This document contains additional implementation details for our method, as well as additional qualitative results from the experiments discussed in the main paper. Please view our included supplementary video for a brief overview of our method, qualitative results with smoothly-moving novel light and camera paths, and demonstrations of additional graphics applications.

A. BRDF Parameterization

We use the standard microfacet bi-directional reflectance distribution function (BRDF) described by Walter *et al.* [55] as our reflectance function, and incorporate some of the simplifications discussed in the BRDF implementations of the Filament [15] and Unreal Engine [20] renderering engines. The BRDF $R(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ we use is defined for any 3D location $\mathbf{x}$, incoming lighting direction $\boldsymbol{\omega}_i$, and outgoing reflection direction $\boldsymbol{\omega}_o$ as:

$$\begin{aligned} R(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) =& \frac{D(\mathbf{h}, \mathbf{n}, \gamma) F(\boldsymbol{\omega}_i, \mathbf{h}) G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \gamma)}{4(\mathbf{n} \cdot \boldsymbol{\omega}_o)} \\ &+ (\mathbf{n} \cdot \boldsymbol{\omega}_i)(1 - F(\boldsymbol{\omega}_i, \mathbf{h}))\left(\frac{\mathbf{a}}{\pi}\right), \end{aligned} \tag{17}$$

$$D(\mathbf{h}, \mathbf{n}, \gamma) = \frac{\rho^2}{\pi((\mathbf{n} \cdot \mathbf{h})^2(\rho^2 - 1) + 1)^2}, \tag{18}$$

$$F(\boldsymbol{\omega}_i, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\boldsymbol{\omega}_i \cdot \mathbf{h}))^5, \tag{19}$$

$$G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \gamma) = \frac{(\mathbf{n} \cdot \boldsymbol{\omega}_o)(\mathbf{n} \cdot \boldsymbol{\omega}_i)}{((\mathbf{n} \cdot \boldsymbol{\omega}_o)(1-k)+k)((\mathbf{n} \cdot \boldsymbol{\omega}_i)(1-k)+k)}, \tag{20}$$

$$\rho = \gamma^2, \qquad \mathbf{h} = \frac{\boldsymbol{\omega}_o + \boldsymbol{\omega}_i}{\|\boldsymbol{\omega}_o + \boldsymbol{\omega}_i\|}, \qquad k = \frac{\gamma^4}{2}, \tag{21}$$

where $\mathbf{a}$ is the diffuse albedo, γ is the roughness, and $\mathbf{n}$ is the surface normal at 3D point $\mathbf{x}$. We use $F_0 = 0.04$, which is the typical value of dielectric (non-conducting) materials. Note that our definition of the BRDF includes the multiplication by the Lambert cosine term $(\mathbf{n} \cdot \boldsymbol{\omega}_i)$ in order to simplify the equations in the main paper.

B. Additional Qualitative Results

Figure 10 shows additional renderings from NeRV and other baseline methods. We see that NeRV is able to recover effective relightable 3D scene representations from images of scenes with complex illumination conditions. Prior work such as Bi *et al.* [3] are unable to recover accurate representations from images with lighting conditions more complex than a single point light. Latent code methods (representative of "NeRF in the Wild" [28]) are unable to generalize to simulate lighting conditions unlike those seen during training.

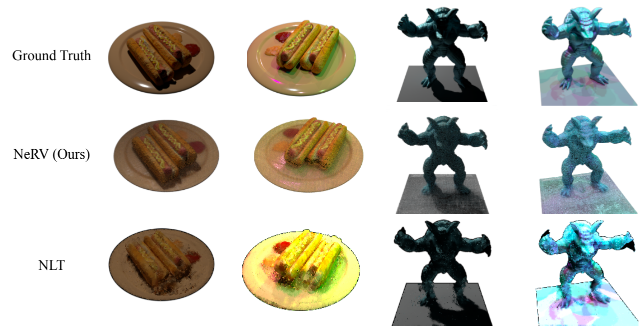

Figure 9: Additional qualitative results, specifically comparing images rendered by NeRV to those rendered by the Neural Light Transport [61] (NLT) baseline. Note that NLT uses a controlled laboratory lighting setup with eight times as many images as used by NeRV, and an input proxy geometry (which is recovered by training a NeRF [32] model on a set of images with fixed illumination). The artifacts seen in the shadows of NLT's renderings demonstrate the difference between recovering geometry that works well for view synthesis (as NLT does) and recovering geometry that works well for both view synthesis and relighting (as NeRV does).

C. Limitations

Recovering a NeRV is a straightforward optimization problem: we optimize the parameters of the MLPs that comprise a NeRV scene representation to minimize the error of re-rendering the input images. NeRV currently does not incorporate any priors into the optimization problem, so a promising direction for future work would be to integrate priors on geometry and reflectance (such as learned priors or simple hand-crafted priors to encourage smooth geometry or reflectance predictions) into the NeRV optimization so that a relightable 3D scene representation could be recovered from fewer viewpoints or fewer observed lighting conditions.

