Is Vanilla MLP in Neural Radiance Field Enough for Few-shot View Synthesis?

Hanxin Zhu[1], Tianyu He[2], Xin Li[1], Bingchen Li[1], Zhibo Chen[1]
[1]University of Science and Technology of China
[2]Microsoft Research Asia
hanxinzhu@mail.ustc.edu.cn, tianyuhe@microsoft.com,
{lixin666, lbc31415926}@mail.ustc.edu.cn, chenzhibo@ustc.edu.cn

Abstract

Neural Radiance Field (NeRF) has achieved superior performance for novel view synthesis by modeling the scene with a Multi-Layer Perception (MLP) and a volume rendering procedure, however, when fewer known views are given (i.e., few-shot view synthesis), the model is prone to overfit the given views. To handle this issue, previous efforts have been made towards leveraging learned priors or introducing additional regularizations. In contrast, in this paper, we for the first time provide an orthogonal method from the perspective of network structure. Given the observation that trivially reducing the number of model parameters alleviates the overfitting issue, but at the cost of missing details, we propose the multi-input MLP (mi-MLP) that incorporates the inputs (i.e., location and viewing direction) of the vanilla MLP into each layer to prevent the overfitting issue without harming detailed synthesis. To further reduce the artifacts, we propose to model colors and volume density separately and present two regularization terms. Extensive experiments on multiple datasets demonstrate that: 1) although the proposed mi-MLP is easy to implement, it is surprisingly effective as it boosts the PSNR of the baseline from 14.73 *to* 24.23*. 2) the overall framework achieves state-of-the-art results on a wide range of benchmarks.*

1. Introduction

Neural Radiance Field (NeRF) has emerged as one of the most promising methods for novel view synthesis, owing to its remarkable ability to represent 3D scenes. By utilizing a Multi-Layer Perception (MLP) in conjunction with classical volume rendering, NeRF can produce photorealistic novel views from multiple 2D images captured from different views [21]. Various works extends NeRF to different tasks such as surface reconstruction [39, 42, 44, 54], dynamic scenes [11, 24, 25, 27] and 3D generation [6, 16, 26, 37, 49, 53], etc. However, these NeRF-based methods require a large number of input views (*e.g.*, 100) [21]. In cases where only a few input views are available (*i.e.*, few-shot view synthesis), NeRF brings severe artifacts and thus leads to a dramatic performance drop [12, 28].

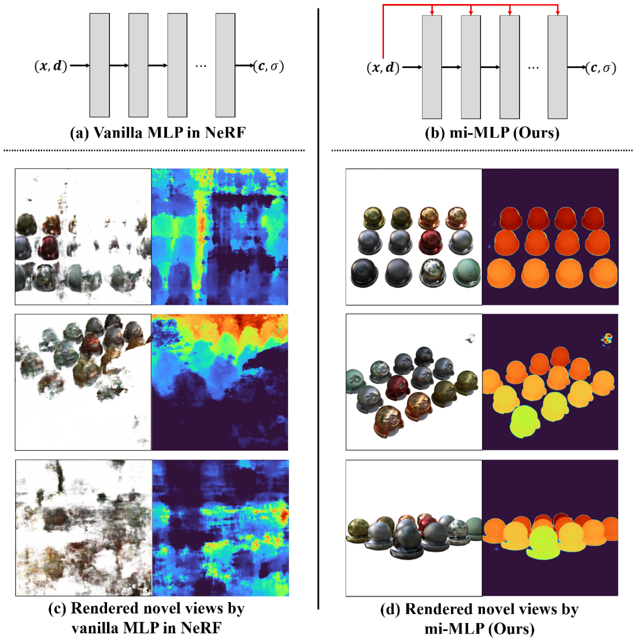

(x, d)
(c, σ)
(a) Vanilla MLP in NeRF
(x, d)
(c, σ)
(b) mi-MLP (Ours)
(c) Rendered novel views by vanilla MLP in NeRF
(d) Rendered novel views by mi-MLP (Ours)

Figure 1. Illustration of **vanilla MLP vs. mi-MLP**. Although mi-MLP is easy to implement, it is surprisingly effective as it boosts the PSNR of the baseline from 14.73 to 24.23.

Two primary challenges arise in the context of few-shot view synthesis. Firstly, due to the limited amount of training data available, the model is prone to overfitting input views, resulting in the estimated geometry being distributed on 2D planes instead of 3D volumes [12, 14, 23]. Secondly, the presence of artifacts such as ghosting and floating effects significantly limit the fidelity and 3D consistency of rendered novel views [23, 50].

To address the aforementioned issues, mainstream approaches can be categorized into two strategies: prior-based [4, 8, 43, 51] and regularization-based [12, 14, 23, 50] methods. Prior-based methods aim to generalize NeRF to different scenes using techniques such as multi-view stereo [10] or image-based rendering [34], where a large-scale dataset is utilized to learn scene priors. Regularization-based methods incorporate additional 3D inductive bias, *e.g.*, frequency [50] and depth [23] regularizations, for the purpose of stronger constraints. Despite achieving remarkable results, none of these methods take the network structure into account and still adhere to the vanilla MLP [21]. In this paper, we challenge this common practice and ask: *is vanilla MLP in NeRF enough for few-shot view synthesis*?

To answer this question, we investigate the overfitting issue and have two key observations: 1) FreeNeRF [50] illustrates that the vanilla NeRF is prone to over-fastly converge to high-frequency details. In this way, the model quickly memorizes input views instead of inferring the underlying geometry. Therefore, to avoid overfitting, a direct solution is to decrease the model capacity by reducing the model parameters (*e.g.*, reducing the number of layers); 2) however, as presented in DietNeRF [12], though the overfitting issue can be alleviated by reducing the model parameters, the details are missed in the generated results. This indicates that model capacity should be preserved for the network.

Capitalizing on the above observations, we propose the multi-input MLP (mi-MLP) that incorporates the inputs (*i.e.*, location and viewing direction) of the vanilla MLP into each layer (as illustrated in Fig. 1). mi-MLP reveals three key insights: **1)** incorporating the inputs into each layer enables shorter paths between inputs and outputs, allowing synthesis with fewer parameters in an end-to-end way; **2)** we keep the model capacity unchanged as it is beneficial to synthesizing high-frequency details; **3)** we keep the inputs and outputs unchanged to make it a plug-and-play solution to the current NeRF-based pipelines.

