FlipNeRF: Flipped Reflection Rays for Few-shot Novel View Synthesis

Seunghyeon Seo Yeonjin Chang Nojun Kwak
Seoul National Univeristy
{zzzlssh, yjean8315, nojunk}@snu.ac.kr

Abstract

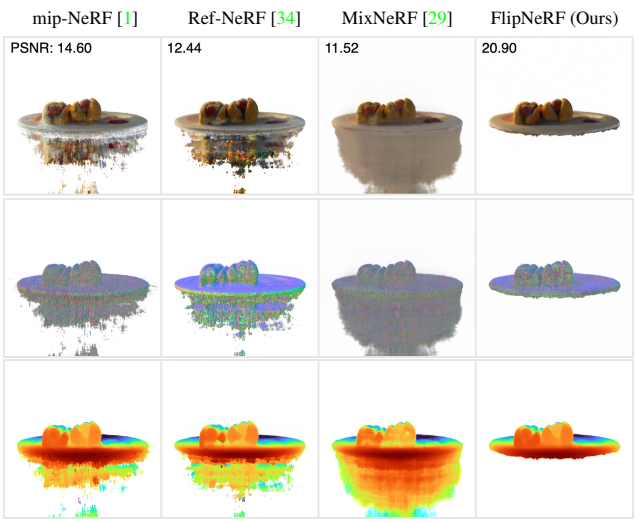
Neural Radiance Field (NeRF) has been a mainstream in novel view synthesis with its remarkable quality of rendered images and simple architecture. Although NeRF has been developed in various directions improving continuously its performance, the necessity of a dense set of multi-view images still exists as a stumbling block to progress for practical application. In this work, we propose FlipNeRF, a novel regularization method for few-shot novel view synthesis by utilizing our proposed flipped reflection rays. The flipped reflection rays are explicitly derived from the input ray directions and estimated normal vectors, and play a role of effective additional training rays while enabling to estimate more accurate surface normals and learn the 3D geometry effectively. Since the surface normal and the scene depth are both derived from the estimated densities along a ray, the accurate surface normal leads to more exact depth estimation, which is a key factor for few-shot novel view synthesis. Furthermore, with our proposed Uncertainty-aware Emptiness Loss and Bottleneck Feature Consistency Loss, FlipNeRF is able to estimate more reliable outputs with reducing floating artifacts effectively across the different scene structures, and enhance the feature-level consistency between the pair of the rays cast toward the photo-consistent pixels without any additional feature extractor, respectively. Our FlipNeRF achieves the SOTA performance on the multiple benchmarks across all the scenarios.

1. Introduction

Neural Radiance Field (NeRF) [22] has achieved great success in rendering photo-realistic images from novel viewpoints. However, the necessity of a dense set of training images remains as a practical bottleneck since it suffers from significant performance degradation when trained with sparse views.

There are two mainstreams for few-shot novel view synthesis: *pre-training* and *regularization* methods, both of which focus on learning the 3D geometry efficiently from sparse inputs. The pre-training methods [41, 4, 5, 37, 18, 11, 17, 27, 33, 13] require large-scale datasets consisting of different scenes with multi-view images for injecting prior knowledge during the pre-training, while the regularization methods [23, 29, 14, 10, 28, 7, 15] are optimized per scene, exploiting additional training resources, *e.g.* unseen viewpoints [23, 14, 15], depth map generation [7, 28], off-the-

(a) Sparse training views



(b) Synthesized unseen views

Figure 1: **Synthesis results from sparse inputs.** Our FlipNeRF significantly improves rendering quality compared to other baselines. Compared to the vanilla mip-NeRF [1] and MixNeRF [29], which is the state-of-the-art regularization method, ours reduces the noises and floating artifacts noticeably with superior surface normal estimation. Although Ref-NeRF [34] estimates smooth normal vectors, it shows much inferior rendering results to ours with a large chunk of noise under the few-shot setting.

shelf models [10, 23], and so on, for an effective regularization to alleviate overfitting. Although the prior arts achieved promising results in novel view synthesis from sparse inputs, there still exist hurdles to overcome. The large-scale datasets, which are used for pre-training methods, are expensive to collect and the NeRF model is prone to performance degradation for the out-of-distribution dataset. On the other hand, the regularization methods heavily rely on additional training resources which might not always be available and require many heuristic factors, *e.g*. the choice of off-the-shelf models, the hyperparameters for sampling unseen viewpoints, and so on.

In this paper, we propose *FlipNeRF*, which is an effective regularization method exploiting the flipped reflection rays[1] as additional training resources without any heuristic factors for the generation process. We derive a batch of flipped reflection rays from the original ray directions and estimated surface normals so that they are cast toward the same target pixels of the original input ray. Compared to the existing regularization methods which have mainly focused on the accurate depth estimation from limited input views [29, 23, 14, 28, 7], our FlipNeRF is trained to reconstruct surface normals accurately by learning to generate effective reflection rays to be used in training. Since both estimated surface normals and depths are derived from the volume densities representing underlying 3D geometry, accurately estimating the surface normals of an object naturally leads to more accurate depth maps.

Furthermore, we propose an effective regularization loss, *Uncertainty-aware Emptiness Loss (UE Loss)*, to reduce the floating artifacts effectively while considering the uncertainty of the model's outputs by using the estimated scale parameters for mixture models. Since our FlipNeRF is built upon MixNeRF [29], which is a regularization method achieving promising results by modeling input rays with mixture density models [2], we are able to apply our proposed loss without any modification of the architecture by using the estimated scale parameters of each sample along a ray, which stand for the uncertainty of the samples' estimated probability density distributions.

