Empowering Sparse-Input Neural Radiance Fields with Dual-Level Semantic Guidance from Dense Novel Views

Yingji Zhong[1], Kaichen Zhou[2], Zhihao Li[2], Lanqing Hong[2], Zhenguo Li[2], Dan Xu[1]

[1]The Hong Kong University of Science and Technology
[2]Huawei Noah's Ark Lab

Abstract

Neural Radiance Fields (NeRF) have shown remarkable capabilities for photorealistic novel view synthesis. One major deficiency of NeRF is that dense inputs are typically required, and the rendering quality will drop drastically given sparse inputs. In this paper, we highlight the effectiveness of rendered semantics from dense novel views, and show that rendered semantics can be treated as a more robust form of augmented data than rendered RGB. Our method enhances NeRF's performance by incorporating guidance derived from the rendered semantics. The rendered semantic guidance encompasses two levels: the supervision level and the feature level. The supervision-level guidance incorporates a bi-directional verification module that decides the validity of each rendered semantic label, while the feature-level guidance integrates a learnable codebook that encodes semantic-aware information, which is queried by each point via the attention mechanism to obtain semantic-relevant predictions. The overall semantic guidance is embedded into a self-improved pipeline. We also introduce a more challenging sparse-input indoor benchmark, where the number of inputs is limited to as few as 6. Experiments demonstrate the effectiveness of our method and it exhibits superiority compared to existing approaches.

Introduction

Neural Radiance Fields (NeRF) (Mildenhall et al. 2020; Barron et al. 2021; Liu et al. 2020; Barron et al. 2022; Müller et al. 2022; Barron et al. 2023) have shown remarkable improvement on novel view synthesis. Albeit impressive, dense input views are required to train the NeRF. Given sparse input views, it is likely for the NeRF to learn a trivial solution that can explain all training views but fail to model accurate geometry and appearance of the scene, known as the shape-radiance ambiguity (Zhang et al. 2020). The ambiguity is caused by the incapability of the model to build up correspondences across training views from sparse RGB supervision, causing severe artifacts in novel view synthesis.

Previous works applied different methods to improve the performance in sparse-input setting (Deng et al. 2022; Niemeyer et al. 2022; Truong et al. 2023; Wynn and Turmukhambetov 2023). Although great improvements have been achieved, the above methods typically overlook the fact that, for a NeRF trained from sparse inputs, its novel-view rendered images can be treated as augmented data. Since the camera poses of novel views are arbitrary, the augmented data can be dense. Another important observation is that, despite the existing artifacts in rendered novel view images, we can tell the semantics of each part in the image. It also holds for a trained NeRF, i.e., it can render basically accurate semantics in novel views, as shown in Fig. 1. The rich semantics in the dense novel views can be used as augmented data to train another NeRF, which can facilitate the model to overcome the ambiguity of the view correspondences, thus leading to improvements in novel view synthesis.

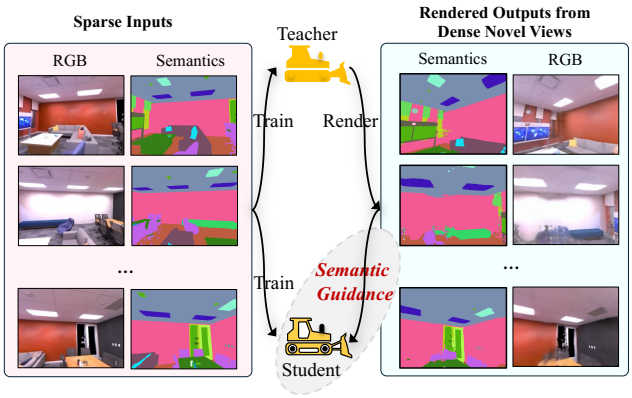

Figure 1: Our method exploits rendered semantics from dense novel views to boost sparse-input NeRF performance. The rendered semantics is exploited by semantic guidance, encompassing supervision-level and feature-level. Our method is embedded into a self-improved framework.

In this paper, we embed the aforementioned motivation into a self-improved framework, dubbed as Dense **S**emantic Guidance for Neural Radiance Fields from **S**parse Inputs with **S**elf-Improvement (S^3NeRF). As shown in Fig. 1, after the teacher NeRF has been trained with sparse inputs of RGB and semantics (Zhi et al. 2021), we can obtain densely rendered outputs of novel views, including rendered RGB and semantics. Along with sparse inputs, we further train the student NeRF with the *rendered semantic guidance*. Note that we do not claim the self-improved framework as our main contribution, which is already explored in (Bai et al. 2023; Jung et al. 2023). Our main contribution lies in *the first time to use rendered semantics* in self-improved frame-

work for sparse-input NeRF, rather than considering rendered RGB as in (Bai et al. 2023; Jung et al. 2023), and accordingly design different guidance methods to improve the effectiveness of using the rendered semantics. Applying rendered RGB as the augmented data might hamper the NeRF training due to artifacts in rendered images, while the rendered semantics can intuitively serve as more reliable augmented data, as we can clearly observe the rendered semantics of different regions regardless of the pixel artifacts of shifted colors or blurs, as illustrated in Fig. 1.

