NeROIC: Neural Rendering of Objects from Online Image Collections

ZHENGFEI KUANG∗, University of Southern California, USA
KYLE OLSZEWSKI, Snap Inc., USA
MENGLEI CHAI, Snap Inc., USA
ZENG HUANG, Snap Inc., USA
PANOS ACHLIOPTAS, Snap Inc., USA
SERGEY TULYAKOV, Snap Inc., USA

Online Images Outputs Novel View Synthesis Relighting Composition

Fig. 1. Our Object Capture Results from Online Images. Our modular NeRF-based approach requires only sparse, coarsely segmented images depicting an object captured under widely varying conditions (1st column). We first infer the geometry as a density field using neural rendering, and then compute the object’s surface material properties and per-image lighting conditions (2nd column). Our model not only can synthesize novel views, but can also relight and composite the captured object in novel environments and lighting conditions (3-5th column).

We present a novel method to acquire object representations from online image collections, capturing high-quality geometry and material properties of arbitrary objects from photographs with varying cameras, illumination, and backgrounds. This enables various object-centric rendering applications such as novel-view synthesis, relighting, and harmonized background composition from challenging in-the-wild input. Using a multi-stage approach extending neural radiance fields, we first infer the surface geometry and refine the coarsely estimated initial camera parameters, while leveraging coarse foreground object masks to improve the training efficiency and geometry quality. We also introduce a robust normal estimation technique which eliminates the effect of geometric noise while retaining crucial details. Lastly, we extract surface material properties and ambient illumination, represented in spherical harmonics with extensions that handle transient elements, e.g., sharp shadows. The union of these components results in a highly modular and efficient object acquisition framework. Extensive evaluations and comparisons demonstrate the advantages of our approach in capturing high-quality geometry and appearance properties useful for rendering applications.

CCS Concepts:
• Computing methodologies → Rendering; Computer vision representations.

Additional Key Words and Phrases: neural rendering, multi-view & 3D, reflectance & shading models

ACM Reference Format:
Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. 2022. NeROIC: Neural Rendering of Objects from Online Image Collections. ACM Trans. Graph. 41, 4, Article 56 (July 2022), 12 pages. https://doi.org/10.1145/3528223.3530177

1 INTRODUCTION

Numerous collections of images featuring identical objects, e.g., furniture, toys, vehicles, can be found online on shopping websites or through a simple image search. The ability to isolate these objects from their surroundings and capture high-fidelity structure and appearance is highly desired, as it would enable applications such as digitizing an object from the images and blending it into a new background. However, individual images of the objects in these collections are typically captured in highly variable backgrounds, illumination conditions, and camera parameters, making object digitization approaches specifically designed for data from controlled environments unsuitable for such an in-the-wild setup. In this work, we seek to address this challenge by developing an approach for...
capturing and re-rendering objects from unconstrained image collections by extending the latest advances in neural object rendering.

Among the more notable recent works using neural field is the Neural Radiance Fields (NeRF) model [Mildenhall et al. 2020], which learns to represent the local opacity and view-dependent radiance of a static scene from sparse calibrated images, allowing high-quality novel view synthesis (NVS). While substantial progress has been made to improve the quality and capabilities of NeRF (e.g. moving or non-rigid content [Li et al. 2021; Park et al. 2021; Pumarola et al. 2021; Xian et al. 2021]), some non-trivial requirements still remain – to synthesize novel views of an object the background and illumination conditions should be seen and fixed, and the multi-view images or video sequences should be captured in a single session.

Recently, several works [Boss et al. 2021a,b; Guo et al. 2020; Martin-Brualla et al. 2021; Yang et al. 2021; Zhang et al. 2021b] have extended NeRF and achieved impressive progress in decomposing the renderings of a scene into semantically meaningful components, including geometry, reflectance, material, and lighting, enabling a flexible interaction with any of these components, e.g. relighting and swapping the background. Unfortunately, none of them built a comprehensive solution to work with the limitations of objects captured from real-world, in-the-wild image collections. In this work, we propose NeROIC, a novel approach to Neural Rendering of objects from Online Image Collections. Our object capture and rendering approach builds upon neural radiance fields with several key features that enable high-fidelity capture from sparse images captured under wildly different conditions, which is commonly seen in online image collections with individual images taken with varying lightings, cameras, environments, and poses. The only expected annotation for each image is a rough foreground segmentation and coarsely estimated camera parameters, which crucially we can obtain in an unsupervised way from structure-from-motion frameworks such as COLMAP [Schönberger and Frahm 2016].

Key to our learning-based method is the introduction of a modular approach, in which we first optimize a NeRF model to estimate the geometry and refine the camera parameters, and then infer the surface material properties and per-image lighting conditions that best explain the captured images. The decoupling of these stages allows us to use the depth information from the first stage to do more efficient ray sampling in the second stage, which improves material and lighting estimation quality and training efficiency. Furthermore, due to the modularity of our approach we can also separately exploit the surface normals initialized from the geometry in the first stage, and innovate with a new normal extraction layer that enhances the accuracy of acquiring materials of the underlying object. An overview of our approach is shown in Fig. 2 (b).

To evaluate our approach, we create several in-the-wild object datasets, including images captured by ourselves in varying environments, as well as images of objects collected from online resources. The comparisons with state-of-the-art alternatives, in these challenging setups, indicate that our approach outperforms the alternatives qualitatively and quantitatively, while still maintaining comparable training and inference efficiency. Fig. 1 presents a set of example object capturing and application results by our approach.