Additionally, inspired by [5, 10, 14] which address the feature-level consistency of targets under the sparse input setting, we encourage the consistency for the pairs of bottleneck features between the original input rays and flipped reflection rays. We leverage a Jensen-Shannon Divergence, which is based on the similarity between the probability distributions, to make the pairs of bottleneck feature distributions of original and flipped reflection rays more similar to each other improving feature consistency.

We demonstrate the effectiveness of our proposed FlipNeRF through the experiments on the multiple benchmarks, *e.g*. Realistic Synthetic 360° [22], DTU [12], and LLFF [21]. Our method achieves state-of-the-art (SOTA) performances compared to other baselines. Especially, ours outperforms other baselines by a large margin with more accurate surface normals under the extremely sparse settings such as 3/4-view setting which are the most challenging ones. Our contributions are summarized as follows:

- We propose an effective training framework for NeRF with sparse training views, called *FlipNeRF*. It leverages flipped reflection rays to provide additional training resources, resulting in more precise surface normals and eliminating the need for heuristic factors when sampling unseen views.
- We also propose an effective regularization loss, *Uncertainty-aware Emptiness Loss (UE Loss)*, which reduces floating artifacts with considering the uncertainty of outputs, leading to more reliable estimation.
- We enhance the consistency of bottleneck features between the original input rays and flipped reflection rays by Jensen-Shannon Divergence, coined as *Bottleneck Feature Consistency Loss (BFC Loss)*, improving the robustness for rendering from unseen viewpoints.
- Our FlipNeRF achieves SOTA performance over the multiple benchmarks. Especially, ours outperforms other baselines by a large margin in more challenging scenarios, *e.g*. 3/4-view.

2. Related Works

2.1. Neural Radiance Field

Recently, Neural Radiance Field (NeRF) [22] has shown impressive performance and potential in the novel view synthesis task. NeRF represents a scene with an MLP, mapping coordinates and viewing directions to its colors and volume density, and then creates a novel view through volume rendering. Subsequent studies have developed NeRF in several directions, *e.g*. using conical frustums instead of rays [1], reparameterizing an input viewing direction as its reflection direction [34], and so on. These works have made significant progress by addressing the various issues in novel view synthesis, but there still exists a limitation in that NeRF requires a dense set of training images and a lengthy training time. Many studies have addressed these issues [9, 26, 4], including the utilization of various data structures for faster training and inference [8, 40] and attempts to train NeRF with only a few training images. Our work focuses on enhancing the performance of NeRF when a sparse set of views are provided as training images.

2.2. Few-Shot Novel View Synthesis

There are two main approaches for a few-shot novel view synthesis: the *pre-training* and the *regularization* method. The pre-training methods require a large dataset of multi-

[1]The term 'flipped' is used because the reflected ray has an opposite direction (from an object to a camera).

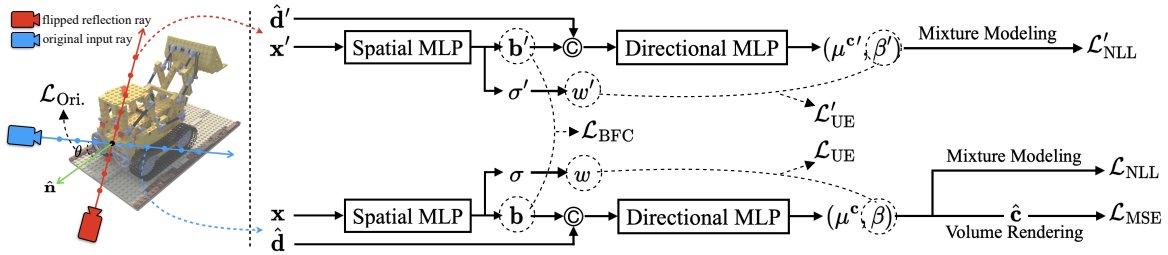


Figure 2: **Overall framework of FlipNeRF.** Our FlipNeRF utilize the newly generated flipped reflection rays with our proposed UE Loss and BFC Loss as well as existing MSE, NLL, and Orientation losses. See Sec. 3 and Fig. 3 for more details about generation process of flipped reflection rays and the loss terms.

view scenes to provide prior knowledges of 3D geometry to a NeRF model and then optionally finetune on the target scene [4, 5, 11, 17, 27, 33, 37, 41]. On the contrary, the regularization methods [29, 14, 10, 23, 7, 28, 15] are optimized per scene without pre-training process by exploiting additional training resources, *e.g.* depth maps [7, 28] and semantic consistency [10], as an extra supervision. Among them, [23, 14, 15] adopt an unseen viewpoint sampling strategy to make up for insufficient training views. However, these sampling processes require many hand-designed factors such as the ranges of rotation, translation, jittering, and so on, which can introduce artificial biases. Our work proposes a novel regularization approach to derive a set of flipped reflection rays from estimated surface normals and utilizes these for regularization, which can eliminate heuristic factors of existing methods, resulting in more effective training strategy with limited inputs.

2.3. Surface Normal Reconstruction

There is a line of research to recover accurate textures and lighting conditions of objects with NeRF [24, 39, 36, 6, 32, 42, 3]. Although these earlier studies successfully reconstruct the high-quality isosurfaces derived from the scene representations, their rendering quality for novel views is still inferior to the NeRF-like models. Meanwhile, Ref-NeRF [34] achieved superior performance with remarkable quality of surface normals compared to the existing NeRF models. Since the normal vectors utilized in NeRF framework are derived from the negative normalized density gradients [32, 3, 34], which represent the underlying geometry of 3D scenes, learning an accurate density distribution along a ray is a key factor for surface normal reconstruction. However, for the few-shot novel view synthesis, the prior works mostly focus on the accurate depth estimation without attention to the surface normals, both of which are derived from the estimated volume densities. In this work, we approach the few-shot novel view synthesis problem with focusing on the surface normals, which is another critical factor for an effective learning of 3D scene geometry. To the best of our knowledge, our work is the first attempt to focus on the surface normal estimation for few-shot novel view synthesis.