We fully exploit the rendered novel-view semantics as guidance for learning the student NeRF from two perspectives. One is a *supervision-level guidance* and the other is a *feature-level guidance*, as illustrated in Fig. 2. Specifically, **(i)** for the supervision-level guidance, the rendered semantics serves as an additional supervision signal for the student NeRF. It helps the NeRF to alleviate the ambiguity by building up the correspondences across views with the assistance of semantic labels. However, it is unavoidable that there exist incorrect rendered labels, which may harm the learning of correspondences. Therefore, we propose a Bi-Directional Verification (BDV) module to tackle the issue of incorrect semantic labels. The rendered semantic label of a pixel in the novel view is considered as valid only if a consensus constraint based on projection is satisfied. Hence, the student NeRF is trained to be a decent semantic field, but with limited constraints on the colors in novel views. **(ii)** To further exploit the supervision from semantic labels, we propose a feature-level guidance module that incorporates a learnable codebook in the MLP. The codebook learns semantic-aware patterns, which are expected to encode the correlation among semantics, colors, and densities. This module further compensates at the feature level for 3D points that receive only the semantic supervision.

Previous works on sparse-input NeRF target at face-forwarding scenarios (Jensen et al. 2014; Mildenhall et al. 2019), or 360° scenarios (Mildenhall et al. 2020). In these scenarios, views are sampled in an "outside-in" or "face-forwarding" manner, meaning that a certain amount of overlap exists across views. Roessle *et al.* (Roessle et al. 2022) introduce a setting on ScanNet (Dai et al. 2017), where the viewing direction follows an "inside-out" pattern, while the input view number is 18. To validate the effectiveness of our method in a more challenging scenario, we introduce a setting based on the indoor scenes of ScanNet++ (Yeshwanth et al. 2023) and Replica (Straub et al. 2019), where the viewing direction is also "inside-out", but we largely reduce the number of input views to 6. It is sufficient to cover the entire scene but with remarkably less overlap across views.

Our contributions are: **(i)** We introduce S^{3}NeRF, the first work leveraging rendered semantics from dense novel views for sparse-input NeRF, which is built upon a self-improved framework; **(ii)** The semantic guidance is incorporated from two perspectives, i.e., supervision-level and feature-level. The former is implemented with a BDV module, while the latter is realized by a learnable semantic-aware codebook; **(iii)** We present a sparse-input indoor benchmark utilizing as few as 6 input views, which is significantly more challenging compared to current sparse-input settings.

Related Works

NeRF from Sparse Inputs. Though recent works (Mildenhall et al. 2020; Barron et al. 2021, 2022; Müller et al. 2022; Barron et al. 2023) achieve impressive results on novel view synthesis, dense inputs are typically required. To tackle sparse inputs, some works learn a generalizable NeRF (Yu et al. 2021; Wang et al. 2021; Chen et al. 2021) on multi-view datasets. Other works train a scene-specific NeRF by regularization (Jain, Tancik, and Abbeel 2021; Kim, Seo, and Han 2022; Deng et al. 2022; Roessle et al. 2022; Kwak, Song, and Kim 2023; Yang, Pavone, and Wang 2023; Wang et al. 2023; Zhong et al. 2024). We also train a scene-specific NeRF from sparse inputs. Different from current works, we use rendered semantics from a trained NeRF as augmented data to guide another NeRF in a self-improved manner.

NeRF for 3D Scene Understanding. Recent works have explored to utilize NeRF for 3D scene understanding (Zhi et al. 2021; Zhang et al. 2023; Kundu et al. 2022; Siddiqui et al. 2023). These methods target 3D scene understanding with NeRF from dense-view inputs. However, we focus on learning a NeRF from sparse inputs. We also encode semantics into the NeRF, but with a clearly different purpose: we use the semantics to guide the NeRF to improve the geometry and appearance modeling in the sparse-input setting.

