DivCon-NeRF: Diverse and Consistent Ray Augmentation for Few-Shot NeRF

Ingyun Lee , Jae Won Jang , Seunghyeon Seo , Nojun Kwak*
Seoul National University
{ig.lee, pert0407, zzzlssh, nojunk}@snu.ac.kr

Abstract

Neural Radiance Field (NeRF) has shown remarkable performance in novel view synthesis but requires numerous multi-view images, limiting its practicality in few-shot scenarios. Ray augmentation has been proposed to alleviate overfitting caused by sparse training data by generating additional rays. However, existing methods, which generate augmented rays only near the original rays, exhibit pronounced floaters and appearance distortions due to limited viewpoints and inconsistent rays obstructed by nearby obstacles and complex surfaces. To address these problems, we propose DivCon-NeRF, which introduces novel sphere-based ray augmentations to significantly enhance both diversity and consistency. By employing a virtual sphere centered at the predicted surface point, our method generates diverse augmented rays from all 360-degree directions, facilitated by our consistency mask that effectively filters out inconsistent rays. We introduce tailored loss functions that leverage these augmentations, effectively reducing floaters and visual distortions. Consequently, our method outperforms recent few-shot NeRF approaches on the Blender, LLFF, and DTU datasets. Furthermore, DivCon-NeRF demonstrates strong generalizability by effectively integrating with both regularization- and framework-based few-shot NeRFs.

1 Introduction

Neural Radiance Field (NeRF) [Mildenhall *et al.*, 2020], a neural network-based method, has significantly advanced the performance of novel view synthesis. However, NeRF-based models face the common deep learning challenge of performance degradation in few-shot scenarios. A limited number of training rays with sparse viewpoints causes overfitting. Therefore, the need for numerous images from different angles presents a substantial challenge for the practical application of NeRF-based models. To address the few-shot view synthesis problem, three main approaches are actively researched: prior-, framework-, and regularization-based methods. Each approach takes a different direction and is relatively orthogonal. Prior-based methods [Yu *et al.*, 2021; Chen *et al.*, 2021] leverage 3D knowledge through pre-training on large-scale datasets, but they incur high computational costs. Meanwhile, framework-based methods [Zhu *et al.*, 2024a; Zhong *et al.*, 2024] modify or extend model architectures, which can increase model complexity.

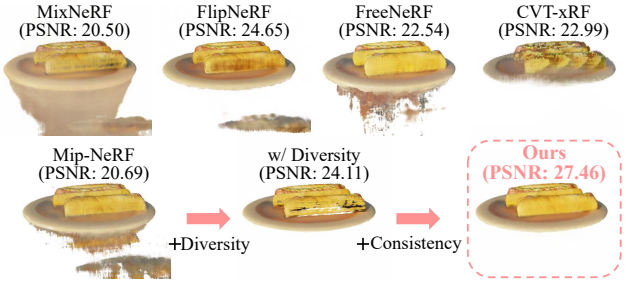


Figure 1: Rendering results of recent few-shot NeRFs on the hotdog scene of the Blender dataset using 4 input views. Notable floaters and appearance distortions are visible in other methods, while ours effectively addresses these artifacts by enhancing both diversity and consistency.

Our proposed method is a regularization-based approach that uses explicit regularizers [Xu *et al.*, 2024; Yang *et al.*, 2023] or auxiliary training sources [Wang *et al.*, 2023]. In particular, ray augmentation generates additional rays from a limited training set, enabling the model to learn from both original and augmented rays, thereby reducing overfitting. We define two key properties for ray augmentation: ray diversity, the range of viewpoints from which augmented rays are generated, and ray consistency, the requirement that each augmented ray converges to the same surface point as its original. Existing methods [Seo *et al.*, 2023b; Seo *et al.*, 2023a] sample rays only near the originals to avoid occlusion. However, as shown in Fig. 1, these approaches suffer from significant floaters owing to limited viewpoint variation (low diversity) and training on inconsistent rays obstructed by nearby obstacles and complex surfaces (low consistency). Furthermore, they introduce geometric and color distortions.

To address these limitations, we propose DivCon-NeRF,

*The corresponding author.

which enhances diversity while preserving consistency. In general data augmentation, various studies [Cubuk *et al.*, 2019; Cubuk *et al.*, 2020; Xie *et al.*, 2020; Lee *et al.*, 2024; Hendrycks *et al.*, 2020] have focused on diversifying data while maintaining consistency. Based on this observation, we hypothesize that enhancing ray diversity while preserving consistency is critical for improving performance in few-shot view synthesis. To achieve this, DivCon-NeRF consists of surface-sphere and inner-sphere augmentations. As shown in Fig. 1, our method effectively reduces floaters and appearance distortions compared to existing approaches.

We first define a virtual 3D sphere centered at the inferred surface point, with its radius set to the distance between the camera and that point. We cast rays from both the surface and interior of the sphere toward the inferred surface point. In surface-sphere augmentation, a consistency mask is generated by comparing the order of high-probability surface points between the original and augmented rays, allowing the model to filter out inconsistent rays easily without relying on precise rendered depth. We propose a ray consistency loss that aligns surface points by comparing the similarity of blending weight distributions, applying a temperature factor to assign more weight to regions with higher surface probabilities. Additionally, we introduce a positional constraint to the bottleneck feature loss to account for the relative positions of points. For greater diversity, inner-sphere augmentation utilizes randomized angles and distances, providing a broader range of viewpoints.

Our DivCon-NeRF effectively increases diversity while maintaining consistency, resulting in improved rendering quality compared to that of recent NeRF-based methods. These results support our hypothesis that both diversity and consistency are essential for enhancing rendering performance. Our main contributions are as follows:

- We experimentally demonstrate the importance of ray diversity and consistency in augmentation, analyzing their effects in both object-centric scenes and those with diverse depth ranges.
- We propose DivCon-NeRF, a novel ray augmentation method that simultaneously enhances diversity and preserves consistency.
- DivCon-NeRF significantly reduces floaters and appearance distortions, outperforming recent few-shot NeRFs on the Blender, LLFF, and DTU datasets.
- Our method is compatible with other regularization- and framework-based methods, demonstrating strong generalizability.