3. Method

In this work, we propose an effective regularization method for few-shot novel view synthesis with flipped reflection rays. Our FlipNeRF is built upon MixNeRF [29] which leverages a mixture model framework (Sec. 3.1). We derive a batch of flipped reflection rays and cast them toward the identical target pixels as additional training rays (Sec. 3.2). Furthermore, we propose the *Uncertainty-aware Emptiness Loss* and *Bottleneck Feature Consistency Loss* to alleviate the floating artifacts adaptively based on the uncertainty and enhance the consistency between the bottleneck feature distributions of the original and flipped reflection rays, respectively (Sec. 3.3 and Sec. 3.4). Finally, our FlipNeRF is trained to minimize the MSE and NLL losses as well as the proposed regularization loss terms with their corresponding balancing weights (Sec. 3.5). Fig. 2 shows an overview of our FlipNeRF.

3.1. Preliminaries

NeRF. The NeRF [22], which is an MLP-based neural network, represents a 3D scene as a continuous radiance field of RGB color and volume density. For every point sampled along a ray, the 3D coordinates $\mathbf{x} = (x, y, z)$ and viewing directions (θ, ϕ) are mapped to the colors $\mathbf{c} = (r, g, b)$ and densities σ:

$$F(\gamma(\mathbf{x}), \gamma(\hat{\mathbf{d}})) \rightarrow (\mathbf{c}, \sigma), \tag{1}$$

where $F(\cdot)$, $\gamma(\cdot)$, and $\hat{\mathbf{d}}$ indicate an MLP, the positional encoding for the inputs, and the 3D Cartesian unit vector used as an input viewing direction in practice, respectively.

The volumetric radiance field is rendered by alpha compositing the RGB values along an input ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ [20], where $\mathbf{o}$ and $\mathbf{d}$ denote the camera origin and unnormalized direction vector, *i.e.* $\mathbf{d} = \|\mathbf{d}\|_2 \cdot \hat{\mathbf{d}}$, respectively. The volume rendering integrals are denoted as follows:

$$\hat{\mathbf{c}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \hat{\mathbf{d}})\, dt, \tag{2}$$

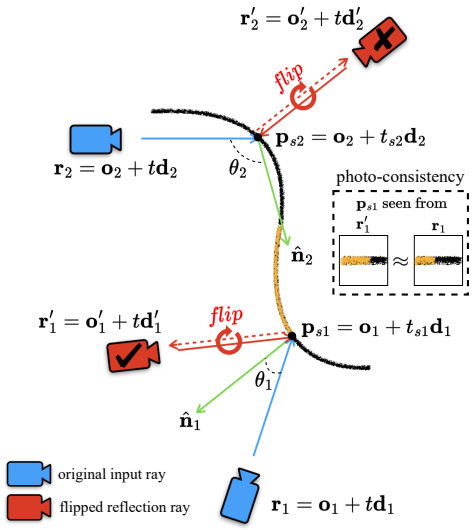


Figure 3: **Flipped reflection ray generation.** Our FlipNeRF generates the flipped reflection ray $\mathbf{r}'$ from the estimated normal vector $\hat{\mathbf{n}}$ and original input ray direction $\mathbf{d}$. With our masking strategy, $\mathbf{r}'_2$ is filtered out since it does not satisfy the photo-consistency condition, *i.e.* θ_2 is bigger than 90°. The smaller θ is, the more photo-consistent the target pixel is, where the pair of $\mathbf{r}$ and $\mathbf{r}'$ are cast.

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(s)\, ds)$ indicates the degree of transparency. In practice, it is approximated with numerical quadrature [22] by sampling points along a ray.

The radiance field is trained to minimize the mean squared error (MSE) between rendered and GT pixels:

$$\mathcal{L}_{\text{MSE}} = \sum_{\mathbf{r}\in\mathcal{R}} ||\hat{\mathbf{c}}(\mathbf{r}) - \mathbf{c}^{\text{GT}}(\mathbf{r})||_2^2, \tag{3}$$

where $\mathcal{R}$ is denoted as a set of rays.

MixNeRF. Built upon mip-NeRF [1], which leveraged a cone tracing method and proposed an integrated positional encoding to address an aliasing problem, MixNeRF [29] estimates the joint probability distribution of color values and models a ray with a mixture of densities:

$$p(\mathbf{c}|\mathbf{r}) = \sum_{i=1}^{M} \pi_i \mathcal{F}(\mathbf{c}; \mu_i^{\mathbf{c}}, \beta_i), \tag{4}$$

where M is the number of sampled points, $\mathcal{F}(\mathbf{c}; \mu_i^{\mathbf{c}}, \beta_i)$ denotes the Laplacian distribution of RGB $\mathbf{c}$ with location parameter $\mu_i^{\mathbf{c}} \in \{\mu_i^r, \mu_i^g, \mu_i^b\}$, *i.e.* estimated RGB values of sample, and scale parameter $\beta_i \in \{\beta_i^r, \beta_i^g, \beta_i^b\}$. The mixture coefficient π_i is derived from the estimated volume density σ_i as follows:

$$\pi_i = \frac{w_i}{\sum_{m=1}^{M} w_m} = \frac{T_i(1-\exp(-\sigma_i\delta_i))}{\sum_{m=1}^{M} T_m(1-\exp(-\sigma_m\delta_m))}, \tag{5}$$

where w_i and δ_i indicate the alpha blending weight and sample interval, respectively. Thanks to the mixture model's capacity of representing complex distributions, MixNeRF learns the density distribution effectively with sparse inputs by minimizing the negative log-likelihood (NLL) in Eq. 4.