To further reduce the artifacts, motivated by the assumption that geometry is typically smoother than appearance [23], instead of using a shared model to model the colors and volume density like NeRF, we propose to model them separately to enable positional encoding [21] with different frequencies. We also propose a novel regularization term to reduce the background artifacts in object-centric scenes and a sampling-annealing strategy to address near-field artifacts in forwarding-facing scenes.

Our main contributions can be summarized as follows:

- To address the overfitting issue, we introduce mi-MLP to tackle few-shot view synthesis from the perspective of network structure by incorporating the inputs into each layer.
- To achieve better geometry, we propose to model the colors and volume density separately to enable positional encoding with different frequencies.
- We propose two regularization terms to improve the quality of rendered novel views.
- Through comprehensive experiments, we demonstrate that our method attains superior performance compared with multiple state-of-the-art methods.

To the best of our knowledge, this is the first work that tackles NeRF-based few-shot novel view synthesis from the perspective of network structure, opening up a new direction for further research in other fields such as 3D generation.

2. Related Works

2.1. Neural Radiance Field

Neural Radiance Field (NeRF) [21] has become increasingly popular due to its impressive 3D representation capabilities, where photorealistic novel views can be rendered with 2D posed images. One of the keys to NeRF's success lies in the usage of an MLP to reason about scene properties, where a mapping from input embeddings to outputs is learned, allowing for continuous scene representation and view interpolation. Numerous researchers have extended NeRF to a variety of areas, including faster training and rendering [9, 13, 22], dynamic scenes [11, 24, 25, 27], generable scenes [7, 18, 38, 40], and 3D generation [16, 26, 29], etc. However, the practical utility of these NeRF-based methods is limited due to the need for a large number of input views. In this paper, we propose a novel method that targets few-shot view synthesis through a well-designed network structure.

2.2. Few-shot View Synthesis

Prior-based methods. Prior-based approaches enable NeRF for few-shot view synthesis either by training a generalized model through large datasets of different scenes or by introducing off-the-shelf pre-trained models. Early works [4, 38, 43, 51] extracted convolutional features from input views as conditions to render novel views, using classical graphics pipelines such as image-based rendering [3, 34] or multi-view stereo [10, 31]. Vision-NeRF [17], however, used vision transformers to extract both local and global features for occlusion-aware rendering. DSNeRF [8] and DDP-NeRF [28] further used depth information obtained from Structure-From-Motion [30] or pre-trained depth completion models to incorporate explicit 3D priors. More recently, SparseNeRF [41] proposed to utilize depth priors obtained from real-world inaccurate observations. DiffusioNeRF [47] learned priors over scene geometry and colors through a more powerful diffusion model, which is trained on RGBD patches. While these methods can produce photorealistic novel views, they often require expensive pre-training costs, and the pre-trained scenes may not be suitable for the target scene.

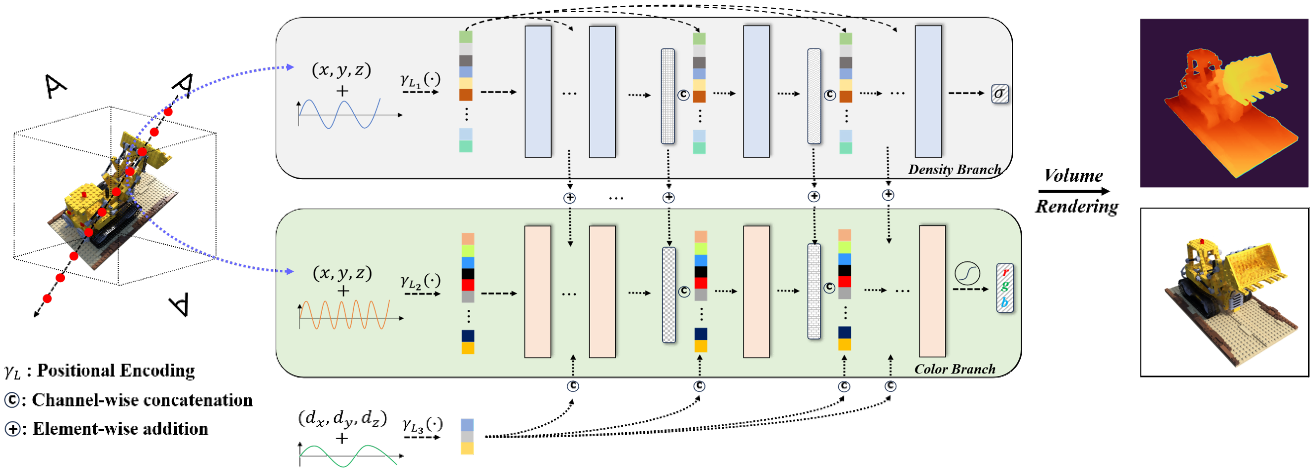


Figure 2. **Network structure of our proposed method.** To avoid the overfitting issue in few-shot view synthesis, we propose multi-input MLP (mi-MLP) that incorporates inputs (*i.e*., location (x, y, z) and viewing direction (d_x, d_y, d_z)) into each layer of the MLP (Sec. 4.1.1). To further improve geometry recovery, we model volume density and colors separately with different frequencies (Sec. 4.1.2).

Regularization-based methods. Regularization-based methods instead obey a per-scene optimization manner similar to vanilla NeRF [21], and introduce additional regularization terms or training sources for better novel view synthesis. Specifically, semantic consistency loss [12], depth-smoothing loss [23], and ray-entropy loss [14] were first introduced to constrain unseen views for better geometry recovery. To increase the number of training views available, several works [1, 5, 15, 48] proposed to use depth-warping to generate novel view images as pseudo labels. Recently, FreeNeRF [50] followed a coarse-to-fine manner through a novel frequency annealing strategy on positional encoding. MixNeRF [33] modeled rays as mixtures of Laplacianssians, followed by FlipNeRF [32] which uses flipped reflection rays as additional training sources. SimpleNeRF [35] proposed to use augmented models to avoid overfitting, which performs well on forward-facing scenes. Though remarkable results have been achieved, all these methods still use the network structure proposed by vanilla NeRF. In contrast, in this paper, we achieve the few-shot view synthesis from the perspective of designing a better network structure.

