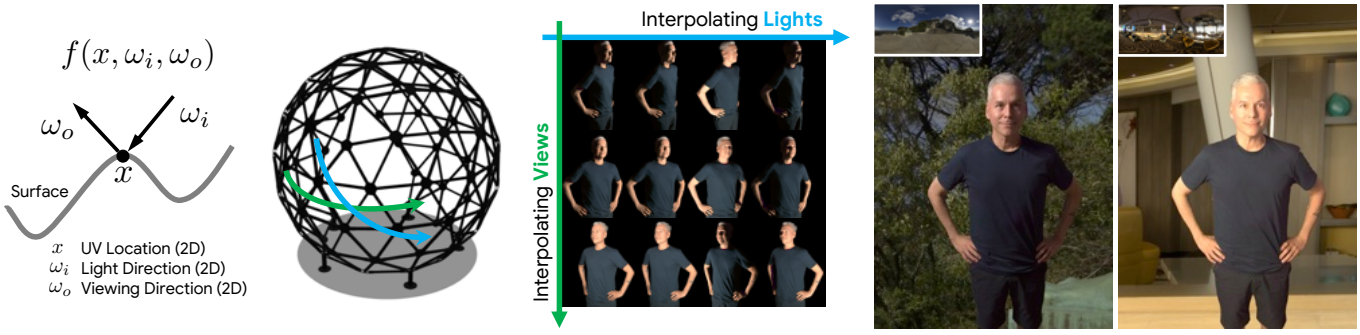
Neural Light Transport for Relighting and View Synthesis

XIUMING ZHANG, Massachusetts Institute of Technology
SEAN FANELLO and YUN-TA TSAI, Google
TIANCHENG SUN, University of California, San Diego
TIANFAN XUE, ROHIT PANDEY, SERGIO ORTS-ESCOLANO, PHILIP DAVIDSON, CHRISTOPH RHEMANN, PAUL DEBEVEC, and JONATHAN T. BARRON, Google
RAVI RAMAMOORTHI, University of California, San Diego
WILLIAM T. FREEMAN, Massachusetts Institute of Technology & Google



(A) The 6D Light Transport Function Being Learned | (B) Capture Setup: multiple views; one light at a time | (C) Simultaneous Relighting and View Synthesis | (D) HDRI Relighting

Fig. 1. (A) Neural Light Transport (NLT) learns to interpolate the 6D light transport function of a surface as a function of the UV coordinate (2 DOFs), incident light direction (2 DOFs), and viewing direction (2 DOFs). (B) The subject is imaged from multiple viewpoints when lit by different directional lights; a geometry proxy is also captured using active sensors. (C) Querying the learned function at different light and/or viewing directions enables simultaneous relighting and view synthesis of this subject. (D) The relit renderings that NLT produces can be combined according to HDRI maps to perform image-based relighting.

The light transport (LT) of a scene describes how it appears under different lighting conditions from different viewing directions, and complete knowledge of a scene's LT enables the synthesis of novel views under arbitrary lighting. In this paper, we focus on image-based LT acquisition, primarily for human bodies within a light stage setup. We propose a semi-parametric approach for learning a neural representation of the LT that is embedded in a texture atlas of known but possibly rough geometry. We model all non-diffuse and global LT as residuals added to a physically-based diffuse base rendering. In particular, we show how to fuse previously seen observations of illuminants and views to synthesize a new image of the same scene under a desired lighting condition from a chosen viewpoint. This strategy allows the network to learn complex material effects (such as subsurface scattering) and global illumination (such as diffuse interreflection), while guaranteeing the physical correctness of the diffuse LT (such as hard shadows). With this learned LT, one can relight the scene photorealistically with a directional light or an HDRI map, synthesize novel views with view-dependent effects, or do both simultaneously, all in a unified framework using a set of sparse observations. Qualitative and quantitative experiments demonstrate that our Neural Light Transport (NLT) outperforms state-of-the-art solutions for relighting and view synthesis, without requiring separate treatments for both problems that prior work requires. The code and data are available at http://nlt.csail.mit.edu.

ACM Reference Format:
Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, Jonathan T. Barron, Ravi Ramamoorthi, and William T. Freeman. 2020. Neural Light Transport for Relighting and View Synthesis. *ACM Trans. Graph.* 40, 1 (December 2020), 16 pages. https://doi.org/10.1145/3446328

Authors' addresses: Xiuming Zhang, xiuming@csail.mit.edu, Massachusetts Institute of Technology; Sean Fanello; Yun-Ta Tsai, Google; Tiancheng Sun, University of California, San Diego; Tianfan Xue; Rohit Pandey; Sergio Orts-Escolano; Philip Davidson; Christoph Rhemann; Paul Debevec; Jonathan T. Barron, Google; Ravi Ramamoorthi, University of California, San Diego; William T. Freeman, billf@mit.edu, Massachusetts Institute of Technology & Google.

0730-0301/2020/12-ART $15.00
https://doi.org/10.1145/3446328

1 INTRODUCTION

The light transport (LT) of a scene models how light interacts with objects in the scene to produce an observed image. The process by which geometry and material properties of the scene interact with global illumination to result in an image is a complicated but well-understood consequence of physics [Pharr et al. 2016]. Much progress in computer graphics has been through the development of more expressive and more efficient mappings from a scene model (geometry, materials, and lighting) to an image. In contrast, *inverting* this process is ill-posed and therefore more difficult: acquiring

the LT of a scene from images of that scene requires untangling the myriad interconnected effects of occlusion, shading, shadowing, interreflections, scattering, etc. Solving this task of inferring aspects of LT from images is an active research area, and even partial solutions have significant practical uses such as phototourism [Snavely et al. 2006], telepresence [Orts-Escolano et al. 2016], storytelling [Kelly et al. 2019], and special effects [Debevec 2012]. A less obvious, but equally important application of inferring LT from images consists of generating groundtruth data for machine learning tasks: many works rely on high-quality renderings of relit subjects under arbitrary lighting conditions and from multiple viewpoints to perform relighting [Meka et al. 2019; Sun et al. 2019], view synthesis [Pandey et al. 2019], re-enacting [Kim et al. 2018], and alpha matting [Sengupta et al. 2020].

Previous work has shown that it is possible to construct a light stage [Debevec et al. 2000], plenoptic camera [Levoy and Hanrahan 1996], or gantry [Murray-Coleman and Smith 1990] that directly captures a subset of the LT function and thereby enables the image-based rendering thereof. These techniques are widely used in film productions and within the research community. However, these systems can only provide *sparse* sampling of the LT limited to the number of LEDs (∼300 on a spherical dome) and the number of cameras (∼50-100 around the subject), resulting in the inability to produce photorealistic renderings outside the supported camera/light locations. Indeed, traditional image-based rendering approaches are usually designed for fixed viewpoints and are unable to synthesize unseen (novel) views under a desired illumination.

In this paper, we learn to interpolate the *dense* LT function of a given scene from *sparse* multi-view, One-Light-at-A-Time (OLAT) images acquired in a light stage [Debevec et al. 2000], through a semi-parametric technique that we dub Neural Light Transport (NLT) (Figure 1). Many prior works have addressed similar tasks (as will be discussed in Section 2), with classic works tending to rely on physics to recover analytical and interpretable models, and recent works using neural networks to infer a more direct mapping from input images to an output image.

Traditional rendering methods often make simplifying assumptions when modeling geometry, BRDFs, or complex inter-object interactions in order to make the problem tractable. On the other hand, deep learning approaches can tolerate geometric and reflectance imperfections, but they often require many aspects of image formation (even those guaranteed by physics) be learned "from scratch," which may necessitate a prohibitively large training set. NLT is intended to straddle this divide: we construct a classical model of the subject being imaged (a mesh and a diffuse texture atlas per Lambertian reflectance), but then we embed a neural network within the parameterization provided by that classical model, construct the inputs and outputs of the model in ways that leverage domain knowledge of classical graphics techniques, and train that network to model all aspects of LT—including those not captured by a classical model. By leveraging a classical model this way, NLT is able to learn an accurate model of the complicated LT function for a subject from a small training dataset of sparse observations.

A key novelty of NLT is that our learned model is embedded within the texture atlas space of an existing geometric model of the subject, which provides a novel framework for *simultaneous* relighting and view interpolation. We express the 6D LT function (Figure 1) at each location on the surface of our geometric model as simply the output of a deep neural network, which works well (as neural networks are smooth and universal function approximators [Hornik 1991]) and obviates the need for a complicated parameterization of spatially-varying reflectance. We evaluate on joint relighting and view synthesis using sparse image observations of scanned human subjects within a light stage, and show state-of-the-art results as well as compelling practical applications.

In summary, our main contributions are:

- An end-to-end, semi-parametric method for learning to interpolate the 6D light transport function per-subject from real data using convolutional neural networks (Section 3.3);
- A unified framework for simultaneous relighting and view synthesis by embedding networks into a parameterized texture atlas and leveraging as input a set of One-Light-at-A-Time (OLAT) images (Section 3.5);
- A set of augmented texture-space inputs and a residual learning scheme on top of a physically accurate diffuse base, which together allow the network to easily learn non-diffuse, higher-order light transport effects including specular highlights, subsurface scattering, and global illumination (Section 3.2 and Section 3.4).

The proposed method allows for photorealistic free-viewpoint rendering under controllable lighting conditions, which not only is a key aspect in compelling user experiences in mixed reality and special effects, but can be applied to a variety of machine learning tasks that rely on photorealistic groundtruth data.

2 RELATED WORK

Our method addresses the problem of recovering a model of light transport from a sparse set of images of some subject, and then predicting novel images of that subject from unseen views and/or under unobserved illuminations. This is a broad problem statement that relates to and subsumes many tasks in graphics and vision.

Single observation. The most sparse sampling is just a single image, from which one could attempt to infer a model (geometry, reflectance, and illumination) of the physical world that resulted in that image [Barrow and Tenenbaum 1978], usually via handcrafted [Barron and Malik 2015; Li et al. 2020] or learned priors [Saxena et al. 2008; Eigen et al. 2014; Sengupta et al. 2018; Li et al. 2018; Kanamori and Endo 2018; Kim et al. 2018; Gardner et al. 2019; LeGendre et al. 2019; Alldieck et al. 2019; Wiles et al. 2020; Zhang et al. 2020]. Though practical, the quality gap between what can be accomplished by single-image techniques and what has been demonstrated by multi-image techniques is significant. Indeed, none of these methods shows complex light transport effects such as specular highlights or subsurface scattering [Kanamori and Endo 2018; Kim et al. 2018]. Moreover, these methods are usually limited to a single task, such as relighting [Kanamori and Endo 2018; Kim et al. 2018; Sengupta et al. 2018] or viewpoint change [Alldieck et al. 2019; Wiles et al. 2020; Li et al. 2020], and some support only a limited range of viewpoint change [Kim et al. 2018; Tewari et al. 2020a].