In summary, our main contributions are:

- A novel, modular pipeline for inferring geometric and material properties from objects captured under varying conditions, using only sparse images, foreground masks, and coarse camera poses as additional input,
- A new multi-stage architecture where we first extract the geometry and refine the input camera parameters, and then infer the object’s material properties, which we show is robust to unrestricted inputs,
- A new method for estimating normals from neural radiance fields that enables us to better estimate material properties and relight objects than more standard alternative techniques,
- Datasets containing images of objects captured in varying and challenging environments and conditions,
- Extensive evaluations, comparisons and results using these and other established datasets demonstrating the state-of-the-art results obtained by our approach.

Our code, pre-trained models, and training datasets are released at https://formyfamily.github.io/NeROIC/.

2 RELATED WORK

Neural Rendering for Novel View Synthesis. One of the more recent advances in novel view synthesis is NeRF [Mildenhall et al. 2020]. A set of multilayer perceptrons (MLPs) are used to infer the opacity and radiance for each point and outgoing direction in the scene by sampling camera rays and learning to generate the corresponding pixel color using volume rendering techniques, allowing for high-quality interpolation between training images. However, this framework requires well-calibrated multi-view datasets of static scenes as input, with no variation in the scene content and lighting conditions. Many subsequent works build upon this framework to address these and other issues. NeRF++ [Wang et al. 2021b], SCN-NeRF [Jeong et al. 2021] and BARF [Lin et al. 2021b] infer the camera parameters while learning a neural radiance field, to allow for novel view synthesis when these parameters are unknown. iNeRF [Lin et al. 2021a] estimate poses by inverting a trained neural radiance field to render the input images. Other works focus on improving the training or inference performance and computational efficiency [Lindell et al. 2021; Liu et al. 2020; Neff et al. 2021; Reiser et al. 2021; Wang et al. 2021a; Yu et al. 2021]). Related approaches [Bergman et al. 2021; Kellnhofer et al. 2021] use a signed-distance function to represent a surface that can be extracted as a mesh for fast rendering and novel view synthesis. However, these works only display high-quality results for a limited range of interpolated views, and do not perform the level of material decomposition and surface reconstruction needed for high-quality relighting and reconstruction.

Learning from Online Image Collections. Online image collections have been used for various applications, such as reconstructing the shape and appearance of buildings [Snavely et al. 2008] or human faces [Kemelmacher-Shlizerman and Seitz 2011; Liang et al. 2016]. However, such approaches typically require many available photographs, making them applicable only to landmarks or celebrities, and are designed explicitly to work with specific subjects with domain features, rather than arbitrary objects. Recently, neural rendering has been combined with the use of generative
adversarial networks [Goodfellow et al. 2014] to allow for generatively sampling different objects within a category and rendering novel views [Nguyen-Phuoc et al. 2019; Niemeyer and Geiger 2021; Schwarz et al. 2020], using only a single image of each training object. However, such approaches do not allow for rendering novel views of a target object, e.g. from one or more images, with controllable shape, appearance, and environmental conditions, and the quality of the sampled images varies. Other approaches learn pose, shape, and texture from images for certain categories of objects [Jang and Agapito 2021; Meshry et al. 2019], or interpolation, view synthesis, and segmentation of sampled category instances [Xie et al. 2021]. However, none of these approaches allow for the level of structure and material decomposition suitable for high-fidelity rendering and relighting.

**Image Content Decomposition and Relighting.** Many recent works focus on decomposing the lighting condition and intrinsic properties of the objects from the training images. NeRF-in-the-Wild [Martin-Brualla et al. 2021] learns to render large-scale scenes works, i.e. NeRD [Boss et al. 2021a], Neural-PIL [Boss et al. 2021b], and OsR [Rudnev et al. 2021] use Spherical Harmonic coefficients to represent environmental lighting as in our approach, but is designed for outdoor scene reconstruction (e.g., buildings and landmarks). To the best of our knowledge, we are the first NeRF-based method to infer both geometry and material parameters of the object depicted in our image collection. This, however, is made more challenging due to the varying lighting environments, transient conditions e.g. sharp shadows, varying camera parameters, and coarse camera poses and intrinsics caused by the lack of background context required for accurate camera calibration. Inspired by Martin-Brualla et al. 2021, we employ a pipeline designed to make use of images captured under different conditions, and introduce additional designs to account for the challenging task of aggregating an object representation solely from the isolated foreground region.

3 METHOD

In this section, we outline our approach to object-centric aggregation. We first provide an overview of the approach (Sec. 3.1), followed by a description of the neural radiance fields framework we extend in our method (Sec. 3.2).

3.1 Overview

Fig. 2 provides an overview of our approach. The inputs are a sparse collection of images $I_k : [0, 1]^2 \rightarrow [0, 1]^3$ depicting an object (or instances of an identical object) under varying conditions, and a set of foreground masks $F_k : [0, 1]^2 \rightarrow [0, 1]$ defining the region of the object, where $1 \leq k \leq N$. During the first stage, we estimate the geometry of the object by learning a density field indicating where there is physical content (Sec. 3.3). During this stage, we also learn both static and transient radiance values to allow for image-based supervision, but do not fully decompose this information into material and lighting properties. We also optimize the pose and intrinsic parameters of the cameras to refine the coarse estimates provided as input.

In the second stage, we fix the learned geometry and optimize the surface material and lighting parameters needed to re-render the object in arbitrary illumination conditions (Sec. 3.5). During this stage, we use the estimated distance from the camera to the object surface to improve our point sampling along the camera rays. We also optimize the surface normals, which improves on the coarse estimates that are obtained from our density field (Sec. 3.4).

