Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields

Dor Verbin[1,2] Peter Hedman[2] Ben Mildenhall[2]
Todd Zickler[1] Jonathan T. Barron[2] Pratul P. Srinivasan[2]
[1]Harvard University [2]Google Research

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-6946-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CVPR52688.2022.00541

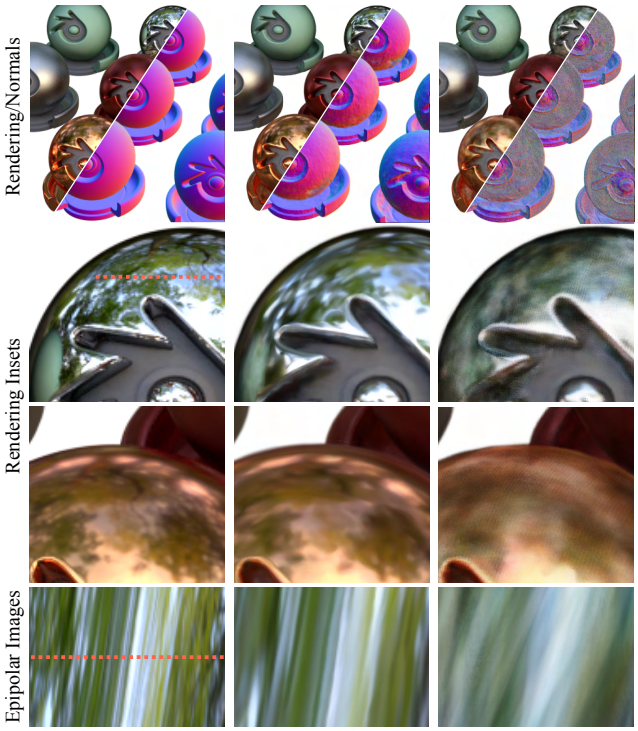

Figure 1. Ref-NeRF significantly improves normal vectors (top row) and visual realism (remaining rows) compared to mip-NeRF, the previous top-performing neural view synthesis model. Ref-NeRF's improvements are apparent in rendered frames (Rows 2 & 3), and even more in rendered videos (bottom row epipolar plane images and supplementary video), where its glossy highlights shift realistically across views instead of blurring and fading like mip-NeRF's. Image PSNR (higher is better) and surface normal mean angular error (lower is better) shown as insets.

is poorly-suited for interpolation. Figure 2 illustrates that, even for a simple toy setup, the scene's true radiance function varies quickly with view direction, especially around specular highlights. As a consequence, NeRF is only able to accurately render the appearance of scene points from

978-1-6654-6946-3/22/$31.00 ©2022 IEEE
DOI 10.1109/CVPR52688.2022.00541

the specific viewing directions obs
ages, and its interpolation of gloss
viewpoints is poor. Second, NeRF
reflections using isotropic emitters
of view-dependent radiance emitte
resulting in objects with semitrans

Our key insight is that structu
tion of view-dependent appearanc
ing function simpler and easier to
a model, which we call Ref-NeI
NeRF's directional MLP by provi
viewing vector about the local no
stead of the viewing vector itself.
illustrates that, for a toy scene com
under distant illumination, this *ref*
is constant across the scene (ignc
and interreflections) because it is
in surface orientation. Consequen
MLP acts as an interpolation ker
able to "share" observations of ap
points to render more realistic viev
terpolated views. We additionally
Directional Encoding technique, a
radiance into explicit diffuse and
allow the reflected radiance functi
spite variation in material and textu

While these improvements cru
to accurately interpolate view-dep
rely on the ability to reflect viewi
vectors estimated from NeRF's volumetric geometry. This presents a problem, as NeRF's geometry is foggy and not tightly concentrated at surfaces, and its normal vectors are too noisy to be useful for computing reflection directions (as shown in the right column of Figure 1). We ameliorate this issue with a novel regularizer for volume density that significantly improves the quality of NeRF's normal vectors and encourages volume density to concentrate around surfaces, enabling our model to compute accurate reflection vectors and render realistic specular reflections, shown in Figure 1.

To summarize, we make the following contributions:

1. A reparameterization of NeRF's outgoing radiance, based on the reflection of the viewing vector about the local normal vector (Section 3.1).
2. An Integrated Directional Encoding (Section 3.2) that, when coupled with a separation of diffuse and specular colors (Section 3.3), enables the reflected radiance function to be smoothly interpolated across scenes with varying materials and textures.
3. A regularization that concentrates volume density around surfaces and improves the orientation of NeRF's normal vectors (Section 4).

We apply these changes on top of mip-NeRF [2], currently the top-performing neural representation for view synthesis. Our experiments demonstrate that Ref-NeRF produces state-of-the-art renderings of novel viewpoints, and substantially improves upon the quality of previous top-performing view synthesis methods for highly specular or glossy objects. Furthermore, our structuring of outgoing radiance produces interpretable components (normal vectors, material roughness, diffuse texture, and specular tint) that enable convincing scene editing capabilities.