3. Preliminaries: NeRF

Different from classical explicit scene representation methods such as mesh, voxel, and point cloud, Neural Radiance Field (NeRF) [21] utilizes an MLP F_θ to represent scenes implicitly and compactly. For a ray $\boldsymbol{r}$ cast from camera origin $\boldsymbol{o}$ through a pixel $\boldsymbol{p}$ along direction $\boldsymbol{d}$, a point $\boldsymbol{r}_t = \boldsymbol{o} + t\boldsymbol{d}$ is first sampled from the ray, where $t \in [t_{\text{near}}, t_{\text{far}}]$. Subsequently, $\boldsymbol{r}_t$ is sent to F_θ to estimate the scene properties, *i.e*., the corresponding color $\boldsymbol{c}$ and volume density σ, which is denoted as:

$$\boldsymbol{c}, \sigma = F_\theta(\gamma_L(\boldsymbol{r}_t), \gamma_L(\boldsymbol{d})), \tag{1}$$

where γ is the positional encoding operation aimed at obtaining high-frequency details that is formulated as follows:

$$\gamma_L(\boldsymbol{x}) = (\sin(2^0\boldsymbol{x}), \cos(2^0\boldsymbol{x}), \cdots, \sin(2^{L-1}\boldsymbol{x}), \cos(2^{L-1}\boldsymbol{x})), \tag{2}$$

where L is a hyperparameter that controls the frequencies.

Given the color and volume density of $\boldsymbol{r}_t$, the color of ray $\boldsymbol{r}$ can be estimated using the following equation:

$$\mathbf{C}(\boldsymbol{r}) = \int_{t_{\text{near}}}^{t_{\text{far}}} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})dt, \tag{3}$$

where $T(t) = \exp\left(-\int_{t_{\text{near}}}^{t} \sigma(\boldsymbol{r}(s))ds\right)$ represents the accumulated transmittance. The NeRF is then optimized using common reconstruction loss, *i.e*.,

$$\mathcal{L} = \frac{1}{|\mathcal{R}|}\sum_{\boldsymbol{r}\in\mathcal{R}} \|\mathbf{C}(\boldsymbol{r}) - \mathbf{C}_{\text{gt}}\|_2^2, \tag{4}$$

where $\mathcal{R}$ is a batch of sampling rays, $\mathbf{C}(\boldsymbol{r})$ is obtained by Eq. 3 and $\mathbf{C}_{\text{gt}}$ represents the ground-truth color.

4. Methods

Motivation. As mentioned in Sec. 1, when only a few input views are available, NeRF faces a significant challenge of overfitting. To solve this problem, we drew inspiration from two key observations: 1) as illustrated in FreeNeRF [50], the overfitting issue is caused by the over-fast convergence speed of NeRF on high-frequency details. In this way, the model quickly memorizes input views instead of correctly inferring the underlying geometry. Hence, to avoid overfitting, a direct solution is to decrease the model capacity by reducing the model parameters (*e.g.*, reducing MLP layers); 2) however, though such a simple operation can alleviate overfitting to some extent, as presented in DietNeRF [12], this simplified NeRF is hardly to recover accurate details, resulting in blurry novel views.

Based on the two observations above, to achieve few-shot view synthesis, our intuition is that in the initial stages of training, the model capacity should be restricted to prevent NeRF from memorizing input views and thus avoid overfitting. However, during the later stage of training, the model capacity should be preserved for detailed rendering.

4.1. Network Structure

Our network consists of two designs as elaborated in Sec. 4.1.1 and Sec. 4.1.2 respectively. The resulting architecture is illustrated in Fig. 2.

4.1.1 Per-layer Inputs Incorporation

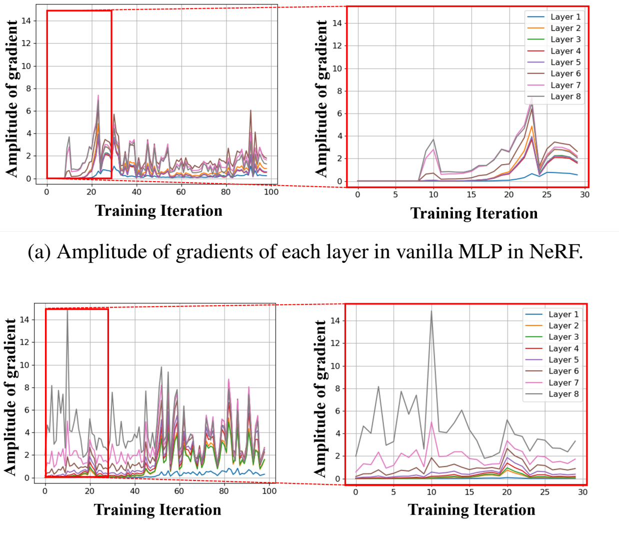
We address the overfitting problem in the few-shot view synthesis from the perspective of network structure. Specifically, as shown in Fig. 1(b), we propose multi-input MLP (mi-MLP) that incorporates inputs (*i.e.*, 3D location and 2D viewing direction) into each layer of the MLP, which is formulated as follows:

$$\boldsymbol{f}_i = \phi_i(\boldsymbol{f}_{i-1}, \gamma_L(\boldsymbol{x})),\ \ \boldsymbol{f}_1 = \phi_1(\gamma_L(\boldsymbol{x})), \tag{5}$$

where ϕ_i is the i-th ($i > 2$) layer of the MLP, $\boldsymbol{f}_i$ is the corresponding output feature, $\boldsymbol{x}$ is the input 5D coordinate and $\gamma_L(\boldsymbol{x})$ represents the encoded input embeddings (Eq. 2).

In contrast to vanilla NeRF, which uses all layers to learn mappings from input embeddings to outputs as shown in Fig. 1(a), our formulation ensures that each layer of the MLP is aware of the input embeddings explicitly. This allows the mappings from input embeddings to outputs with varying number of layers. We hypothesize that such flexible connections between inputs and outputs play a significant role in alleviating the overfitting issue. The analysis is provided below.