3.2 Preliminaries

In Neural Radiance Fields (NeRF) [Mildenhall et al. 2020], a set of networks are trained to infer radiance and density for arbitrary 3D points, and generate images from novel viewpoints using volumetric rendering. Specifically, it employs two MLP functions: a density function $\sigma(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^+$ and a color function $c(x) : \mathbb{R}^3 \rightarrow [0, 1]^3$. For each ray $r = (r_o, r_d)$ emitted from the camera origin $r_o$ in direction $r_d$, NeRF samples $N_p$ 3D points along the ray $x_i = r_o + d_i r_d$ ($0 \leq i \leq N_p$), and integrates the pixel color as follows:

$$C(r) = \sum_{i=1}^{N_p} a_i (1 - w_i) c(x_i),$$

where $w_i = \exp(-(d_i - d_{i-1}) \sigma(x_i))$ represents the transmittance of the ray segment between sample points $x_{i-1}$ and $x_i$, and $d_i = \prod_{j=1}^{i-1} w_i$ is the ray attenuation from the origin $r_o$ to the sample point $x_i$. In addition to the volumetric rendering function, NeRF also introduces an adaptive coarse-to-fine pipeline which uses a coarse model to guide the 3D point sampling of the fine model.

3.3 Geometry Networks

In our first stage, we seek to reconstruct the geometry of the target object depicted in our image collection. This, however, is made more challenging due to the varying lighting environments, transient conditions e.g. sharp shadows, varying camera parameters, and coarse camera poses and intrinsics caused by the lack of background context required for accurate camera calibration. Inspired by Martin-Brualla et al. 2021, we employ a pipeline designed to make use of images captured under different conditions, and introduce additional designs to account for the challenging task of aggregating an object representation solely from the isolated foreground region.
Thus, we employ a two-branch pipeline which handles transient and static content separately, and assigns unique embedding vectors $z_k^{(r)}$ and $z_k^{(s)}$ to each image to represent the transient geometry and changing lighting. Our model for this stage thus consists of four functions instead of two: $\sigma_k^{(s)}(x)$, $\sigma_k^{(r)}(x)$, and $c_k^{(s)}(x)$, $c_k^{(r)}(x)$. The volumetric rendering function in Mildenhall et al. [2020] is re-formulated as:

$$C_k(r) = \sum_{i=1}^{N_p} \alpha_{ki}(1 - w_{ki}^{(s)})c_k^{(s)}(x_i) + (1 - w_{ki}^{(r)})c_k^{(r)}(x_i), \quad (2)$$

where $w_{ki}^{(s,r)} = \exp(-(d_i - d_{i-1})\sigma_k^{(s,r)}(x_i))$, and $\alpha_{ki} = \prod_{j=1}^{p} w_{ki}^{(s)}w_{ki}^{(r)}$. We also adopt the Bayesian learning framework of Kendall and Gal [2017], predicting an uncertainty $\beta_k(x)$ for transient geometry when accounting for the image reconstruction loss.

Eq. 2 serves as the rendering function when training this network. As in Martin-Brualla et al. [2021], we use a color reconstruction loss $L_c$, incorporated with $\beta_k$, and a transient regularity loss $L_{tr}$. However, to accurately capture the geometry corresponding to our target object, we found it essential to incorporate additional losses designed for our particular use case.

**Silhouette Loss.** We use the input foreground masks to help the networks focus on the object inside the silhouette, thus preventing ambiguous geometry from images with varying backgrounds. While we mask out the background in each image and replace it with pure white, a naive approach will still fail to discriminate the object from the background, thus producing white artifacts around the object and occluding it in novel views. To avoid this issue, we introduce a silhouette loss $L_{sil}$, defined by the binary cross entropy (BCE) between the predicted ray attenuation $\alpha_k$ and the ground-truth foreground mask $F_k$ to guide the geometry learning process. As seen in the ablation study in Tab. 3, the silhouette loss significantly improves our results on testing data.

**Adaptive Sampling.** We also introduce an adaptive sampling strategy in our model using these masks. At the beginning of every training epoch, we randomly drop out part of the background rays from the training set, so that to ensure the ratio of the foreground rays is above 1/3. This seemingly simple strategy significantly increases the training efficiency, and balances the silhouette loss and prevents $\alpha_k$ from converging to a constant. Our ablation study in Tab. 3 demonstrates that without this adaptive sampling, the model produces much worse results during testing.

**Camera Optimization.** While our input images come from multiple sources, the lack of a consistent background leads to poor camera pose registration. In practice, though, we use COLMAP [Schönberger and Frahm 2016] on images with the backgrounds removed, the poses for some objects are still inaccurate, as seen in Fig. 3. To address this issue, we jointly optimize the camera poses during training, in a manner similar to Wang et al. [2021b]. More specifically, we incorporate camera parameters $(\delta_R, \delta_t, \delta_f)$ for rotation, translation, and focal length, respectively. We use an axis-angle representation for rotation, while the others are in linear space. We also add a regularity loss $L_{cam}$ for the camera parameters, which is simply an L2 loss on these parameters.

As a summary, the final loss we use for this stage is:

$$L_{geo} = L_c + \lambda_{tr}L_{tr} + \lambda_{sil}L_{sil} + \lambda_{cam}L_{cam}. \quad (3)$$

where the weights $\lambda_{tr}$, $\lambda_{sil}$, and $\lambda_{cam}$ are 0.01, 0.1, and 0.01, respectively, in our experiments.

### 3.4 Normal Extraction Layer

With the learned geometry from our first stage, we then extract the surface normals of the object as the supervision to the next stage, which helps reduce the ambiguity of the lighting and material estimation task. While many previous works [Bi et al. 2020; Boss et al. 2021a; Zhang et al. 2021b] choose to use the gradient of the density function (i.e., $\nabla\sigma^{(s)}(x)$) as an approximation of normals, we find that this approach may produce incorrect results in certain areas, due to the challenging issues with unconstrained, real data. Therefore, we propose a novel normal extraction method by leveraging the geometry network and using novel views to help differentiate between the object and background.