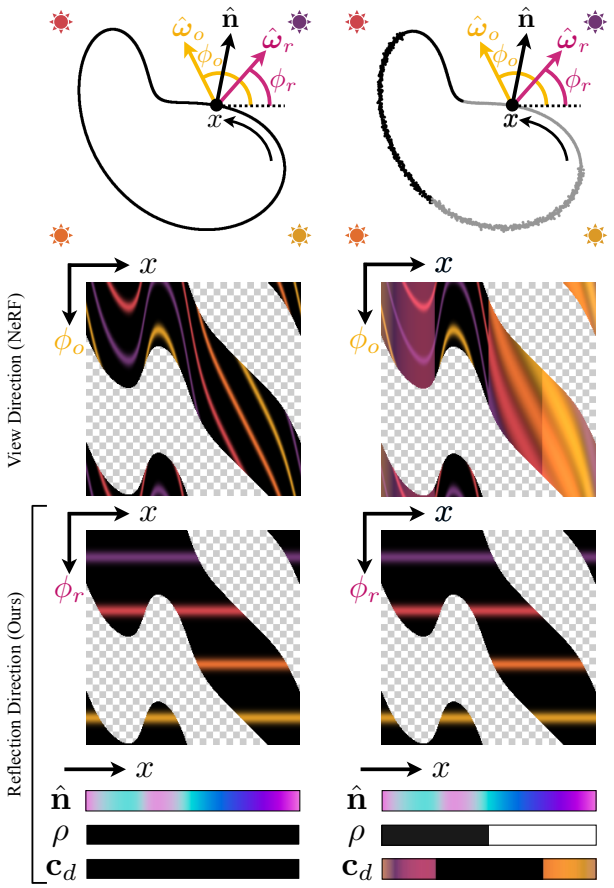

Figure 2. Visualizations of outgoing radiance in NeRF and Ref-NeRF, using 2D position-angle slices of radiance along an x-parameterized surface curve on a glossy object under colored lights. Because NeRF (middle row) uses view angle ϕ_o as input, when presented with glossy reflectances (left) or spatially-varying materials (right) it must interpolate between highly complicated functions like the irregularly curved colored lines shown here. In contrast, Ref-NeRF (bottom row) parameterizes radiance using a normal vector $\hat{\mathbf{n}}$ and a reflection angle ϕ_r, and adds diffuse color $\mathbf{c}_d$ and roughness ρ to its spatial MLP, which collectively makes radiance functions simple to model even for shiny or spatially-varying materials. The gray checkerboard indicates directions below the surface at position x.

2. Related Work

We review NeRF and related methods for photorealistic view synthesis, as well as techniques from computer graphics for capturing and rendering specular appearance.

3D scene representations for view synthesis View synthesis, the task of using observed images of a scene to render images from novel unobserved camera viewpoints, is a longstanding research problem within the fields of computer vision and graphics. In situations where it is possible to densely capture images of the scene, simple light field interpolation techniques [11, 16] can render novel views with high fidelity. However, exhaustive sampling of the light field is impractical in most scenarios, so methods for view synthesis from sparsely-captured images reconstruct 3D scene geometry in order to reproject observed images into novel viewpoints [7]. For scenes with glossy surfaces, some methods explicitly build virtual geometry to explain the motion of reflections [15, 31, 33]. Early approaches used triangle meshes as the geometry representation, and rendered novel views by reprojecting and blending multiple captured images with either heuristic [6, 8, 40] or learned [12, 30] blending algorithms. Recent works have used volumetric representations such as voxel grids [18] or multiplane images [9,21,35,39,46], which are better suited for gradient-based optimization than meshes. While these discrete volumetric representations can be effective for view synthesis, their cubic scaling limits their ability to represent large or high resolution scenes.

The recent paradigm of *coordinate-based neural representations* replaces traditional discrete representations with an MLP that maps from any continuous input 3D coordinate to the geometry and appearance of the scene at that location. NeRF [22] is an effective coordinate-based neural representation for photorealistic view synthesis that represents a scene as a field of particles that block and emit view-dependent light. NeRF has inspired many subsequent works, which extend its neural volumetric scene representation to application domains including dynamic and deformable scenes [24], avatar animation [10, 25], and even phototourism [19]. Our work focuses on improving a core component of NeRF: the representation of view-dependent appearance. We believe that the improvements presented here can be used to improve rendering quality in many of the applications of NeRF described above.

A key component of our approach considers the reflection of camera rays off NeRF's geometry. This idea is shared by recent works that extend NeRF to enable relighting by decomposing appearance into scene lighting and materials [3–5, 34, 43, 45]. Crucially, our model structures the scene into components that are *not* required to have precise physical meanings, and is thus able to avoid the strong simplifying assumptions (such as known lighting [3, 34], no self-occlusions [4, 5, 43], single-material scenes [43]) that these works need to make to recover explicit parametric representations of lighting and material. Our work also focuses on improving the smoothness and quality of normal vectors extracted from NeRF's geometry. This goal is shared by recent works that combine NeRF's neural volumetric representation with neural implicit surface representations [23,37,41]. However, these methods primarily focus on the quality of isosurfaces extracted from their representation as opposed to the quality of rendered novel views, and as such their view synthesis performance is significantly worse than top-performing NeRF-like models.

Efficient rendering of glossy appearance Our work takes inspiration from seminal approaches in computer graphics for representing and rendering view-dependent specular and reflective appearance, particularly techniques based on precomputation [27]. The reflected radiance function encoded in our directional MLP is similar to prefiltered environment maps [14, 29], which were introduced for real-time rendering of specular appearance. Prefiltered environment maps leverage the insight that the outgoing light from a surface can be seen as a spherical convolution of the incoming light and the (radially-symmetric) bidirectional reflectance distribution function (BRDF) that describes the material properties of the surface [28]. After storing this convolution result, rays intersecting the object can be rendered efficiently by simply indexing into the prefiltered environment maps with the reflection direction of the viewing vector about the normal vector.