Self-Training NeRF. Self-training methods firstly train a teacher model with sparsely labeled data. The teacher model is then used to generate pseudo labels for unlabeled data, which is combined with the labeled data to train a student model. There are also attempts of employing the self-training paradigram on NeRF (Bai et al. 2023; Jung et al. 2023), which apply rendered RGB to train the student NeRF. However, the rendered RGB values are mostly unreliable, especially in the sparse-input settings where floating artifacts, blurs, and color shiftings widely exist. Since NeRF is trained by RGB correspondences across views, the unreliable rendered RGB values might even exacerbate the ambiguity and thus hamper the modeling performance. In this paper, we instead utilize rendered semantics from a trained NeRF as the pseudo labels. We show in the experiments that the semantic guidance is more effective than RGB values.

Methodology

Preliminary

NeRF represents a scene with an MLP, which predicts the color $\mathbf{c}$ and density σ of each 3D point $\mathbf{x}$. NeRF renders a pixel by applying volume rendering along a ray: $C(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_\mathrm{i}\delta_\mathrm{i}))\mathbf{c}_\mathrm{i}$, where δ_i refers to the distance between two adjacent samples and T_i is the accumulated transmittance. With the ground truth colors $C_{\mathrm{gt}}(\mathbf{r})$, MLP is trained with a reconstruction loss:

$$\mathcal{L}_{\mathrm{recon}} = \frac{1}{|\mathbf{r}|}\sum_{\mathbf{r}} \|C(\mathbf{r}) - C_{\mathrm{gt}}(\mathbf{r})\|^2. \tag{1}$$

Semantic NeRF. Zhi *et al.* (Zhi et al. 2021) extends the above framework by encoding semantics into the NeRF. They use the MLP to predict additional semantic logits $\mathbf{g}$, as illustrated in Fig. 4 (a). Volume rendering is then applied on the logits of each point along the ray: $G(\mathbf{r}) =$

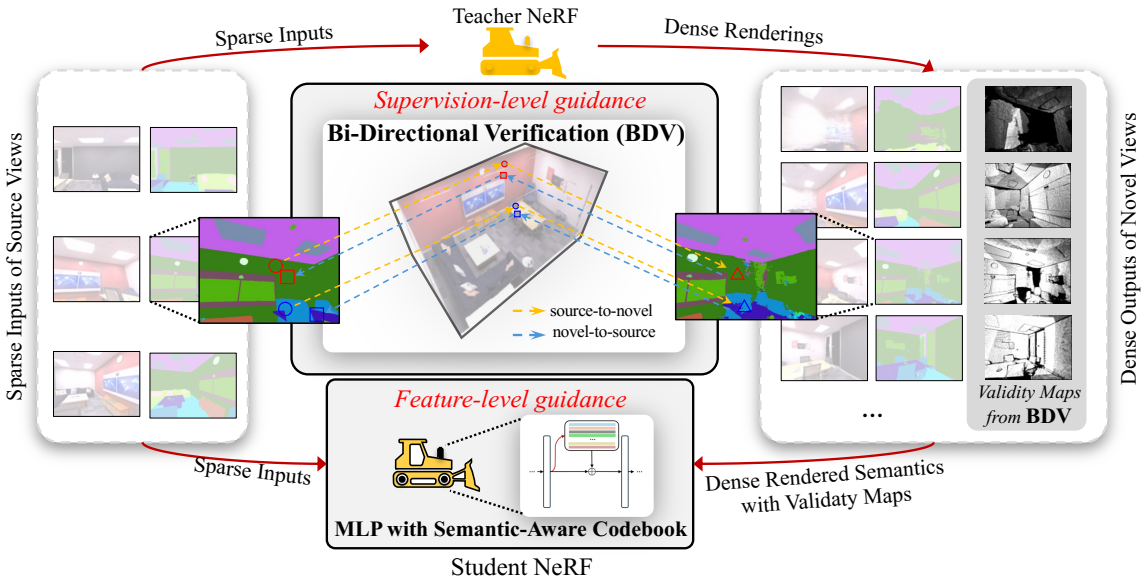

Figure 2: Overview of the proposed self-improved S^{3}NeRF, which exploits rendered semantics from the teacher NeRF by two levels of guidance: supervision-level guidance with Bi-Directional Verification (BDV), and feature-level guidance with semantic-aware codebook. BDV returns validity maps for each semantic map, indicating the correctness of semantic labels for robust supervision. The semantic-aware codebook encodes correlation among densities, colors, and semantics, further exploiting the underlying information embedded in the semantic labels. The codebook is integrated into the MLP of the student NeRF.

