefore learn a network that performs this light pre-integration, thereby converting the costly integral computation into a simple network query. The architecture of the Neural-PIL is visualized in Fig. 4. The Neural-PIL takes as input the illumination embedding $\boldsymbol{z}^l$, the incoming light direction $\boldsymbol{\omega}_i$ and the roughness b_r at a point, and directly predicts the pre-integrated light $\tilde{L}_i(\boldsymbol{\omega}_r, b_r)$. The aim of the Neural-PIL is to first decode the illumination along the incoming mirror direction $\boldsymbol{\omega}_r$ from the given embedding $\boldsymbol{z}^l$ and then mimic the light pre-integration process for the surface roughness b_r. Following this general intuition, the Neural-PIL takes $\boldsymbol{\omega}_r$ as input, and we condition the first few layers of the network with illumination $\boldsymbol{z}^l$, and condition a later layer with roughness b_r. The first few layers are intended to decode all required illumination information for the given direction, and the later layers are intended to perform light integration conditioned on the material roughness.

For the Neural-PIL design, we leverage a pi-GAN-like [12] architecture with FiLM-SIREN layers. Each FILM-SIREN layer [12] takes the modulating parameters (a scalar λ_0 and two vectors $\boldsymbol{\beta}$ and

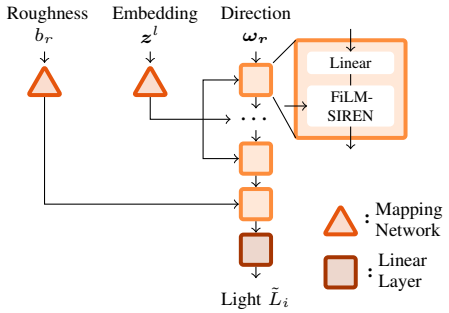


Figure 4: **Neural-PIL.** A coordinate-based MLP returns the pre-integrated radiance for the query direction, where roughness determines the integration footprint.

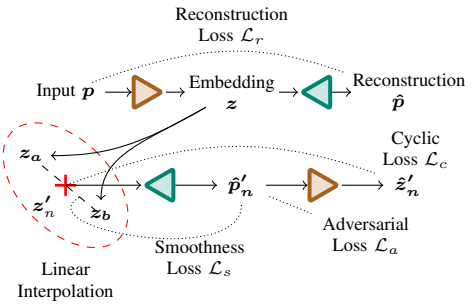


Figure 5: **Smooth manifold auto-encoder.** By imposing specific losses on interpolations between input samples, our Smooth Manifold Autoencoder encourages a smooth embedding space.

γ) to modulate the output $\boldsymbol{y}$ of the earlier linear layer followed by sine computation as follows: $\phi(\boldsymbol{y}) = \sin(\lambda_0 \boldsymbol{\gamma} \odot \boldsymbol{y} + \boldsymbol{\beta})$, where $\odot$ denotes a Hadamard product. In our Neural-PIL, we employ two mapping networks to predict modulating parameters($\boldsymbol{\gamma}$, $\boldsymbol{\beta}$) for the FiLM-SIREN layers. The first mapping network generates the modulating parameters for the first layers from the illumination embedding $\boldsymbol{z}^l$, while the second one for the penultimate layer from the given roughness $\boldsymbol{b}_r$.

In addition to converting a costly light integration process in the rendering into a simple network query, our Neural-PIL has another key advantage. State-of-the-art differentiable rendering frameworks [11, 22, 38, 63] that work with BRDFs use either spherical Gaussian (SG) or spherical harmonic (SH) light representations, which both suffer from lack of fine details. Although one could represent fine illumination details with a large number of SG or SH bands, this parameter increase would also make the rendering prohibitively slow with high memory costs. In contrast, our Neural-PIL is an MLP that directly produces pre-integrated light required for the rendering. Our experiments also demonstrate that our Neural-PIL can represent finer details in illumination compared to SG representation (Sec. 4 - Fig. 6). See the supplementary for more Neural-PIL architecture details.