Using the geometry network, in which we infer the lighting conditions (represented as spherical harmonics coefficients), surface material properties (using the Phong rendering model), and high-quality surface normals (c).

---

For details on these losses and their use, please see the supplementary document.
grid center $x$, we remap its density value as:

$$a'_x = \frac{1}{\lambda} \left( 1 - \exp(-\lambda \sigma_x) \right).$$

This function remaps the density value from $[0, +\infty]$ to $[0, \frac{1}{\lambda}]$. The derivative gradually decays as the density value increases, which helps to filter out noise and obtain smoother predictions. $\lambda$ is a controllable parameter to adjust the sharpness of the normal. As $\lambda$ decreases, the derivative converges to a constant value of 1, and thus the mapping function itself converges to the identity function. After remapping, we estimate the gradient of the density field $d\sigma'/dx$ by applying a 3D convolution with a Sobel kernel $\mathcal{K}(x) = x/\|x\|^2$ of size 5 to the density grid.

Finally, we divide the convolution output $n_x^{(g)} = -\mathcal{K}(\sigma_x)$ by max(1, $\|n_x^{(g)}\|_2^2$), producing a normal supervision vector with length no larger than 1. We treat its length as the confidence of the estimation, which becomes the weight of its supervising loss in the following stage. We show the results of each step in Fig. 4.

### 3.5 Rendering Networks

The purpose of our final stage is to estimate the lighting of each input image and the material properties of the object, given the geometry shape and surface normals from previous stages. Since extracting object materials in unknown lighting is highly ill-posed [Ramamoorthi and Hanrahan 2001; Zhang et al. 2021b], we use low-order Spherical Harmonics (SH) to represent our lighting model and optimize its coefficients. We use the standard Phong BRDF [Phong 1975] to model the object material properties, which are controlled by three parameters: $K_d$ for the base color, $K_s$ for the specularularity and $g$ for the glossiness. According to Ramamoorthi and Hanrahan [2001], this light transportation between a Phong BRDF surface and a SH environment map can be efficiently approximated, and we thus employ these rendering equations in our pipeline.

### Hybrid Color Prediction using Transience

Although the spherical harmonics illumination model typically works well on scenes with ambient environment illumination, it lacks the ability to represent sharp shadows and shiny highlights from high-frequency light sources. While we believe it is quite impractical to acquire high-frequency details of lighting and material with respect to our unconstrained input, we hope to eliminate the effect caused by those components, and to learn an unbiased result at lower frequencies. To achieve that, we introduce a hybrid method that combines color prediction with neural networks and parametric models. As in the geometry network described in Sec. 3.3, we employ the concept of transience. However, here we do not learn a separate transient geometry in this model, as our geometry is fixed at this point. We use the volumetric rendering in Eq. 1, but replace the color function with:

$$c_k(x) = \text{lerp}(c_k^{(\tau)}(x), c_k^{(\text{SH})}(x), \exp(-a_k^{(\tau)}(x))).$$

where $c_k^{(\text{SH})}(x)$ is the output color of our SH renderer.

### Estimated Depth for Acceleration

Compared to our geometry networks where color is predicted by neural networks, the rendering stage requires more computation to calculate the color of each
sample point due to the more complex rendering equations. On the other hand, however, the learned geometry from the first networks can be used to filter out sampling points that are far away from the object, thus accelerating the whole training process. We develop a hybrid sampling strategy that can speed up the training without introducing any significant artifact.

For a group of $N_p$ sample points $x_i = r_o + d_i r_d$ on a ray, we build a discrete distribution along the ray with the probability of each point proportional to $\sigma_i (1 - w_i)$. Then, we calculate the expectation and variance on $d_i$ w.r.t. to this distribution, denoted as $E(d)$ and $V(d)$. If the variance $V(d)$ is smaller than a threshold $\tau_d$, we then calculate the $\sigma$ point at depth $E(d)$ and only use this point for the color calculation. Otherwise, we use all sample points. Please refer to our supplementary material for more details.

**Neural Normal Estimation w/ Supervision.** Our network also predicts the final surface normals $n(x)$, supervised by the output of the normal extraction layer in Sec 3.4, with the reconstruction loss $\mathcal{L}_n$ defined by:

$$\mathcal{L}_n = \left\| \left( |n_n^{(g)}|_2 \right) \cdot n(x) - n_n^{(g)} \right\|^2_2,$$

(6)

We also adopt the normal smoothing loss $\mathcal{L}_{sm}$ in Zhang et al. [2021b] to improve the smoothness of the predicted normals.

**Tone-Mapping.** Since our renderer calculates the radiance in linear HDR space, we also apply a tone-mapping process to the rendered results, defined as:

$$\mathcal{T}_k(x) = x^{(1/y_k)},$$

(7)

where $y_k$ is a trainable parameter assigned to image $I_k$, and is initialized from 2.4, the default value of common sRGB curves. However, we neither apply exposure compensation nor white balance to our renderer’s output, assuming that our SH renderer can automatically adapt to these variances during the optimization.