Multiple views. Multiview geometry techniques recover a textured 3D model that can be rendered using conventional graphics or

photogrammetry techniques [Hartley and Zisserman 2003], but have material and shading variation baked in, and do not enable relighting. Image-based rendering techniques such as light fields [Levoy and Hanrahan 1996] or lumigraphs [Gortler et al. 1996] can be used to directly sample and render the plenoptic function [Adelson and Bergen 1991], but the accuracy of these techniques is limited by the density of sampled input images. For unstructured inputs, reprojection-based methods [Buehler et al. 2001; Eisemann et al. 2008] assume the availability of a geometry proxy (so does our work), reproject nearby views to the query view, and perform image blending in that view. However, such methods rely heavily on the quality of the geometry proxy and cannot synthesize pixels that are not visible in the input views. A class-specific geometry prior (such as that of a human body [Shysheya et al. 2019]) can be used to increase the accuracy of a geometry proxy [Carranza et al. 2003], but none of these methods enables relighting.

Recently, deep learning techniques have been used to synthesize new images from sparse sets of input images, usually by training neural networks to synthesize some intermediate geometric representation that is then projected into the desired image [Zhou et al. 2018; Sitzmann et al. 2019a,b; Srinivasan et al. 2019; Flynn et al. 2019; Mildenhall et al. 2019, 2020; Thies et al. 2020]. Some techniques even entirely replace the rendering process with a learned "neural" renderer [Thies et al. 2019; Martin-Brualla et al. 2018; Pandey et al. 2019; Lombardi et al. 2019, 2018; Tewari et al. 2020b]. Despite effective, these methods generally do not attempt to explicitly model light transport and hence do not enable relighting—though they are often capable of preserving view-dependent effects for the fixed illumination condition under which the input images were acquired [Thies et al. 2019; Mildenhall et al. 2020]. Additionally, neural rendering often breaks "backwards compatibility" with existing graphics systems, while our approach infers images directly in texture space that can be re-sampled by conventional graphics software (e.g., Unity, Blender, etc.) to synthesize novel viewpoints. Recently, Chen et al. [2020] propose to learn relightable view synthesis from dense views (200 vs. 55 in this work) under image-based lighting; using spherical harmonics as the lighting representation, the work is unable to produce hard shadow caused by a directional light as in this work.

Multiple illuminants. Similar to the multi-view task is the task of photometric stereo [Woodham 1980; Basri et al. 2007] (as cameras function analogously to illuminants in some contexts [Sen et al. 2005]): repeatedly imaging a subject with a fixed camera but under different illuminations and then recovering the surface normals. However, most photometric stereo solutions assume Lambertian reflectance and do not support relighting with non-diffuse light transport. More recently, Ren et al. [2015], Meka et al. [2019], Sun et al. [2019], and Sun et al. [2020] show that neural networks can be applied to relight a scene captured under multiple lighting conditions from a fixed viewpoint. Nestmeyer et al. [2020] decompose an image into shaded albedo (so no cast shadow) and residuals, unlike this work that models cast shadow as part of a physically accurate diffuse base. None of these works supports view synthesis. Xu et al. [2019] perform free-viewpoint relighting, but unlike our approach, they require running the model of Xu et al. [2018] as a second stage.

Multiple views and illuminants. Garg et al. [2006] utilize the symmetry of illuminations and view directions to collect sparse samples of an 8D reflectance field, and reconstruct a complete field using a low-rank assumption. Perhaps the most effective approach for addressing sparsity in light transport estimation is to circumvent this problem entirely, and densely sample whatever is needed to produce the desired renderings. The landmark work of Debevec et al. [2000] uses a light stage to acquire the full reflectance field of a subject by capturing a One-Light-at-A-Time (OLAT) scan of that subject, which can be used to relight the subject by linear combination according to some High-Dynamic-Range Imaging (HDRI) environment map. Despite its excellent results, this approach lacks an explicit geometric model, so rendering is limited to a fixed set of viewpoints. Although this limitation has been partially addressed by the work of Ma et al. [2007] that focuses on facial capture, a recent system of Guo et al. [2019] builds a full volumetric relightable model using two spherical gradient illumination conditions [Fyffe et al. 2009]. This system supports relighting and view synthesis, but assumes pre-defined BRDFs and therefore cannot synthesize more complex light transport effects present in real images. Zickler et al. [2006] also pose the problem of appearance synthesis as that of high-dimensional interpolation, but they use radial basis functions on smaller-scale data.

Our work follows the convention of the nascent field of "neural rendering" [Thies et al. 2019; Lombardi et al. 2019, 2018; Sitzmann et al. 2019a; Tewari et al. 2020b; Mildenhall et al. 2020], in which a separate neural network is trained for each subject to be rendered, and all images of that subject are treated as "training data." These approaches have shown great promise in terms of their rendering fidelity, but they require per-subject training and are unable to generalize across subjects yet. Unlike prior work that focuses on a specific task (e.g., relighting or view synthesis), our texture-space formulation allows for simultaneous light and view interpolation. Furthermore, our model is a valuable training data generator for many works that rely on high-quality renderings of subjects under arbitrary lighting conditions and from multiple viewpoints, such as [Meka et al. 2019; Sun et al. 2019; Pandey et al. 2019; Kim et al. 2018; Sengupta et al. 2020].

3 METHOD

Our framework is a semi-parametric model with a residual learning scheme that aims to close the gap between the diffuse rendering of the geometry proxy and the real input image (Figure 2). The semi-parametric approach is used to fuse previously recorded observations to synthesize a novel, photorealistic image under any desired illumination and viewpoint.

The method relies on recent advances in computer vision that have enabled accurate 3D reconstructions of human subjects, such as the technique of Collet et al. [2015] which takes as input several images of a subject and produces as output a mesh of that subject and a UV texture map describing its albedo. At first glance, this appears to address the entirety of our problem: given a textured mesh, we can perform simultaneous view synthesis and relighting by simply re-rendering that mesh from some arbitrary camera location and under some arbitrary illumination. However, this simplistic

Fig. 2. **Gap in photorealism between a traditional rendering and a real image.** Even when high-quality geometry and albedo can be captured (e.g., by Relightables [Guo et al. 2019]), photorealistic rendering remains challenging, because any geometric inaccuracy will show up as visual artifacts (e.g., black rims/holes in the hair), and manually creating spatially-varying, photorealistic materials is onerous, if possible at all. NLT aims to close this gap by learning directly from real images the residuals that account for geometric inaccuracies and non-diffuse LT, such as global illumination.

model of reflectance and illumination only permits equally simplistic relighting and view synthesis, assuming Lambertian reflectance:

$$\tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_o) = \rho(\mathrm{x}) L_i(\mathrm{x}, \boldsymbol{\omega}_i)\,(\boldsymbol{\omega}_i \cdot \mathrm{n}(\mathrm{x}))\,. \tag{1}$$

Here $\tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_o)$ is the diffuse rendering of a point x with a surface normal $\mathrm{n}(\mathrm{x})$ and albedo $\rho(\mathrm{x})$, lit by a directional light $\boldsymbol{\omega}_i$ with an incoming intensity $L_i(\mathrm{x}, \boldsymbol{\omega}_i)$ and viewed from $\boldsymbol{\omega}_o$. This reflectance model is only sufficient for describing matte surfaces and direct illumination. More recent methods (such as Relightables [Guo et al. 2019]) also make strong assumptions about materials by modeling reflectance with a cosine lobe model. The shortcomings of these methods are obvious when compared to a more expressive rendering approach, such as the rendering equation [Kajiya 1986], which makes far fewer simplifying assumptions:

$$L_o(\mathrm{x}, \boldsymbol{\omega}_o) = L_e(\mathrm{x}, \boldsymbol{\omega}_o) + \int_\Omega f_s(\mathrm{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\, L_i(\mathrm{x}, \boldsymbol{\omega}_i)\,(\boldsymbol{\omega}_i \cdot \mathrm{n}(\mathrm{x}))\,\mathrm{d}\,\boldsymbol{\omega}_i. \tag{2}$$

From this we observe the many limitations in computing $\tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_o)$: it assumes a single directional light instead of integrating over the hemisphere of all incident directions Ω, it approximates an object's BRDF $f_s(\cdot)$ as a single scalar, and it ignores emitted radiance $L_e(\cdot)$ (in addition to scattering and transmittance, which this rendering equation does not model either). The goal of our learning-based model is to close the gap between $L_o(\mathrm{x}, \boldsymbol{\omega}_o)$ and $\tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_o)$, and furthermore between $L_o(\mathrm{x}, \boldsymbol{\omega}_o)$ and the observed image.

Though not perfect for relighting, the geometry and texture atlas provided by Guo et al. [2019] offers us a mapping from each image of a subject onto a canonical texture atlas that is shared across all views of that subject. This motivates the high-level approach of our model: we use this information to map the input images of the subject from "camera space" (XY pixel coordinates) to "texture space" (UV texture atlas coordinates), then use a semi-parametric neural network *embedded* in this texture space to fuse multiple observations and synthesize an RGB texture atlas for the desired relit and/or novel-view image. This is then warped back into the camera space of the desired viewpoint, thereby giving us an output rendering of the subject under the desired illumination and viewpoint.

In Section 3.1 and Section 3.2, we describe our data acquisition setup and the input data to our framework. In Section 3.3, we detail the texture-space two-path neural network architecture at the core of our model, which consists of: 1) "observation paths" that take as input a set of observed RGB images that have been warped into the texture space and produce a set of intermediate neural activations, and 2) a "query path" that uses these activations to synthesize a texture-space rendering of the subject according to some desired light and/or viewing direction.

The texture-space inputs encode a rudimentary geometric understanding of the scene and correspond to the arguments of the 6D LT function (i.e., UV location on the 3D surface x, incident light direction $\boldsymbol{\omega}_i$, and viewing direction $\boldsymbol{\omega}_o$). By using a skip-link between the query path's diffuse base image and its output as described in Section 3.4, our model is encouraged to learn a residual between the provided Lambertian rendering with geometric artifacts and the real-world appearance, which not only guarantees the physical correctness of the diffuse LT, but also directs the network's attention towards learning higher-order, non-diffuse LT effects. In Section 3.6, we explain how our model is trained end-to-end to minimize a photometric loss and a perceptual loss in the camera space. Our model is visualized in Figure 4.