Smooth manifold auto-encoder (SMAE). Since jointly estimating 3D shape, materials and lighting is a highly underconstrained problem, in order to converge to plausible solutions, we must regularize optimization towards likely illuminations and BRDFs. For this, we learn low-dimensional smooth manifolds that capture the data distribution of BRDFs and illuminations. In addition to acting as strong priors, optimizing on smooth manifolds (as opposed to directly optimizing on standard BRDF space, which need not be smooth) allows for more effective gradient-based optimization of reflectance decomposition for a given scene. Fig. 5 illustrates our SMAE that we use to learn separate low-dimensional manifolds to represent BRDF and illumination embeddings. Specifically, we use Interpolating Autoencoders [5] with several additional loss functions. The encoder network E takes input $\boldsymbol{p}$ (either BRDF or light environment map) and generates the latent embedding $\boldsymbol{z}$, which is then passed onto the decoder network G that generates an input reconstruction $\boldsymbol{p}'$. We then randomly sample two latent vectors from the mini-batch: $\boldsymbol{z}_a$ and $\boldsymbol{z}_b$; followed by sampling $m \in \mathbb{N}$ linearly interpolated embeddings that are uniformly spaced between $\boldsymbol{z}_a$ and $\boldsymbol{z}_b$: $\{\boldsymbol{z}'_n\}|n = 1, 2, \ldots, m$. We pass each of these interpolated latents $\boldsymbol{z}'_n$ through the decoder G and the encoder E to obtain $\hat{\boldsymbol{p}}'_{\boldsymbol{n}}$ and $\hat{\boldsymbol{z}}'_{\boldsymbol{n}}$, respectively. Using the four losses depicted in Fig. 5 the encoder and decoder networks are trained jointly. One is the standard reconstruction loss $\mathcal{L}_r$ between input $\boldsymbol{p}$ and reconstruction $\boldsymbol{p}'$. In addition, we add a discriminator network on $\hat{\boldsymbol{p}}'_{\boldsymbol{n}}$ and use the standard adversarial loss $\mathcal{L}_a$ used in LSGAN [41], which ensures that the interpolated latent vectors can generate plausible data. We enforce a bijective mapping of the encoder and the decoder with a cyclic loss $\mathcal{L}_c$ which is the L_2-loss between the interpolated latents $\{\boldsymbol{z}'_n\}$ and their re-estimated counterparts $\{\hat{\boldsymbol{z}}'_n\}$. Lastly, to ensure that the learned embedding space is smooth, we impose a smoothness loss $\mathcal{L}_s$ on the gradient of decoder G w.r.t. the interpolating scalar value α: $\mathcal{L}_s = 1/m \sum_n (\nabla_\alpha G(\boldsymbol{z}'_n))^2$. The total loss to train SMAE is a combination of the 4 losses: $\mathcal{L} = \mathcal{L}_r + \lambda_1 \mathcal{L}_a + \lambda_2 \mathcal{L}_c + \lambda_3 \mathcal{L}_s$.

Despite being only 7-dimensional, the space of the Cook-Torrence BRDF representation [15] that we use is too unconstrained for our task, and imposes strong correlations between the diffuse and specular terms of real-world materials. We therefore train a BRDF-SMAE with an MLP encoder

and decoder that maps these 7D parameters into 4D latent embeddings $\boldsymbol{z}^b \in \mathbb{R}^4$ using a dataset of real-world BRDF material collections [10]. Similarly, real-world illuminations exhibit significant statistical regularities: lights are more likely to be tinted blue or yellow, and brighter light is more likely to coming from above than below. To capture this regularity, we train a Light-SMAE with CNN encoder and decoder on a dataset of 320 environment maps from [61]. We then map the 128×256 2D environment maps onto a 128-dimensional smooth latent space, $\boldsymbol{z}^l \in \mathbb{R}^{128}$. We provide more network and training details for BRDF-SMAE and Light-SMAE in the supplementary.

Training. Since we have several networks in our decomposition learning pipeline, we will briefly explain the overall training protocol here with more details in the supplementary. We first train Light-SMAE and BRDF-SMAE with a dataset of environment maps and BRDFs respectively. With the frozen BRDF-SMAE's decoder in the decomposition network and with Neural-PIL in the rendering step, we jointly optimize both the coarse network and the decomposition network for a given set of scene images. For stability, we only optimize the illumination embedding z^l via decomposition network and we do not backpropagate the loss signal onto illumination in the coarse network. More training details can be found in the supplement.