Additionally, to reduce the ambiguity between the material properties and the lighting, we add a regularity loss $\mathcal{L}_\text{reg}$ on both the lighting parameters and material properties. The regularity loss is defined as:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{spec}} \|K_k\|_2^2 + \lambda_{\text{gamma}} \frac{1}{N} \sum_{k=1}^N \|y_k - 2.4\|_2^2$$

$$+ \lambda_{\text{light}} \frac{1}{N_t} \sum_{t=1}^{N_t} \|\text{ReLU}( -L_{k_t}(\omega_t) - \tau_{\text{light}})\|_2^2,$$

(8)

where $L_k$, is the environment map of image $k_t$ derived from its SH coefficients, and $\lambda_{\text{spec}}, \lambda_{\text{gamma}}, \lambda_{\text{light}}$ are coefficients set to 0.1, 5, and 5 respectively. The last term is for light regularization, designed to prevent negative values (lower than $-\tau_{\text{light}}$, with $\tau_{\text{light}}$ set to 0.01) in the SH lighting model, which may happen during training due to overfitting to sharp shadows. For each iteration, we randomly sample $N_t$ incoming light directions $\omega_t$ and image indices $k_t$, and evaluate the corresponding incoming light values for the loss calculation. $N_t$ is set to be identical to the batch size in our experiments.

In summary, the total loss of this stage is defined as:

$$\mathcal{L}_{\text{render}} = \mathcal{L}_c + \lambda_{\text{tr}} \mathcal{L}_{\text{tr}} + \lambda_n \mathcal{L}_n + \lambda_{\text{sm}} \mathcal{L}_{\text{sm}} + \mathcal{L}_{\text{reg}},$$

(9)

where the weights $\lambda_{\text{tr}}, \lambda_n$, and $\lambda_{\text{sm}}$ are set to 1, 5, and 0.5, respectively, in our experiments.

### 3.6 Network Structure

In our first stage, the geometry network, the input position vector $x$ is embedded using the positional encoding method introduced in Vaswani et al. [2017], then fed into an 8-layer MLP with a hidden vector dimension of 256. The resulting embedding, $z_x$, is then fed into three branches: a branch consisting of one layer to predict static density $\sigma_k^{(r)}$, a branch consisting of one layer to predict static color $c_k^{(s)}$, which also takes the positional-embedded view direction $d$ and appearance embedding $z_k^{(a)}$ as input; and a branch of another 4-layer MLP with a hidden vector dimension of 128, followed by several output layers to predict transient density $\sigma_k^{(t)}$, transient color $c_k^{(t)}$ and uncertainty $\beta_k$, where the transient embedding $z_k^{(t)}$ is also provided as input.

Our second stage, the rendering network, shares the same structure as the first stage on most components, except for the static color prediction branch. For this branch, we use a new 4-layer MLP with the hidden vector dimension of 128, which takes $x$ and $z_x$ as input, followed by several output layers to generate the normal $n$, base color $K_d$, specularity $K_s$, and glossiness $g$.

We choose ReLU as the activation function for all intermediate layers. For the outputs layers, we adopt SoftPlus for density functions, uncertainty, and glossiness; Sigmoid for static/transient/base color and specularity; and a vector normalizing layer for normal estimation.

In addition to our network parameters, we also jointly optimize the light coefficients $L_{k,lm}$, the camera parameters $(\delta_R, \delta_t, \delta_f)_k$, and the tone-mapping parameter $y_k$ for each image $I_k$.

### 4 EVALUATIONS

#### 4.1 Implementation details

**Training.** We use a modified version of MLP structure following Martin-Brualla et al. [2021], Mildenhall et al. [2020] as our networks. In the training, we use the Adam optimizer [Kingma and Ba 2015] to learn all of our parameters, and our initial learning rate is set to $4 \cdot 10^{-4}$. Our training and inference experiments are implemented using the PyTorch framework [Paszke et al. 2019]. We train our model on 4 NVIDIA V100s with the batch size of 4096, and test our model on a single NVIDIA V100. In the first stage, we train our model with 30 epochs (60K-220K iterations), in roughly 6 to 13 hours. We then apply our normal extraction layer to generate the surface normals for all training images within 1 hour. For the second stage, we freeze the static MLP for density prediction and train the rest of our model from scratch. Approximately 2 to 4 hours are required for 10 epochs of training.

**Datasets.** We use datasets of 13 objects in our evaluations, collected from three different sources: image collections found on the Internet; objects we captured ourselves; and data published in NeRD [Boss et al. 2021a]. The number of training images of each object varies...
We first show comparisons between our model and NeRF [Mildenhall et al. 2020] in 7 offline captured objects (Milk, Figure, TV, Gnome, Head, Cape, MotherChild). We adopt the commonly used metrics Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) in our evaluation. Considering that the illumination conditions of the testing images are unknown, we propose two settings to evaluate our model: using the lighting parameters (e.g. the embedded vector for the geometry network, and the SH coefficients and the gamma value for the rendering network) from another training image in the same scene; or freezing the networks and optimize the lighting parameters with a Stochastic Gradient Descent optimizer for 1000 steps. As shown in Tab. 1 and Fig. 5, our method outperforms NeRF in both settings by a considerable margin, except on the NeRD Head dataset. We believe that this is because it is a relatively simple case in which the object is rotating in front of a camera with a fixed pose. Since NeRF has fewer components and parameters to optimize, it is more likely to converge to a sharper geometry and get better testing results in such a trivial scenario. In qualitative results, our model generates more consistent and smooth results than NeRF does. Aside from the comparison with NeRF, we would like to highlight that our rendering networks produce competitive results compared to the first stage, while it also supports relighting on unseen environments.

We also compare our model with NeRF on the TV dataset with fewer training images. In this experiment, we train our model and NeRF on three training sets consisting of all images, 20 images and 10 images respectively. We then test them on the same testing set. The results are shown in Tab. 2. As scores of both model are both decreased when training images get fewer, our model are less affected than NeRF, especially on LPIPS score. This result proves that our model is able to keep the output accurate in a perceptual manner even if only sparse training images are given.