Instead of rendering predefined 3D assets, our work leverages these computer graphics insights to solve a computer vision problem where we are recovering a renderable model of the scene from images. Furthermore, our directional MLP's representation of reflected radiance improves upon the prefiltered environment map representations used in computer graphics in a critical way: our directional MLP can represent spatial variation in reflected radiance due to spatial variation in both lighting and scene properties such as material roughness and texture, while the techniques described previously require computing and storing discrete prefiltered radiance maps for each possible material.

Our work is also inspired by a long line of works in computer graphics that reparameterize directional functions such as BRDFs [29, 32] and outgoing radiance [40] for improved interpolation and compression.

2.1. NeRF Preliminaries

NeRF [22] represents a scene as a volumetric field of particles that emit and absorb light. Given any input 3D position $\mathbf{x}$, NeRF uses a *spatial* MLP to output the density $\tau(\mathbf{x})$ of volumetric particles as well as a "bottleneck" vec-

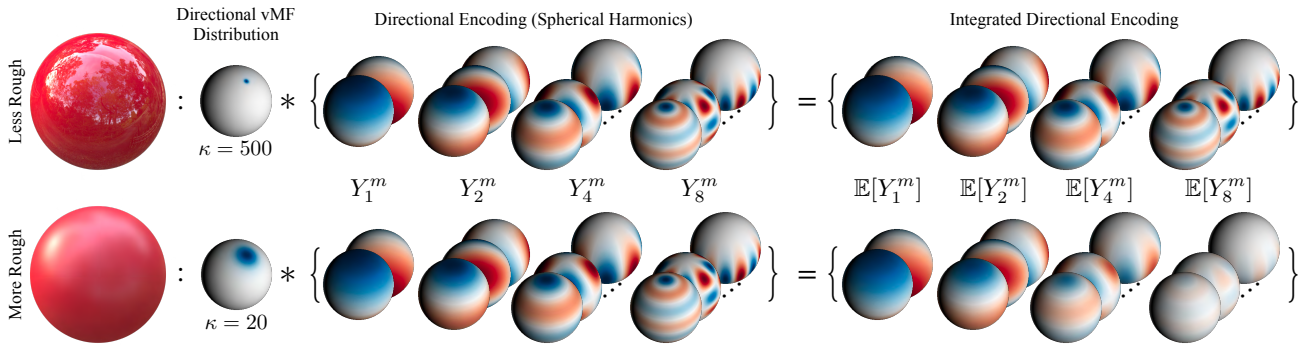

Figure 3. We enable the directional MLP to represent reflected radiance functions for any continuously-valued roughness using our integrated directional encoding. Each component of the encoding is a spherical harmonic function convolved with a vMF distribution with concentration parameter κ, output by our spatial MLP (equivalent to the expectation of the spherical harmonic under the vMF). Less rough locations receive higher-frequency encodings (top), while more rough regions receive encodings with attenuated high frequencies. Our IDE allows lighting information to be shared between locations with different roughnesses, and lets reflectance be edited.

tor $\mathbf{b}(\mathbf{x})$ which, along with the view direction $\hat{\mathbf{d}}$, is provided to a second *directional* MLP that outputs the color $\mathbf{c}(\mathbf{x}, \hat{\mathbf{d}})$ of light emitted by a particle at that 3D position at direction $\hat{\mathbf{d}}$ (see Figure 4 for a visualization). Note that Mildenhall *et al*. [22] use a single-layer directional MLP in their work, and that prior work often describes the combination of NeRF's spatial and directional MLPs as a single MLP.

The two MLPs are queried at points $\mathbf{x}_i = \mathbf{o} + t_i\hat{\mathbf{d}}$ along a ray originating at $\mathbf{o}$ with direction $\hat{\mathbf{d}}$, and return densities $\{\tau_i\}$ and colors $\{\mathbf{c}_i\}$. These densities and colors are alpha composited using numerical quadrature [20] to obtain the color of the pixel corresponding to the ray:

$$\mathbf{C}(\mathbf{o}, \hat{\mathbf{d}}) = \sum_i w_i \mathbf{c}_i \,, \tag{1}$$

$$\text{where} \quad w_i = e^{-\sum_{j<i} \tau_j (t_{j+1} - t_j)} \left(1 - e^{-\tau_i (t_{i+1} - t_i)}\right) .$$

The MLP parameters are optimized to minimize the L2 difference between each pixel's predicted color $\mathbf{C}(\mathbf{o}, \hat{\mathbf{d}})$ and its true color $\mathbf{C}_{\text{gt}}(\mathbf{o}, \hat{\mathbf{d}})$ taken from the input image:

$$\mathcal{L} = \sum_{\mathbf{o}, \hat{\mathbf{d}}} \|\mathbf{C}(\mathbf{o}, \hat{\mathbf{d}}) - \mathbf{C}_{\text{gt}}(\mathbf{o}, \hat{\mathbf{d}})\|^2 \,. \tag{2}$$

In practice, NeRF uses two sets of MLPs, one coarse and one fine, in a hierarchical sampling fashion, where both are trained to minimize the loss in Equation 2.

Prior NeRF-based models define a normal vector field in the scene by either using the spatial MLP to predict unit vectors [3, 45] at any 3D location, or by using the gradient of the volume density with respect to 3D position [4, 34]:

$$\hat{\mathbf{n}}(\mathbf{x}) = -\frac{\nabla \tau(\mathbf{x})}{\|\nabla \tau(\mathbf{x})\|} \,. \tag{3}$$

3. Structured View-Dependent Appearance

In this section, we describe how Ref-NeRF structures the outgoing radiance at each point into (prefiltered) incoming radiance, diffuse color, material roughness, and specular tint, which are better-suited for smooth interpolation across the scene than the function of outgoing radiance parameterized by view direction. By explicitly using these components in our directional MLP (Figure 4), Ref-NeRF can accurately reproduce the appearance of specular highlights and reflections. In addition, our model's decomposition of outgoing radiance enables scene editing.