4 Experiments

We evaluate our approach w.r.t. different baselines on three different aspects: BRDF and light estimation, view synthesis and relighting.

Baselines. Currently, there exists no published work that tackles the same problem of estimating shape, illumination and BRDF from images of varying illumination. The closest one is NeRD [11] which is a concurrent work and forms our primary comparison across different evaluations. For view synthesis, we also compare with NeRF [44] as we base our main network design on NeRF. For BRDF and light evaluations, we also compare with Li *et al*. [37] which does BRDF decomposition from a single image. In addition, we combine Li *et al*. [37] with NeRF [44] to create a baseline that is closer to our problem setting compared to only [37].

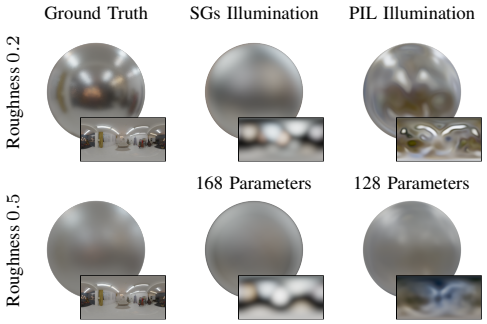
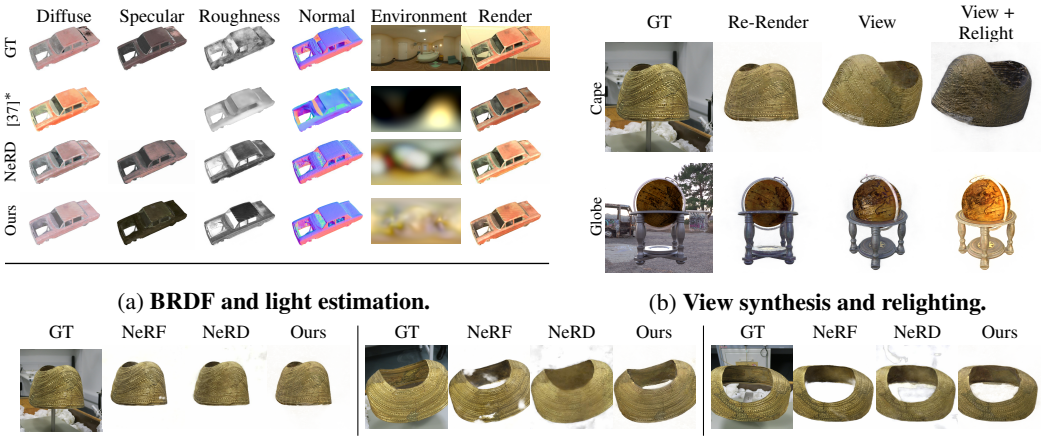


Figure 6: **Neural-PIL vs. spherical Gaussian (SG) renderings.** With known geometry and reflectance, we optimize for SGs as well as our latent illumination via Neural-PIL. This figure shows final renderings with the optimized light parameters, while the recovered illumination is shown in the insets. Despite Neural-PIL having fewer parameters, it is able to recover more detailed environment maps and thereby produce accurate renderings.

Datasets. To enable the comparisons with NeRD [11], we use the publicly released dataset used in [11] which provides 3 synthetic (Chair, Globe, Car) and 4 real-world scenes. Two real-world datasets consist of multi-view captures with fixed, unknown illumination (Head, Cape) and two others where the illumination varies with each image (Gnome, MotherChild). In addition, we also present view synthesis results on datasets (Ship, Chair, Lego) used in NeRF [44].