How mi-MLP works? Intuitively, the per-layer inputs incorporation enables shorter paths between inputs and outputs, allowing synthesis with fewer parameters in an end-to-end way. It also encourages that the amplitude of gradients of the shallower layer be smaller than that of the deeper layer. As demonstrated in Fig. 3, at the beginning of the training stage, in contrast to vanilla MLP that results in a similar amplitude of gradients for each layer (Fig. 3(a)), mi-MLP enables that the deeper layers (*i.e.*, layers close to the outputs) are updated with large gradients while the shallower layers are updated with extremely small ones (Fig. 3(b)).



(a) Amplitude of gradients of each layer in vanilla MLP in NeRF.

(b) Amplitude of gradients of each layer in our proposed mi-MLP.

Figure 3. Illustration of the averaged amplitude of gradients of each layer in MLP at the beginning of training. (a) All layers in vanilla MLP have a similar amplitude of gradients. (b) In contrast, mi-MLP enables that the deeper layers (*i.e.*, layers close to the outputs) are updated with large gradients while the shallower layers are updated with extremely small ones.

More theoretical, assuming $\gamma_L(\boldsymbol{x}) \in \mathbb{R}^{d_1 \times 1}$, $\boldsymbol{f}_i \in \mathbb{R}^{d_2 \times 1}$, the bias vector and weight matrix of ϕ_i are $\boldsymbol{b}_i \in \mathbb{R}^{d_2 \times 1}$ and $\boldsymbol{w}_i = (\boldsymbol{w}_i^1, \boldsymbol{w}_i^2, \dots, \boldsymbol{w}_i^{d_2})^T$ respectively, where $\boldsymbol{w}_i^j = (\boldsymbol{w}_i^{j0} \in \mathbb{R}^{1\times d_1}, \boldsymbol{w}_i^{j1} \in \mathbb{R}^{1 \times d_2})^T$. Thus Eq. 5 is equivalent to

$$\begin{aligned}\phi_i^j(\gamma_L(\boldsymbol{x})) &= \epsilon\{\boldsymbol{w}_i^j[\gamma_L(\boldsymbol{x}), \phi_{i-1}(\gamma_L(\boldsymbol{x}))]^T + \boldsymbol{b}_i\} \\ &= \epsilon\{\boldsymbol{w}_i^{j0}[\gamma_L(\boldsymbol{x})] + \boldsymbol{w}_i^{j1}[\phi_{i-1}(\gamma_L(\boldsymbol{x}))] + \boldsymbol{b}_i\},\end{aligned} \tag{6}$$

where ϕ_i^j is the j-th element of $\boldsymbol{f}_i$, ϵ denotes the activation function whose default setting is ReLU. It can be proved that the closed-form solution that represents the ratio of the amplitude of gradients of two adjacent layers can be formulated as follows, where $\mathcal{L}$ means the loss function:

$$\begin{aligned}&\|\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i}\|_1 / \|\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{i-1}}\|_1 = \frac{1}{d_2}\sum_{j=1}^{d_2}\|\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i^j}\|_1 / \|\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{i-1}^j}\|_1 \\ =&\frac{1}{d_2}\sum_{j=1}^{d_2}\frac{\|\gamma_L(\boldsymbol{x})\|_1 + \|\phi_{i-1}(\gamma_L(\boldsymbol{x}))\|_1}{\|\sum \boldsymbol{w}_i^{j1}\|_1 \cdot \{\|\gamma_L(\boldsymbol{x})\|_1 + \|\phi_{i-2}(\gamma_L(\boldsymbol{x}))\|_1\}},\end{aligned} \tag{7}$$

Accordingly, $\|\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i}\|_1 / \|\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{i-1}}\|_1 \geq 1$ holds true in a high probability during the early stage of training when

$\|\sum \boldsymbol{w}_i^{j1}\|_1 \in (0,1]$ and $\|\phi_{i-1}(\gamma_L(\boldsymbol{x}))\|_1 \approx \|\sum \boldsymbol{w}_i^{j1}\|_1 \cdot \|\phi_{i-2}(\gamma_L(\boldsymbol{x}))\|_1$. In practice, we find that the default initialization provided by PyTorch can meet the requirements, where the amplitude of gradients of each layer is shown in Fig. 3. Please refer to the supplementary materials for more details.

4.1.2 Modeling Colors and Volume Density Separately

Although mi-MLP alone can perform comparably to several prior methods, the rendered novel views still contain noticeable artifacts, as shown in Fig. 7. To address this issue and further improve geometry recovery, we propose to model volume density and colors separately.

Specifically, it is widely accepted that geometry (represented by the volume density) is not as detailed as appearance (represented by the colors), since geometry is usually piecewise smooth [23]. To prioritize low-frequency information in volume density, we propose to reduce the dimensions of input embeddings for volume density in comparison to those for colors, considering that the dimensions of the encoded input embeddings obtained by Eq. 2 decide how detailed the output is [21, 50].

To this end, different from NeRF which uses one shared MLP to predict colors and volume density synchronously, we instead use two separate MLPs to estimate them individually, dubbed the Color Branch C_θ and Density Branch D_θ, where the dimensions of input embeddings for different branches are not the same. As shown in Fig. 2, the whole network structure can thus be formulated as follows:

$$\sigma = D_\theta(\gamma_{L_1}(\boldsymbol{x})), \boldsymbol{c} = C_\theta(\gamma_{L_2}(\boldsymbol{x}), \gamma_{L_3}(\boldsymbol{d})), \tag{8}$$

where σ and $\boldsymbol{c}$ denote the estimated volume density and colors respectively, $\boldsymbol{x}$ is the input 3D point coordinate, $\boldsymbol{d}$ is viewing direction vector, L_1, L_2, and L_3 are hyperparameters that control the frequencies of positional encoding which satisfy $L_3 \le L_1 \le L_2$.