3.1 Hardware Setup and Data Acquisition

Our method relies on directional light images (One-Light-at-A-Time [OLAT] images) in the form of texture-space UV buffers. This requires us to acquire training images under known illumination conditions alongside a parameterized geometric model to obtain the UV buffers. We use a light stage similar to Sun et al. [2019] to acquire OLAT images where only one (known) directional light source is active in each image. For each session we captured 331 OLAT images for 64 RGB cameras placed around the performer. When a light is pointing towards a given camera or gets blocked by the subject, the resultant image is either "polluted" by the glare or is overly dark. As such, for a given camera, there are approximately 130 usable OLAT images. These OLAT images are sparse samples of the 6D light transport function, which NLT learns to interpolate. We visualize samples of these OLAT images in Figure 3.

Following previous works [Meka et al. 2019; Sun et al. 2019], we ask the subject to stay still during the acquisition phase, which lasts ~6 seconds for a full OLAT sequence. Since it is nearly impossible for the performer to stay perfectly still, we align all the images using the optical flow technique of Meka et al. [2019]: we capture "all-lights-on" images throughout the scan that are used as "tracking frames," and compute 2D flow fields between each tracking frame and a reference tracking frame taken from the middle of the sequence. These flow fields are then interpolated from the tracking frames to the rest of the images to produce a complete alignment.

Following the approach of Guo et al. [2019], we use 32 high-resolution active IR cameras and 16 custom dot illuminator projectors to construct a high-quality parameterized base mesh of each subject fully automatically. These data are critical to our approach, as the estimated geometry provided by this system provides the substrate that our learned model is embedded within in the form of a texture atlas. However, this captured 3D model is far from perfect

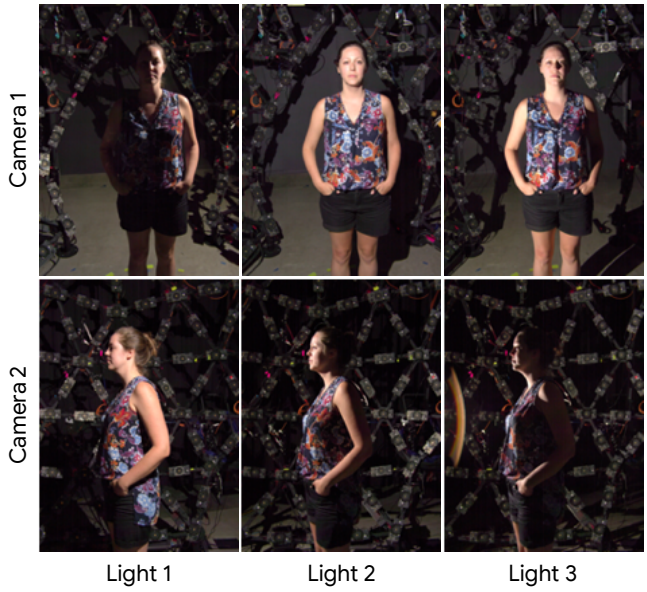


Fig. 3. **Sample images used for model training.** We train our model with multi-view, One-Light-at-A-Time (OLAT) images, in each of which only one (known) directional light is active at a time. A *proxy* of the underlying geometry is also required by NLT, but it can be as rough as 500 vertices (see Section 4.3). These images are sparse samples of the 6D light transport function that NLT learns to interpolate.

due to approximations in the mesh model (that cannot accurately model fine structures such as hair) and hand-crafted priors in the reflectance estimation (that relies on a cosine lobe BRDF model). This is demonstrated in Figure 2. Our model overcomes these issues and enables photorealistic renderings, as demonstrated in Section 4. Additionally, we demonstrate in Section 4.3 that our neural rendering approach is robust to geometric degradation and can work with geometry proxies of as few as 500 vertices.

We collect a dataset of 70 human subjects with fixed poses, each of which provides ∼18,000 frames under 331 lighting conditions and 55 viewpoints (before filtering out glare-polluted and overly dark frames, as aforementioned). We randomly hold out 6 lighting conditions and 2 viewpoints for training. Subjects are selected to maximize diversity in terms of clothing, skin color, and age. By training our model to reproduce held-out images from these light stage scans, it is able to learn a general LT function that can be used to produce renderings for arbitrary viewpoints and illuminations. Because our scans do not share the same UV parameterization, we train a separate model for each subject.

3.2 Texture-Space Inputs

In order to perform light and view interpolation, we use as input to our model a set of OLAT images, the subject's diffuse base, and the dot products of the surface normals with the desired or observed viewing directions or light directions (a.k.a. "cosine maps"), all in the UV texture space. This augmented input allows our learned model to leverage insights provided by classic graphics models, as the dot products between the normals and the viewing or lighting directions are the standard primitives in parametric reflectance models (Equations 1, 2, etc.).

These augmented texture-space input buffers superficially resemble the "G-buffers" used in deferred shading models [Deering et al. 1988] and used with neural networks in Deep Shading [Nalbach et al. 2017]. But unlike Nalbach et al. [2017], our goal is to train a model for view and light interpolation using *real images*, instead of renderings from a CG model. This different goal motivates additional novelties of our approach, such as the embedding of our model in UV space (which removes the dependency on viewpoints and implicitly provides aligned correspondence across multiple views) and our use of a residual learning scheme (to encourage training to focus on higher-order LT effects). Li et al. [2019] also successfully employ deep learning in the texture space and regress PRT coefficients for deformable objects, but they learn only predefined diffuse and glossy light transport from synthetic rendering. We use three types of buffers in our model, as described below.

Cosine map. Assuming directional light sources, we calculate the cosine map of a light as the dot product between the light's direction $\boldsymbol{\omega}$ and the surface's normal vector $\mathrm{n}(\mathrm{x})$. For each view and light (both observed and queried), we compute two cosine maps: a view cosine map $\mathrm{n}(\mathrm{x}) \cdot \boldsymbol{\omega_o}$ and a light cosine map $\mathrm{n}(\mathrm{x}) \cdot \boldsymbol{\omega_i}$. Crucially, these maps are masked by visibility computed via ray casting from each camera onto the geometry proxy, such that the light cosines also provide rough understanding of cast shadows (texels with zero visibility; see Figure 4), leaving the network an easier task of adding, e.g., global illumination effects to these hard shadows.

Diffuse base. The diffuse base is obtained by summing up all OLAT images for each view or equivalently, illuminating the subject from all directions simultaneously (because light is additive). These multiple views are then averaged together in the texture space, which mitigates the view-dependent effects and produces a texture map that resembles albedo. Note that multiplying the diffuse base by a light cosine map produces the diffuse rendering (with hard cast shadows) for that light $\tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_i)$. The construction of this diffuse base is visualized in the bottom middle of Figure 4.

Residual map. We compute the difference between each observed OLAT image and the aforementioned diffuse base, thereby capturing the "non-diffuse and non-local" residual content of each input image. These residual maps are available only for the *sparsely* captured OLAT from fixed viewpoints. To synthesize a novel view for any desired lighting condition, our network uses a semi-parametric approach that interpolates previously seen observations and their residual maps, generating the final rendering.

3.3 Query and Observation Networks

Our semi-parametric approach is shown in Figure 4: the network takes as input multiple UV buffers in two distinct branches, namely a "query path" and "observation paths." The query path takes as input a set of texture maps that can be generated from the captured geometry, i.e., view/light cosine maps and a diffuse base. The observation paths represent the semi-parametric nature of our framework and have access to non-diffuse residuals of the captured OLAT images. The two branches are merged in an end-to-end fashion to synthesize an unseen lighting condition from any desired viewpoint.

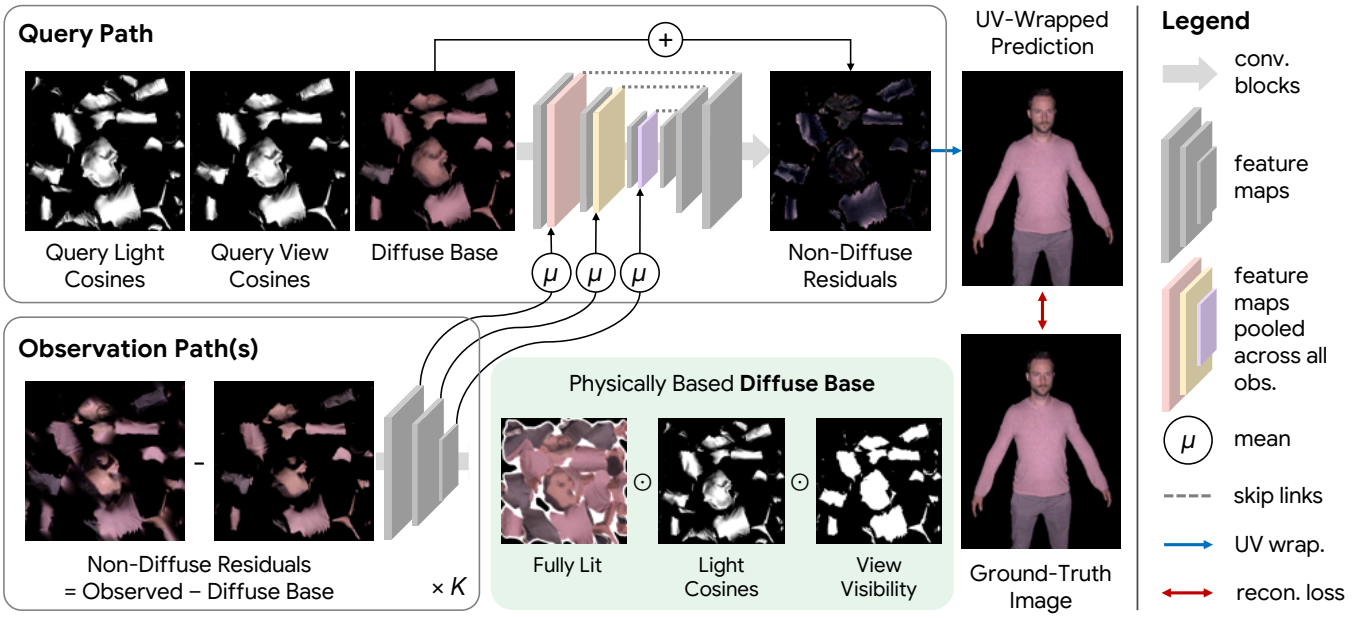


Fig. 4. **Model.** Our network consists of two paths. The "observation paths" take as input K nearby observations (as texture-space residual maps) sampled around the target light and viewing directions, and encode them into multiscale features that are pooled to remove the dependence on their order and number. These pooled features are then concatenated to the feature activations of the "query path," which takes as input the desired light and viewing directions (in the form of cosine maps) as well as the physically accurate diffuse base (also in the texture space). This path predicts a residual map that is added to the diffuse base to produce the texture rendering. With the (differentiable) UV wrapping pre-defined by the geometry proxy, we then resample the texture-space rendering back into the camera space, where the prediction is compared against the ground-truth image. Because the entire network is embedded in the texture space of a subject, the same model can be trained to perform relighting, view synthesis, or both simultaneously, depending on the input and supervision.