2 Related Work

2.1 Neural Scene Representations

Neural scene representations have become a prominent method for encoding 3D scenes, achieving impressive results in tasks such as novel view synthesis [Mildenhall *et al.*, 2020; Barron *et al.*, 2022; Mildenhall *et al.*, 2022; Müller *et al.*, 2022] and 3D scene generation [Jain *et al.*, 2022; Poole *et al.*, 2022; Lin *et al.*, 2023]. These methods typically employ neural networks to model scene properties continuously, rather than relying on traditional discretized representations, such as meshes [Kanazawa *et al.*, 2018; Liao *et al.*, 2018; Pan *et al.*, 2019], voxels [Gadelha *et al.*, 2017; Xie *et al.*, 2019; Yu *et al.*, 2022], or point clouds [Achlioptas *et al.*, 2018; Thomas *et al.*, 2019]. Neural scene representations enable highly detailed and photorealistic scene reconstructions [Park *et al.*, 2021; Martin-Brualla *et al.*, 2021]. Among the various approaches, NeRF [Mildenhall *et al.*, 2020] has gained attention for its ability to synthesize novel views. However, NeRF-based models experience a significant performance drop when trained on sparse image data [Zhong *et al.*, 2024; Xu *et al.*, 2024], posing a critical challenge in practical applications.

2.2 Few-Shot View Synthesis

Various studies have aimed to address few-shot view synthesis in NeRF-based models. Prior-based methods [Chen *et al.*, 2021; Yu *et al.*, 2021] pre-train models on large-scale datasets to provide prior knowledge of 3D scenes. However, these methods incur substantial costs due to their reliance on large-scale datasets, and performance can degrade when the pre-training and target data distributions differ.

Framework-based methods modify network structures with models such as Transformers [Vaswani *et al.*, 2017] and CNNs [Krizhevsky *et al.*, 2012] and use these modified models during rendering. mi-MLP [Zhu *et al.*, 2024a] incorporates input embeddings into each MLP layer for flexible learning, while CVT-xRF [Zhong *et al.*, 2024] employs an in-voxel Transformer to refine radiance properties within voxels. However, these methods increase model complexity and limit integration with other framework-based methods. Additionally, appearance distortions persist due to sparse viewpoints (e.g., CVT-xRF; see Fig. 1).

Regularization-based methods optimize each scene using regularization terms [Deng *et al.*, 2022] or auxiliary training sources [Wang *et al.*, 2023]. FreeNeRF [Yang *et al.*, 2023] implements a frequency domain to mitigate artifacts. AR-NeRF [Xu *et al.*, 2024] adapts rendering loss to match the frequency progression of positional encoding. However, data scarcity remains unresolved, as these methods do not directly increase the number of training rays, leading to several floaters (e.g., FreeNeRF; see Fig. 1).

Additionally, recent few-shot 3D Gaussian Splatting (3DGS) methods [Zhang *et al.*, 2024; Li *et al.*, 2024; Zhu *et al.*, 2024b] have demonstrated promising performance in general few-shot scenarios. However, they encounter challenges in specific scenarios, such as background-free object rendering, which is crucial for practical applications like e-commerce and digital advertising.

2.3 Ray Augmentation for Sparse Images

Ray augmentation is a regularization-based approach that generates augmented rays from the originals. This approach [Chen *et al.*, 2022b; Seo *et al.*, 2023b; Truong *et al.*, 2023] effectively addresses the shortage of training rays by directly increasing their quantity. InfoNeRF [Kim *et al.*, 2022] reduces ray entropy to improve consistency across close views, while FlipNeRF [Seo *et al.*, 2023a] employs flipped reflection rays to improve 3D geometry estimation.

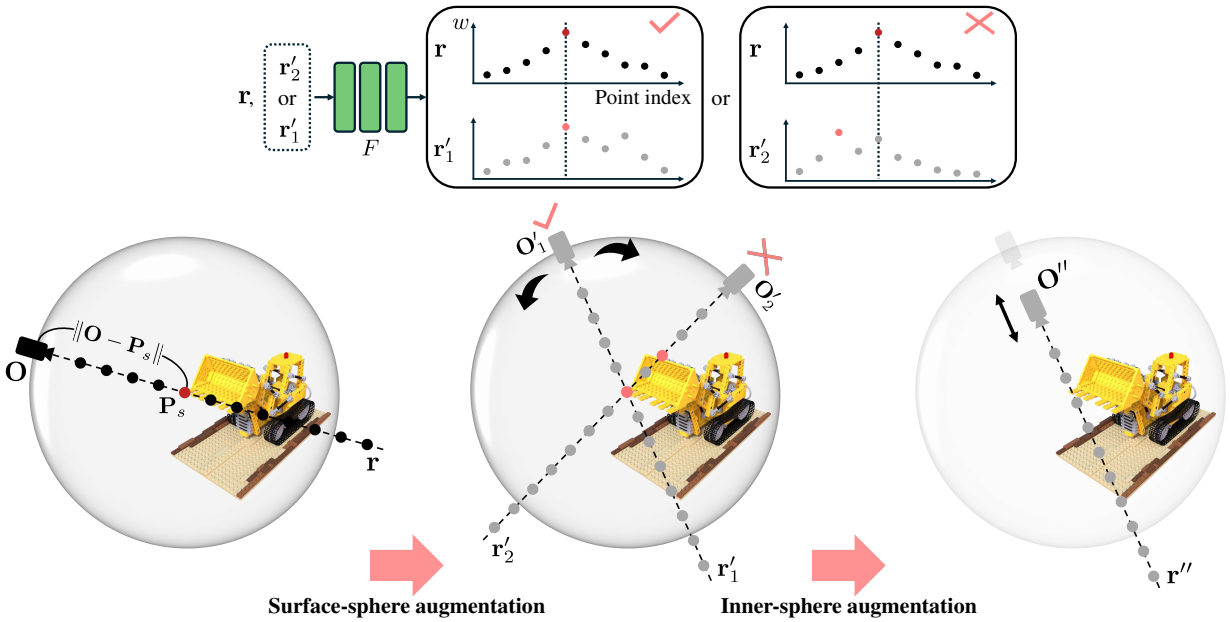


Figure 2: Overall structure of DivCon-NeRF. (Left) First, the virtual sphere is defined, centered at the predicted surface point $\mathbf{P}_s$. (Center) A surface-sphere augmented ray $\mathbf{r}'$ is then cast from a randomly selected position on the sphere's surface. (Right) To further enhance diversity, an inner-sphere augmented ray $\mathbf{r}''$ is generated with a randomly selected radius within the sphere and the same random angle as the surface-sphere augmented ray. (Top) To maintain consistency, the consistency mask filters out inconsistent rays (*e.g.*, $\mathbf{r}'_2$) if the sampled point with the highest blending weight does not have the same index as the original.