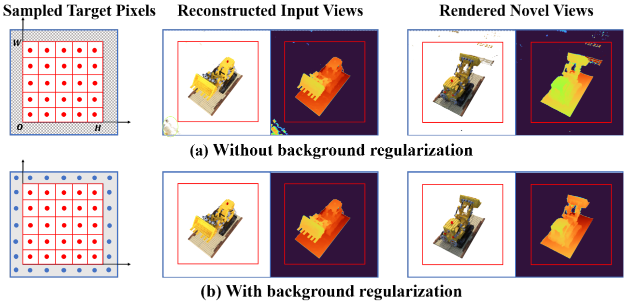
Overall, we adopt both per-layer incorporation and separate modeling of colors and volume density in our network design. Therefore, for the Density Branch, as illustrated in Sec. 4.1.1, we incorporate inputs into each layer, *i.e.*,

$$\boldsymbol{f}_i^D = \phi_i^D(\boldsymbol{f}_{i-1}^D, \gamma_{L_1}(\boldsymbol{x})), \ \boldsymbol{f}_1^D = \phi_1^D(\gamma_{L_1}(\boldsymbol{x})), \tag{9}$$

where ϕ_i^D is the i-th ($i \ge 2$) layer of the Density Branch MLP, $\boldsymbol{f}_i^D$ is the corresponding output feature. For the Color Branch, we empirically find that an interaction between the Color Branch and the Density Branch is beneficial to better geometry recovery, which is denoted as follows:

$$\begin{aligned} \boldsymbol{f}_{i-1}^C &= \phi_{i-1}^C(\boldsymbol{f}_{i-2}^C, \gamma_{L_3}(\boldsymbol{d})) + \boldsymbol{f}_{i-1}^D \\ \boldsymbol{f}_i^C &= \phi_i^C(\boldsymbol{f}_{i-1}^C, \gamma_{L_3}(\boldsymbol{d})), \ \boldsymbol{f}_1^C = \phi_1^C(\gamma_{L_2}(\boldsymbol{x})), \end{aligned} \tag{10}$$

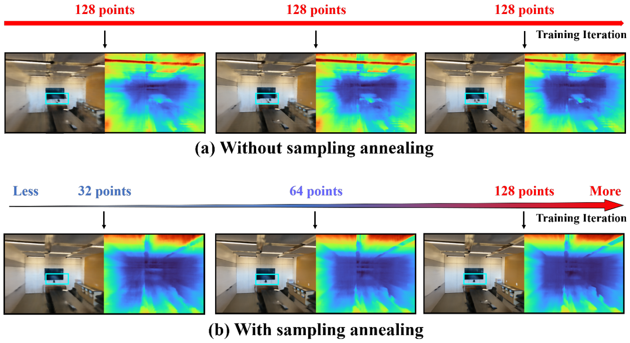
where ϕ_i^C is the i-th ($i \ge 2$) layer of the Color Branch MLP, $\boldsymbol{f}_i^C$ is the corresponding output feature.



(a) Without background regularization

(b) With background regularization

Figure 4. **Background regularization.** In addition to sampling target pixels within the image space (*i.e.*, the red dots) to generate training rays, we also sample target pixels outside the image space (*i.e.*, the blue dots) to address background artifacts in object-centric scenes.



(a) Without sampling annealing

(b) With sampling annealing

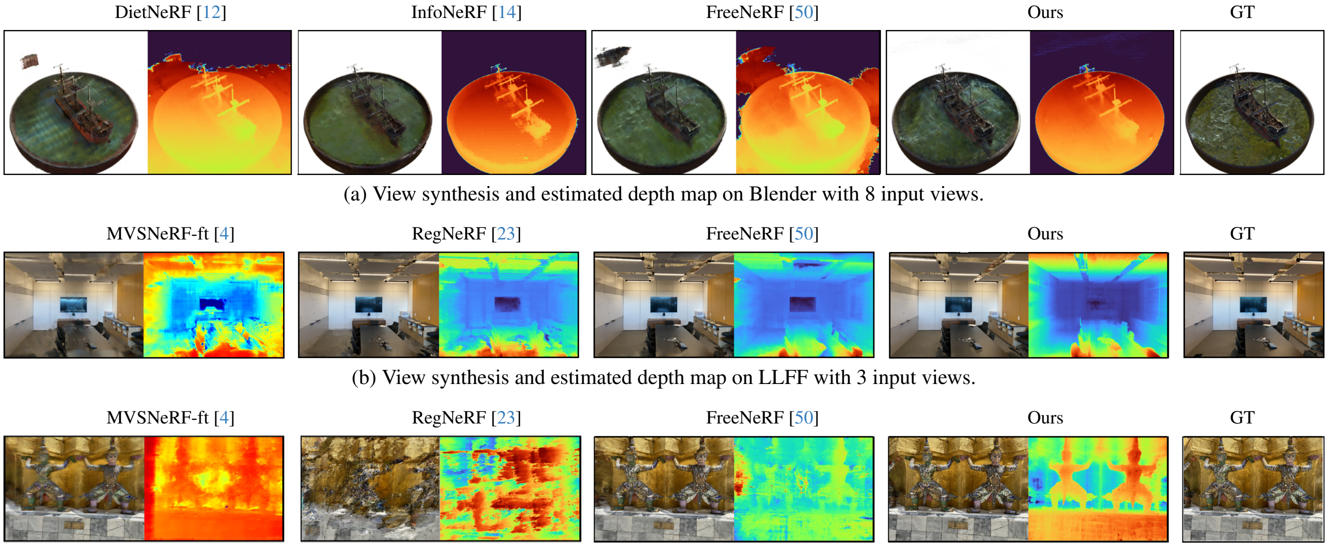
Figure 5. **Sampling annealing.** During the early stage of training, fewer points are sampled along a ray to make the network more focused on coarse geometry estimation, while more sampling points are utilized during the later stage for details recovery.

4.2. Background Regularization

A common failure mode for rendering scenes centered on a single object is the presence of background artifacts for both reconstructed input views and rendered novel views, as shown in Fig. 4(a).