To synthesize a new image of the subject under a desired lighting and viewpoint, we have access to potentially all the OLAT images from multiple viewpoints. The goal of the observation paths is to combine these images and extract meaningful features that are passed to the query path to perform the final rendering. However, using all these observations as input is not practical during training due to memory and computational limits. Therefore, for a desired novel view and light condition, we randomly select only $K = 1$ or 3 OLAT images from the "neighborhood" as observations (the precise meaning of "neighborhood" will be clarified in Section 3.6). The random sampling prevents the network from "cheating" by memorizing fixed neighbors-to-query mappings and encourages it to learn that for a given query, different observation selections should lead to the same prediction (also observed by Sun et al. [2020]).

These observed images (in the form of UV-space residual maps as shown in Figure 4) are then fed in parallel (i.e., processed as a "batch") into the observation paths of our network, which can alternatively be thought of as K distinct networks that all share the same weights. The resulting set of K network activations are then averaged across the set of images by taking their arithmetic mean*, thereby becoming invariant to their cardinality and order, and are then passed to the query path.

While the goal of the observation paths is to process input images and glean reflectance information from them, the goal of the query path is to take as input information that encodes the non-diffuse residuals of nearby lights/views and then predict radiance values of the queried light and view positions at each UV location. We therefore concatenate the aggregated activations from the observation paths to the self-produced activations of the query path using a set of cross-path skip-connections. The query path then decodes a texture-space rendering of the subject under the desired light and viewing directions, which is then resampled to the perspective of the desired viewpoint using conventional, differentiable UV wrapping.

Our proposed architecture has several advantages over a single-path network that would take as input all the available observations, which would be prohibitively expensive in terms of memory and computation. Because our observation paths do not depend on a fixed order or number of images, during training, we can simply select a dynamic subset of whatever observations that are best suited to the desired lighting and viewpoint. This ability is useful because the lights and cameras in our dataset are sampled at different rates—lights are ~4× denser than cameras. The superiority of this dual-path design is demonstrated by both qualitative and quantitative experiments in Section 4.4.

3.4 Residual Learning of High-Order Effects

When synthesizing the output texture-space image in the query path of our network, we do not predict the final image directly. Instead, we have a residual skip-link [He et al. 2016] from the input diffuse base to the output of our network. Formally, we train

*In practice, we observe no improvement when we replace the uniform weights with the barycentric coordinates of the query w.r.t. its $K = 3$ observations.

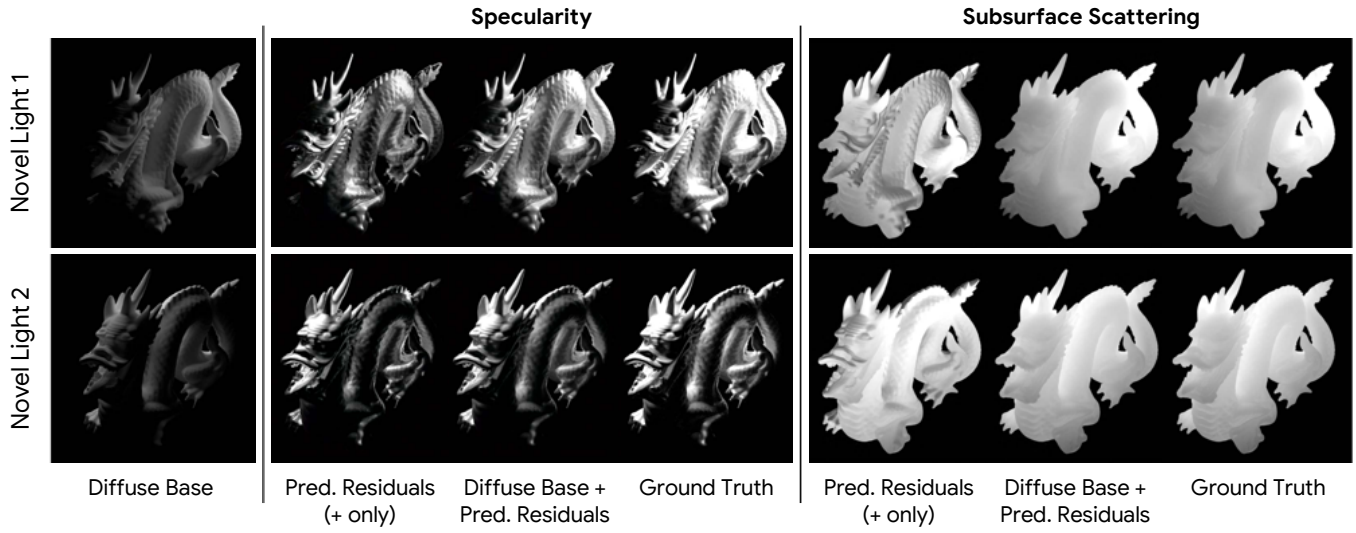


Fig. 5. **Modeling non-diffuse BSSRDFs as residuals in relighting.** A diffuse base (left) captures all diffuse LT (e.g., hard shadows) under a novel point light. By learning a residual on top of this base rendering, NLT can reproduce non-diffuse LT (here, specularities and subsurface scattering) from the actual scene appearance. When predicting specularities (center), the NLT emits exclusively positive residuals (negative part hence not shown) to add bright highlights to the diffuse base. When predicting scattering (right), the additive residuals represent additional illumination provided by nearby subsurface light transport.

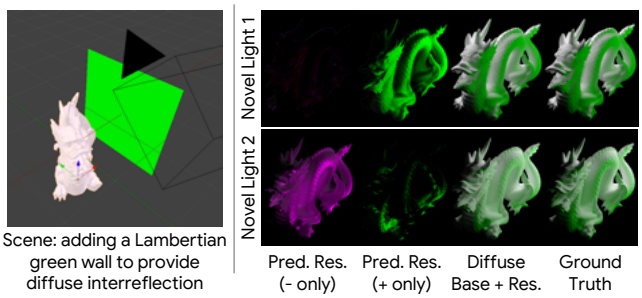


Fig. 6. **Modeling global illumination as residuals in relighting** (same diffuse bases as in Figure 5). In addition to intrinsic material properties, NLT can also learn to express global illumination (e.g., diffuse interreflection) as residuals. Here we add a diffuse green wall to the right of the scene (left). Under Novel Light 1 (right top), the wall provides additional green indirect illumination, so the residuals are green and mostly positive. Notably, the residuals are not necessarily all positive: under Novel Light 2 (right bottom), the residuals are mostly negative and high in blue and red, effectively casting "negative purple" indirect illumination that results in a greenish tinge.

our deep neural network to synthesize a residual ΔL that is then added to our diffuse base $\tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_o)$ to produce our final predicted rendering $L_o(\mathrm{x}, \boldsymbol{\omega}_o) = \Delta L + \tilde{L}_o(\mathrm{x}, \boldsymbol{\omega}_o)$. This approach of adding a physically-based diffuse rendering allows our network to focus on learning higher-order, non-diffuse, non-local light transport effects (specularities, scattering, etc.) instead of having to "re-learn" the fundamentals of image formation (basic colors, rough locations and shapes of cast shadows, etc.). Because these residuals are the unconstrained output of a network, this model is able to describe any output image: positive residuals can be added to represent specularities, and negative residuals can be added to represent shading or shadowing. This residual approach causes our model to be implicitly regularized towards a simplified but physically-plausible diffuse model – the network can "fall back" to the diffuse base rendering by simply emitting zeros.

We demonstrate that our method is capable of modeling complicated lighting effects including specular highlights (BRDFs), subsurface scattering (BSSRDFs), and diffuse interreflection (global illumination), in the context of relighting a toy dragon scene. We consider a 3D model with perfect geometry and known material properties and render it in a virtual scene similar to a light stage setup using Cycles (Blender's built-in, physically-based renderer). We produce a diffuse render of the scene as a baseline, and then re-render it using both our model and Blender with three lighting effects: specular highlights, subsurface scattering, and diffuse interreflections, to demonstrate that NLT is capable of modeling those effects. The results are shown in Figure 5 and Figure 6.

Specular highlights. In Blender, we mix a glossy shader into the dragon's diffuse shader and re-render the scene, resulting in a render with highlights. We then train our model to infer these residuals for relighting. In Figure 5 (center), we show the NLT renderings under two novel light directions (unseen during training) alongside with the ground-truth renderings. The residual image predicted by our model correctly models the specular highlights, and our rendering closely resembles the ground truth.

Subsurface scattering. Our model can capture lighting effects that cannot be captured by a BRDF, such as subsurface scattering. We mix a subsurface scattering shader into the dragon's diffuse shader, and then train our model to learn these effects in relighting. As shown in Figure 5 (right), the NLT results are almost identical to the ground truth.

Diffuse interreflection. To demonstrate global illumination, in Figure 6 we place a matte green wall into the scene, and we see that

NLT is able to accurately predict the non-local light transport of a green glow cast by the wall onto the dragon.

3.5 Simultaneous Relighting and View Synthesis

Embedded in the texture space, NLT is a unified framework that can perform relighting, view synthesis, or both simultaneously. The architecture described in Section 3.3 takes as input the cosine maps that encode the light and viewing directions, as well as a set of observed residual maps from nearby lights and/or views (neighbor selection scheme in Section 3.6). Since there is no model design specific to relighting or view synthesis, the model is agnostic to which task it is solving other than interpolating the 6D LT function. Therefore, by varying both lights and views in the training data, the model can be trained to render the subject under any desired illumination from any camera position (a.k.a. simultaneously relighting and view synthesis). We demonstrate this capability in Section 4.2 and the supplemental video.