FrugalNeRF [Lin *et al.*, 2025] enforces novel view consistency by jointly optimizing poses and voxel features. However, these methods, which augment rays only near the originals, struggle with complex surfaces and nearby obstacles, causing inconsistencies. Furthermore, these methods generate augmented rays from limited viewpoints. GeCoNeRF [Kwak *et al.*, 2023] applies a depth-based mask to every pixel of a warped patch rather than directly to a ray. This mask relies on image warping, which causes interpolation issues and fails to maintain equal distances from surface points to both the original and pseudo cameras for each pixel, resulting in coarse filtering. Therefore, GeCoNeRF depends on pseudo-views near the training views. These limitations in existing methods lead to pronounced floaters and appearance distortions due to low diversity and inadequate consistency. In contrast, our proposed method enhances both diversity and consistency, effectively addressing these limitations.

3 Preliminaries: NeRF

NeRF [Mildenhall *et al.*, 2020] utilizes MLPs to synthesize novel views of a scene from densely sampled images. The NeRF network F maps 3D coordinates $\mathbf{X} = (x, y, z)$ and viewing direction $\mathbf{d}$ to color $\mathbf{c}$ and volume density σ:

$$F : (\mathbf{X}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma). \tag{1}$$

NeRF casts a single ray into 3D space for each pixel in the image. Because the ray is continuous, points along the ray are sampled. These sampled points determine the final pixel color C through volume rendering. The ray is expressed as $\mathbf{r}(t) = \mathbf{O} + t\mathbf{d}$, where $\mathbf{O}$ is the camera origin. The predicted pixel color $C(\mathbf{r})$ is calculated as follows:

$$C(\mathbf{r}) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) \mathbf{c}_i = \sum_{i=1}^{N} w_i \mathbf{c}_i, \tag{2}$$

where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$, and $\delta_i = t_{i+1} - t_i$. Blending weight w_i represents the contribution of the color at the i-th point to the pixel color $C(\mathbf{r})$. The NeRF network F is optimized using a mean square error (MSE) loss to ensure that the predicted pixel color $C(\mathbf{r})$ matches the ground truth color $C_{\text{gt}}(\mathbf{r})$ of the training rays:

$$\mathcal{L}_{\text{MSE}} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| C_{\text{gt}}(\mathbf{r}) - C(\mathbf{r}) \right\|^2, \tag{3}$$

where $\mathcal{R}$ represents a set of rays. However, NeRF requires dense training views and performs poorly with sparse inputs.

4 Method

We propose DivCon-NeRF, a ray augmentation method that simultaneously enhances ray diversity and preserves ray consistency. In contrast to depth-based image warping methods [Kwak *et al.*, 2023; Chen *et al.*, 2022a] that require a rendering for augmentation and suffer from interpolation issues, DivCon-NeRF augments rays directly in 3D space. This method combines surface-sphere augmentation (Sec. 4.1), which generates rays on a sphere, and inner-sphere augmentation (Sec. 4.2), which introduces random distance variations within the sphere, both directed toward the inferred surface point of the original ray. We introduce a consistency mask and three losses tailored to these augmentations. The overall scheme of DivCon-NeRF is depicted in Fig. 2.

4.1 Surface-Sphere Augmentation

In contrast to the typical approach [Mildenhall *et al.*, 2020], which positions the virtual sphere at the center of the object or 3D scene to determine camera positions, we center the virtual sphere on the predicted surface point $\mathbf{P}_s$ that the original ray hits. When the original and augmented rays cast from the sphere intersect at the same surface point, the augmented ray is unobstructed. We verify the consistency of the augmented rays by comparing the surface points hit by both rays.

For surface-sphere augmentation, the sphere's center is set at $\mathbf{P}_s$, the most likely surface point predicted by the model: $\mathbf{P}_s = \mathbf{O} + t_s\mathbf{d}$, where $s = \text{argmax}_i(w_i)$. Here, w_i denotes the blending weights of the original ray $\mathbf{r}$. The radius is defined as $\|\mathbf{O} - \mathbf{P}_s\|$. Consequently, the augmented ray's camera coordinates $\mathbf{O}'$ are given by:

$$\begin{aligned} x_{o'} &= \|\mathbf{O} - \mathbf{P}_s\| \sin(\theta)\cos(\phi) \\ y_{o'} &= \|\mathbf{O} - \mathbf{P}_s\| \sin(\theta)\sin(\phi) \\ z_{o'} &= \|\mathbf{O} - \mathbf{P}_s\| \cos(\theta), \end{aligned} \tag{4}$$

where (θ, ϕ) are spherical coordinates, with $\theta \sim U[0, \pi]$ and $\phi \sim U[0, 2\pi)$.

During training, the model progressively improves its prediction of $\mathbf{P}_s$, refining the accuracy of the virtual sphere. Notably, regardless of the point on the sphere from which an augmented ray is cast, all rays share the same distance R $(= t_s\|\mathbf{d}\|)$ and converge toward $\mathbf{P}_s$. This property of surface-sphere augmentation allows for the use of a consistency mask by estimating the index of the peak blending weight along the ray, eliminating the need for precise depth prediction.