We assume that this is caused by insufficient constraints on the background. Specifically, during the training process of NeRF, the sampled target pixels $\boldsymbol{p} = (p_x, p_y)$ that generate training rays $\boldsymbol{r}$ are all distributed inside the input image space, where $p_x \in [0, H], p_y \in [0, W]$, H and W represent the height and width of input images. For object-centric scenes, it is reasonable to assume that the corresponding pixel colors outside the image space should be the same as the background color. However, as shown in Fig. 4(a), when only a few input views are available, the extrapolated input image contains apparent artifacts, especially in the areas that lie outside the image space.

Motivated by this observation, we propose a regularization technique for background artifact removal, which is de-



(a) View synthesis and estimated depth map on Blender with 8 input views.

(b) View synthesis and estimated depth map on LLFF with 3 input views.

(c) View synthesis and estimated depth map on Shiny with 3 input views.

Figure 6. **Qualitive comparisons on the Blender, LLFF, and Shiny dataset.** Our proposed method can achieve both photorealistic novel views and accurate depth estimation, ft indicates the results fine-tuned on each scene individually.

noted as follows:

$$\mathcal{L}_{\text{BR}} = \frac{1}{|\mathcal{R}_o|} \sum_{\boldsymbol{r} \in \mathcal{R}_o} \|\mathbf{C}(\boldsymbol{r}) - \mathbf{C}_{\text{bk}}\|_2^2, \tag{11}$$

where $\mathcal{R}_o$ is a batch of sampling rays generated from target pixels outside the input image space, $\mathbf{C}(\boldsymbol{r})$ is the rendered color and $\mathbf{C}_{\text{bk}}$ is the background color. As shown in Fig. 4(b), the regularization obtains clear images effectively.

4.3. Sampling Annealing

In the context of real-world scenes, as shown in Fig. 5(a), we also observe that the floating artifacts are distributed in close proximity to the camera [23, 50], which are referred to near-field artifacts.

To solve this problem, we propose a sampling annealing strategy, where the number of sampling points along a ray increases linearly during training, which is formulated as follows:

$$N_t = \min(N_{\max}, \lfloor u/\eta \rfloor + N_{start}), \tag{12}$$

where u denotes the current training iteration, N_t indicates the number of sampling points along one ray at the u-th iteration, $N_{\max}$ is the maximum number of sampling points, N_{start} is the number of sampling points at the start of training, η is a hyperparameter that controls the increasing speed of sampling points.

5. Experiments

Datasets and metrics. We evaluate our proposed method on three popular datasets: Blender [21], LLFF [20], and Shiny [46]. Blender consists of 8 synthetic 360° object-centric scenes with white background. LLFF and Shiny individually contain 8 real-world forward-facing scenes, while Shiny is much more complex due to its view-dependent effects, such as reflections and refraction. We follow the experimental protocols provided by [12, 23].

We use PSNR, SSIM [45], and LPIPS [52] to measure the quantitative results of our proposed methods. We also report the geometric average following [23] for an easier comparison.

5.1. Comparison with State-of-the-art Methods

Blender. Our proposed method achieves state-of-the-art performance on the Blender dataset for both 4 and 8 input views, as shown in Tab. 1 and Fig. 6(a). Notably, for methods such as [12] and [14] that impose additional regularizations on unseen views, though reasonable results can be obtained, the rendered novel views include unexpected imaginary contents. For FreeNeRF [50], since the regularization is only applied to known input views, the estimated geometry contain severe floating artifacts, as demonstrated from the death map in Fig. 6(a). In contrast, our proposed method can achieve photorealistic novel view synthesis as well as clear geometry estimation.

LLFF. We also perform experiments on the LLFF dataset with 3/6/9 known input views. As shown in Tab. 2 and Fig. 6(b), our method generally outperforms other baselines across all settings. For prior-based methods, severe artifacts will be generated due to the domain gap between the training dataset and the testing set. Compared to regularization-based methods, ours can achieve the best performance, except for the PSNR metric when 6 input views are available.

Method	Setting	PSNR↑		SSIM↑		LPIPS↓		Average↓	
		4-view	8-view	4-view	8-view	4-view	8-view	4-view	8-view
Vanilla-NeRF [21]	Baselines	-	14.73	-	0.734	-	0.451	-	0.199
Mip-NeRF [2]		14.12	18.74	0.722	0.828	0.382	0.238	0.221	0.121
Ref-NeRF [39]		18.09	24.00	0.764	0.879	0.269	0.106	0.150	0.059
NV [19]	Regularization-based	-	17.85	-	0.741	-	0.245	-	0.127
Simplified NeRF [12]		-	20.09	-	0.822	-	0.179	-	0.090
DietNeRF [12]		15.42	23.14	0.730	0.866	0.314	0.109	0.201	0.063
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
MixNeRF [33]		18.99	23.84	0.807	0.878	0.199	0.103	0.113	0.060
FreeNeRF [50]		19.70	24.26	0.812	0.883	0.175	0.098	0.093	0.058
Ours	Network-based	20.38	24.70	0.828	0.885	0.156	0.087	0.084	0.047

Table 1. **Quantitative Comparison on Blender.** Our proposed method can achieve state-of-the-art performance on all metrics. The best, second-best, and third-best entries are marked in red, orange, and yellow, respectively. Our baseline is marked in gray.

Method	Setting	PSNR ↑			SSIM ↑			LPIPS ↓			Average ↓		
		3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view
Vanilla-NeRF [21]	Baselines	13.34	-	-	0.373	-	-	0.451	-	-	0.255	-	-
Mip-NeRF [2]		14.62	20.87	24.26	0.351	0.692	0.805	0.495	0.255	0.172	0.246	0.114	0.073
SRF [7]	Prior-based	12.34	13.10	13.00	0.250	0.293	0.297	0.591	0.594	0.605	0.313	0.293	0.296
PixelNeRF [51]		7.93	8.74	8.61	0.272	0.280	0.274	0.682	0.676	0.665	0.461	0.433	0.432
MVSNeRF [4]		17.25	19.79	20.47	0.557	0.656	0.689	0.356	0.269	0.242	0.171	0.125	0.111
SRF-ft [7]		17.07	16.75	17.39	0.436	0.438	0.465	0.529	0.521	0.503	0.203	0.207	0.193
PixelNeRF-ft [51]		16.17	17.03	18.92	0.438	0.473	0.535	0.512	0.477	0.430	0.217	0.196	0.163
MVSNeRF-ft [4]		17.88	19.99	20.47	0.584	0.660	0.695	0.327	0.264	0.244	0.157	0.122	0.111
DietNeRF [12]	Regularization-based	14.94	21.75	24.28	0.370	0.717	0.801	0.496	0.248	0.183	0.240	0.105	0.073
RegNeRF [23]		19.08	23.10	24.86	0.587	0.760	0.820	0.336	0.206	0.161	0.146	0.086	0.067
FreeNeRF [50]		19.63	23.73	25.13	0.612	0.779	0.827	0.308	0.195	0.160	0.134	0.075	0.064
Ours	Network-based	19.75	23.57	25.15	0.614	0.788	0.834	0.300	0.163	0.140	0.125	0.069	0.055