Our FlipNeRF is built upon MixNeRF leveraging the mixture modeling framework while achieving superior rendering quality with noticeably fewer artifacts and more accurate surface normals to MixNeRF.

3.2. Auxiliary Flipped Reflection Ray

As shown in Fig. 3, we exploit a batch of flipped reflection rays $\mathbf{r}' \in \mathcal{R}'$ as extra training resources, which are derived from the original input ray directions $\mathbf{d}$ and estimated surface normals $\hat{\mathbf{n}}$. First, we derive a flipped reflection direction $\mathbf{d}'$ from $\mathbf{d}$ and $\hat{\mathbf{n}}$:

$$\mathbf{d}' = 2(\mathbf{d}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}} - \mathbf{d}, \tag{6}$$

where $\hat{\mathbf{n}}$ denotes the weighted sum of blending weights and estimated normal vectors along a ray, *i.e.* $\hat{\mathbf{n}} = \sum_{i=1}^{M} w_i \mathbf{n}_i$. [2] Note that we use the gradient of volume density as estimated surface normals following [3, 32, 34].

To generate the additional training rays based on $\mathbf{d}'$, we need a set of imaginary ray origins $\mathbf{o}'$ located in a suitable space considering the hitting point and the original input ray origins $\mathbf{o}$. Since the vanilla NeRF models, which are trained with a dense set of images, tend to have the blending weight distribution whose peak is located on the point around the object surface $\mathbf{p}_s = \mathbf{o} + t_s\mathbf{d}$ [7, 29], *i.e.* the s-th sample whose blending weight is the highest along a ray. Therefore, we place $\mathbf{o}'$ so that the s-th sample of $\mathbf{r}'$ is $\mathbf{p}_s$:

$$\mathbf{o}' = \mathbf{p}_s - t_s\mathbf{d}', \tag{7}$$

resulting in our proposed flipped reflection ray, $\mathbf{r}'(t) = \mathbf{o}' + t\mathbf{d}'$. Compared to the previous unseen viewpoint sampling strategies [14, 23, 15], our proposed strategy does not rely on the randomness of unseen viewpoint sampling and reduces the heuristic factors for sampling schemes, *e.g.* the range of rotation, translation, and so on. Furthermore, since our newly generated $\mathbf{r}'$ are cast on the identical object surfaces where the original input rays $\mathbf{r}$ are cast, *i.e.* the target pixels are photo-consistent for the pair of $\mathbf{r}$ and $\mathbf{r}'$ without any sophisticated viewpoint sampling process, we are able to train $\mathbf{r}'$ effectively with the same GT pixels of $\mathbf{r}$.

However, since $\hat{\mathbf{n}}$, which are used to derive $\mathbf{d}'$, are not the ground truth but the estimation, there exists a concern that even miscreated $\mathbf{r}'$, which do not satisfy photo-consistency, can be used for training. As a result, it might lead to performance degradation while providing misleading training

[2] Technically, $\hat{\mathbf{n}}$ is not guaranteed to be a unit vector without an explicit normalization process. However, we empirically found that the normalization rather destabilizes the training and leads to the performance degradation. Kindly refer to our supplementary material for related experiments.

cues. To address this problem, we mask the ineffective $\mathbf{r}'$ by considering the angle θ between $\hat{\mathbf{n}}$ and $-\hat{\mathbf{d}}$ as follows:

$$M(\mathbf{r}') = \begin{cases} 1 & \text{if } \arccos\left(-(\hat{\mathbf{d}} \cdot \hat{\mathbf{n}})\right) < \tau \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where $-(\hat{\mathbf{d}} \cdot \hat{\mathbf{n}})$ amounts to $\cos\theta$ of original input rays and normal vectors, and τ indicates the threshold for filtering the invalid rays, which we set as 90° unless specified. Through this masking process, only $\mathbf{r}'$ which are cast toward the photo-consistent point can be remained as we intend. Finally, our proposed flipped reflection rays are modeled by mixture density like the original input rays:

$$p(\mathbf{c}|\mathbf{r}') = \sum_{i=1}^{M} \pi_i' \mathcal{F}(\mathbf{c}; \mu_i^{\mathbf{c}'}, \beta_i'). \tag{9}$$

Additionally, we leverage the *Orientation Loss* proposed in Ref-NeRF [34] to penalize the backward-facing normal vectors for learning accurate surface normals:

$$l_{\text{Ori.}}(\mathbf{r}) = \sum_{i=1}^{M} w_i \max(0, \mathbf{n}_i \cdot \hat{\mathbf{d}})^2. \tag{10}$$

Unlike Ref-NeRF, we penalize underlying density gradient normal $\mathbf{n}_i$ instead of predicted normals.

Note that our FlipNeRF is fundamentally different compared to Ref-NeRF since ours generates additional training rays through the derivation of reflection direction without modification to original representations while Ref-NeRF replaced the input viewing direction with its reflection direction, reparameterizing the outgoing radiance.

3.3. Uncertainty-aware Regularization

Several regularization techniques have been proposed to reduce the floating artifacts present in synthesized images, which is one of the major problems of NeRF. Among them, we leverage the *Emptiness Loss* [35] which penalizes the small blending weights along a ray as follows:

$$l_{\text{Emp.}}(\mathbf{r}) = \frac{1}{M} \sum_{i=1}^{M} \log(1 + \eta \cdot w_i), \tag{11}$$

where the bigger η is, the steeper the loss function becomes around 0.