Consistency Mask

To generate the consistency mask for augmented rays without constraining angle and distance, we consider all points on the virtual sphere as potential positions for $\mathbf{O}'$. In each iteration, a point on the sphere is randomly selected as $\mathbf{O}'$ for each original ray $\mathbf{r}$. The coordinates of the sampled points along the surface-sphere augmented ray $\mathbf{r}'$ are computed as $\mathbf{r}'(t) = \mathbf{O}' + t\mathbf{d}'$, where $\mathbf{d}'$ denotes the augmented ray's camera viewing direction:

$$\mathbf{d}' = \frac{\|\mathbf{d}\|\,(\mathbf{P}_s - \mathbf{O}')}{\|\mathbf{P}_s - \mathbf{O}'\|}. \tag{5}$$

The magnitude of $\mathbf{d}'$ is set equal to that of $\mathbf{d}$ to match the distribution of the original training data. The sampled points along $\mathbf{r}'$ are then fed into the model F. The point with the highest blending weight is identified as the most probable surface point along the augmented ray: $\mathbf{P}_{s'} = \mathbf{O}' + t_{s'}\mathbf{d}'$, where $s' = \text{argmax}_i(w'_i)$, with the blending weight w'_i derived from $\mathbf{r}'$. Note that this process occurs during coarse sampling, where points are sampled at equal intervals. Because $\|\mathbf{P}_s - \mathbf{O}\| = \|\mathbf{P}_s - \mathbf{O}'\|$ and $\delta = \delta'$, where δ and δ' represent the distances between adjacent samples on the original and augmented rays, respectively, $\mathbf{P}_{s'}$ of the consistent ray should be identical to $\mathbf{P}_s$. To exclude occluded rays from training, we define the following consistency mask:

$$M(\mathbf{r}, \mathbf{r}') = \begin{cases} 1 & \text{if } |s - s'| \le \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where s and s' denote the indices of the maximum blending weights along the rays $\mathbf{r}$ and $\mathbf{r}'$, respectively. We introduce the parameter ϵ as an error tolerance, given that this process occurs during coarse sampling. If the difference between the indices exceeds ϵ, the augmented ray is filtered out, indicating that the ray reaches a different surface point.

Accurately estimating w_i and w'_i at every point is challenging for a model during training. For example, GeCoNeRF [Kwak *et al.*, 2023], which relies on depth for masking warped patches, employs an additional depth smoothness loss to improve rendered depth accuracy. In contrast, determining the point with the highest blending weight via the argmax function is more straightforward and effective, resulting in fewer floaters and better performance. This claim is empirically validated in Sec. 5.3 and the supplement.

Ray Consistency Loss

In addition to the consistency mask, we introduce a ray consistency loss to further align the surface points of $\mathbf{r}$ and $\mathbf{r}'$, enhancing overall consistency. We apply a Kullback–Leibler (KL) divergence loss between the blending weight distributions of the original and augmented rays, using a temperature-scaled softmax. The ray consistency loss for each ray is defined as:

$$l_{\text{RC}}(\mathbf{r}, \mathbf{r}') = KL\left(P_T \| Q_T\right), \tag{7}$$

where

$$P_T(i) = \frac{\exp(w_i/T)}{\sum_j \exp(w_j/T)} \text{ and } Q_T(i) = \frac{\exp(w'_i/T)}{\sum_j \exp\left(w'_j/T\right)}.$$

The temperature T sharpens these softmax distributions, emphasizing their peak regions. As KL divergence primarily penalizes mismatches in high-probability regions, it encourages the peak locations of these distributions to align during training. As this loss functions as a regularizer, it allows flexibility in low-probability regions. Note that augmented rays far from the originals may not exhibit strong global similarity but should exhibit consistent peak locations near the surface.

In the LLFF dataset [Mildenhall *et al.*, 2019], which contains many objects within a scene, augmented rays farther from the original rays tend to have more dissimilar overall distributions. For these augmented rays located beyond a certain range, we apply clipping to w_i and w'_i by setting them to 0 when $i > s$, and then obtain $P_T(i)$ and $Q_T(i)$. Finally, the overall ray consistency loss $\mathcal{L}_{\text{RC}}$ for a set of rays $\mathcal{R}$ is defined as:

$$\mathcal{L}_{\text{RC}} = \sum_{\mathbf{r} \in \mathcal{R}} M(\mathbf{r}, \mathbf{r}') \cdot l_{RC}(\mathbf{r}, \mathbf{r}'). \tag{8}$$

In contrast to InfoNeRF [Kim *et al.*, 2022], which only applies information regularization to augmented rays with minimal viewpoint variations, our approach leverages temperature scaling and clipping to enforce a strong consistency constraint near the surface, utilizing augmented rays across all ranges for $\mathcal{L}_{\text{RC}}$. Additional comparison experiments are detailed in the supplementary material.

Positional Bottleneck Feature Loss

We introduce a positional constraint to the bottleneck feature loss as proposed alongside FlipNeRF [Seo *et al.*, 2023a]. In

contrast to FlipNeRF, which fails to maintain a consistent positional relationship among sampled points due to the varying normal vector magnitudes, we utilize sampled points from the original and augmented rays that are equidistant from the predicted surface point. The positional bottleneck feature loss is computed using Jensen-Shannon divergence as follows:

$$l_{\mathrm{PBF}}\left(\mathbf{r}, \mathbf{r}'\right) = \sum_{i=1}^{N} \frac{1}{N} JSD\left(\psi\left(h\left(\mathbf{X}_i\right)\right), \psi\left(h\left(\mathbf{X}_i'\right)\right)\right), \quad (9)$$

where $\|\mathbf{P}_s - \mathbf{X}_i\| = \|\mathbf{P}_s - \mathbf{X}_i'\|$, ψ denotes the softmax function, and $h(\cdot)$ maps the input to the last layer's feature space. This constraint effectively reduces geometric discrepancies between $\mathbf{X}_i$ and $\mathbf{X}_i'$ for reliable feature matching, as demonstrated in Sec. 5.3. Therefore, we obtain l_{PBF} through the coarse sampling of surface-sphere augmentation, excluding inner-sphere augmentation. The overall positional bottleneck feature loss $\mathcal{L}_{\mathrm{PBF}}$ is defined as:

$$\mathcal{L}_{\mathrm{PBF}} = \sum_{\mathbf{r}\in\mathcal{R}} M\left(\mathbf{r}, \mathbf{r}'\right) \cdot l_{\mathrm{PBF}}\left(\mathbf{r}, \mathbf{r}'\right). \quad (10)$$

4.2 Inner-Sphere Augmentation

In addition to surface-sphere augmentation, inner-sphere augmentation randomly samples the distance R within the sphere in each iteration. This further enhances diversity and improves the model's robustness to variations in the camera's distance from the surface. When a pseudo camera is positioned farther from the surface point than the original camera, a single pixel represents a larger area, covering both the ground truth and its neighboring pixels, making it more difficult to maintain color consistency (see supplementary material for further analysis). Therefore, the inner-sphere augmented ray's camera coordinates $\mathbf{O}''$ are given by:

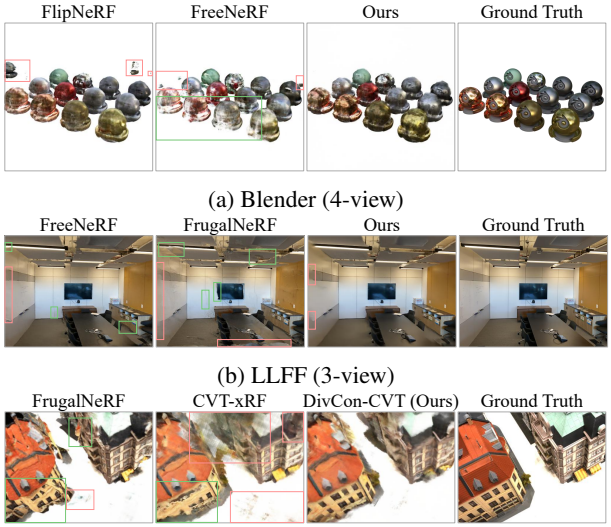
$$\begin{aligned} x_{o''} &= r\|\mathbf{O} - \mathbf{P}_s\| \sin(\theta)\cos(\phi) \\ y_{o''} &= r\|\mathbf{O} - \mathbf{P}_s\| \sin(\theta)\sin(\phi) \\ z_{o''} &= r\|\mathbf{O} - \mathbf{P}_s\| \cos(\theta), \end{aligned} \quad (11)$$

where $r \sim U(0, 1]$, and θ and ϕ are common random variables shared with surface-sphere augmentation, resulting in the same viewing direction as $\mathbf{d}'$. Therefore, we apply the same consistency mask, which detects obstacles within the sphere along the line containing both augmented rays. Although this may filter out a few consistent rays, it significantly reduces the likelihood of training on inconsistent rays.

Instead of using MSE loss, we extend the NLL loss from MixNeRF [Seo *et al.*, 2023b] to our inner-sphere augmented ray $\mathbf{r}''$, enabling the model to maintain color consistency across diverse viewpoints:

$$\mathcal{L}_{\mathrm{MNLL}} = -\sum_{\mathbf{r}\in\mathcal{R}} M\left(\mathbf{r}, \mathbf{r}'\right) \cdot \log p\left(C_{\mathrm{gt}} \mid \mathbf{r}''\right), \quad (12)$$

where C_{gt} is the ground truth color of $\mathbf{r}$, and $p(C_{\mathrm{gt}} \mid \mathbf{r}'') = \sum_{i=1}^{N} \pi_i \mathcal{F}(C_{\mathrm{gt}} \mid \mathbf{r}'')$. Here, $\pi_i = \frac{w_i''}{\sum_{m=1}^{N} w_m''}$, w_i'' denotes the blending weights of $\mathbf{r}''$, and $\mathcal{F}$ denotes a Laplacian distribution centered at the predicted color $C(\mathbf{r}'')$. Although the rays converge at the same surface point, the consistent augmented ray does not necessarily have the same ground truth color as the original ray due to view-dependent effects. As the ground truth for $\mathbf{r}''$ is unavailable, we utilize the predicted color $C(\mathbf{r}'')$ from coarse sampling and a probability-based NLL loss to handle view-dependent variations while maintaining color consistency. This allows the augmented ray to produce a visually consistent color that adapts naturally across views.



(a) Blender (4-view)

(b) LLFF (3-view)

(c) DTU (3-view)

Figure 3: Qualitative comparisons on the Blender, LLFF, and DTU datasets. Red boxes indicate severe floaters, and green boxes indicate significant visual distortions. Our method exhibits fewer floaters and visual distortions than other NeRF-based methods across all three datasets.

4.3 Training Loss

The overall training loss is defined as follows, with λ terms representing balancing weights for the respective losses:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\mathrm{MSE}} + \lambda_1\mathcal{L}_{\mathrm{RC}} + \lambda_2\mathcal{L}_{\mathrm{PBF}} \\ + \lambda_3\mathcal{L}_{\mathrm{MNLL}} + \lambda_4\mathcal{L}_{\mathrm{NLL}} + \lambda_5\mathcal{L}_{\mathrm{UE}}. \end{aligned} \quad (13)$$

Here, $\mathcal{L}_{\mathrm{MSE}}$ is calculated using the original ray, $\mathcal{L}_{\mathrm{NLL}}$ is the NLL loss of the original ray, introduced with MixNeRF [Seo *et al.*, 2023b], and $\mathcal{L}_{\mathrm{UE}}$ refers to the uncertainty-aware emptiness loss introduced with FlipNeRF [Seo *et al.*, 2023a], stabilizing the blending weights to prevent fluctuations. The supplementary material provides more details of both losses.

5 Experiments

5.1 Datasets and Metrics

We evaluate our method against recent NeRF-based methods without external generative priors on the Blender [Mildenhall *et al.*, 2020], LLFF [Mildenhall *et al.*, 2019], and DTU [Jensen *et al.*, 2014] datasets. For reference, we report the results of recent few-shot 3DGS methods on Blender; their performance on the LLFF and DTU datasets is reported in the supplementary material. The Blender dataset comprises 8 synthetic, background-free, object-centric scenes, while LLFF consists of 8 forward-facing real-world scenes.