Table 2. **Quantitative Comparison on LLFF.** Our proposed method outperforms other methods on real-world forward-facing scenes, ft indicates the results fine-tuned on each scene individually.

We believe this is caused by the choice of different baselines, where we use vanilla NeRF as our baseline, while methods like RegNeRF [23] and FreeNeRF [50] choose a more powerful baseline, *i.e.*, MipNeRF [2].

Shiny. On account that the Shiny dataset contains more complex view-dependent effects such as reflection and refraction, most regularization-based methods such as [12, 23] perform even worse than vanilla NeRF, due to the mismatch between introduced regularization terms and actual physical prior, as shown in Tab. 3 and Fig. 6(c). Though FreeNeRF [50] can still work and produce reasonable results, the rendered novel views contain obvious artifacts. In contrast, our proposed method can achieve a significant performance improvement, both quantitatively and qualitatively. More additional results on the three datasets are provided in the supplementary materials.

Method	PSNR↑	SSIM↑	LPIPS↓	Average↓
Vanilla-NeRF [21]	14.37	0.309	0.610	0.264
MVSNeRF [4]	16.45	0.375	0.506	0.208
MVSNeRF-ft [4]	17.08	0.408	0.475	0.192
DietNeRF [12]	13.12	0.341	0.646	0.295
InfoNeRF [14]	12.86	0.332	0.681	0.307
RegNeRF [23]	12.76	0.287	0.621	0.302
FreeNeRF [21]	17.20	0.411	0.454	0.187
Ours	18.25	0.475	0.416	0.165

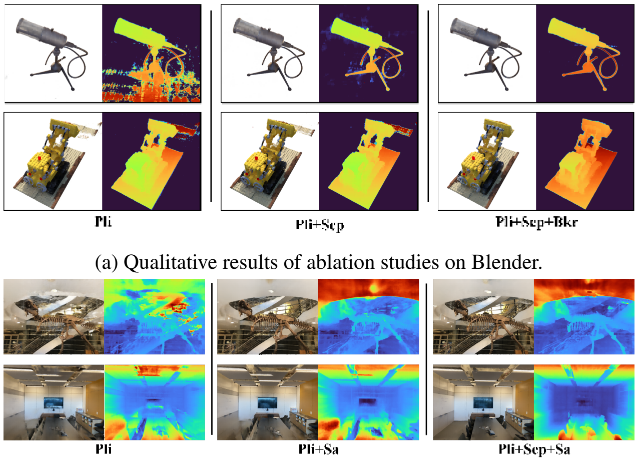
Table 3. **Quantitative Comparison on Shiny.** On the more challenging scenes with complex view-dependent effects such as reflection, our proposed method can still obtain a significant performance improvement when only 3 input views are available, ft indicates the results fine-tuned on each scene individually.

5.2. Ablation Studies

To showcase the effectiveness of our design choices, we conduct both quantitative and qualitative ablation studies, as shown in Tab. 4 and Fig. 7. With only per-layer inputs incorporation, a dramatic performance gain against our baseline (*i.e.*, vanilla NeRF) can be achieved, where we observe a 9.5dB PSNR improvement for the Blender dataset and a 4.8dB PSNR improvement for the LLFF dataset. For object-centric scenes like Blender, the separate modeling of volume density and colors is beneficial to clear geometry recovery, and the background regularization is able to further improve the performance by removing background arti-

	Pli	Sep	Bkr	Sa	PSNR↑	SSIM↑	LPIPS↓	Average↓
Blender								
NeRF	✗	✗	✗	✗	14.73	0.734	0.451	0.199
Ours	✓	✗	✗	✗	24.12	0.879	0.114	0.053
	✓	✗	✓	✗	24.42	0.882	0.092	0.048
	✓	✓	✗	✗	24.23	0.881	0.108	0.052
	✓	✓	✓	✗	24.70	0.885	0.087	0.046
LLFF								
NeRF	✗	✗	✗	✗	13.34	0.373	0.451	0.255
Ours	✓	✗	✗	✗	18.12	0.512	0.417	0.164
	✓	✓	✗	✗	16.87	0.463	0.441	0.187
	✓	✗	✗	✓	18.88	0.543	0.391	0.150
	✓	✓	✗	✓	19.75	0.614	0.300	0.125

Table 4. **Ablation Studies.** We perform ablation studies on Blender with 8 input views and LLFF with 3 input views, where **Pli** means per-layer inputs incorporation, **Sep** means separate modeling of colors and volume density, **Bkr** means background regularization, and **Sa** means sampling annealing.



(a) Qualitative results of ablation studies on Blender.

(b) Qualitative results of ablation studies on LLFF.

Figure 7. Qualitative results of ablation studies on Blender and LLFF, where **Pli** means per-layer inputs incorporation, **Sep** means separate modeling of colors and volume density, **Bkr** means background regularization, and **Sa** means sampling annealing.

facts. For forward-facing scenes like LLFF, we find that the sampling annealing strategy is crucial for accurate geometry estimation. By combining both the sampling annealing strategy and the separate modeling of volume density and colors, we are able to achieve the best performance. Moreover, we also try a classical approach to avoid overfitting, *i.e.*, Dropout [36], which we find a comparable performance with DietNeRF [12] can be achieved. Kindly refer to the supplementary materials for more results.