Aside from NeRF, we compare two state-of-the-art decomposition methods with our approach: NeRD [Boss et al. 2021a] and NerFactor [Zhang et al. 2021b]. Results are shown in Fig.7 and Fig.6. We found that our approach performed well compared to NeRD when using their data, and outperforms both NeRD and NerFactor on the less constrained images we use, which contain a wider range of backgrounds and lighting conditions. For more comparisons and discussions of these works, please consult the supplementary material.

We also compare our model with NeRF on the TV dataset with fewer training images. In this experiment, we train our model and NeRF on three training sets consisting of all images, 20 images and 10 images respectively. We then test them on the same testing set. The results are shown in Tab. 2. As scores of both model are both decreased when training images get fewer, our model are less affected than NeRF, especially on LPIPS score. This result proves that our model is able to keep the output accurate in a perceptual manner even if only sparse training images are given.

4.2 Comparisons

We first show comparisons between our model and NeRF [Mildenhall et al. 2020] in 7 offline captured objects (Milk, Figure, TV, Gnome, Head, Cape, MotherChild). We adopt the commonly used metrics Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) in our evaluation. Considering that the illumination conditions of the testing images are unknown, we propose two settings to evaluate our model: using the lighting parameters (e.g. the embedded vector for the geometry network, and the SH coefficients and the gamma value for the rendering network) from another training image in the same scene; or freezing the networks and optimize the lighting parameters with a Stochastic Gradient Descent optimizer for 1000 steps. As shown in Tab. 1 and Fig. 5, our method outperforms NeRF in both settings by a considerable margin, except on the NeRD Head dataset. We believe that this is because it is a relatively simple case in which the object is rotating in front of a camera with a fixed pose. Since NeRF has fewer components and parameters to optimize, it is more likely to converge to a sharper geometry and get better testing results in such a trivial scenario. In qualitative results, our model generates more consistent and smooth results than NeRF does. Aside from the comparison with NeRF, we would like to highlight that our rendering networks produce competitive results compared to the first stage, while it also supports relighting on unseen environments.

We also compare our model with NeRF on the TV dataset with fewer training images. In this experiment, we train our model and NeRF on three training sets consisting of all images, 20 images and 10 images respectively. We then test them on the same testing set. The results are shown in Tab. 2. As scores of both model are both decreased when training images get fewer, our model are less affected than NeRF, especially on LPIPS score. This result proves that our model is able to keep the output accurate in a perceptual manner even if only sparse training images are given.

Aside from NeRF, we compare two state-of-the-art decomposition methods with our approach: NeRD [Boss et al. 2021a] and NerFactor [Zhang et al. 2021b]. Results are shown in Fig.7 and Fig.6. We found that our approach performed well compared to NeRD when using their data, and outperforms both NeRD and NerFactor on the less constrained images we use, which contain a wider range of backgrounds and lighting conditions. For more comparisons and discussions of these works, please consult the supplementary material.

4.3 Ablations

To help understand the importance and effectiveness of our contributions, we further conduct four ablatative studies on our model.

Ablation Study on Novel View Synthesis. In the first study, we quantitatively compare our geometry and rendering model with variants on the MotherChild dataset. For our geometry model, the variants we evaluated are: Our model trained without silhouette loss (Model w/o sil); without adaptive sampling (Model w/o ada); and without transient components (Model w/o tr). Variants of our rendering model are: Our model trained without regularity loss (Model w/o reg); without transient components (Model w/o tr); and without the normal extraction layer (Model w/o NEL). We report the PSNR score for all models, and the mean squared error between the attenuation map and the ground-truth foreground mask (denoted as MMSE) for all geometry models. The results are shown in Tab. 3. Our model generally outperforms all variants on both stages, except in two cases. First, removing the transient component from the geometry model did not significantly affect its performance on the PSNR metric. We believe this is because, in most images, the areas with occlusions and incorrect masking are relatively small compared to the object, and thus have a minor effect on the color reconstruction. However, it still decreases the accuracy of the geometry silhouette by one order of magnitude. Second, removing the normal extraction layer from the rendering model did not notably decrease its optimization-free PSNR score either. We believe this is because the effect on the normal is canceled out by other components when rendering the testing view. We will directly compare the normal maps in our next experiment.

Ablation Study on Decomposition. Our second ablation study aims to demonstrate that our specific approach generates smooth and accurate material properties in the second stage. We qualitatively compare our full model with three variants mentioned above on the TV and Head datasets. As shown in Fig. 8, using the normals without proper processing results in worse normal predictions for both datasets; removing the transient component leads to unwanted artifacts on the TV’s smooth surface, where mirroring reflections are more likely to appear; and removing the regularity loss results

![Fig. 5. Qualitative Comparisons with NeRF. Result of Ours-Full are rendered with SH rendering only. Some shadows and highlights are handled as transient component, thus not appearing in Ours-Full.](Image)
Table 1. Comparison with NeRF [Mildenhall et al. 2020]. We report the PSNR, SSIM and LPIPS scores of 7 tested scenes. We also ablate on w/ only the first stage of our approach (Ours-Geom) and w/o appearance optimization (w/o Opt). We highlight the best and second best results of each column in orange and yellow, respectively. The last column contains the mean result across all target objects.