$\sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{g}_i$. The MLP is then trained with the $\mathcal{L}_{\text{recon}}$ and the semantic loss $\mathcal{L}_{sem}$:

$$\mathcal{L}_{\text{sem}} = \frac{1}{|\mathbf{r}|}\sum_{\mathbf{r}} \text{CE}(\text{softmax}(G(\mathbf{r})), S_{\text{gt}}(\mathbf{r})), \qquad (2)$$

where $\text{softmax}(\cdot)$ converts logits into a distribution, $\text{CE}(\cdot)$ and $S_{\text{gt}}(\mathbf{r})$ refers to the cross-entropy loss and the ground truth semantic label. This extends the radiance field into a semantic field. In this paper, we use semantics to guide the modeling of geometry and color for sparse-input NeRF.

Self-Improving with Rendered Semantics

The self-improved framework applied in this work involves training a teacher and a student NeRF, respectively. The outputs of novel views rendered from the teacher NeRF are treated as the augmented data to train the student NeRF.

Given N sparse inputs, i.e., $\{\mathbf{I}_i, \Phi_i, \mathbf{S}_i\}_{i=1}^{N}$, where three values refer to RGB values, the camera pose, and semantic labels of each image, we firstly train the teacher NeRF with the reconstruction loss (Eq. (1)) and the semantic loss (Eq. (2)). We observe that limited semantics of the sparse inputs degrade the performance of novel view synthesis. Thus, we detach the semantic branch of Fig. 4 (a) from the main branch. After training, the teacher NeRF can render images as well as their semantic maps from novel views. The rendered outputs are denoted by $\{\hat{\mathbf{I}}_j, \hat{\Phi}_j, \hat{\mathbf{S}}_j, \hat{\mathbf{D}}_j\}_{j=1}^{T}$, where T denotes the number of novel views, which is typically larger than N, and $\{\hat{\mathbf{D}}_j\}_{j=1}^{T}$ refers to the rendered depth maps.

Previous works (Bai et al. 2023; Jung et al. 2023) exploit rendered RGB values $\{\hat{\mathbf{I}}_j\}_{j=1}^{T}$ as the augmented data to train the student NeRF. However, the rendered RGB values are unreliable due to the existence of color shiftings and blurry regions in the rendered images. Though previous works design strategies for filtering, it is essentially difficult to detect them, whether in feature space or by value differences.

We propose using rendered semantics $\{\hat{\mathbf{S}}_j\}_{j=1}^{T}$ as the augmented data. The rendered semantic labels exhibit greater robustness, as the trained teacher NeRF can render accurate semantic labels for most regions, even for those blurry ones. To utilize the rendered semantics, a straightforward way is to supervise the rays of novel views with their rendered semantic labels by the semantic loss $\mathcal{L}_{\text{sem}}$. Although the rendered semantic labels show higher robustness, there still exists misclassified labels that might hamper the training. We thus introduce a weighting factor to adjust the impact of the incorrect labels. The training objective of the student NeRF is then formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \frac{\lambda_{\text{sem}}}{|\mathbf{r}|+|\hat{\mathbf{r}}|}\cdot\left(\sum_{\mathbf{r}\in\mathcal{R}}\text{CE}(\text{softmax}(G(\mathbf{r})), S_{\text{gt}}(\mathbf{r})) + \sum_{\hat{\mathbf{r}}\in\hat{\mathcal{R}}} w(\hat{\mathbf{r}})\cdot\text{CE}(\text{softmax}(G(\hat{\mathbf{r}})), \hat{S}_{\text{gt}}(\hat{\mathbf{r}}))\right), \qquad (3)$$

where the semantic guidance is balanced by λ_{sem}. $\mathcal{R}$ and $\hat{\mathcal{R}}$ denote the sets of rays from sparse input and novel views, respectively. $\hat{\mathbf{r}}$ refers to sampled rays from rendered novel views. $w(\hat{\mathbf{r}})$ and $\hat{S}_{\text{gt}}(\hat{\mathbf{r}})$ represent the weights and pseudo semantic labels for the rays. In the following, we elaborate on the detailed strategy of deciding $w(\hat{\mathbf{r}})$ for the rays.