Fildelity of Neural-PIL. Since Neural-PIL forms the key component of our decomposition framework, we evaluate its learned light representation with a more commonly used spherical Gaussians (SG). We render a simple metallic sphere using an unseen environment as shown in Fig. 6, with two different roughness levels 0.2 and 0.5. Assuming known roughness and shape, we optimize for SG illumination using the SG-based differentiable rendering used in NeRD [11]. Similarly, we optimize the latent illumination representation using our Neural-PIL-based renderer. The resulting SG illumination and Neural-PIL illumination are shows in Fig. 6. Compared to the SG illumination model with 24 lobes and 168 parameters our recovered illumination vector $\boldsymbol{z}^l$ with only 128 dimensions captures more details, especially in the high-frequency light panels. This leads to a significantly reduced rendering error for both roughness values even though the illumination prediction is more ambiguous for rougher materials. Besides the improved quality, Neural-PIL based rendering is also much faster. Rendering 1×10^6 samples with our Neural-PIL network takes just $1.86 \times 10^{-3}\,\mathrm{s}$ compared to $0.21\,\mathrm{s}$ rendering with 24 SGs.



(a) **BRDF and light estimation.**

(b) **View synthesis and relighting.**

(c) **View synthesis with NeRF, NeRD and our approach.**

Figure 7: **Visual comparisons.** (a) Our model produces more accurate BRDF and illumination estimates, which results in more faithful rendering results. (b) To evaluate view synthesis and relighting we keep the camera and light fixed (col 2), then move the camera (col 3), and then adjust the lighting (col 4). (c) Even when using a single illumination (the problem setting used by NeRF) our method produces shape estimates with fewer artifacts and more detail than both NeRF or NeRD.

BRDF and light evaluations. Following the results in NeRD [11], Table 1a shows the BRDF and light estimation metrics for different baseline techniques computed on the scenes Globe, Car and Chair. Our approach with Neural-PIL reduces the error on estimated illumination by a factor of 2 in comparison to Li *et al*. [37]. When compared with NeRD, our approach resulted in better diffuse and roughness parameters. Only the prediction of the specular parameter is worse compared to NeRD. This may be due to NeRD's basecolor-metallic parameterization, which can reduce some ambiguity but also limits the space of expressible materials. A visual comparison is shown in Fig. 7a demonstrating clear visual improvements w.r.t. [37]. One can observe higher frequency details in the estimated environment map using our approach compared to NeRD. The final renderings also show that our result is more close to GT rendering (top-right) compared to NeRD. Refer to the supplementary for more visual results.

(a) **BRDF decomposition**

Parameter	[37]	[37]+[44]	[11]	Ours
Diffuse↓	0.7835	0.7679	0.0150	**0.0095**
Specular↓	—	—	**0.0027**	0.0207
Roughness↓	0.0191	0.0187	0.0321	**0.0033**
Light↓	0.6240	—	N/A	**0.2475**

(b) **View synthesis**

	Synthetic		Real-World	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NeRF	**31.3966**	0.9280	23.1013	0.8936
NeRD	28.7277	0.9381	25.4678	0.8517
Ours	27.1061	**0.9402**	**27.4254**	**0.9507**

(c) **View synthesis and relighting**

	Synthetic		Real-World	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NeRF	21.2860	0.9108	22.2083	0.8972
NeRD	22.8390	0.9160	25.5525	0.9415
Ours	**28.5946**	**0.9471**	**27.4254**	**0.9507**

Table 1: **Comparisons with baselines.** (a) A comparison against methods for BRDF decomposition under unknown illuminations, where we see that our model performs well (consistent with our improved relighting performance). (b) An evaluation of view-synthesis (without relighting) under a single illumination, where our model performs well despite this not being our primary task. (c) Here, input images are taken under different illumination conditions, so joint relighting and view synthesis are required. Our model outperforms both baselines by a significant margin: NeRF (which is not intended to address this task) but also NeRD (which targets this same problem statement).

View synthesis and relighting. On the datasets with fixed illumination (Cape, Head, NeRF-Ship, NeRF-Chair, NeRF-Lego), we can directly compare our renderings with existing novel view synthesis techniques (both NeRF [44] and NeRD [11] here). Table 1b shows novel view evaluation metrics on these datasets with fixed illumination. Results show that our results are better than both NeRF and NeRD showing the better capture of view-dependent effects. However, the fixed illumination might in general limit the decomposition capabilities, as shadows do appear always at the same surface locations and therefore might not be correctly disentangled from the BRDF.

On datasets with varying illumination across images (Gnome, MotherChild, Chair, Car, Globe), we need to do both view synthesis and relighting to generate novel unseen test views. Table 1c shows the results on these datasets. NeRF [44] can not do relighting and thus is included as a very weak baseline here. The results are significantly better than NeRD showing that our Neural-PIL network can more faithfully estimate the underlying scene parameters resulting in better relighting under novel illumination conditions.