3.1. Reflection Direction Parameterization

While NeRF directly uses view direction, we instead reparameterize outgoing radiance as a function of the reflection of the view direction about the local normal vector:

$$\hat{\boldsymbol{\omega}}_r = 2(\hat{\boldsymbol{\omega}}_o \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \hat{\boldsymbol{\omega}}_o, \tag{4}$$

where $\hat{\boldsymbol{\omega}}_o = -\hat{\mathbf{d}}$ is a unit vector pointing from a point in space to the camera, and $\hat{\mathbf{n}}$ is the normal vector at that point. As demonstrated in Figure 2, this reparameterization makes specular appearance better-suited for interpolation.

For BRDFs that are rotationally-symmetric about the reflected view direction, *i.e*. ones that satisfy $f(\hat{\boldsymbol{\omega}}_i, \hat{\boldsymbol{\omega}}_o) = p(\hat{\boldsymbol{\omega}}_r \cdot \hat{\boldsymbol{\omega}}_i)$ for some lobe function p (which includes BRDFs such as Phong [26]), and neglecting phenomena such as interreflections and self-occlusions, view-dependent radiance is a function of the reflection direction $\hat{\boldsymbol{\omega}}_r$ only:

$$L_{\text{out}}(\hat{\boldsymbol{\omega}}_o) \propto \int L_{\text{in}}(\hat{\boldsymbol{\omega}}_i) p(\hat{\boldsymbol{\omega}}_r \cdot \hat{\boldsymbol{\omega}}_i) d\hat{\boldsymbol{\omega}}_i = F(\hat{\boldsymbol{\omega}}_r). \tag{5}$$

Thus, by querying the directional MLP with the reflection direction, we are effectively training it to output this integral as a function of $\hat{\boldsymbol{\omega}}_r$. Because more general BRDFs may

vary with the angle between the view direction and normal vector due to phenomena such as Fresnel effects [14], we also input $\hat{\mathbf{n}} \cdot \hat{\boldsymbol{\omega}}_o$ to the directional MLP to allow the model to adjust the shape of the underlying BRDF.

3.2. Integrated Directional Encoding

In realistic scenes with spatially-varying materials, radiance cannot be represented as a function of reflection direction alone. The appearance of rougher materials changes slowly with reflection direction, while the appearance of smoother or shinier materials changes rapidly. We introduce a technique, which we call an *Integrated Directional Encoding (IDE)*, that enables the directional MLP to efficiently represent the function of outgoing radiance for materials with any continuously-valued roughness. Our IDE is inspired by the integrated positional encoding introduced by mip-NeRF [2] which enables the spatial MLP to represent prefiltered volume density for anti-aliasing.

First, instead of encoding directions with a set of sinusoids, as done in NeRF, we encode directions with a set of spherical harmonics $\{Y_\ell^m\}$. This encoding benefits from being stationary on the sphere, a property which is crucial to the effectiveness of positional encoding in Euclidean space [22, 36] (more details in our supplement).

Next, we enable the directional MLP to reason about materials with different roughnesses by encoding a distribution of reflection vectors instead of a single vector. We model this distribution defined on the unit sphere with a von Mises-Fisher (vMF) distribution (also known as a normalized spherical Gaussian), centered at reflection vector $\hat{\boldsymbol{\omega}}_r$, and with a concentration parameter κ defined as inverse roughness $\kappa = {}^1\!/\!_\rho$. The roughness ρ is output by the spatial MLP (using a softplus activation) and determines the roughness of the surface: a larger ρ value corresponds to a rougher surface with a wider vMF distribution. Our IDE encodes the distribution of reflection directions using the expected value of a set of spherical harmonics under this vMF distribution:

$$\text{IDE}(\hat{\boldsymbol{\omega}}_r, \kappa) = \left\{ \mathbb{E}_{\hat{\boldsymbol{\omega}} \sim \text{vMF}(\hat{\boldsymbol{\omega}}_r, \kappa)} [Y_\ell^m(\hat{\boldsymbol{\omega}})] \colon (\ell, m) \in \mathcal{M}_L \right\},$$
$$\text{with} \quad \mathcal{M}_L = \{(\ell, m) \colon \ell = 1, ..., 2^L, m = 0, ..., \ell\}. \tag{6}$$

In our supplement, we show that the expected value of any spherical harmonic under a vMF distribution has the following simple closed-form expression:

$$\mathbb{E}_{\hat{\boldsymbol{\omega}} \sim \text{vMF}(\hat{\boldsymbol{\omega}}_r, \kappa)} [Y_\ell^m(\hat{\boldsymbol{\omega}})] = A_\ell(\kappa) Y_\ell^m(\hat{\boldsymbol{\omega}}_r), \tag{7}$$

and that the ℓth attenuation function $A_\ell(\kappa)$ can be well-approximated using a simple exponential function:

$$A_\ell(\kappa) \approx \exp\left(-\frac{\ell(\ell+1)}{2\kappa} \right). \tag{8}$$

Figure 3 illustrates that our integrated directional encoding has an intuitive behavior; increasing the roughness of a material by lowering κ corresponds to attenuating the encoding's spherical harmonics with high orders ℓ, resulting in a wider interpolation kernel that limits the high frequencies in the represented view-dependent color.