5.3. Orthogonality of mi-MLP

We also perform experiments to demonstrate the proposed mi-MLP is orthogonal to current works. For this purpose, we select three representative methods: FreeNeRF [50], InfoNeRF [14], and DietNeRF [12], and replace their network structure with our proposed method. As shown in Tab. 5 and Fig. 8, for a scene randomly chosen from the Blender dataset, better performance can always be achieved when combined with mi-MLP. Such a result reflects the potential of our proposed method to serve as a backbone for NeRF. Additionally, we extend mi-MLP to 3D generation.

Method	Known Views	PSNR↑	SSIM↑	LPIPS↓	Average↓
FreeNeRF [50]	4	18.88	0.777	0.179	0.102
FreeNeRF+Ours		**19.36**	**0.787**	**0.173**	**0.097**
InfoNeRF [14]	8	24.27	0.868	0.112	0.053
InfoNeRF+Ours		**24.77**	**0.877**	**0.104**	**0.049**
DietNeRF [12]	8	23.70	0.850	0.130	0.060
DietNeRF+Ours		**23.90**	**0.857**	**0.125**	**0.057**

Table 5. **Orthogonality of mi-MLP.** We choose 3 baselines, and replace their network structure with ours to demonstrate the proposed mi-MLP is orthogonal to current works.

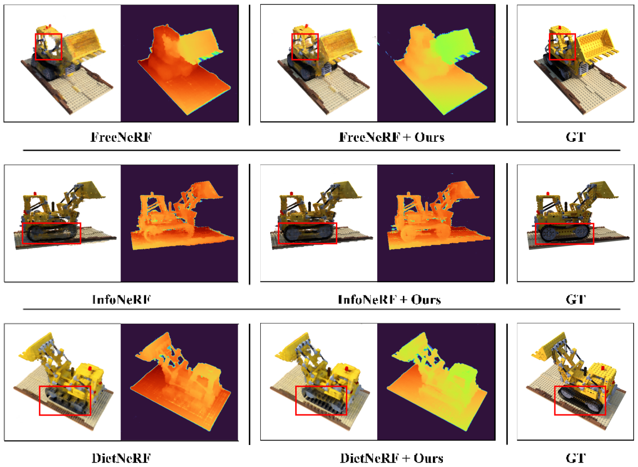


Figure 8. The proposed mi-MLP is orthogonal to current works since an improved performance can always be achieved for different methods when combined with our proposed mi-MLP.

6. Conclusion

In this paper, we have presented a novel method for few-shot view synthesis from the perspective of network structure for the first time. Specifically, to address the overfitting problem, motivated by the observation that a reduced model capacity is beneficial to alleviating overfitting while at the cost of missing details, we propose the mi-MLP that incorporates inputs into each layer of the MLP. Subsequently, based on the assumption that geometry is smoother than appearance, we propose to model colors and volume density separately for better geometry recovery. Additionally, we also provide two regularization terms to improve the quality of rendered novel views. Experiments have demonstrated that our proposed method can achieve state-of-the-art performance on multiple datasets. Considering the orthogonality of our proposed method, mi-MLP also opens up a new direction to other fields such as 3D generation. This work was supported in part by NSFC under Grant 62371434, 62021001.

References

[1] Young Chun Ahn, Seokhwan Jang, Sungheon Park, Ji-Yeon Kim, and Nahyup Kang. Panerf: Pseudo-view augmentation for improved neural radiance fields based on few-shot inputs. *arXiv preprint arXiv:2211.12758*, 2022. 3

[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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 7

[3] SC Chan, Heung-Yeung Shum, and King-To Ng. Image-based rendering and synthesis. *IEEE Signal Processing Magazine*, 24(6):22–33, 2007. 2

[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. 2, 6, 7

[5] Di Chen, Yu Liu, Lianghua Huang, Bin Wang, and Pan Pan. Geoaug: Data augmentation for few-shot nerf with geometry constraints. In *European Conference on Computer Vision*, pages 322–337. Springer, 2022. 3

[6] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. *arXiv preprint arXiv:2304.06714*, 2023. 1

[7] 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. 2, 7

[8] 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. 2

[9] 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

[10] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 2

[11] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 1, 2

[12] 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, 4, 6, 7, 8

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 2

[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, 6, 7, 8

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

[16] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 1, 2

[17] Kai-En Lin, Yen-Chen Lin, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 806–815, 2023. 2

[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. 2

[19] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 7

[20] 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. 6

[21] 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, 5, 6, 7

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[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, 5, 6, 7

[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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 1, 2

[25] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Ani-

matable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. 1, 2

[26] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2

[27] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1, 2

[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

[29] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 2

[30] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[31] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 519–528. IEEE, 2006. 2

[32] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22883–22893, 2023. 3

[33] Seunghyeon Seo, Donghoon Han, Yeonjin Chang, and Nojun Kwak. Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20659–20668, 2023. 3, 7

[34] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, pages 2–13. SPIE, 2000. 2

[35] Nagabhushan Somraj, Adithyan Karanayil, and Rajiv Soundararajan. Simplenerf: Regularizing sparse input neural radiance fields with simpler solutions. *arXiv preprint arXiv:2309.03955*, 2023. 3

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 8

[37] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 1

[38] 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. 2

[39] 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, 7

[40] Dan Wang, Xinrui Cui, Septimiu Salcudean, and Z Jane Wang. Generalizable neural radiance fields for novel view synthesis with transformer. *arXiv preprint arXiv:2206.05375*, 2022. 2

[41] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *arXiv preprint arXiv:2303.16196*, 2023. 2

[42] 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. 1

[43] 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. 2

[44] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Pet-neus: Positional encoding tri-planes for neural surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12598–12607, 2023. 1

[45] 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

[46] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 6

[47] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4180–4189, 2023. 2

[48] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 736–753. Springer, 2022. 3

[49] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4479–4489, 2023. 1

[50] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023. 1, 2, 3, 5, 6, 7, 8

[51] 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. 2, 7

[52] 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

[53] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12588–12597, 2023. 1

[54] Bingfan Zhu, Yanchao Yang, Xulong Wang, Youyi Zheng, and Leonidas Guibas. Vdn-nerf: Resolving shape-radiance ambiguity via view-dependence normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 35–45, 2023. 1