Supervision-level Guidance

In this section, we propose a Bi-Directional Verification (BDV) module to determine the validity of each rendered

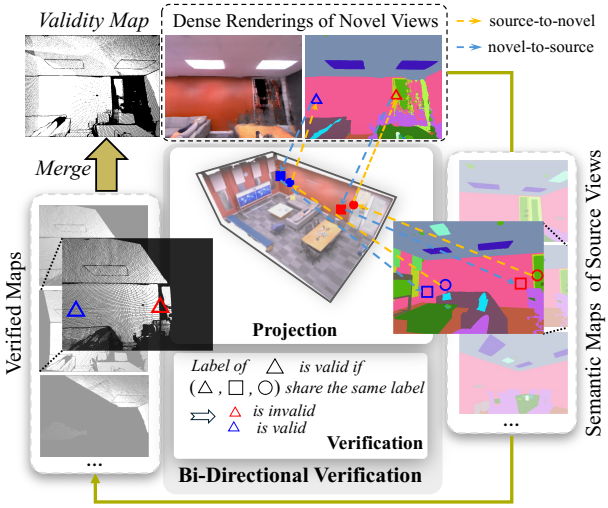

Figure 3: Illustration of the Bi-Directional Verification (BDV) module. For each rendered semantic map, BDV is applied between each source view and it by projection and verification. A validity map is created by merging verified maps from all source views.

semantic label. Thus, the weighting factor $w(\hat{\mathbf{r}})$ in Eq. (3) can be either 1 or 0, indicating whether the label is valid or invalid. The BDV module guides the semantic loss using a 0-1 weighting scheme, and thus it is called a supervision-level guidance. Subsequently, the term "source views" is utilized to refer to the sparse inputs.

As shown in Fig. 2, BDV is applied between each source view and each rendered novel view. Fig. 3 presents more details about the BDV module. Concretely, the inputs into the BDV module consist of $\{\Phi, \mathbf{S}, \mathbf{D}\}$ and $\{\hat{\Phi}, \hat{\mathbf{S}}, \hat{\mathbf{D}}\}$, where each of the symbols in the triplet refers to the camera pose, the semantic map and the depth map, respectively. Note that $\mathbf{D}$ is also rendered by the teacher NeRF. For clarity in the following statements, we drop the subscripts to consider one source view and one novel view.

The motivation behind BDV is straightforward: if a rendered semantic label is accurate, we should be able to identify the same semantic label in corresponding regions of at least one specific source view. BDV consists of two key steps: projection and verification.

Projection. Given the rendered depth maps $\mathbf{D}$ and $\hat{\mathbf{D}}$, we employ projection to establish correspondences between regions of the source view and the novel view. Our objective is to determine the validity of each semantic label in the novel view, by identifying its corresponding region in the source view, and verifying whether it matches the semantic label assigned to that source view region. While the depth maps may not provide accurate pixel-to-pixel correspondences, we have observed that they are generally precise enough to establish pixel-to-region correspondences in most cases, as depicted in Fig. 3. Our proposed BDV employs two directions of projection: source-to-novel and novel-to-source projections, based on which the verification step is applied.

Source-to-novel projection: Given the camera poses Φ and $\hat{\Phi}$, along with the rendered depth maps $\mathbf{D}$, for the coordinates of each pixel $\mathbf{p}$ in the source view, we can project it onto the novel view, and obtain the projected coordinates $\hat{\mathbf{p}}_{\text{src}\to\text{nov}}$ using the following formula:

$$\hat{\mathbf{p}}_{\text{src}\to\text{nov}} \sim K T_{\Phi\to\hat{\Phi}} \mathbf{D}(\mathbf{p}) K^{-1} \mathbf{p}, \tag{4}$$

where K refers to the camera intrinsic matrix, and $T_{\Phi\to\hat{\Phi}}$ represents the pose transition from the source view pose to the novel view pose. $\mathbf{p}$ is input to Eq. (4) in the form of homogeneous coordinates but we omit it for simplicity.

Intuitively, if $\hat{\mathbf{p}}_{\text{src}\to\text{nov}}$ in the novel view is located at the same region with $\mathbf{p}$ in the source view, the correctness of the semantic label can be determined by checking if $\hat{\mathbf{S}}(\hat{\mathbf{p}}_{\text{src}\to\text{nov}})$ is identical to $\mathbf{S}(\mathbf{p})$. However, the depth map $\mathbf{D}$ of the source view may contain large errors in some regions, which will result in wrong projections. The semantic label may still be incorrect even if it passes the above checking. Therefore, we also include the rendered depth of novel views to comprehensively determine the correctness of the semantic labels, by utilizing the inverse projection, i.e., *novel-to-source projection*, to get $\mathbf{p}_{\text{nov}\to\text{src}}$.