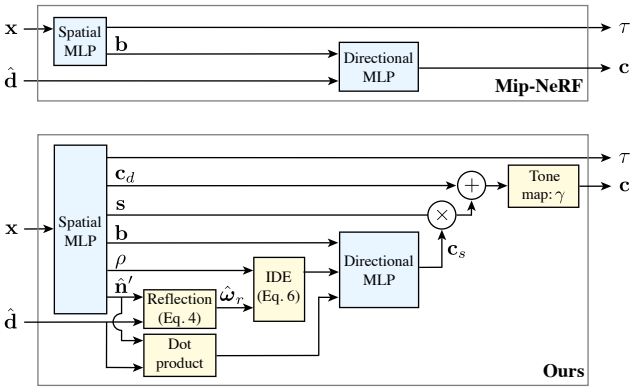

Figure 4. A visualization of mip-NeRF's and our architectures.

3.3. Diffuse and Specular Colors

We further simplify the function of outgoing radiance by separating the diffuse and specular components, using the fact that diffuse color is (by definition) a function of only position. We modify the spatial MLP to output a diffuse color $\mathbf{c}_d$ and a specular tint $\mathbf{s}$, and we combine this with the specular color $\mathbf{c}_s$ provided by the directional MLP to obtain a single color value:

$$\mathbf{c} = \gamma(\mathbf{c}_d + \mathbf{s} \odot \mathbf{c}_s), \tag{9}$$

where $\odot$ denotes elementwise multiplication, and γ is a fixed tone mapping function that converts linear color to sRGB [1] and clips the output color to lie in $[0, 1]$.

3.4. Additional Degrees of Freedom

Effects such as interreflections and self-occlusion of lighting cause illumination to vary spatially over a scene. We therefore additionally pass a bottleneck vector $\mathbf{b}$, output by the spatial MLP, into the directional MLP so the reflected radiance can change with 3D position.

4. Accurate Normal Vectors

While the structuring of outgoing radiance described in the previous section provides a better parameterization for the interpolation of specularities, it relies on a good estimation of volume density for facilitating accurate reflection direction vectors. However, the volume density field recovered by NeRF-based models suffers from two limitations: 1) normal vectors estimated from its volume density gradient as in Equation 3 are often extremely noisy (Figures 1 and 5); and 2) NeRF tends to "fake" specular highlights by embedding emitters inside the object and partially occluding them with a "foggy" diffuse surface (see Figure 5). This

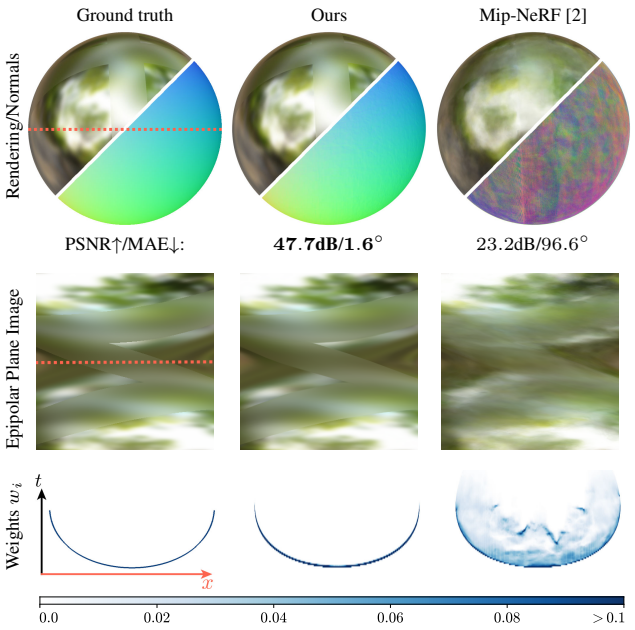

Figure 5. Prior top-performing NeRF-based approaches can fail catastrophically in highly-reflective scenes. Mip-NeRF (right column) produces blurry renderings of reflections that are inconsistent over different views (see EPI), and does not correctly simulate the appearance of the two different surface roughnesses. Our model (middle column) reconstructs the object almost perfectly. The accumulated normals and rendering weights w_i along the central scanline of the image (bottom row) show that mip-NeRF mimics specularities using emitters inside the object, while Ref-NeRF correctly recovers a concentrated surface.

is a suboptimal explanation, as it requires diffuse content on the surface to be semitransparent so that the embedded emitter can "shine through".

We address the first issue by using predicted normals for computing reflection directions: for each position $\mathbf{x}_i$ along a ray we output a 3-vector from the spatial MLP, which we then normalize to get a predicted normal $\hat{\mathbf{n}}'_i$. We tie these predicted normals to the underlying density gradient normal samples along each ray $\{\hat{\mathbf{n}}_i\}$ using a simple penalty:

$$\mathcal{R}_\mathrm{p} = \sum_i w_i \|\hat{\mathbf{n}}_i - \hat{\mathbf{n}}'_i\|^2, \tag{10}$$

where w_i is the weight of the ith sample along the ray, as defined in Equation 1. These MLP-predicted normals tend to be smoother than gradient density normals because the gradient operator acts as a high-pass filter on the MLP's effective interpolation kernel [36].

We address the second issue by introducing a novel regularization term that penalizes normals that are "backfacing", *i.e.* oriented away from the camera, at samples along the ray that contribute to the ray's rendered color:

$$\mathcal{R}_\mathrm{o} = \sum_i w_i \max(0, \hat{\mathbf{n}}'_i \cdot \hat{\mathbf{d}})^2. \tag{11}$$

This regularization acts as a penalty on "foggy" surfaces: samples are penalized when they are "visible" (high w_i) and the volume density is decreasing along the ray (*i.e.* the dot product between $\hat{\mathbf{n}}'_i$ and ray direction $\hat{\mathbf{d}}$ is positive). This normal orientation penalty prevents our method from explaining specularities as emitters hidden beneath a semi-transparent surface, and the resulting improved normals enable Ref-NeRF to compute accurate reflection directions for use in querying the directional MLP.