3.6 Network Architecture, Losses, and Other Details

Both paths of our architecture are modifications of the U-Net [Ronneberger et al. 2015], where our query path is a complete encoder-decoder architecture (with skip-connections) that decodes the final predicted image, while our observation path is just an encoder. Following standard conventions, each scale of the network consists of two convolutional layers (except at the very start and end), where downsampling and upsampling are performed using strided (possibly transposed) convolutions, and the channel number of the feature maps is doubled after each downsampling and halved after each upsampling. Detailed descriptions of the architectures of these two networks are provided in Tables 1 and 2. $\mathrm{conv}(d, w \times h, s)$ denotes a two-dimensional convolutional layer (a.k.a. "conv2D") with d output channels, a filter size of $(w \times h)$, and a stride of s, and is always followed by a leaky ReLU [Maas et al. 2013] activation function. $\mathrm{conv}^{\mathrm{T}}$is the transpose of conv and is also followed by a leaky ReLU. No normalization is used. Note that the activations from the observation paths are appended to the query path before its internal skip connections, meaning that observation activations are effectively skip-connected to the decoder of the query network.

We trained our model to minimize losses in the image space between the predicted image $L_o(\mathrm{x}_i, \boldsymbol{\omega}_o)$ and the ground-truth captured image. To this end, we first resample the UV-space prediction back to the camera space, and then compute the total loss as a combination of a robust photometric loss [Barron 2019] and a perceptual loss (LPIPS) [Zhang et al. 2018].

We use the loss function of Barron [2019] with $\alpha = 1$ (also known as pseudo-Huber loss) applied to a CDF9/7 wavelet decomposition [Cohen et al. 1992] in the YUV color space:

$$\mathcal{L}_{\mathcal{I}} = \sum_i \sqrt{\left(\frac{\mathrm{CDF}\left(\mathrm{YUV}\left(L_o(\mathrm{x}_i, \boldsymbol{\omega}_o) - L_o^*(\mathrm{x}_i, \boldsymbol{\omega}_o)\right)\right)}{c}\right)^2 + 1} - 1. \tag{3}$$

We empirically set the scale hyperparameter $c = 0.01$. As was demonstrated in Barron [2019], we found that imposing a robust loss in this YUV wavelet domain produced reconstructions that better captured both high-frequency and low-frequency details.

Table 1. The neural network architecture of our "observation path."

ID	Operator	Output Shape		
O1	conv(16, 1×1, 1)	H	W	16
O2	conv(16, 3×3, 2)	$H/2$	$W/2$	16
O3	conv(16, 3×3, 1)	$H/2$	$W/2$	16
O4	conv(32, 3×3, 2)	$H/4$	$W/4$	32
O5	conv(32, 3×3, 1)	$H/4$	$W/4$	32
...	...	...		
O14	conv(1024, 3×3, 2)	$H/128$	$W/128$	1024
O15	conv(1024, 3×3, 1)	$H/128$	$W/128$	1024
O16	conv(2048, 3×3, 2)	$H/256$	$W/256$	2048
O17	conv(2048, 3×3, 1)	$H/256$	$W/256$	2048

Table 2. The neural network architecture of our "query path." The layers that reflect skip connections from the activations of our observation path in Table 1 are highlighted in blue, and U-Net-like skip-links within the query path are highlighted in green.

ID	Operator	Output Shape		
Q1	conv(16, 1×1, 1)	H	W	16
Q2	append (mean(O1))	H	W	32
Q3	conv(16, 3×3, 2)	$H/2$	$W/2$	16
Q4	conv(16, 3×3, 1)	$H/2$	$W/2$	16
Q5	append (mean(O3))	$H/2$	$W/2$	32
Q6	conv(32, 3×3, 2)	$H/4$	$W/4$	32
Q7	conv(32, 3×3, 1)	$H/4$	$W/4$	32
Q8	append (mean(O5))	$H/4$	$W/4$	64
...	...	...		
Q44	append (Q8)	$H/4$	$W/4$	80
Q45	$\mathrm{conv}^{\mathrm{T}}$(8, 3×3, 2)	$H/2$	$W/2$	8
Q46	$\mathrm{conv}^{\mathrm{T}}$(8, 3×3, 1)	$H/2$	$W/2$	8
Q47	append (Q5)	$H/2$	$W/2$	40
Q48	$\mathrm{conv}^{\mathrm{T}}$(4, 3×3, 2)	H	W	4
Q49	$\mathrm{conv}^{\mathrm{T}}$(4, 3×3, 1)	H	W	4
Q50	append (Q2)	H	W	36
Q51	$\mathrm{conv}^{\mathrm{T}}$(3, 1×1, 1)	H	W	3

The perceptual loss $\mathcal{L}_{\mathcal{P}}$ [Zhang et al. 2018] is defined as the ℓ_2 distance in feature space extracted with a VGG network [Simonyan and Zisserman 2014] pre-trained on ImageNet [Deng et al. 2009]. The final loss function is simply the sum of the two losses $\mathcal{L} = \mathcal{L}_{\mathcal{I}} + \mathcal{L}_{\mathcal{P}}$. Empirically, we found that using the same weight for both losses achieved the best results.

We trained our model by minimizing $\mathcal{L}$ using Adam [Kingma and Ba 2015] with a learning rate of 2.5×10^{-4}, a batch size of 1, and the following optimizer hyperparameters: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$. Our model is implemented in TensorFlow [Abadi et al. 2016] and trained on a single Nvidia Tesla P100, which takes less than 12 hours for the real scenes and much less for synthetic scenes.

Observations. Our observation paths are designed to be invariant to the number and order of observations, so that one can input as many observations into the path as the task requires or memory constraints permit. During training, the observations are the $K = 3$ neighbors randomly sampled from a cone around the query light direction with an apex angle of 30° for relighting, or from a pool of nearby cameras that have large visibility overlap in the UV space

(per-texel "and" operation) with the query camera for view synthesis. In training simultaneous models, we use $K = 1$ neighbor with both the nearest camera and light as the observation. At test time, thanks to the framework's invariance to the number and order of observations, we use a fixed set of observations to reduce flickering caused by frequent neighborhood switching, and observe no visual difference with different sets of observations. One may also use all of the physical lights and cameras as observations at test time.

Resolutions. For relighting and view synthesis, our texture-space images have a resolution of 1024×1024, and the camera-space images have a resolution of 1536×1128. For simultaneous relighting and view synthesis, the resolutions used are 512×512 in the UV space and 1024×752 in the camera space.

4 RESULTS

We perform multiple quantitative and qualitative experiments to demonstrate the efficacy of our model for relighting (Section 4.1) and view synthesis (Section 4.2), which are further clarified through performance analysis (Section 4.3) and ablation studies (Section 4.4). Additionally, we present qualitative results for HDRI relighting and for simultaneous view synthesis and relighting. All results are on unseen target light and viewing directions unless stated otherwise.

4.1 Relighting

We start our evaluation demonstrating that the method can synthesize new lighting conditions of the subject. Once we have learned the Neural Light Transport (NLT) of a given scene, we can relight the scene with a novel incident light direction $\boldsymbol{\omega}_i$ by simply querying our model.

Table 3. **Relighting.** NLT (or its variant) outperforms all baselines in terms of PSNR and LPIPS [Zhang et al. 2018]. Although barycentric blending achieves a similar SSIM score [Wang et al. 2004], it produces inaccurate renderings (such as ghosting shadows) upon visual inspection (Column 4 of Figure 7). Ablating the perceptual loss slightly increases PSNR, but degrades rendering quality by losing high-frequency details (such as the facial specularities shown in Figure 7). The numbers are means and 95% confidence intervals.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Diffuse Base	30.21 ± .08	.878 ± .003	.102 ± .003
Barycentric Blending	34.28 ± .20	.942 ± .002	.051 ± .002
Deep Shading [2017]	33.67 ± .27	.918 ± .009	.106 ± .012
Xu et al. [2018]	31.94 ± .09	.923 ± .003	.089 ± .003
Relightables [2019]	31.03 ± .08	.891 ± .003	.090 ± .003
NLT (ours)	33.99 ± .19	.942 ± .002	.045 ± .002
NLT w/o residuals	33.65 ± .25	.928 ± .006	.063 ± .007
NLT w/o observations	32.56 ± .15	.925 ± .002	.064 ± .002
NLT w/o LPIPS [2018]	34.43 ± .24	.939 ± .005	.066 ± .006

First, we quantitatively evaluate our model against the state-of-the-art relighting solutions and ablations of our model, and report our results in Table 3 in terms of Peak Signal-to-Noise Ratio (PSNR), SSIM [Wang et al. 2004], and LPIPS [Zhang et al. 2018]. We see that our method outperforms all baselines and ablations, although simple baselines such as diffuse rendering and barycentric blending also obtain high scores. This appears to be due to these metrics underemphasizing high frequency details and high-order light transport effects. These results are more easily interpreted using the visualization in Figure 7, where we see that the renderings produced by our approach more closely resemble the ground truth than those of other models. In particular, our method synthesizes shadows, specular highlights, and self-occlusions with higher precision when compared against simple barycentric blending, as well as state-of-art neural rendering algorithms such as Nalbach et al. [2017] and Xu et al. [2018]. Our approach also produces more realistic results than the geometric 3D capture pipeline of Guo et al. [2019]. See the supplemental video for more examples.

HDRI relighting. Our model's ability to synthesize images corresponding to arbitrary light directions allows us to render subjects under arbitrary HDRI environment maps. To do this, we synthesize 331 directional OLAT images that cover the whole light stage dome. These images are then converted to light stage weights by approximating each light with a Gaussian around its center, and we produce the HDRI relighting results by simply using a linear combination of the rendered OLAT images [Debevec et al. 2000]. As shown in Figure 8, we are able to reproduce view-dependent effects as well as specular highlights with high fidelity, and generate compelling composites of the subjects in virtual scenes. See the supplemental video for more examples.