Verification. With the bi-level projection, we have a projection chain that composes of a triplet, i.e., $(\mathbf{p}, \hat{\mathbf{p}}_{\text{src}\to\text{nov}}, \mathbf{p}_{\text{nov}\to\text{src}})$, which is respectively represented by the circle, triangle, and square in Fig. 3. To obtain the verified map $\hat{\mathbf{M}}$ which verifies the correctness of the semantic labels of the novel view $\hat{\mathbf{S}}$, we apply the simplest verification step based on consensus:

$$\hat{\mathbf{M}} = \begin{cases} 1 & \text{if } \mathbf{S}(\mathbf{p}) = \hat{\mathbf{S}}(\hat{\mathbf{p}}_{\text{src}\to\text{nov}}) = \mathbf{S}(\mathbf{p}_{\text{nov}\to\text{src}}) \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

which means that the semantic label in the novel view is considered valid only if its assigned triplet shares the same label. Note that we only consider the coordinates on the novel view projected from the source view. A simple illustration of the verified map can be found in Fig. 2.

Merge Verified Maps into a Validity Map: The verified map of Eq. (5) is obtained between one source view and one novel view. However, the correctness of the semantic labels in the novel view should be verified after checking with all N source views. Thus, for each semantic map of the novel view, we repeat the above projection and verification steps with other source views, and obtain totally N verified maps $\{\hat{\mathbf{M}}_i\}_{i=1}^{N}$. We can then obtain the validity map $\hat{\mathbf{V}}$ for the novel view by: $\hat{\mathbf{V}} = \bigcup_{i=1}^{N} \hat{\mathbf{M}}_i$, where $\bigcup$ denotes the "element-wise or" operation. The reason for us to use the "or" is that, once the correctness of a semantic label is verified by a certain source view, and then the label is reliable. As shown in Fig. 2 and Fig. 6, the validity maps can reflect the correctness of semantic labels. Regarding Eq. (3), we set $w(\hat{\mathbf{r}})$ to 0 or 1 according to its value in the validity map.

Feature-level Guidance

Eq. (3) with the proposed supervision-level guidance can alleviate the ambiguity problem in the sparse-input setting by using the rendered semantic labels to build up correspondences across views. As mentioned, the rendered RGB can-

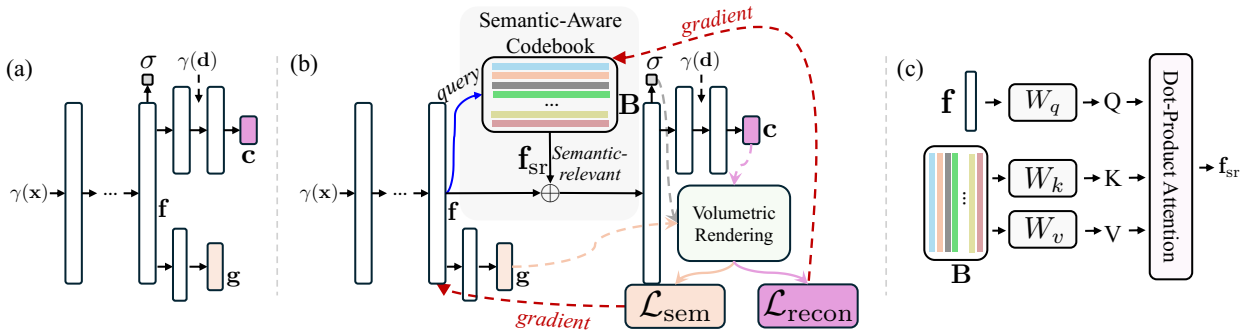

Figure 4: **(a)** MLP of the semantic NeRF (Zhi et al. 2021), where $\mathbf{c}$, σ and $\mathbf{g}$ refer to color, density, and semantic logits of each input 3D point. $\mathbf{f}$ is the implicit feature used to predict the density. **(b)** Our proposed MLP for feature-level guidance incorporates a semantic-aware codebook to encode the correlation among densities, colors, and semantics. Each 3D point queries the codebook via an attention in **(c)**. The codebook $\mathbf{B}$ is updated by the gradient from the reconstruction loss, while the semantic field is learned from the semantic loss.

not be considered as reliable augmented data and we do not use them for supervision. This makes the trained student NeRF a decent semantic field, exhibiting improved rendering of multi-view consistent semantics, but with limited enhancement in view synthesis (i.e., appearance) due to the absence of RGB supervision for augmented rays.

We observe that the semantics are correlated to the color, e.g., walls in a scene only exhibit limited colors. Moreover, the encoded semantics of 3D points in the semantic field are even related to their densities. For instance, for a point encoded with certain semantics, it density can be high. Therefore, the rendered semantics not only provide semantic labels for the student NeRF, enabling it to function as a reliable semantic field, but also convey the inner semantic-relevant information, such as colors, which is not employed by the supervision-level guidance. By leveraging the correlation between semantics and colors, we can model a more accurate radiance distribution of the scene with the guidance from the correlation. Moreover, it can compensate for novel views that lack RGB supervision to achieve better color predictions, thus improving the performance of view synthesis.