Throughout the paper, we use the gradient density normals for visualization and quantitative evaluation, as they directly demonstrate the quality of the underlying recovered scene geometry.

5. Experiments

We implement our model on top of mip-NeRF [2], an improved version of NeRF that reduces aliasing. We use the same spatial MLP architecture as mip-NeRF (8 layers, 256 hidden units, ReLU activations), but we use a larger directional MLP (8 layers, 256 hidden units, ReLU activations) than mip-NeRF to better represent high-frequency reflected radiance distributions. Please refer to our supplement for additional baseline implementation details.

We use the same quantitative metrics as previous view synthesis works [2, 22, 43]: PSNR, SSIM [38], and LPIPS [44] are used for evaluating rendering quality, and mean angular error (MAE) is used for evaluating estimated normal vectors.

Shiny Blender Dataset Though the "Blender" dataset used by NeRF [22] contains a variety of objects with complex geometry, it is severely limited in terms of material variety: most scenes are largely Lambertian. To probe more challenging material properties, we have created an additional "Shiny Blender" dataset with 6 different glossy objects rendered in Blender under conditions similar to NeRF's dataset (100 training and 200 testing images per scene). The quantitative results in Table 1 highlight the significant advantage of our model over mip-NeRF, the previous top-performing technique, for rendering novel views of these highly specular scenes. We also include three improved versions of mip-NeRF, all of which have an 8-layer directional MLP and, respectively: 1) no additional components; 2) normal vectors appended to the view direction in the directional MLP (as was done in IDR [42] and VolSDF [41]); and 3) our orientation loss applied to mip-NeRF's density gradient normal vectors. Our method significantly outperforms all of these improved variants of this previously top-performing neural view synthesis method, both in terms of novel view rendering quality and normal vector accuracy. Although PhySG [43] recovers more accurate normals, it requires ground-truth object masks (all

	PSNR ↑	SSIM ↑	LPIPS ↓	MAE° ↓
PhySG [43] (requires object masks)	26.21	0.921	0.121	8.46
Mip-NeRF [2]	29.76	0.942	0.092	60.38
Mip-NeRF, 8 layers	31.59	0.956	0.072	58.07
Mip-NeRF, 8 layers, w/ normals	31.39	0.955	0.074	58.27
Mip-NeRF, 8 layers, w/ $\mathcal{R}_o$	31.48	0.955	0.073	57.37
Ours, no reflection	29.47	0.944	0.084	16.19
Ours, no $\mathcal{R}_o$	31.62	0.954	0.078	52.56
Ours, no pred. normals	30.91	0.936	0.105	30.67
Ours, concat. viewdir	35.42	0.966	0.061	21.25
Ours, fixed lobe	35.52	0.965	0.061	26.46
Ours, no diffuse color	33.32	0.962	0.067	26.13
Ours, no tint	35.45	0.965	0.060	22.70
Ours, no roughness	33.39	0.963	0.065	25.96
Ours, standard encoding	35.90	0.968	0.058	20.31
Ours	35.96	0.967	0.058	18.38

Table 1. Baseline comparisons and ablation study on our "Shiny Blender" dataset.

	PSNR ↑	SSIM ↑	LPIPS ↓	MAE° ↓
PhySG [43] (requires object masks)	20.60	0.861	0.144	29.17
VolSDF [41]	27.96	0.932	0.096	19.45
NSVF [17]	31.74	0.953	0.047	–
NeRF [22]	32.38	0.957	0.046	–
Mip-NeRF [2]	33.09	0.961	0.043	38.30
Ours, standard encoding	33.90	0.965	0.039	24.16
Ours	33.99	0.966	0.038	23.22

Table 2. Results for our method compared to previous approaches on the Blender dataset [22].

other methods only require RGB images) and produces significantly worse renderings. Figure 5 showcases the impact of our approach using one object from our dataset: while mip-NeRF [2] fails to recover the geometry and appearance of this simple metallic sphere with two roughnesses, our method produces a nearly perfect reconstruction. Figure 9 displays another visual example from this dataset that showcases our model's improvements to recovered normal vectors and rendered specularities.

Table 1 also contains a quantitative ablation study of our model. If we use view directions instead of reflection directions as the directional MLP's input ("no reflection"), our method's reconstruction metrics drop significantly, showing the benefits of our reflected radiance parameterization. Removing the orientation loss ("no $\mathcal{R}_o$") results in severely degraded normals and renderings, and applying the orientation loss directly to the density field's normals and using those to compute reflection directions ("no pred. normals") also reduces performance. Including the view direction as input to the directional MLP in addition to the IDE ("concat. viewdir") slightly decreases performance, demonstrating the difficulty of parameterizing specular appearance as a function of viewing direction. On the other hand, not feeding $\hat{\mathbf{n}}' \cdot \hat{\boldsymbol{\omega}}_o$ to the directional MLP ("fixed lobe") also slightly reduces performance. Finally, removing our structural components of outgoing radiance (roughness, diffuse color, or tint) and replacing our IDE with a non-integrated directional encoding or NeRF's standard positional encoding (PE) all slightly decrease performance.