However, the naive application of existing regularization techniques with limited training views might not be consistently helpful across the different scenes due to the scene-by-scene different structure, resulting in overall performance degradation. To address this problem, we propose *Uncertainty-aware Emptiness Loss (UE Loss)* developed upon the Emptiness Loss, which reduces the floating artifacts consistently over the different scenes by considering the output uncertainty:

$$\begin{aligned} l_{\text{UE}}(\mathbf{r}) &= \frac{1}{M} \sum_{i=1}^{M} \log(1 + \rho \cdot \eta \cdot w_i), \\ \text{where} \quad \rho &= \frac{1}{3} \sum_{c}^{\{r,g,b\}} \sum_{i=1}^{M} \beta_i^c. \end{aligned} \tag{12}$$

ρ amounts to the average of the summation of estimated scale parameters of RGB color distributions from all samples along a ray, which we use as the uncertainty of a ray. By our proposed UE Loss, we are able to regularize the blending weights adaptively, *i.e.* the more uncertain a ray is, the more penalized the blending weights along the ray are. It is able to reduce floating artifacts consistently across the scenes with different structures and enables to synthesize more reliable outputs by considering uncertainty.

3.4. Bottleneck Feature Consistency

Motivated by previous works addressing the feature-level consistency of multiple views for few-shot novel view synthesis [5, 10, 14], we encourage the consistency of bottleneck feature distributions between $\mathbf{r}$ and $\mathbf{r}'$, which are intermediate feature vectors, *i.e.* outputs of the spatial MLP of NeRF, by Jensen-Shannon Divergence (JSD):

$$l_{\text{BFC}}(\mathbf{r}, \mathbf{r}') = JSD(\psi(\mathbf{b}), \psi(\mathbf{b}')), \tag{13}$$

where $\psi(\cdot)$, $\mathbf{b}$ and $\mathbf{b}'$ denote the softmax function, the bottleneck features of $\mathbf{r}$ and $\mathbf{r}'$, respectively. While the existing methods [5, 10] rely on off-the-shelf feature extractors like 2D CNN or CLIP [25] to address high-level feature consistency, we regulate the pair of features effectively by enhancing consistency between bottleneck features without depending on additional feature extractors.

3.5. Total Loss

Our FlipNeRF is not only trained to maximize the log-likelihood of the target pixel $\mathbf{c}_{\text{GT}}$ for a set of original input rays $\mathcal{R}$, but also for flipped reflection rays $\mathcal{R}'_M$, where the ineffective rays are excluded from the total flipped reflection rays $\mathcal{R}'$ by our masking strategy in Eq. 8. Likewise, the UE Losses are applied for both $\mathcal{R}$ and $\mathcal{R}'_M$.

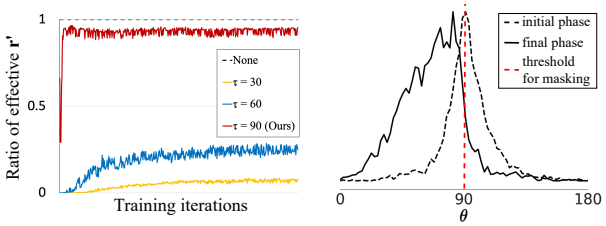
Aggregating all, our total loss over a batch is as follows:

$$\begin{aligned} \mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{MSE}} &+ \lambda_1 \mathcal{L}_{\text{NLL}} + \lambda_2 \mathcal{L}'_{\text{NLL}} \\ &+ \lambda_3 \mathcal{L}_{\text{UE}} + \lambda_4 \mathcal{L}'_{\text{UE}} + \lambda_5 \mathcal{L}_{\text{BFC}} + \lambda_6 \mathcal{L}_{\text{Ori.}}, \end{aligned} \tag{14}$$

where a set of λ's are balancing weight terms for the losses. More details about our proposed losses and training schemes are provided in the supp. material.

τ	PSNR ↑	SSIM ↑	LPIPS ↓	Average Err. ↓
30°	18.62	0.747	0.206	0.121
60°	18.12	0.723	0.237	0.126
90°	**19.55**	**0.767**	**0.180**	**0.101**
180° (No masking)	18.76	0.755	0.190	0.111

Table 1: **Comparison of masking conditions.** Our masking strategy with τ of 90° achieves the best results, filtering out the ineffective flipped reflection rays successfully.



(a) Effective $\mathbf{r}'$ ratio by τ (b) Distribution of θ

Figure 4: **Analysis of the masking strategy.**

4. Experiments

4.1. Experimental Settings

Datasets and metrics. We evaluate the performance of our FlipNeRF and baselines on the representative benchmarks: Realistic Synthetic 360° [22], DTU [12], and LLFF [21]. Realistic Synthetic 360° contains 8 synthetic scenes, each consisting of 400 multi-view rendered images with white background. For DTU, which provide various scenes including objects put on a white table with a black background, we conduct a series of experiments for the analysis of FlipNeRF under 3-view setting as well as comparison against other baselines. Additionally, we compare our FlipNeRF against other baselines on LLFF consisting of real forward-facing scenes, which is often tested as an out-of-distribution dataset for pre-training methods. We follow the experimental protocols from [41, 22, 29].

For the quantitative evaluation for rendered images, we adopt the mean of PSNR, SSIM [38], LPIPS [43], and the geometric average [1]. Furthermore, we also adopt the mean angular error (MAE°) [34] and NLL [31, 30, 19, 16] for evaluating the surface normals and uncertainty, respectively. Further details of experimental protocols and evaluation metrics are provided in our supp. material.

Baselines. We compare our FlipNeRF against the SOTA regularization methods [10, 14, 23, 29] on Realistic Synthetic 360° as well as the vanilla mip-NeRF [1] and Ref-NeRF [34], which is known for achieving promising results with accurate surface normals. Furthermore, we compare ours against the representative pre-training methods [4, 5, 41] as well as regularization methods on DTU and LLFF. The pre-training baselines exploit the DTU and LLFF as pre-training dataset and out-of-distribution test set, respectively, while the regularization methods, mip-NeRF and Ref-NeRF are optimized per scene. Note that we report the quantitative results of other baselines on DTU and LLFF from [23], which achieved better results than its original papers by the modified training scheme, and those on Realistic Synthetic 360° from [29], which trained the baselines with the identical training views for a fair comparison.