The rendered semantic labels from dense novel views can train the MLP to represent a decent semantic field, using the vanilla structure as depicted in Fig. 4 (a). Thus, the implicit feature $\mathbf{f}$ is trained to be well-encoded with semantics. To further exploit the rendered semantic guidance from the teacher NeRF, we propose a feature-level guidance that guides the learning of the implicit feature not only for well-encoded semantics, but also for better predictions of colors and densities. To implement this, we incorporate a learnable codebook within the MLP, as illustrated in Fig. 4 (b). The codebook learns semantic-aware patterns, and captures the correlation among semantics, colors, and densities. Consequently, the predictions of the density and color are based on the semantic-relevant information that is queried by the semantics encoded in the feature $\mathbf{f}$, leading to more accurate predictions. We denote the codebook as: $\mathbf{B} \in \mathbb{R}^{K \times d}$, which contains K learnable embeddings of dimension d.

For each 3D point that inputs to the MLP, after obtaining the implicit feature $\mathbf{f}$ through the feed-forwarding, we extract the semantic-relevant information from the codebook with a query operation: $\mathbf{f}_{\text{sr}} = \text{Query}(\mathbf{f}, \mathbf{B})$, where $\mathbf{f}_{\text{sr}}$ denotes the queried feature. The query operation is implemented with an attention process as shown in Fig. 4 (c). While $\mathbf{f}_{\text{sr}}$ contains semantic-relevant information, using it directly for predictions may not yield satisfactory outcomes, due to the inherent heterogeneity within each semantic class. For instance, the walls may show different colors in a scene. To address this issue, as depicted in Fig. 4 (b), we employ element-wise addition between $\mathbf{f}$ and $\mathbf{f}_{\text{sr}}$, combing the position-specific with the semantic-relevant information to obtain a more comprehensive representation, which is used for color and density predictions. The codebook can be learned from the gradient of the reconstruction loss (Eq. (1)), as illustrated in Fig. 4 (b). Fig. 5 qualitatively demonstrates that, with the feature-level guidance, the student NeRF can render more accurate colors.

Experiments

Experimental Settings

Datasets. Our experiments are conducted on two indoor datasets: Replica (Straub et al. 2019) and ScanNet++ (Yeshwanth et al. 2023), using sparse inputs of 6 images. We also evaluate our method on LLFF (Mildenhall et al. 2019) and DTU (Jensen et al. 2014) following the standard 3-view setting (Niemeyer et al. 2022).

Baseline. In our experiments, we adopt Mip-NeRF (Barron et al. 2022) with the monocular depth regularization (Yu et al. 2022) as the baseline, and we validate the effectiveness of our approach on it. Accordingly, we treat the baseline method as the teacher NeRF within our self-improved framework, and it remains fixed throughout the experiments. For more details about the training of the baseline, as well as specific details regarding implementation and hyperparameter settings, we refer readers to the supplementary material.

Performances. In the tables below, the reported results regarding our method pertain to the performance of the student NeRF, except the entry labeled "baseline", which is the performance of the teacher NeRF.

(a)	PSNR↑	SSIM↑	LPIPS↓
baseline	21.17	0.779	0.395
+supervision-level	**21.69**	**0.780**	**0.387**
w/o rendered labels	20.49	0.747	0.416
w/o verification	21.35	0.774	0.409

(b)	PSNR↑	SSIM↑	LPIPS↓
baseline+supervision-level	21.69	0.780	0.387
+feature-level (S^{3}NeRF)	**22.21**	**0.787**	**0.364**
w/o $\mathcal{L}_{sem}$	21.69	0.783	0.376
w pre-trained codebook	20.28	0.764	0.404

(c)	PSNR↑	SSIM↑	LPIPS↓
baseline-12view	23.30	0.814	0.295
Ours-12view	**24.00**	**0.818**	**0.277**
baseline-18view	24.96	0.827	0.282
Ours-18view	**25.80**	**0.837**	**0.253**

Table 1: Ablation studies on the **(a) supervision-level** and **(b) feature-level** guidance on the ScanNet++ dataset. **(c)** Effectiveness of our method with more input views on the Replica dataset.

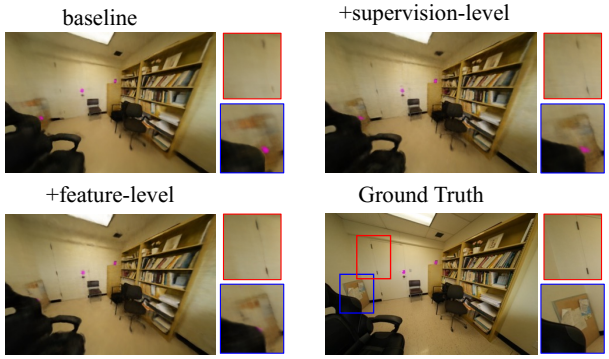

Figure 5: Efficacy of the supervision-level and feature-level guidance. The supervision-level guidance improves the quality in object boundary regions. The feature-level guidance boosts the quality by rendering more accurate colors.