Table 4. **View synthesis.** NLT outperforms all baselines (top) and ablations (bottom) in terms of the LPIPS perceptual metric [Zhang et al. 2018]. UV-space barycentric blending and DNR [Thies et al. 2019] are competitive in terms of PSNR and SSIM [Wang et al. 2004], but barycentric blending is unable to hallucinate missing pixels that are not observed from any given view (in contrast to our learning-based approach), and DNR does not provide a means for relighting (simultaneous or otherwise). For fair comparisons, we circumvent the need to learn viewpoints for screen-space methods, such as barycentric blending and Deep Shading [Nalbach et al. 2017], by performing their operations in the UV space. The numbers are means and 95% confidence intervals.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Diffuse Base	31.45 ± .268	.917 ± .005	.070 ± .004
Bary. Blending (UV)	34.97 ± .273	.960 ± .002	.035 ± .002
Deep Shading (UV)	34.77 ± .405	.950 ± .009	.058 ± .009
DNR	35.49 ± .315	.966 ± .002	.039 ± .002
Relightables	32.24 ± .246	.922 ± .005	.065 ± .004
NLT (ours)	34.83 ± .259	.959 ± .002	.030 ± .001
NLT w/o residuals	34.49 ± .258	.958 ± .002	.032 ± .001
NLT w/o obs.	34.43 ± .286	.953 ± .003	.037 ± .002
NLT w/o LPIPS	34.36 ± .273	.959 ± .003	.041 ± .002

4.2 View Synthesis

Here we present results for the view synthesis task: synthesizing novel viewpoints while also capturing view-dependent effects. A quantitative analysis is presented in Table 4, where we see that our approach outperforms the baselines and is comparable with Thies et al. [2019], which (unlike our technique) only performs view synthesis and does not enable relighting. A qualitative analysis is

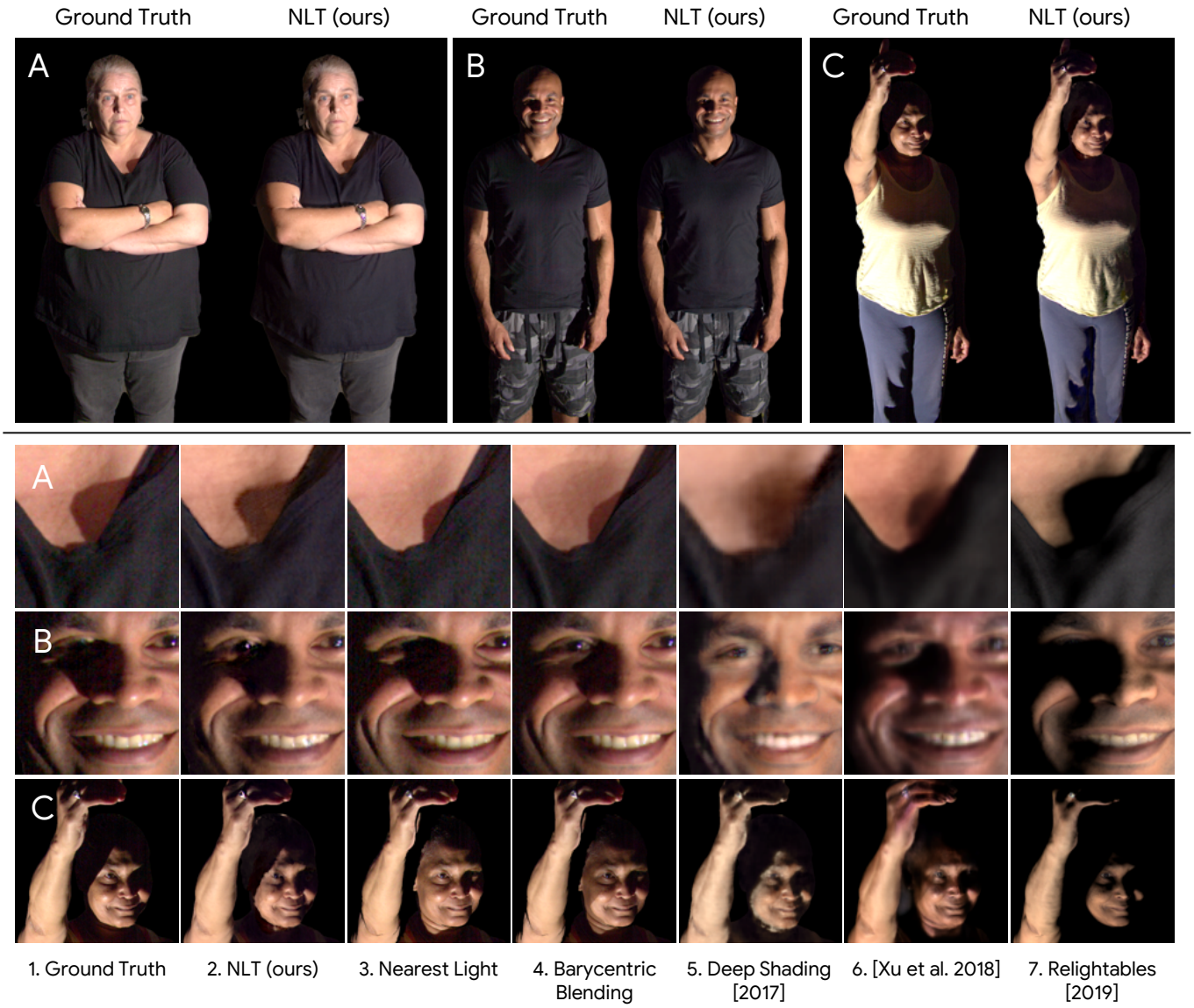


Fig. 7. **Relighting by a directional light.** Here we visualize the performance of our NLT for the task of relighting using directional lights. We show representative examples of full-body subjects with zoom-ins detail focusing on cast shadows (A, C) and facial specular highlights (B). Note how our method is able to outperform all the other approaches with sharper and ghosting-free results that are drastically different from the nearest neighbors.

visualized in Figure 9. We see that the inferred residuals produced by NLT are able to account for the non-diffuse, non-local light transport and mitigate the majority of artifacts in the diffuse base caused by geometric inaccuracy. We see that renderings from NLT exhibit accurate specularities and sharper details, especially when compared with other machine learning methods, thereby demonstrating that our model is able to capture view-dependent effects. See the supplementary video for more examples.

Simultaneous relighting and view synthesis. In Figure 10, we show the unique ability of our model to synthesize illumination and viewpoints simultaneously with an unprecedented quality for human capture. Note that our model's ability to naturally handle this simultaneous task is a direct consequence of embedding our neural network within the UV texture atlas space of the subject. All that is required to enable simultaneous relighting and view interpolation is interleaving the training data for both tasks and training a single instance of our network (more details in Section 3.5). Figure 10 shows that our method accurately models shadows and global illumination, while correctly capturing high-frequency details such as specular highlights. See the supplementary video for more examples.

The recent work of Mildenhall et al. [2020], Neural Radiance Fields (NeRF), achieves impressive view synthesis given approximately 100 views of an object. Here we qualitatively compare NLT against

Fig. 8. **HDRI relighting.** Because NLT can relight a subject with any directional light, it can be used to render OLAT "bases" that can then be linearly combined to relight the scene for a given HDRI map (shown as insets) [Debevec et al. 2000]. The relit subjects exhibit realistic specularities and shadows.

NeRF with 10 levels of positional encoding for the location and 4 for the viewing direction. NeRF does not require any proxy geometry, but in this particular setting, it has to work with a limited number of views (around 55), which are insufficient to capture the full volume. As Figure 11 (left) shows, NLT synthesizes more realistic facial and eye specularity as well as higher-frequency hair details.

We also attempt to extend NeRF to perform simultaneous relighting and view synthesis, and compare NLT with this extension, "NeRF + Light." To this end, we additionally condition the output radiance on the light direction (with 4 levels of positional encoding) along with the original viewing direction. As shown in Figure 11 (right), NeRF + Light struggles to synthesize hard shadows or specular highlights, and produces significantly more blurry results than NLT, which demonstrates the importance of a proxy geometry when there is lighting variation and only sparse viewpoints.

4.3 Performance Analysis

Here we analyze how our model performs with respect to different factors. We show that as the geometry degrades, our neural rendering approach consistently outperforms traditional reprojection-based methods, which heavily rely on the geometry quality. In relighting, we show that our model performs reasonably when the number of illuminants is reduced, demonstrating the potential applicability of NLT to smaller light stages.

View synthesis. Because NLT leverages a geometry proxy to generate a texture parameterization, we study its robustness against geometry degradation in the context of view synthesis. We decimate our mesh progressively from the original 100k vertices down to only 500 vertices (Figure 12 bottom left). At each mesh resolution, we train one NLT model with $K = 3$ nearby views and evaluate it on the held-out views. With the geometry proxy, one can also reproject nearby observed views to the query view, followed by different types of blending [Eisemann et al. 2008; Buehler et al. 2001]. We compare renderings from NLT with those of Eisemann et al. [2008] at each decimation level. As Figure 12 shows, even at the extreme decimation level of 500 vertices, NLT produces reasonable

Fig. 9. **View synthesis.** Here we visualize the NLT results for the task of synthesizing unseen views of a subject. The diffuse base (Column 1) fails to capture fine geometry (hair, chins, etc.), non-Lambertian material effects (specularities and subsurface scattering), and global illumination, all of which are corrected for by the residuals (Column 2) predicted by NLT (Column 3). NLT is able to handle view-dependent specularities (eyes, nose tips, cheeks), high-frequency geometry variation (Subjects B's and D's hair), and global illumination (Subjects A, B, and C's shirts). We see a substantial improvement over the state-of-the-art view synthesis method of Thies et al. [2019] (Column 5), which tends to produce blurry results (the missing specularities in Subject B's eyes) and over the recent geometric approach of Guo et al. [2019] (Column 7), which lacks non-Lambertian material effects. We also compare against an enhanced version of Deep Shading [Nalbach et al. 2017] that has been trained in our texture space (à la Li et al. [2019]) so that the model does not need to learn cross-view correspondences. As Column 6 shows, images synthesized by this enhanced baseline are less faithful to the ground truth (Column 4) than NLT.

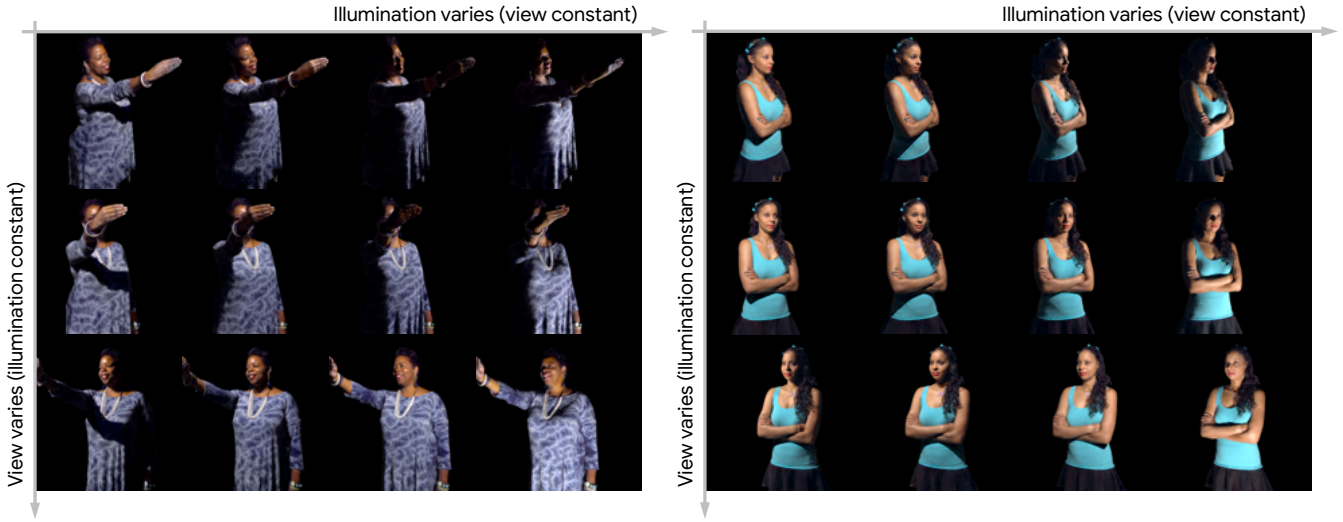


Fig. 10. **Simultaneous relighting and view synthesis.** Our model is able to perform simultaneous relighting and view synthesis, and produces accurate renderings (including view- and light-dependent effects) for unobserved viewpoints and light directions. In the x-axis we vary illumination, and in the y-axis we vary the view. This functionality is enabled by our decision to embed our neural network architecture within the texture atlas of a subject.