Fig. 7b shows a couple of results with view synthesis and relighting. The renderings demonstrate realistic view synthesis including re-lighting. Fig. 7c shows novel view synthesis comparison with NeRF and NeRD on Cape scene captured with fixed illumination. Despite NeRF being a strong baseline it could not recover the complete surface due to the reflective nature of the surfaces. On the other hand, our view synthesis results are more close to the GT on unseen views compared to both NeRF and NeRD.

5 Conclusion

We presented a novel reflectance decomposition technique that can estimate shape, per-image illumination and BRDF from images captured in unknown illumination. The key innovation is the neural-PIL network that can replace costly light integration during rendering with a simple network query resulting in a fast and practical differentiable rendering with high-fidelity illumination. In addition, we propose novel learning techniques for SMAEs that can learn effective low-dimensional smooth manifolds for both BRDF and light representations. Experiments on both synthetic and real-world scenes demonstrate superior decomposition results along with better novel view synthesis and relighting in comparison to prior art.

Limitations While our techniques make significant strides in the areas of differentiable rendering as well as shape and material decomposition, several challenges still remain in this complex problem setting. Our approach can not handle inter-reflections. Concurrent works such as NeRV [51] are capable of handling inter-reflections and shadowing but only for known illumination. Due to the large ambiguity between the interplay of all effects solving everything jointly is an extremely challenging problem. Another limitation of our method is that we can not guarantee to converge to the correct underlying BRDF, reflectance and illumination. Our loss is only photometric and, therefore, we find one solution which explains all input images, *e.g.*, adding a new input image might converge to a different representation, as new effects are visible. Also, while our neural-PIL network is capable of producing higher frequency illumination with fewer parameters compared to standard representations such as SGs, mirror-like reflections are still not possible and therefore can limit the reconstruction quality when mirror-like surfaces are present in the scene.

Broader impact As is generally the case in machine learning, biases in the data used during training may result in biases in the learned model. This bias is relatively well-explored in conventional machine learning tasks such as machine translation or object recognition, but is perhaps less intuitive in the context of NeRF-like coordinate-based models, wherein the input to the network is a single coordinate and the output is some physical quantity. These models do not explicitly reason about "semantics" or "categories", which may ameliorate many concerns about bias or equity, but caution is still warranted. Our pre-trained networks for BRDFs and incident illumination serve as priors on materials and lighting conditions, and so any bias in the training data used for pre-training those models may result in bias in our estimations of materials and illumination. If the presented techniques were applied to human subjects (which we do not do here) the performance of the model might vary as a function of the subject's skin color for skewed training distributions.

The purpose of our model is to better enable the creation of highly accurate 3D models from photographs, which could then be used as visual effects in film or television, or in video games. Currently, the creation or acquisition of 3D assets is largely the domain of specialized CGI artists. Improved tools for automating this task may lower the barrier to entry into these careers, which may be seen as harming job opportunities for artists already working in this area. Despite this, we are hopeful that the commoditization of tools for 3D model acquisition will have a net positive impact by allowing a wider range of people to automatically construct high-fidelity 3D models from their image collections.

References

[1] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. In *ACM Transactions on Graphics (ToG)*, 2018.

[2] Rachel Albert, Dorian Yao Chan, Dan B. Goldman, and James F. O'Brian. Approximate svBRDF estimation from mobile phone video. In *Eurographics Symposium on Rendering*, 2018.

[3] Louis-Philippe Asselin, Denis Laurendeau, and Jean-François Lalonde. Deep SVBRDF estimation on real materials. In *International Conference on 3D Vision (3DV)*, 2020.

[4] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2015.

[5] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *International Conference on Learning Representations (ICLR)*, 2019.

[6] 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 e-prints*, 2020.

[7] 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. In *European Conference on Computer Vision (ECCV)*, 2020.

[8] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] Mark Boss, Fabian Groh, Sebastian Herholz, and Hendrik P. A. Lensch. Deep Dual Loss BRDF Parameter Estimation. In *Workshop on Material Appearance Modeling*, 2018.