	PSNR ↑	SSIM ↑	LPIPS ↓	Average ↓	NLL ↓
MixNeRF [29]	18.95	0.744	0.203	0.113	9.99
FlipNeRF (Ours)					
w/o $\mathcal{L}_{\text{Emp.}}$ or $\mathcal{L}_{\text{UE}}$	19.30	0.758	0.196	0.108	4.88
w/ $\mathcal{L}_{\text{Emptiness}}$	18.62	0.749	0.204	0.118	4.92
w/ $\mathcal{L}_{\text{UE}}$	**19.55**	**0.767**	**0.180**	**0.101**	**2.56**

Table 2: **Effectiveness of $\mathcal{L}_{\text{UE}}$.** Our proposed $\mathcal{L}_{\text{UE}}$ improves the rendering quality consistently across the scenes with considering the uncertainty.

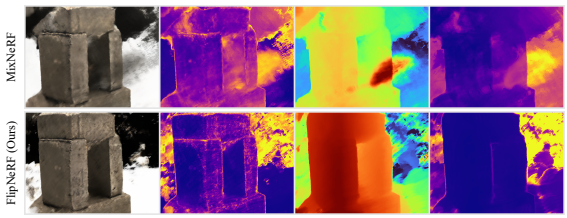


Figure 5: **Comparison of FlipNeRF and MixNeRF.** Our FlipNeRF renders the images from novel views with much fewer artifacts and more accurate depth maps than MixNeRF. Considering the standard deviation of RGB and depths, our FlipNeRF is able to estimate more reliable outputs than MixNeRF. For the std. map, the darker the pixel is, the more certain the output is.

4.2. Analysis of FlipNeRF

Analysis of flipped reflection rays. As shown in Tab. 1, our masking strategy of filtering out the ineffective flipped reflection rays with the threshold (τ) of 90° achieves the best performance among different options. With τ of 30° and 60°, the rendering quality is rather degraded since the newly generated rays are overly-filtered and do not provide enough additional supervision as demonstrated in Fig. 4a. On the other hand, when we exploit all the flipped reflection rays without masking, there exists a little improvement of performance compared to 30° and 60° masking, but it is still much inferior to 90° due to the negative impact from the ineffective rays. Additionally, Fig. 4b shows the distribution of θ, *i.e.* angles between the input viewing directions and normal vectors. As the smaller θ is, the more photo-consistent the target pixel is, *i.e.* the more effective the newly generated flipped reflection rays are for training. Since our FlipNeRF is trained to estimate the accurate normal vectors, we are able to exploit a set of more effective flipped reflection rays through the training, which are cast on the more photo-consistent target pixels. Furthermore, at the initial training phase, the invalid additional rays are filtered effectively by our masking strategy, leading to high-

	PSNR ↑	SSIM ↑	LPIPS ↓	Average Err. ↓
w/o $\mathcal{L}_{\text{BFC}}$	18.79	0.755	0.200	0.115
w/ $\mathcal{L}_{\text{BFC}}$				
MSE on $\mathbf{b}$	18.83	0.755	0.197	0.113
Cos. Sim. on $\mathbf{b}$	18.77	0.755	0.193	0.115
JSD on $\mathbf{b}_{\hat{\mathbf{d}}}$	18.43	0.749	0.204	0.120
JSD on $\mathbf{b}$	**19.55**	**0.767**	**0.180**	**0.101**

Table 3: **Comparison of different strategies for $\mathcal{L}_{\text{BFC}}$.** Our $\mathcal{L}_{\text{BFC}}$ achieves a significant performance gain compared to other regularization schemes for feature consistency.

	$\mathcal{L}_{\text{NLL}}$	$\mathcal{L}'_{\text{NLL}}$	$\mathcal{L}_{\text{UE}}{}^{\dagger}$	$\mathcal{L}_{\text{BFC}}$	$\mathcal{L}_{\text{Ori.}}$	PSNR ↑	SSIM ↑	LPIPS ↓	Average ↓
(1)	✓					17.44	0.729	0.187	0.130
(2)	✓	✓				16.94	0.727	0.217	0.143
(3)	✓	✓	✓			17.89	0.736	0.206	0.130
(4)	✓	✓	✓	✓		19.01	0.755	0.181	0.107
(5)	✓	✓	✓		✓	18.79	0.755	0.200	0.115
(6)	✓	✓		✓	✓	19.30	0.758	0.196	0.108
(7)	✓	✓	✓	✓	✓	**19.55**	**0.767**	**0.180**	**0.101**

Table 4: **Ablation study.** † indicate that the losses are applied to both $\mathbf{r}$ and $\mathbf{r}'$.

quality supervision and stabilizing the training.

Uncertainty-aware regularization. Tab. 2 shows the effectiveness of our proposed UE Loss. Compared to MixNeRF [29] which models a ray with mixture of distributions as our FlipNeRF, ours consistently achieves more reliable rendering results with much lower NLL. Without $\mathcal{L}_{\text{Emp.}}$ [35] or our proposed $\mathcal{L}_{\text{UE}}$, ours already outperforms MixNeRF by a large margin. However, ours with naively leveraged $\mathcal{L}_{\text{Emp.}}$ rather shows inferior results to MixNeRF. It shows that naive application of the existing regularization technique for reducing artifacts under the few-shot setting can lead to overall performance degradation due to the scene-by-scene various structures. As illustrated in Fig. 5, with our proposed $\mathcal{L}_{\text{UE}}$, ours improves both of the rendering quality and the reliability of the model outputs by a large margin compared to MixNeRF.