| Methods                  | Cape | Gnome | Head | MotherChild |
|--------------------------|------|-------|------|-------------|
|                          | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| NeRF [2020]              | 22.95 | 0.78 | 0.218 | 18.73 | 0.82 | 0.240 | 29.13 | 0.92 | 0.140 | 25.08 | 0.95 | 0.106 |
| Ours-Geom w/o Opt        | 23.22 | 0.82 | 0.180 | 26.95 | 0.89 | 0.120 | 26.37 | 0.92 | 0.136 | 23.03 | 0.95 | 0.068 |
| Ours-Full w/o Opt        | 22.82 | 0.79 | 0.198 | 25.30 | 0.87 | 0.132 | 26.08 | 0.90 | 0.146 | 25.69 | 0.96 | 0.069 |
| Ours-Geom                | 24.70 | 0.83 | 0.178 | 28.11 | 0.89 | 0.119 | 26.80 | 0.92 | 0.136 | 28.98 | 0.97 | 0.058 |
| Ours-Full                | 23.47 | 0.80 | 0.197 | 26.11 | 0.88 | 0.129 | 26.35 | 0.91 | 0.145 | 29.02 | 0.97 | 0.062 |

Table 2. Comparison with NeRF with fewer images. We highlight the best and second best results in each column in orange and yellow.

| Methods                  | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
|--------------------------|-------|-------|--------|-------|-------|--------|-------|-------|--------|-------|-------|--------|
| NeRF [2020]              | 19.40 | 0.92 | 0.145 | 21.54 | 0.90 | 0.159 | 19.03 | 0.90 | 0.145 | 22.26 | 0.88 | 0.164 |
| Ours-Geom w/o Opt        | 21.41 | 0.94 | 0.059 | 22.89 | 0.92 | 0.121 | 20.31 | 0.92 | 0.114 | 23.45 | 0.90 | 0.114 |
| Ours-Full w/o Opt        | 23.00 | 0.95 | 0.066 | 23.69 | 0.92 | 0.133 | 21.88 | 0.92 | 0.122 | 24.06 | 0.90 | 0.124 |
| Ours-Geom                | 27.51 | 0.96 | 0.052 | 24.41 | 0.93 | 0.118 | 25.06 | 0.93 | 0.110 | 26.51 | 0.91 | 0.110 |
| Ours-Full                | 28.87 | 0.95 | 0.056 | 24.72 | 0.92 | 0.130 | 26.52 | 0.93 | 0.107 | 26.43 | 0.90 | 0.118 |

Table 3. Ablation Study on Novel View Synthesis. We show PSNR and MMSE scores on our geometry network models (top 4 rows) and rendering network models (bottom 4 rows). For the geometry network, from top to bottom: model trained without silhouette loss; model trained without adaptive sampling; model trained without transient model; full model. For rendering network, from top to bottom: model trained without regularity loss; model trained without transient model; model trained without normal estimation layer; full model. Please notice that MMSE is a geometry-based metric and thus not affected by model optimization nor rendering network training.

| Stage       | Methods         | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
|-------------|-----------------|-------|-------|--------|-------|-------|--------|-------|-------|--------|
| Geometry    | Model w/o opt   | 21.30 | 23.38 | 0.180  | 16.52 | 0.89 | 0.167  | 17.51 | 0.88 | 0.244  |
|             | Model w/o aha   | 13.60 | 13.78 | 0.073  |       |       |        |       |       |        |
|             | Model w/o tr    | 21.34 | 30.27 | 0.030  |       |       |        |       |       |        |
|             | Full model      | 25.03 | 28.98 | 0.003  |       |       |        |       |       |        |
| Rendering   | Model w/o reg   | 25.57 | 28.00 |        | N/A   |       |        |       |       |        |
|             | Model w/o tr    | 24.95 | 28.03 |        |       |       |        |       |       |        |
|             | Model w/o NEL   | 25.79 | 28.53 |        |       |       |        |       |       |        |
|             | Full model      | 25.69 | 29.02 |        |       |       |        |       |       |        |

in a biased albedo for the Head data. Our full model addresses each of these problems, producing the most appropriate results.

Ablation Study on Imperfect Masks. While our model performs well on our data with pre-processed foreground masks, we also evaluate our approach’s capacity to handle improperly segmented input images, testing our model in an alternative, more challenging setting with imperfect masks. In this experiment, we dilate the foreground masks of all training images of TV data by multiple fixed radii (10, 20 and 40 pixels), and evaluate our full geometry model and our model trained without transient components. This dilation process aims to simulate the common error of roughly annotated masks, where the mask contains the object but does not fit the object silhouette perfectly. We also emphasize that this setting is more challenging than many real cases. As all training images share an identical type of mask dilation, it becomes harder for the model to distinguish the foreground object from the wrongly annotated background. The results are shown in Fig.9. We note that our model generally reconstructs the geometry of the TV well with mask dilations within 40 pixels, and removing the transient component significantly affects the robustness of our model.

Ablation Study on Camera Optimization. Our last ablation study focuses on evaluating the effect of the camera optimization technique in our geometry model in improving its robustness to pose perturbations. We use the synthetic datasets from NeRD (Chair, Car and Globe), which contain ground-truth poses. For each dataset, we only use 40 images of its training set to train our model. Three types of perturbations are added to the poses: (1) rotational perturbation within a certain degree range (10° - 20°); (2) translational perturbation within a certain distance (0.1 - 0.2, after the poses are normalized); and (3) the combination of the first two perturbations. We test our full model and our model trained without camera optimization (Model w/o cam-opt), and report the average PSNR scores, the number of scenes where each model succeed to converge (Succ No.), and an error defined by the average mean squared error.
Fig. 6. **Comparison on Decomposition.** We show diffuse albedo, surface normal, roughness/glossiness maps and final rendered result of the TV and Figure datasets, predicted by NeRFactor [Zhang et al. 2021b], NeRD [Boss et al. 2021a] and our method. Note that the normal and roughness/glossiness maps of each method are encoded in different ways from respective code bases, and NeRFactor does not explicitly predict a roughness map. We also note that NeRFactor was originally designed for datasets with constant lighting environments.

between the ground truth fundamental matrices and the predicted ones (FMSE). Specifically, this error is defined as:

$$E_{FMSE} = \frac{1}{N_t(N_t - 1)} \sum_{i,j \in N_t} || F_{ij}^{gt} - F_{ij}^{pred} ||^2,$$

where $F_{ij}$ is the fundamental matrix derived from the poses of the $i$-th and $j$-th images. This error is to measure the accuracy of the camera poses without the affect of global transformations of the scene or changes to the camera intrinsic parameters, which can occur during a joint optimization of the model and the camera poses.