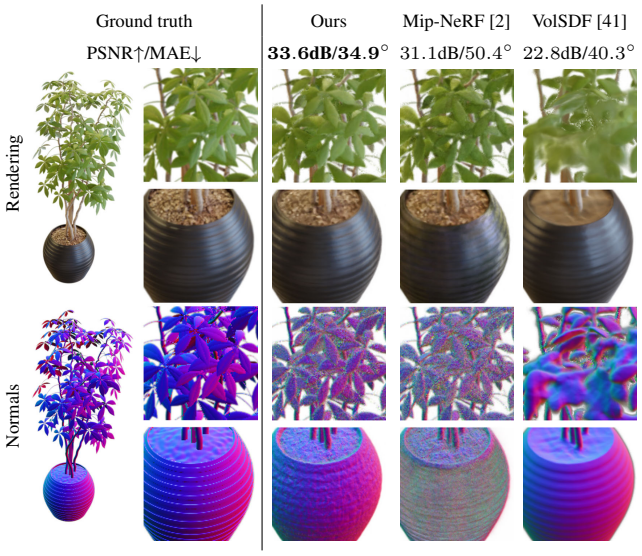

Figure 6. Our model renders accurate glossy appearance and recovers fine geometric details. VolSDF [42] estimates accurate specularities and normal vectors but often fails to capture fine-scale details, such as leaves. Mip-NeRF [2] is able to capture fine structures but fails to faithfully render specular highlights (such as those on the pot and leaves) in novel views, and does not recover accurate normal vectors.

Blender Dataset We also compare Ref-NeRF to recent neural view synthesis baseline methods on the standard Blender dataset from the original NeRF paper [22]. Table 2 shows that our method outperforms all prior work across all image quality metrics. Our method also yields a large (35%) improvement in the MAE of its normal vectors relative to mip-NeRF. While the hybrid surface-volume VolSDF representation [41] recovers slightly more accurate normal vectors (15% lower MAE), our PSNR is substantially higher (6dB) than theirs. Additionally, VolSDF tends to oversmooth geometry, which makes our results qualitatively superior upon inspection (see Figure 6).

Real Captured Scenes In addition to these two synthetic datasets, we evaluate our model on a set of 3 real captured scenes. We capture a "sedan" scene, and use the "garden spheres" and "toy car" captures from the Sparse Neural Radiance Grids paper [13]. Figure 8 and our supplement demonstrate that our rendered specular reflections and recovered normal vectors are often much more accurate on these real-world scenes.

Scene Editing Our structuring of outgoing radiance enables view-consistent editing of scenes. Although we do not perform a full inverse-rendering decomposition of appearance into BRDFs and lighting, our individual components behave intuitively and enable visually plausible scene editing results which are not attainable from a standard NeRF.

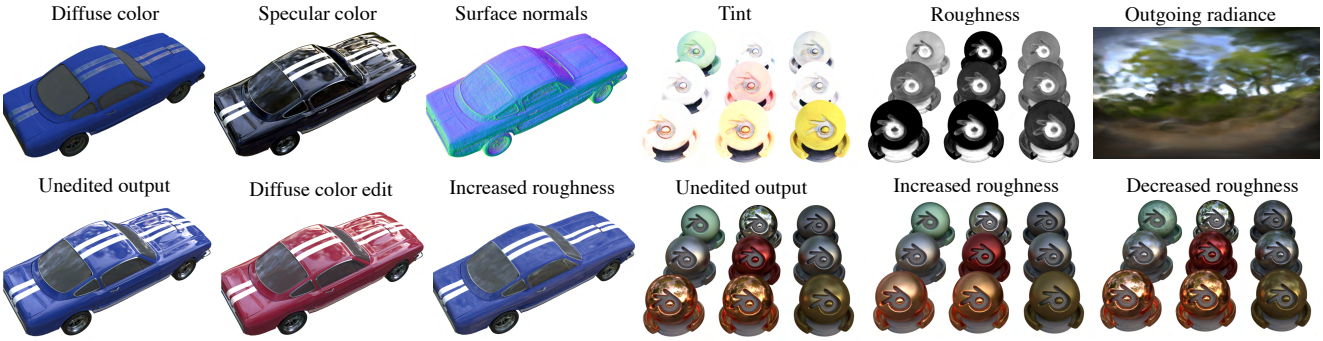

Figure 7. Our model's structuring of outgoing radiance decomposes scene appearance into interpretable components (top row) that enable editing (bottom row). Note that the recovered outgoing radiance function for a point on the chrome material ball (top right) is a plausible reconstruction of the actual scene lighting. We can edit the diffuse color of the car without affecting the specular reflections off its glossy paint, and we can plausibly modify the roughness of the car and material balls by manipulating the κ values used in the IDE.

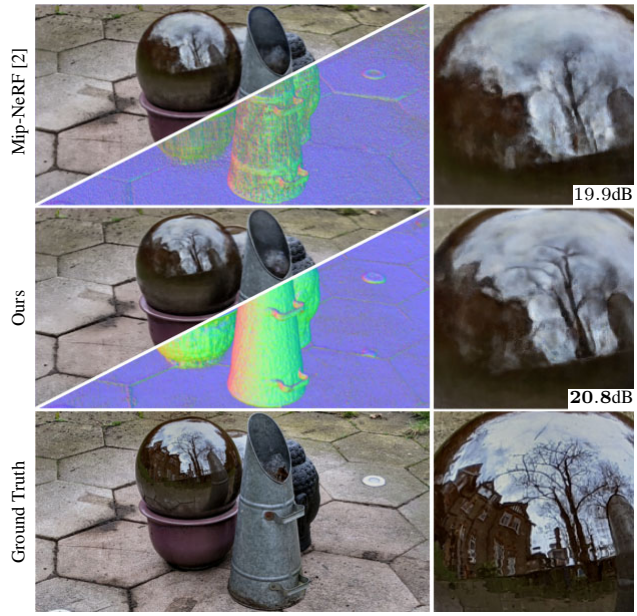

Figure 8. In this "garden spheres" scene, mip-NeRF's foggy geometry (see rendered normals) leads to blurred reflections (see inset, with PSNRs), while our model is able to recover accurate normal vectors and render more realistic reflections.