Successfully recovering a NeRV representation relies on jointly optimizing the geometry, reflectance, and visibility MLPs. We have noticed failure cases where the reflectance MLP seems to converge faster than the geometry and visibility MLPs and is stuck in a local minimum. For example, in cases where the scene is observed under very few illumination conditions, the reflectance MLP sometimes quickly converges to include shadows and light tints in the recovered albedo, and is not able to recover even after the visibility MLP catches up to correctly explain those shadows. Further investigations into the optimization landscape and dynamics of NeRV could help shed light on this issue.

Finally, the NeRV optimization problem trains a geometry MLP along with a visibility MLP that is meant to approximate integrals of the geometry MLP's output. Though we impose a loss that encourages these two MLPs to be consistent with each other, there is no guarantee that these two MLPs will be exactly consistent. Investigating potential strategies to enforce such consistency may be helpful.

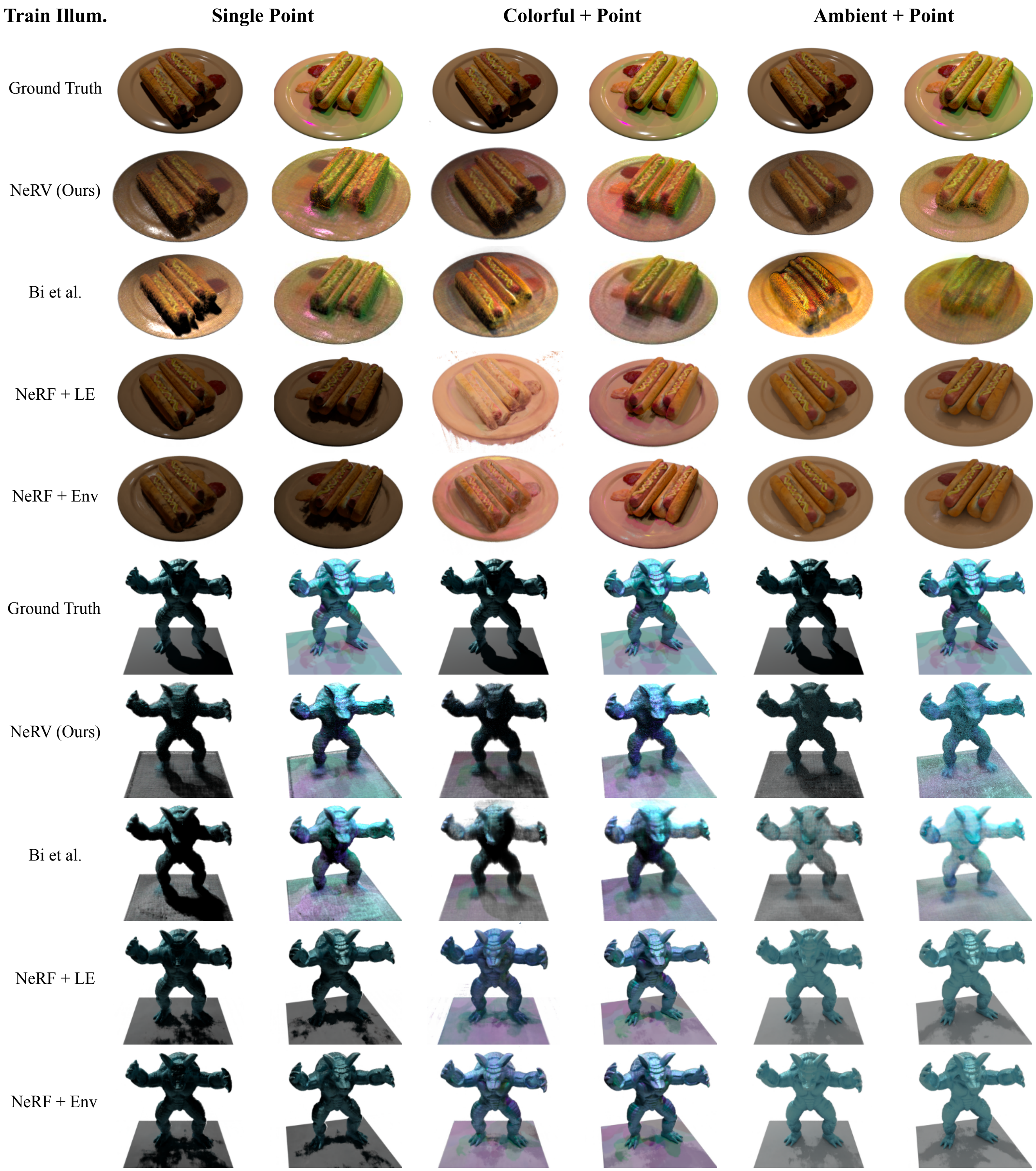

Figure 10: Additional qualitative results from the experiments discussed in the main paper. We can see that NeRV is able to render convincing images from novel viewpoints under novel lighting conditions. The method of Bi *et al*. [3] is unable to recover accurate models when trained with illumination more complex than a single point light (columns 3-6). Methods that use latent codes to explain variation in appearance due to lighting (NeRF+LE, NeRF+Env) are unable to generalize to lighting conditions different than those seen during training.