Bottleneck feature consistency. As shown in Tab. 3, there is no significant impact on the performance with regularizing the bottleneck feature $\mathbf{b}$ using MSE or cosine similarity. By our proposed $\mathcal{L}_{\text{BFC}}$ with JSD, we achieve a considerable performance improvement. Interestingly, the performance rather degrades when we apply $\mathcal{L}_{\text{BFC}}$ with JSD to $\mathbf{b}_{\hat{\mathbf{d}}}$, *i.e.* the bottleneck feature conditioned with input viewing direction $\hat{\mathbf{d}}$. We conjecture that the reduced feature dimension of $\mathbf{b}_{\hat{\mathbf{d}}}$ reduces the capacity of feature representations and prevents the model from improving robustness.

4.3. Ablation Study

The quantitative results of our ablation study are reported in Tab. 4. With only additionally exploiting our proposed flipped reflection rays, our FlipNeRF achieves more degenerate results compared to the baseline ((1) → (2)). However, we are able to achieve performance improvement by a large margin with our proposed $\mathcal{L}_{\text{UE}}$ and $\mathcal{L}_{\text{BFC}}$ ((2) → (3) → (4)). (5) → (7) shows that enhancing the consistency of bottleneck features between the pair of original and flipped reflection rays is considerably effective. Additionally, by leveraging $\mathcal{L}_{\text{Ori.}}$ from Ref-NeRF, we are able to estimate more accurate normal vectors, leading to more effective flipped reflection rays and performance gain ((4) → (7)).

	PSNR ↑	SSIM ↑	LPIPS ↓	Average Err. ↓
mip-NeRF [1]	14.62	0.351	0.495	0.246
DietNeRF [10]	14.94	0.370	0.496	0.240
RegNeRF [23]	19.08	0.587	0.336	0.146
MixNeRF [29]	19.27	0.629	0.236	0.124
FlipNeRF (Ours)	**19.34**	**0.631**	**0.235**	**0.123**

Table 5: **Additional results on LLFF 3-view.** More comparison with pre-training methods is provided in supp. mat.

4.4. Comparison with other SOTA Methods

Realistic Synthetic 360°. As demonstrated in Tab. 6, our FlipNeRF achieves the SOTA performance across all the evaluation metrics. Compared to MixNeRF which leverages a mixture model framework as ours, our FlipNeRF improves the performance by a large margin. Noticeably, our FlipNeRF estimates more accurate surface normals than other baselines, leading to the performance gain with better reconstructed fine details from limited input views as shown in Fig. 6a. Additionally, the vanilla Ref-NeRF, which shows great performance with accurate normal vectors, achieves comparable or even better performance than other regularization methods except ours. From this result, we are able to expect that estimating the accurate surface normals is one of the key factors for learning 3D geometry with sparse inputs. Note that the comparable MAE° of RegNeRF results from the overly-smoothed depth estimation, not indicating the high-quality of rendering results, as shown in Fig. 6a.

DTU. Our FlipNeRF achieves the best results across all the scenarios and most of the evaluation metrics on DTU as shown in Tab. 7. Similar to the results on Realistic Synthetic 360°, ours outperforms other baselines by a large margin especially under the 3-view, which is the most challenging scenario, with reducing the floating artifacts successfully as shown in Fig. 6b. Since the flipped reflection rays are effective training resources for unseen views, the fewer the training views are provided, the more performance gain is expected. Kindly refer to our supp. mat. for the comparison with pre-training methods.

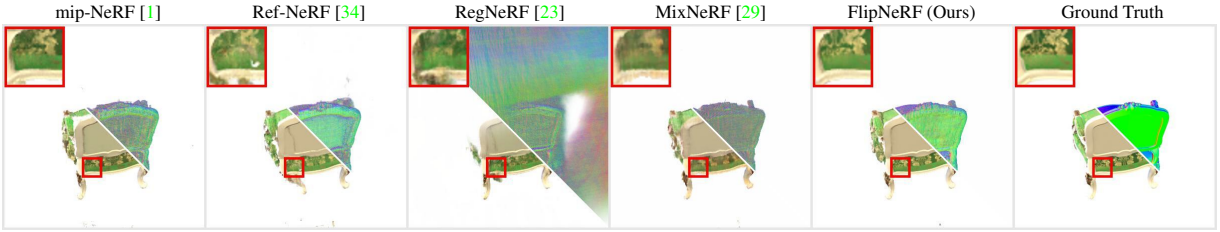
LLFF. Table 5 compares our FlipNeRF against other baselines on LLFF, which is a real forward-facing dataset. Although ours achieves the SOTA performance among other baselines, it is much more marginal than those on Realistic Synthetic 360° and DTU. We conjecture the reason for the marginal improvement of our proposed method can be the fact that a set of flipped reflection rays, which are able to widely cover the unseen views, are not very useful for the scenes in LLFF, where a set of camera poses are much less

	PSNR ↑		SSIM ↑		LPIPS ↓		Average Err. ↓		MAE° ↓	
	4-view	8-view	4-view	8-view	4-view	8-view	4-view	8-view	4-view	8-view
mip-NeRF [1]	14.12	18.74	0.722	0.828	0.382	0.238	0.221	0.121	96.05	101.21
Ref-NeRF [34]	18.09	24.00	0.764	0.879	0.269	0.106	0.150	0.058	65.62	57.93
DietNeRF [10]	15.42	21.31	0.730	0.847	0.314	0.153	0.201	0.086	-	-
InfoNeRF [14]	18.44	22.01	0.792	0.852	0.223	0.133	0.119	0.073	-	-
RegNeRF [23]	13.71	19.11	0.786	0.841	0.346	0.200	0.210	0.122	62.78	60.37
MixNeRF [29]	18.99	23.84	0.807	0.878	0.199	0.103	0.113	0.060	70.90	62.04
FlipNeRF (Ours)	**20.60**	**24.38**	**0.822**	**0.883**	**0.159**	**0.095**	**0.091**	**0.055**	**58.72**	**57.17**