Method	Setting	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
Mip-NeRF [Barron *et al.*, 2021]	Baseline	14.12	0.722	0.382	0.221
InfoNeRF [Kim *et al.*, 2022]	Regularization-based	18.44	0.792	0.217	0.118
RegNeRF [Niemeyer *et al.*, 2022]		13.71	0.786	0.339	0.208
MixNeRF [Seo *et al.*, 2023b]		18.99	0.807	0.199	0.113
FlipNeRF [Seo *et al.*, 2023a]		20.60	0.822	0.159	0.091
FreeNeRF* [Yang *et al.*, 2023]		20.06	0.815	0.141	0.092
CVT-xRF* [Zhong *et al.*, 2024]	Framework-based	19.98	0.819	0.178	0.096
mi-MLP* [Zhu *et al.*, 2024a]		20.38	0.828	0.156†	0.084†
DivCon-NeRF (Ours)	Regularization-based	<u>22.01</u>	**0.843**	**0.127**	**0.073**
DivCon-NeRF* (Ours)		**22.08**	<u>0.839</u>	<u>0.129</u>	<u>0.074</u>
DNGaussian* [Li *et al.*, 2024]	3DGS	19.50	0.808	0.147	0.093
FSGS* [Zhu *et al.*, 2024b]	3DGS	17.60	0.677	0.257	0.140

Table 1: Quantitative comparison on the Blender dataset using 4 views. We report the results of NeRF-based approaches and, for reference, recent few-shot 3DGS methods (bottom). * indicates experiments conducted under DietNeRF [Jain *et al.*, 2021] protocols. † indicates results reported in the original paper, as the official code was not publicly available. The best scores are in bold, and the second-best are underlined. The same notations are used in the following tables.

Method	Setting	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
Mip-NeRF [Barron *et al.*, 2021]	Baseline	15.53	0.363	0.445	0.223
SRF-ft [Chibane *et al.*, 2021]	Prior-based	17.07	0.436	0.496	0.198
PixelNeRF-ft [Yu *et al.*, 2021]		16.17	0.438	0.473	0.210
MVSNeRF-ft [Chen *et al.*, 2021]		17.88	0.584	0.260	0.143
RegNeRF [Niemeyer *et al.*, 2022]	Regularization-based	19.08	0.587	0.263	0.132
FlipNeRF [Seo *et al.*, 2023a]		19.34	0.631	0.235	0.123
FreeNeRF [Yang *et al.*, 2023]		19.63	0.612	0.240	0.122
AR-NeRF [Xu *et al.*, 2024]		<u>19.90</u>	<u>0.635</u>	0.283†	0.126†
FrugalNeRF [Lin *et al.*, 2025]		19.49	0.621	<u>0.229</u>	<u>0.121</u>
mi-MLP [Zhu *et al.*, 2024a]	Framework-based	19.75	0.614	0.300†	0.125†
DivCon-NeRF (Ours)	Regularization-based	**20.46**	**0.662**	**0.221**	**0.109**

Table 2: Quantitative comparison on the LLFF dataset under the 3-view setting. Our method achieves the best performance among all NeRF-based methods. We report the fine-tuning (ft) performance for prior-based methods.

For the DTU dataset, we use 15 scenes following the protocol established by RegNeRF [Niemeyer *et al.*, 2022].

We report PSNR, SSIM [Wang *et al.*, 2004], LPIPS [Zhang *et al.*, 2018], and the average error (AVGE) calculated as the geometric mean of MSE $= 10^{-\text{PSNR}/10}$, $\sqrt{1-\text{SSIM}}$, and LPIPS, following Niemeyer et al. (2022). LPIPS is evaluated using AlexNet [Krizhevsky *et al.*, 2012], following Zhong et al. [2024]. Additional experimental details and extended results with more input views across Blender, LLFF, and DTU are provided in the supplement.

5.2 Comparisons

Blender

Our method significantly outperforms all compared NeRF-based approaches on the Blender dataset, as shown in Tab. 1. For a fair comparison, we follow the experimental protocols of MixNeRF [Seo *et al.*, 2023b] (no mark) and DietNeRF [Jain *et al.*, 2021] (marked by *). In both cases, our method consistently achieves the highest performance by a substantial margin. As illustrated in Fig. 3a, it effectively reduces floaters and appearance distortions compared to existing methods, thereby reinforcing the importance of consistency and diversity, as hypothesized. For completeness, we include the results of recent few-shot 3DGS methods [Zhu *et al.*, 2024b; Li *et al.*, 2024] at the bottom of Tab. 1; these perform notably worse on background-free, object-centric scenes (see Sec. 5.4 and the supplementary material).

Method	Setting	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
Mip-NeRF [Barron *et al.*, 2021]	Baseline	9.02	0.570	0.339	0.309
SRF-ft [Chibane *et al.*, 2021]	Prior-based	15.68	0.698	0.260	0.160
PixelNeRF-ft [Yu *et al.*, 2021]		18.95	0.710	0.242	0.121
MVSNeRF-ft [Chen *et al.*, 2021]		18.54	0.769	0.168	0.107
MixNeRF [Seo *et al.*, 2023b]	Regularization-based	18.95	0.744	0.203	0.113
FlipNeRF [Seo *et al.*, 2023a]		19.55	0.767	0.180	0.101
FreeNeRF [Yang *et al.*, 2023]		19.92	0.787	0.147	0.092
AR-NeRF [Xu *et al.*, 2024]		20.36	0.788	0.187†	0.095†
FrugalNeRF [Lin *et al.*, 2025]		<u>21.06</u>	0.792	0.160	0.087
CVT-xRF [Zhong *et al.*, 2024]	Framework-based	21.00	<u>0.837</u>	<u>0.132</u>	<u>0.077</u>
DivCon-Flip (Ours)	(Reg.+Reg.)-based	19.95	0.769	0.163	0.096
DivCon-CVT (Ours)	(Reg.+Frame.)-based	**22.01**	**0.858**	**0.130**	**0.069**

Table 3: Quantitative comparison on the DTU under the 3-view setting. We evaluate our method's integration with existing approaches. Results are evaluated by masked metrics following Niemeyer et al. [2022].

LLFF

As shown in Tab. 2, our method achieves a substantial performance improvement over other approaches, demonstrating its effectiveness across both synthetic and real-world datasets. As shown in Fig. 3b, our method more effectively prevents geometric distortions compared with other methods owing to enhanced consistency, better preserving the continuity of straight lines on wall surfaces.

DTU

To demonstrate the generalizability of our method, we integrate it with the regularization-based FlipNeRF [Seo *et al.*, 2023a] and the framework-based CVT-xRF [Zhong *et al.*, 2024]. As shown in Tab. 3, our DivCon-Flip outperforms FlipNeRF in all metrics, and our DivCon-CVT achieves the best performance. Integrating our method with CVT-xRF substantially improves PSNR, SSIM, and AVGE while closely maintaining LPIPS. This suggests that our method is architecture-agnostic and effectively complements framework-based methods. As illustrated in Fig. 3c, our method yields cleaner renderings than other methods.