The results are shown in Tab. 4. Our model successfully converged for all scenes when the rotational perturbation is under $10^\circ$ (rotational) and the translational perturbation is under 0.1. Also, removing the camera optimization from our model hugely affects the robustness of our model when handling inaccurate camera poses.

4.4 Relighting and Compositing Results
We provide more results of our model in three different showcases: material decomposition, relighting, and composition with online objects.

Fig. 10 shows our prediction of material properties and rendering components for the Gnome dataset. We would like to highlight that our model successfully disentangled the shadow in the input images from our SH rendering component, and learned unbiased material properties from the whole training set. With the material properties, we are able to re-render the objects with new lighting environments, and the results are shown in Fig. 11.

Finally, we demonstrate results for our target application, rendering objects from online image collections in novel environments and lighting. With images of several items collected from the internet, we can recover their geometry and material properties, and finally re-render and compose them into a new environment. As shown in Fig. 12, even though our input images are captured in vastly differing environments, our model handles this challenging task, producing high-quality and plausible compositing results.

5 LIMITATIONS
As shown in Fig. 13, our approach has a few limitations. Firstly, while our model can handle certain camera perturbations, it still requires an initial camera pose for each training image. As shown in Tab. 4, there is a chance that our model will fail when the angle between
Table 4. **Ablation study on camera optimization.** We highlight the best results in each column in orange. We also would like to mention that our full model converged on all scenes where the model w/o cam-opt converged as well. For PSNR metric, we report the average score of all scenes where our model converged.

| Methods           | $\Delta t = 0.1$ | $\Delta t = 0.2$ | $\Delta t = 10^\circ$ |
|-------------------|------------------|------------------|------------------------|
|                   | PSNR↑ | FMSE↓ | Succ No.↑ | PSNR↑ | FMSE↓ | Succ No.↑ | PSNR↑ | FMSE↓ | Succ No.↑ |
| Model w/o cam-opt |       |       |           |       |       |           |       |       |           |
| 21.20             | 0.00031 | 3/3     | 17.82     | 0.00124 | 1/3     | 19.36     | 0.00264 | 3/3     |
| 24.74             | 0.00029 | 3/3     | 23.81     | 0.00054 | 2/3     | 22.67     | 0.00146 | 3/3     |
| Full Model        |       |       |           |       |       |           |       |       |           |
| 20.38             | 0.01048 | 2/3     | 20.18     | 0.00300 | 3/3     | 15.94     | 0.01190 | 2/3     |
|                   |       |       |           |       |       |           |       |       |           |
| Model w/o cam-opt |       |       |           |       |       |           |       |       |           |
| 18.61             | 0.00860 | 2/3     | 22.22     | 0.00150 | 3/3     | 20.28     | 0.00945 | 2/3     |
| Full Model        |       |       |           |       |       |           |       |       |           |

**Fig. 9. Ablation Study on Imperfect Masks.** Results of our full model and model without transient embedding (Model w/o Transient) on the TV dataset with imperfect masks. The average MMSE scores on all testing images are also shown at the left-bottom corner of each result image.

**Fig. 10. Examples of Our Decomposition Results.** First row: from left to right: diffuse albedo map, normal map, specularity map, and Glossiness map. Second row: from left to right: color rendering w/o transient, transient blending weight, color rendering w/ transient, and ground-truth image.

the ground-truth poses and input poses is above 20 degrees, thus it is unlikely that our model will converge to accurate geometry with a trivial initialization of the camera poses. However, we believe this problem can be addressed by developing a more advanced camera optimization method as future work.

Although our rendering network can process complex and challenging shading in input images, e.g., sharp shadows, it does not support generating such components in novel scenes. While some works [Srinivasan et al. 2021; Zhang et al. 2021b] use terms such as...
| Ground Truth | Prediction | View #1 | View #2 | Base Color |
|--------------|------------|---------|---------|------------|

Fig. 13. Failure cases of our model: **Left Part:** An example of where our model failed to converge with high camera perturbations. The ground truth image is on the left and our prediction is on the right. **Right Part:** An example of where the ambient occlusion shading appears on the predicted base color map. From left to right are: two reference images from different views and the predicted base color map. Note that the shadow inside the grease appears both in the reference images and the base color map.

light visibility to represent these effects in a physically-based way, applying these techniques to data with varying unknown illumination is much more challenging, and requires further investigation. In addition, some shading that appears on many training images, for example ambient occlusion, may be misinterpreted as an object’s material property by our model. We are planning to explore these topics in our future work.

**6 CONCLUSION**

In this paper, we demonstrate that compelling capture, compositing, and relighting results are possible using only online image collections for which calibrated multi-view datasets captured in controlled settings are unavailable. This opens the door for many interesting future applications, such as re-rendering objects that may one day be rare or non-extant, represented in online image collections but never explicitly captured using multi-view techniques.

**REFERENCES**

Wang, A.B., Bergmann, B., Kellnhofer, and Gordon Wetzstein. 2021. Fast Training of Neural Lumigraph Representations using Meta Learning. In NeurIPS.

ACM Trans. Graph., Vol. 41, No. 4, Article 56. Publication date: July 2022.