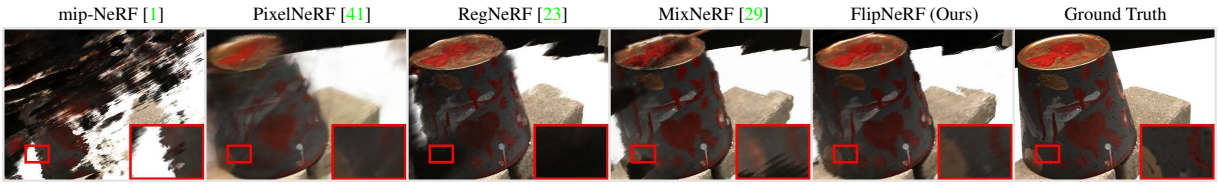
Table 6: **Quantitative results on Realistic Synthetic 360°.** Our FlipNeRF achieves the SOTA performance among other baselines across all the scenarios and metrics.

	PSNR ↑			SSIM ↑			LPIPS ↓			Average Error ↓		
	3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view
mip-NeRF [1]	8.68	16.54	23.58	0.571	0.741	0.879	0.353	0.198	0.092	0.323	0.148	0.056
DietNeRF [10]	11.85	20.63	23.83	0.633	0.778	0.823	0.314	0.201	0.173	0.243	0.101	0.068
RegNeRF [23]	18.89	22.20	24.93	0.745	**0.841**	**0.884**	0.190	0.117	0.089	0.112	0.071	0.047
MixNeRF [29]	18.95	22.30	25.03	0.744	0.835	0.879	0.203	0.102	0.065	0.113	0.066	0.042
FlipNeRF (Ours)	**19.55**	**22.45**	**25.12**	**0.767**	0.839	0.882	**0.180**	**0.098**	**0.062**	**0.101**	**0.064**	**0.041**

Table 7: **Quantitative results on DTU.** Our FlipNeRF outperforms other baselines in every scenario, especially by a large margin under the 3-view setting. More comparison with pre-training methods is provided in supp. material.



(a) Realistic Synthetic 360° 4-view



(b) DTU 3-view

Figure 6: **Qualitative results on Realistic Synthetic 360° and DTU.** More results are provided in the supp. material.

dynamic than other datasets. In other words, our FlipNeRF is able to not only achieve a competitive performance for the scenes consisting of a set of simple camera poses, but also render the novel views in much higher quality for more dynamically captured scenes with only a few shots. The comparison with pre-training methods and rendered images are provided in the supp. material.

5. Conclusion

In this work, we have focused on accurate surface normals, which is another key factor for the few-shot novel view synthesis. Our proposed FlipNeRF utilizes a set of flipped reflection rays as additional training resources, which are simply derived from the estimated normal vectors and the input ray directions. Since it does not require any heuristic factor for unseen view generation, we are able to exploit these additional training resources with much less burden. Furthermore, with our proposed UE Loss, FlipNeRF reduces the floating artifacts consistently across the different scene structures while considering the output uncertainty, leading to more reliable outputs. Also, our proposed BFC Loss enhances the bottleneck feature consistency between the rays cast on the photo-consistent pixels without leveraging the off-the-shelf feature extractor, leading to performance improvement under the few-shot setting. Our FlipNeRF achieves the SOTA performance with limited input views among the other few-shot baselines and vanilla NeRF-like models. We expect that our work is able to open another meaningful direction for the research of few-shot novel view synthesis.

References

[1] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1, 2, 4, 6, 7, 8

[2] Christopher M Bishop. Mixture density networks. 1994. 2

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 3, 4

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 1, 2, 3, 6

[5] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7911–7920, 2021. 1, 2, 3, 5, 6

[6] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022. 3

[7] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1, 2, 3, 4

[8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2

[9] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 2

[10] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 1, 2, 3, 5, 6, 7, 8

[11] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958, 2021. 1, 3

[12] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 2, 6

[13] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022. 1

[14] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022. 1, 2, 3, 4, 5, 6, 8

[15] Minseop Kwak, Jiuhn Song, and Seungryong Kim. Geconerf: Few-shot neural radiance fields via geometric consistency. *arXiv preprint arXiv:2301.10941*, 2023. 1, 3, 4

[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 6

[17] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12578–12588, 2021. 1, 3

[18] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 1

[19] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020. 6

[20] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 3

[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 (TOG)*, 38(4):1–14, 2019. 2, 6

[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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 4, 6

[23] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 1, 2, 3, 4, 6, 7, 8

[24] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 3

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5

[26] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 2

[27] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. Sharf: Shape-conditioned radiance fields from a single view. In *International Conference on Machine Learning*, pages 8948–8958. PMLR, 2021. 1, 3

[28] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 1, 2, 3

[29] Seunghyeon Seo, Donghoon Han, Yeonjin Chang, and Nojun Kwak. Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. *arXiv preprint arXiv:2302.08788*, 2023. 1, 2, 3, 4, 6, 7, 8

[30] Jianxiong Shen, Antonio Agudo, Francesc Moreno-Noguer, and Adria Ruiz. Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 540–557. Springer, 2022. 6

[31] Jianxiong Shen, Adria Ruiz, Antonio Agudo, and Francesc Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In *2021 International Conference on 3D Vision (3DV)*, pages 972–981. IEEE, 2021. 6

[32] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 3, 4

[33] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021. 1, 3

[34] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 1, 2, 3, 4, 5, 6, 8

[35] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022. 5, 7

[36] 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. *arXiv preprint arXiv:2106.10689*, 2021. 3

[37] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 1, 3

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[39] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 3

[40] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[41] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 1, 3, 6, 8

[42] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 3

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6