[10] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz. Two-shot spatially-varying BRDF and shape estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[11] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. NeRD: Neural reflectance decomposition from image collections. In *ArXiv e-prints*, 2021.

[12] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[13] Zhe Chen, Shohei Nobuhara, and Ko Nishino. Invertible neural BRDF for object inverse rendering. In *European Conference on Computer Vision (ECCV)*, 2020.

[14] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1982.

[16] Valentin Deschaintre, Miika Aitalla, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. In *ACM Transactions on Graphics (ToG)*, 2018.

[17] Valentin Deschaintre, Miika Aitalla, Fredo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF capture with a multi-image deep network. In *Eurographics Symposium on Rendering*, 2019.

[18] Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. In *Eurographics Symposium on Rendering*, 2020.

[19] Yue Dong, Guojun Chen, Pieter Peers, Jianwen Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2014.

[20] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2019.

[21] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics (ToG)*, 2017.

[22] Marc-Andre Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. Deep parametric indoor lighting estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[23] Dan B. Goldman, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[24] Tom Haber, Christian Fuchs, Phillipe Bekaer, Hans-Peter Seidel, Michael Goesele, and Hendrik P. A. Lensch. Relighting objects from image collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[25] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis, 2021.

[26] Philipp Henzler, Valentin Deschaintre, Niloy J Mitra, and Tobias Ritschel. Generative modelling of BRDF textures from flash images. *ArXiv e-prints*, 2021.

[27] James T. Kajiya. The rendering equation. In *ACM Transactions on Graphics (SIGGRAPH)*, 1986.

[28] Brian Karis. Real shading in unreal engine 4. Technical report, Epic Games, 2013.

[29] Jan Kautz, Pere-Pau Vázquez Alcocer, Wolfgang Heidrich, and Hans-Peter Seidel. A unified approach to prefiltered environment maps. *Eurographics Symposium on Rendering*, 2000.

[30] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncalibrated neural inverse rendering for photometric stereo of general surfaces. 2021.

[31] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[32] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)*, 2004.

[33] Hendrik P. A. Lensch, Jan Kautz, Michael Gosele, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Conference on Rendering*, 2001.

[34] Hendrik P.A. Lensch, Jochen Lang, M. Sa Asla, and Hans-Peter Seidel. Planned sampling of spatially varying BRDFs. In *Computer Graphics Forum*, 2003.

[35] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. In *ACM Transactions on Graphics (ToG)*, 2017.

[36] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *European Conference on Computer Vision (ECCV)*, 2018.

[37] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018.

[38] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and SVBRDF from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[39] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[40] 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 (ToG)*, 2019.

[41] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[42] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *ArXiv e-prints*, 2020.

[43] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] Ben Mildenhall, Pratul 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 (ECCV)*, 2020.

[45] Giljoo Nam, Diego Gutierrez, and Min H. Kim. Practical SVBRDF acquisition of 3d objects with unstructured flash photography. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2018.

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and SVBRDF estimation. In *European Conference on Computer Vision (ECCV)*, 2020.

[48] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[49] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[50] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[51] 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 *arXiv*, 2020.

[52] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[53] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Eurographics Symposium on Rendering*, 2007.

[54] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[55] Henrique Weber, Prévost. Donald, and Jean-François Lalonde. Learning to estimate indoor lighting from 3d objects. In *International Conference on 3D Vision (3DV)*, 2018.

[56] Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2016.

[57] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 2019.

[58] Zexiang Xu et al. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 2018.

[59] Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Single image surface appearance modeling with self-augmented cnns and inexact supervision. *Computer Graphics Forum*, 2018.

[60] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ArXiv e-prints*, 2021.

[61] Greg Zaal. Hdri haven, 2019. https://hdrihaven.com/.

[62] Jianzhao Zhang, Guojun Chen, Yue Dong, Jian Shi, Bob Zhang, and Enhua Wu. Deep inverse rendering for practical object appearance scan with uncalibrated illumination. In *Advances in Computer Graphics*, 2020.

[63] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians forphysics-based material editing and relighting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[64] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image GANs meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *International Conference on Learning Representations (ICLR)*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Will be made available upon acceptance
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]