5.3 Analysis

Ray Diversity

We evaluate the impact of ray diversity on rendering quality by progressively expanding the ranges of θ, ϕ, and r. As shown in Fig. 4, low diversity results in severe floaters (red boxes), while increased diversity notably reduces these artifacts, indicating the importance of augmented rays from diverse viewpoints. In particular, when the angular range is narrow, more floaters and pronounced visual distortions are observed (green boxes). However, employing a full viewing angle without the consistency mask can lead to challenges in maintaining ray consistency.

Absence of Consistency Mask

To analyze the importance of consistency, we evaluate our method without the consistency mask. In object-centric scenes, as shown in Fig. 5a, reduced consistency leads to significant visual distortions. The corresponding depth map reveals empty regions on the object surfaces, indicating that the rays fail to locate the surface. This suggests that inconsistent rays are used during training, increasing uncertainty in depth

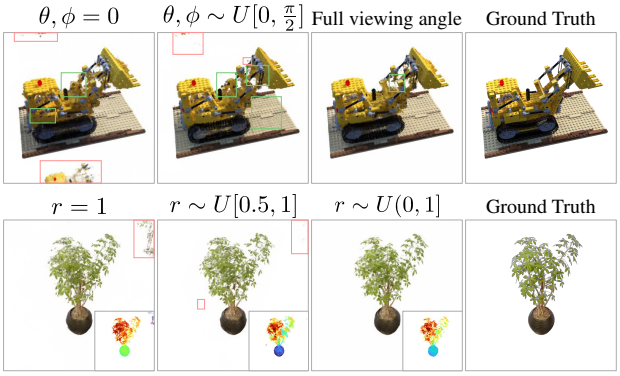
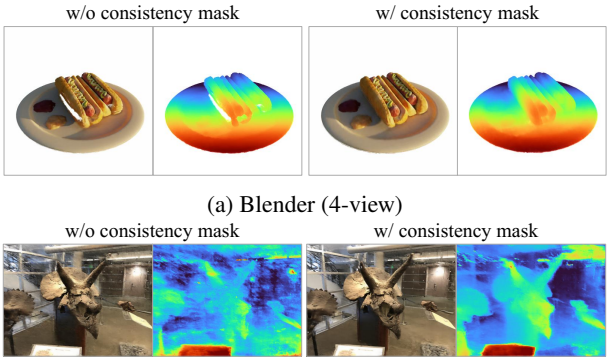


Figure 4: Analysis of diversity for angle and radius on Blender. The insets in the second row show depth maps. Best viewed when enlarged.



(a) Blender (4-view)

(b) LLFF (3-view)

Figure 5: Comparison of rendered RGB and depth images with and without the consistency mask.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
DivCon-NeRF (w/ argmax)	**20.46**	**0.662**	**0.221**
DivCon-NeRF (w/ depth)	19.25	0.607	0.289

Table 4: Comparison of using the argmax function and rendered depth on LLFF.

estimation and resulting in unreliable depth predictions. This highlights the crucial role of consistency in ray augmentation for preventing visual distortions. As shown in Fig. 5b, diversity alone cannot eliminate many floaters in the LLFF scene with varying depths. In this case, consistency masks effectively reduce floaters and appearance distortions. Therefore, our model demonstrates robust performance across both synthetic and real-world scenes by enhancing diversity and consistency.

Limitations of Using Rendered Depth

We compare the performance of our consistency mask when using the argmax function versus rendered depth. Using rendered depth results in lower performance, as shown in Tab. 4. This suggests, that because the depth calculation depends on accurate blending weights for all points, the argmax-based masking is more effective for maintaining consistency.

	Base	$\mathcal{L}_{\text{RC}}$	$\mathcal{L}_{\text{PBF}}$	$\mathcal{L}_{\text{MNLL}}$	$\mathcal{L}_{\text{BF}}$	PSNR ↑	SSIM ↑	LPIPS ↓
(1)	✓					18.86	0.803	0.212
(2)	✓	✓				19.60	0.802	0.182
(3)	✓	✓	✓			20.69	0.820	0.168
(4)	✓	✓	✓	✓		**22.08**	**0.839**	**0.129**
(5)	✓	✓		✓	✓	20.73	0.817	0.149

Table 5: Ablation study. Base includes $\mathcal{L}_{\text{NLL}}$ and $\mathcal{L}_{\text{UE}}$. We highlight the effect of the newly introduced loss terms.

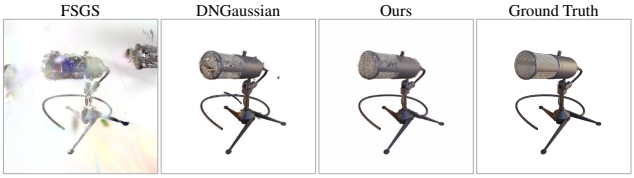


Figure 6: Qualitative comparison of our method with recent few-shot 3DGS methods on Blender in the 4-view setting.

Ablation Study

Tab. 5 shows the impact of each component in our method. In (2), adding only $\mathcal{L}_{\text{RC}}$ to align surface points shows a notable performance improvement, except for SSIM. As shown in (3), incorporating $\mathcal{L}_{\text{PBF}}$ significantly surpasses the baseline across all metrics. Adding $\mathcal{L}_{\text{MNLL}}$ of inner-sphere augmented rays achieves the ideal performance in (4), underscoring the importance of enhancing diversity while preserving consistency.

In (5), $\mathcal{L}_{\text{BF}}$ represents the bottleneck feature loss proposed with FlipNeRF [Seo *et al.*, 2023a]. Substituting $\mathcal{L}_{\text{PBF}}$ with $\mathcal{L}_{\text{BF}}$ results in substantial overall performance degradation, indicating the importance of incorporating relative positional information for reliable feature matching.

5.4 Effectiveness in Background-Free Scenes

Our method is highly effective in background-free, object-centric scenes. In contrast to recent few-shot 3DGS methods [Zhu *et al.*, 2024b; Li *et al.*, 2024] which suffer from significant distortions and floaters (Fig. 6, Tab. 1), our approach produces cleaner renderings with no floaters. This makes it especially suitable for applications, such as e-commerce and digital advertising, where accurate multi-view rendering of isolated products is essential for flexible scene composition. Additional analysis on Blender and other scene types is provided in the supplementary material.