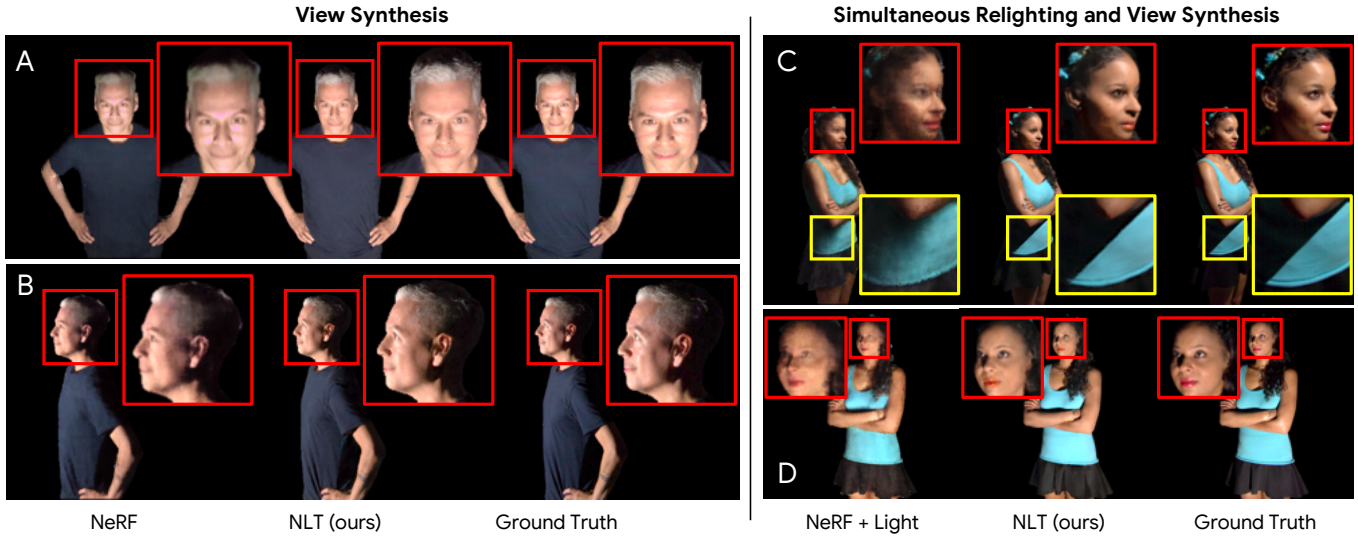


Fig. 11. **Comparisons with NeRF and a NeRF-based extension that supports simultaneous relighting and view synthesis.** In view synthesis (left), NeRF struggles to synthesize realistic facial specularity, high-frequency hair details, and specularity in the eyes (red boxes in A & B). In simultaneous relighting and view synthesis (right), the "NeRF + Light" extension fails to synthesize facial details (red boxes in C & D) and hard shadows (yellow boxes in C).

rendering with no missing pixels, because it has learned to hallucinate pixels that are non-visible from any of the nearby views. In contrast, Floating Textures [Eisemann et al. 2008] leaves missing pixels unfilled (e.g., in the hair) due to reprojection errors stemming from the rough geometry proxy. As the geometry proxy gets more accurate, Floating Textures improves but still struggles to render high-frequency patterns correctly (such as the shadow boundary beside the nose, highlighted by a yellow arrow), even at the original mesh resolution. In comparison, the high-frequency patterns in the NLT rendering match closely the ground truth. Quantitatively, NLT also outperforms Floating Textures in terms of the LPIPS perceptual metric [Zhang et al. 2018] (lower is better) across all mesh resolutions.

Relighting. In this experiment, we artificially downsample the lights on the light stage to study the effects of light density on NLT's relighting performance. We use only 60% of the lights to train a relighting model, which translates to around 75 lights per camera. Although the model is still able to relight the person reasonably in Figure 13, inspection reveals that the relit image has ghosting shadows like those often observed in barycentric blending.

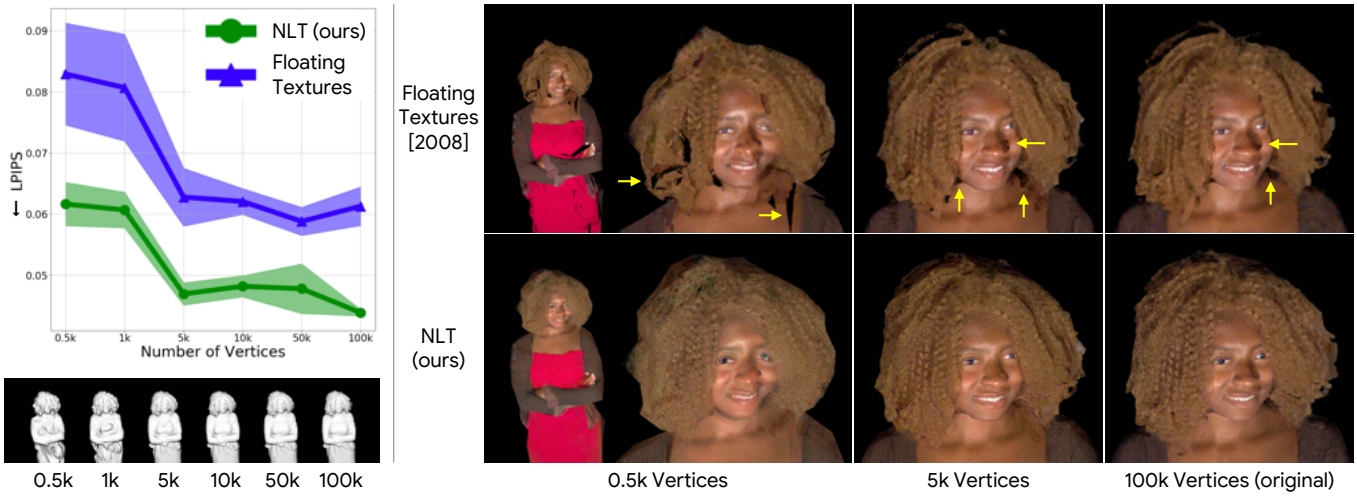


Fig. 12. **View synthesis performance w.r.t. quality of the geometry proxy.** As we decimate the geometry proxy from 100k vertices down to only 500 vertices, NLT remains performant in terms of the LPIPS perceptual metric [Zhang et al. 2018] (lower is better; bands indicate 95% confidence intervals), while Floating Textures [Eisemann et al. 2008], a reprojection-based method, suffers from the low quality of the geometry proxy, producing missing pixels (e.g., in the hair) and misplaced high-frequency patterns (e.g., shadow boundaries), as highlighted by the yellow arrows. Both NLT and Floating Textures use the same three nearby views.

Fig. 13. **Training with sparser lights.** When only 60% lights are used to train a relighting model, we observe ghosting shadows in our relit rendering (yellow arrow), similar to those produced by barycentric blending.

4.4 Ablation Studies

Our quantitative evaluations of relighting and view synthesis (Tables 3 and 4) include ablation studies of our model, in which separate model components are removed to demonstrate that component's contribution.

No observation paths. Instead of our two-path query/observation network (Section 3.3) we can just train the query path of our network without the available observations. As shown in Figure 14, this ablation struggles to synthesize details for each possible view and lighting condition, and produces oversmoothed results.

No residual learning. Instead of using our residual learning approach (Section 3.4) we can allow our network to directly predict the output image. As shown in Figure 14, not using the diffuse base at all reduces the quality of the rendered image, likely because the network is then forced to waste its capacity on inferring shadows and albedo. The middle ground between no diffuse base and our full method is using the diffuse bases only as network input, but not for the skip link. Comparing the "Deep Shading" rows and "NLT w/o obs." rows of Table 3 and Table 4 reveals the importance of the skip connection to diffuse bases: in both relighting and view synthesis, NLT without observations (which has the skip link) outperforms Deep Shading (which uses the diffuse bases only as network input) in LPIPS. Our proposed residual learning scheme allows our model to focus on learning higher-order light transport effects, which results in more realistic renderings.

No perceptual loss. We find that adding a perceptual loss as proposed by Zhang et al. [2018] helps the network produce higher-frequency details (such as the hard shadow boundary in Figure 14). Quantitative evaluations verify this observation: full NLT with the perceptual loss achieves the best perceptual scores in both tasks of relighting and view synthesis.

Fig. 14. **Ablation studies** in the context of relighting. Removing different components of our model reduces rendering quality: no direct access to the diffuse base makes it more challenging for the network to learn hard shadows, having no observation path deprives the network of information from nearby views or lights, and removing the perceptual loss of Zhang et al. [2018] blurs the shadow boundary.

5 LIMITATIONS

Similar to recent works in neural rendering [Lombardi et al. 2019; Thies et al. 2019; Lombardi et al. 2018; Sitzmann et al. 2019a; Mildenhall et al. 2020], our method must be trained individually per scene, and generalizing to unseen scenes is an important future step for the

field. In addition, neural rendering of dynamic scenes is desirable, especially in this case of human subjects. Using a fixed texture atlas may directly enable our method to work for dynamic performers.

Additionally, the fixed 1024×1024 resolution of our texture-space model limits our model's ability to synthesize higher-frequency contents, especially when the camera zooms very close to the subject, or when an image patch is allocated too few texels (see the hair artifact in Figure 9D). This could be solved by training on higher-resolution images, but this would increase memory requirements and likely require significant engineering effort.

Our method has occasional failure modes as shown in Figure 15, where complex light transport effects, such as the ones on the glittery chain, are hard to synthesize, and the final renderings lack high-frequency details.