Figure 7 shows example edits of the scene's components, and our supplementary video contains additional examples that demonstrate the view-consistency of our edited models.

Limitations While Ref-NeRF significantly improves upon previous top-performing neural scene representations for view synthesis, it requires increased computation: evaluating our integrated directional encoding is slightly slower than computing a standard positional encoding, and back-propagating through the gradient of the spatial MLP to compute normal vectors makes our model roughly 25% slower than mip-NeRF. Our reparameterization of outgoing radiance by the reflection direction does not explicitly model interreflections or non-distant illumination, so our improvement upon mip-NeRF is reduced in such cases.

Figure 9. On the "coffee" scene from our "Shiny Blender" dataset our method succeeds at estimating normals and interpolating specularities, whereas mip-NeRF [2] fails at doing both (see reflections on the spoon for example).

6. Conclusion

We have demonstrated that prior neural representations for view synthesis fail to accurately represent and render scenes with specularities and reflections. Our model, Ref-NeRF, introduces a new parameterization and structuring of view-dependent outgoing radiance, as well as a regularizer on normal vectors. These contributions allow Ref-NeRF to significantly improve both the quality of view-dependent appearance and the accuracy of normal vectors in synthesized views of the scene. We believe that this work makes important progress towards capturing and reproducing the rich photorealistic appearance of objects and scenes.

Acknowledgements We would like to thank Lior Yariv and Kai Zhang for helping us evaluate their methods, and Ricardo Martin-Brualla for helpful comments on our text. DV is supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (an NSF AI Institute, `http://iaifi.org`).

References

[1] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet—sRGB. *Color and imaging conference*, 1996. 5

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 1, 2, 5, 6, 7, 8

[3] Sai Bi, Zexiang Xu, Pratul P. Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv*, 2020. 3, 4

[4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. NeRD: Neural reflectance decomposition from image collections. *ICCV*, 2021. 3, 4

[5] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P. A. Lensch. Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. *NeurIPS*, 2021. 3

[6] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. *SIGGRAPH*, 2001. 3

[7] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. *SIGGRAPH*, 1993. 3

[8] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach. *SIGGRAPH*, 1996. 3

[9] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. DeepView: View synthesis with learned gradient descent. *CVPR*, 2019. 3

[10] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. *CVPR*, 2021. 3

[11] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *SIGGRAPH*, 1996. 3

[12] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2018. 3

[13] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021. 7

[14] Jan Kautz and Michael D. McCool. Approximation of glossy reflection with prefiltered environment maps. *Graphics Interface*, 2000. 3, 5

[15] Johannes Kopf, Fabian Langguth, Daniel Scharstein, Richard Szeliski, and Michael Goesele. Image-based rendering in the gradient domain. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2013. 3

[16] Marc Levoy and Pat Hanrahan. Light field rendering. *SIGGRAPH*, 1996. 3

[17] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 7

[18] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (SIGGRAPH)*, 2019. 3

[19] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. *CVPR*, 2021. 3

[20] Nelson Max. Optical models for direct volume rendering. *IEEE TVCG*, 1995. 4

[21] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (SIGGRAPH)*, 2019. 3

[22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 1, 3, 4, 5, 6, 7

[23] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *ICCV*, 2021. 3

[24] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 3

[25] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. *ICCV*, 2021. 3

[26] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 1975. 4

[27] Ravi Ramamoorthi. Precomputation-based rendering. *Foundations and Trends in Computer Graphics and Vision*, 2009. 3

[28] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. *SIGGRAPH*, 2001. 3

[29] Ravi Ramamoorthi and Pat Hanrahan. Frequency space environment map rendering. *SIGGRAPH*, 2002. 3

[30] Gernot Riegler and Vladlen Koltun. Free view synthesis. *ECCV*, 2020. 3

[31] Simon Rodriguez, Siddhant Prakash, Peter Hedman, and George Drettakis. Image-based rendering of cars using semantic labels and approximate reflection flow. *Proc. ACM Comput. Graph. Interact. Tech.*, 2020. 3

[32] Szymon M. Rusinkiewicz. A new change of variables for efficient BRDF representation. *Eurographics Rendering Workshop*, 1998. 3

[33] Sudipta N. Sinha, Johannes Kopf, Michael Goesele, Daniel Scharstein, and Richard Szeliski. Image-based rendering for scenes with reflections. *ACM Transactions on Graphics (SIGGRAPH)*, 2012. 3

[34] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron.

NeRV: Neural reflectance and visibility fields for relighting and view synthesis. *CVPR*, 2021. 3, 4

[35] Pratul P. Srinivasan, Richard Tucker Joand nathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. *CVPR*, 2019. 3

[36] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 5, 6

[37] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 3

[38] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 6

[39] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. NeX: Real-time view synthesis with neural basis expansion. *CVPR*, 2021. 3

[40] Daniel Wood, Daniel Azuma, Wyvern Aldinger, Brian Curless, Tom Duchamp, David Salesin, and Werner Stuetzle. Surface light fields for 3D photography. *SIGGRAPH*, 2000. 3

[41] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 2021. 3, 6, 7

[42] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 2020. 6, 7

[43] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. *CVPR*, 2021. 3, 6, 7

[44] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. 6

[45] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2021. 3, 4

[46] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (SIGGRAPH)*, 2018. 3