6 Conclusion

In this paper, we present DivCon-NeRF, which effectively reduces floaters and visual distortions in NeRF-based few-shot view synthesis. Our consistency mask effectively filters out inconsistent augmented rays, making us the first to utilize pseudo views from all 360-degree directions based on a virtual sphere. Our method simultaneously enhances diversity and preserves consistency, with experimental results confirming its effectiveness in both object-centric scenes and real-world environments. Our results demonstrate that ray diversity and consistency are crucial for few-shot NeRF.

Acknowledgments

This work was supported by the Korean Government through the grants from IITP (RS-2021-II211343 and RS-2025-25442338) and KOCCA (RS-2024-00398320).

References

[Achlioptas *et al.*, 2018] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, pages 40–49, 2018.

[Barron *et al.*, 2021] 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 *ICCV*, pages 5855–5864, 2021.

[Barron *et al.*, 2022] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022.

[Chen *et al.*, 2021] 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 *ICCV*, pages 14124–14133, 2021.

[Chen *et al.*, 2022a] Di Chen, Yu Liu, Lianghua Huang, Bin Wang, and Pan Pan. Geoaug: Data augmentation for few-shot nerf with geometry constraints. In *ECCV*, pages 322–337, 2022.

[Chen *et al.*, 2022b] Tianlong Chen, Peihao Wang, Zhiwen Fan, and Zhangyang Wang. Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations. In *CVPR*, pages 15191–15202, 2022.

[Chibane *et al.*, 2021] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *CVPR*, pages 7911–7920, 2021.

[Cubuk *et al.*, 2019] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pages 113–123, 2019.

[Cubuk *et al.*, 2020] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 702–703, 2020.

[Deng *et al.*, 2022] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, pages 12882–12891, 2022.

[Gadelha *et al.*, 2017] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3DV*, pages 402–411, 2017.

[Hendrycks *et al.*, 2020] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020.

[Jain *et al.*, 2021] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, pages 5885–5894, 2021.

[Jain *et al.*, 2022] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, pages 867–876, 2022.

[Jensen *et al.*, 2014] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *CVPR*, pages 406–413, 2014.

[Kanazawa *et al.*, 2018] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, pages 371–386, 2018.

[Kim *et al.*, 2022] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, pages 12912–12921, 2022.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, volume 25, 2012.

[Kwak *et al.*, 2023] Minseop Kwak, Jiuhn Song, and Seungryong Kim. Geconerf: Few-shot neural radiance fields via geometric consistency. In *ICML*, 2023.

[Lee *et al.*, 2024] Ingyun Lee, Wooju Lee, and Hyun Myung. Domain generalization with vital phase augmentation. In *AAAI*, volume 38, pages 2892–2900, 2024.

[Li *et al.*, 2024] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *CVPR*, pages 20775–20785, 2024.

[Liao *et al.*, 2018] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *CVPR*, pages 2916–2925, 2018.

[Lin *et al.*, 2023] 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 *CVPR*, pages 300–309, 2023.

[Lin *et al.*, 2025] Chin-Yang Lin, Chung-Ho Wu, Chang-Han Yeh, Shih-Han Yen, Cheng Sun, and Yu-Lun Liu. Frugalnerf: Fast convergence for extreme few-shot novel view synthesis without learned priors. In *CVPR*, pages 11227–11238, 2025.

[Martin-Brualla *et al.*, 2021] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021.

[Mildenhall *et al.*, 2019] 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*, 38(4):29:1–29:14, 2019.

[Mildenhall *et al.*, 2020] 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. In *ECCV*, pages 405–421, 2020.

[Mildenhall *et al.*, 2022] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, pages 16190–16199, 2022.

[Müller *et al.*, 2022] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, July 2022.

[Niemeyer *et al.*, 2022] 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 *CVPR*, pages 5480–5490, 2022.

[Pan *et al.*, 2019] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *CVPR*, pages 9964–9973, 2019.

[Park *et al.*, 2021] 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 *ICCV*, pages 5865–5874, 2021.

[Poole *et al.*, 2022] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022.

[Seo *et al.*, 2023a] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *ICCV*, pages 22883–22893, 2023.

[Seo *et al.*, 2023b] Seunghyeon Seo, Donghoon Han, Yeonjin Chang, and Nojun Kwak. Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In *CVPR*, pages 20659–20668, 2023.

[Thomas *et al.*, 2019] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *CVPR*, pages 6411–6420, 2019.

[Truong *et al.*, 2023] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, pages 4190–4200, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

[Wang *et al.*, 2004] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[Wang *et al.*, 2023] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *CVPR*, pages 9065–9076, 2023.

[Xie *et al.*, 2019] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, pages 2690–2698, 2019.

[Xie *et al.*, 2020] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, volume 33, pages 6256–6268, 2020.

[Xu *et al.*, 2024] Qingshan Xu, Xuanyu Yi, Jianyao Xu, Wenbing Tao, Yew-Soon Ong, and Hanwang Zhang. Few-shot nerf by adaptive rendering loss regularization. In *ECCV*, 2024.

[Yang *et al.*, 2023] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, pages 8254–8263, 2023.

[Yu *et al.*, 2021] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. Pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021.

[Yu *et al.*, 2022] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, pages 5501–5510, 2022.

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[Zhang *et al.*, 2024] Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: Sparse-view 3d gaussian splatting via co-regularization. In *ECCV*, pages 335–352, 2024.

[Zhong *et al.*, 2024] Yingji Zhong, Lanqing Hong, Zhenguo Li, and Dan Xu. Cvt-xrf: Contrastive in-voxel transformer for 3d consistent radiance fields from sparse inputs. In *CVPR*, pages 21466–21475, 2024.

[Zhu *et al.*, 2024a] Hanxin Zhu, Tianyu He, Xin Li, Bingchen Li, and Zhibo Chen. Is vanilla mlp in neural radiance field enough for few-shot view synthesis? In *CVPR*, pages 20288–20298, 2024.

[Zhu *et al.*, 2024b] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *ECCV*, pages 145–163, 2024.