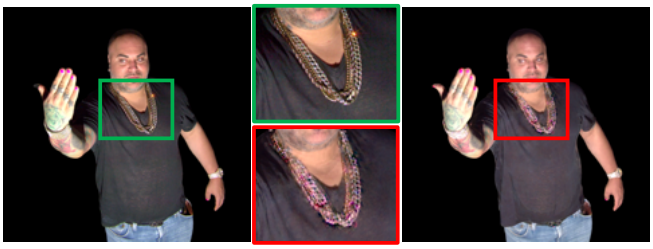


Fig. 15. **A failure case** in view synthesis. NLT may fail to synthesize views of complicated light transport effects, such as those on the glittery chain.

6 CONCLUSION

We have presented Neural Light Transport (NLT), a semi-parametric deep learning framework that allows for simultaneous relighting and view synthesis of full-body scans of human subjects. Our approach is enabled by prior work [Guo et al. 2019] that provides a method for recovering geometric models and texture atlases, and uses as input One-Light-at-A-Time (OLAT) images captured by a light stage. Our model works by embedding a deep neural network into the UV texture space provided by a mesh and texture atlas, and then training that model to synthesize texture-space RGB images corresponding to observed light and viewing directions. Our model consists of a dual-path neural network architecture for aggregating information from observed images and synthesizing new images, which is further enhanced through the use of augmented texture-space inputs that leverage insights from conventional graphics techniques and a residual learning scheme that allows training to focus on higher-order light transport effects such as highlights, scattering, and global illumination. Multiple comparisons and experiments demonstrate clear improvement over previous specialized relighting or view synthesis solutions, and our approach additionally enables simultaneous relighting and view synthesis.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback, Dan B. Goldman for pointing out an image-based relighting issue, Zhoutong Zhang, Graham Fyffe, and Xuaner (Cecilia) Zhang for the fruitful discussions, Qiurui He, Charles Herrmann, and Hayato Ikoma for the generous infrastructure support, David E. Jacobs and Marc Levoy for their constructive comments on an initial draft of this paper, and the actors for appearing in this paper. This work was funded in part by a Google Fellowship, ONR grant N000142012529, and the Ronald L. Graham Chair. We acknowledge support from Shell Research.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, and et al. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *OSDI*.

Edward H Adelson and James R Bergen. 1991. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing* (1991).

Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. 2019. Tex2Shape: Detailed Full Human Body Geometry from a Single Image. In *ICCV*.

Jonathan T Barron. 2019. A General and Adaptive Robust Loss Function. In *CVPR*.

Jonathan T. Barron and Jitendra Malik. 2015. Shape, Illumination, and Reflectance from Shading. *TPAMI* (2015).

H. G. Barrow and J. M. Tenenbaum. 1978. Recovering intrinsic scene characteristics from images. *Computer Vision Systems* (1978).

Ronen Basri, David Jacobs, and Ira Kemelmacher. 2007. Photometric stereo with general, unknown lighting. *IJCV* (2007).

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 425–432.

Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. 2003. Free-viewpoint video of human actors. In *ACM SIGGRAPH*, Vol. 22. 569–577.

Zhang Chen, Anpei Chen, Guli Zhang, Chengyuan Wang, Yu Ji, Kiriakos N Kutulakos, and Jingyi Yu. 2020. A Neural Rendering Framework for Free-Viewpoint Relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5599–5610.

A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. 1992. Biorthogonal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics* (1992).

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality Streamable Free-viewpoint Video. *ACM TOG* (2015).

Paul Debevec. 2012. The Light Stages and Their Applications to Photoreal Digital Actors. (2012).

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*.

Michael Deering, Stephanie Winner, Bic Schediwy, Chris Duffy, and Neil Hunt. 1988. The Triangle Processor and Normal Vector Shader: A VLSI System for High Performance Graphics. In *SIGGRAPH*.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *NIPS*.

Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. 2008. Floating textures. In *Computer graphics forum*.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View Synthesis With Learned Gradient Descent. In *CVPR*.

Graham Fyffe, Cyrus A. Wilson, and Paul Debevec. 2009. Cosine Lobe Based Relighting from Gradient Illumination Photographs. In *SIGGRAPH Poster*.

Marc-Andre Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. 2019. Deep Parametric Indoor Lighting Estimation. (2019).

Gaurav Garg, Eino-Ville Talvala, Marc Levoy, and Hendrik PA Lensch. 2006. Symmetric Photography: Exploiting Data-sparseness in Reflectance Fields. In *Rendering Techniques*. 251–262.

S Gortler, R Grzeszczuk, R Szeliski, and M Cohen. 1996. The Lumigraph. In *SIGGRAPH*.

Kaiwen Guo, Jason Dourgarian, Danhang Tang, Anastasia tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Peter Lincoln, Paul Debevec, Shahram Izad, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, and Rohit Pandey. 2019. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting. In *SIGGRAPH Asia*.

Richard Hartley and Andrew Zisserman. 2003. *Multiple View Geometry in Computer Vision* (2 ed.). Cambridge University Press, New York, NY, USA.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *CVPR* (2016).

Kurt Hornik. 1991. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* (1991).

James T. Kajiya. 1986. The Rendering Equation. In *SIGGRAPH*.

Yoshihiro Kanamori and Yuki Endo. 2018. Relighting Humans: Occlusion-Aware Inverse Rendering for Full-Body Human Images. *TOG* (2018).

Sean Kelly, Samantha Cordingley, Patrick Nolan, Christoph Rhemann, Sean Fanello, Danhang Tang, Jude Osborn, Jay Busch, Philip Davidson, Paul Debevec, Peter Denny, Graham Fyffe, Kaiwen Guo, Geoff Harvey, Shahram Izadi, Peter Lincoln, Wan-Chun Alex Ma, Jonathan Taylor, Xueming Yu, Matt Whalen, Jason Dourgarian, Genevieve Blanchett, Narelle French, Kirstin Sillitoe, Tea Uglow, Brenton Spiteri, Emma Pearson, Wade Kernot, and Jonathan Richards. 2019. AR-ia: Volumetric Opera for Mobile Augmented Reality. In *SIGGRAPH Asia 2019 XR*.

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. 2018. Deep Video Portraits. *TOG* (2018).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul Debevec. 2019. DeepLight: Learning Illumination for Unconstrained Mobile Mixed Reality. In *CVPR*.

Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. In *SIGGRAPH*.

Yikai Li, Jiayuan Mao, Xiuming Zhang, Bill Freeman, Josh Tenenbaum, Noah Snavely, and Jiajun Wu. 2020. Multi-Plane Program Induction with 3D Box Priors. *Advances in Neural Information Processing Systems* 33 (2020).

Yue Li, Pablo Wiedemann, and Kenny Mitchell. 2019. Deep Precomputed Radiance Transfer for Deformable Objects. *ACM CGIT* (2019).

Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. 2018. Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image. In *SIGGRAPH Asia*.

Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep Appearance Models for Face Rendering. *ACM TOG* (2018).

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *SIGGRAPH* (2019).

Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. 2007. Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *EGSR*.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML*.

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, and et al. 2018. LookinGood: Enhancing Performance Capture with Real-Time Neural Re-Rendering. *SIGGRAPH Asia*.

Abhimitra Meka, Jason Dourgarian, Peter Denny, Sofien Bouaziz, Peter Lincoln, Matt Whalen, Geoff Harvey, Jonathan Taylor, Shahram Izadi, Andrea Tagliasacchi, Paul Debevec, Christian Häne, Christian Theobalt, Julien Valentin, Christoph Rhemann, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, and Jay Busch. 2019. Deep reflectance fields: high-quality facial reflectance field inference from color gradient illumination. In *SIGGRAPH*.

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. In *SIGGRAPH*.

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).

J.F. Murray-Coleman and A.M. Smith. 1990. The Automated Measurement of BRDFs and their Application to Luminaire Modeling. *Journal of the Illuminating Engineering Society* (1990).

Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. 2017. Deep Shading: Convolutional Neural Networks for Screen-Space Shading. (2017).

Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, Epic Games, Andreas Lehrmann, and AI Borealis. 2020. Learning Physics-guided Face Relighting under Directional Light. In *CVPR*.

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. 2016. Holoportation: Virtual 3D Teleportation in Real-time. In *UIST*.

Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip Davidson, Cem Keskin, Shahram Izadi, and Sean Fanello. 2019. Volumetric Capture of Humans with a Single RGBD Camera via Semi-Parametric Learning. In *CVPR*.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image Based Relighting Using Neural Networks. *ACM TOG* (2015).

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*.

Ashutosh Saxena, Min Sun, and Andrew Y Ng. 2008. Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI* (2008).

Pradeep Sen, Billy Chen, Gaurav Garg, Stephen R. Marschner, Mark Horowitz, Marc Levoy, and Hendrik P. A. Lensch. 2005. Dual Photography. In *SIGGRAPH*.

Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. 2020. Background Matting: The World is Your Green Screen. In *CVPR*.

Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. 2018. SfSNet: Learning Shape, Refectance and Illuminance of Faces in the Wild. In *CVPR*.

Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor Lempitsky. 2019. Textured Neural Avatars. In *CVPR*.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. 2019a. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *CVPR*.

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*. 1121–1132.

Noah Snavely, Steven M Seitz, and Richard Szeliski. 2006. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH*.

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 *CVPR*.

Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E. Debevec, and Ravi Ramamoorthi. 2019. Single Image Portrait Relighting. In *SIGGRAPH*.

Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T Barron, and Ravi Ramamoorthi. 2020. Light Stage Super-Resolution: Continuous High-Frequency Relighting. *arXiv preprint arXiv:2010.08888* (2020).

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. 2020a. StyleRig: Rigging StyleGAN for 3D Control over Portrait Images, CVPR 2020. In *CVPR*.

Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhoefer. 2020b. State of the Art on Neural Rendering. In *Eurographics*.

Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. 2020. Image-guided neural object rendering. In *International Conference on Learning Representations*.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: image synthesis using neural textures. In *SIGGRAPH*.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE TIP* (2004).

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7467–7477.

Robert J. Woodham. 1980. Photometric Method For Determining Surface Orientation From Multiple Images. *Optical Engineering* (1980).

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

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

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Xuaner Zhang, Jonathan T. Barron, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, and David E. Jacobs. 2020. Portrait Shadow Manipulation. *ACM Transactions on Graphics (TOG)* 39, 4.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. In *SIGGRAPH*.

T. Zickler, R. Ramamoorthi, S. Enrique, and P. N. Belhumeur. 2006. Reflectance sharing: predicting appearance from a sparse set of images of a known shape. *IEEE TPAMI* (2006).