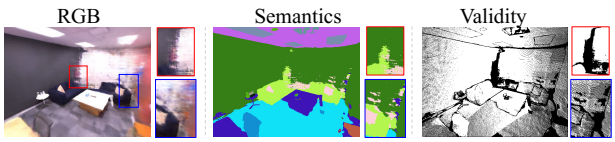

Figure 6: Validity maps from the BDV module. The black regions are recognized as invalid and their enclosed semantic labels are not used in the semantic loss for the student NeRF.

Method	ScanNet++			Replica		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF	19.58	0.755	0.389	18.12	0.707	0.391
InfoNeRF	14.54	0.646	0.495	13.07	0.598	0.552
DietNeRF	19.76	0.719	0.431	18.99	0.676	0.444
FreeNeRF	20.17	0.756	0.368	20.99	0.765	0.324
Mip-NeRF*	21.17	0.779	0.395	21.37	0.785	0.318
DNGaussian	19.01	0.754	0.367	17.63	0.718	0.435
FSGS	17.95	0.730	0.373	20.22	0.760	0.304
S^{3}NeRF (Ours)	22.21	0.787	0.364	22.54	0.800	0.287

Table 2: Comparisons with other methods on the ScanNet++ and Replica datasets. The best, second-best, and third-best entries are marked in red, orange, and yellow.

Method	LLFF			DTU		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF ft	16.17	0.438	0.512	18.95	0.710	0.269
MVSNeRF ft	17.88	0.584	0.327	18.54	0.769	0.197
RegNeRF	19.08	0.587	0.336	18.89	0.745	0.190
GeCoNeRF	18.77	0.596	0.338	-	-	-
FreeNeRF	19.63	0.612	0.308	19.92	0.787	0.182
SparseNeRF	19.86	0.624	0.328	19.55	0.769	0.201
DNGaussian	19.12	0.591	0.294	18.91	0.790	0.176
SE-NeRF	18.10	0.540	0.450	-	-	-
S^{3}NeRF (ours)	19.86	0.589	0.351	21.09	0.835	0.147

Table 3: Comparisons with other methods on the LLFF and DTU datasets. The best, second-best, and third-best entries are marked in red, orange, and yellow.

Ablation Studies

Effect of the supervision-level guidance. The supervision-level guidance assists the training of the student NeRF with the rendered semantic labels that are verified by the BDV module. As presented in Tab. 1 (a), the supervision-level guidance enhances the performance by more than 0.5 PSNR. Fig. 5 demonstrates that the guidance notably increases the quality of the blurry regions around object boundaries. Training with rendered semantic labels without the BDV module may negatively impact the performance, as indicated by the SSIM and LPIPS drops of "w/o verification" in Tab. 1, due to the influence of incorrect labels. To show that the performance gain is attributed to the semantic labels from the rendered novel views, rather than the ones from the source views, we conduct an experiment where NeRF is trained solely with semantic labels from the source views, denoted as "w/o rendered labels", whose performance drops to 20.49, indicating that the semantic labels can assist the learning of NeRF only if they are dense enough to build up the correspondences across views.

Effect of the feature-level guidance. As indicated in Tab. 1 (b), the proposed feature-level guidance brings a PSNR improvement of over 0.5 compared to the model with supervision-level guidance. Fig. 5 shows that, the feature-level guidance improves the synthesis quality, particularly in terms of more accurate color predictions, e.g., the colors of the wall beside the board and the hinges. This is attributed to the correlation between semantics and colors that are encoded in the codebook. We also assess the codebook's contribution by excluding the semantic loss, labeled as "w/o $\mathcal{L}_{sem}$" which shows a performance gain similar to the supervision-level guidance. Combining both losses yields the best result. Previous work (Yin et al. 2022) also utilizes a codebook that is pre-trained from ImageNet. Compared to our method of Fig. 4 (b), they extract the visual clues from the codebook at the MLP input, which cannot utilize the feature with encoded semantics. Additionally, the codebook is pre-trained and is thus semantic-agnostic. Our method outperforms theirs ("w pre-trained codebook") by a large margin, validating the effectiveness of the proposed learnable semantic-aware codebook.

Performance with more input views. We also examine the

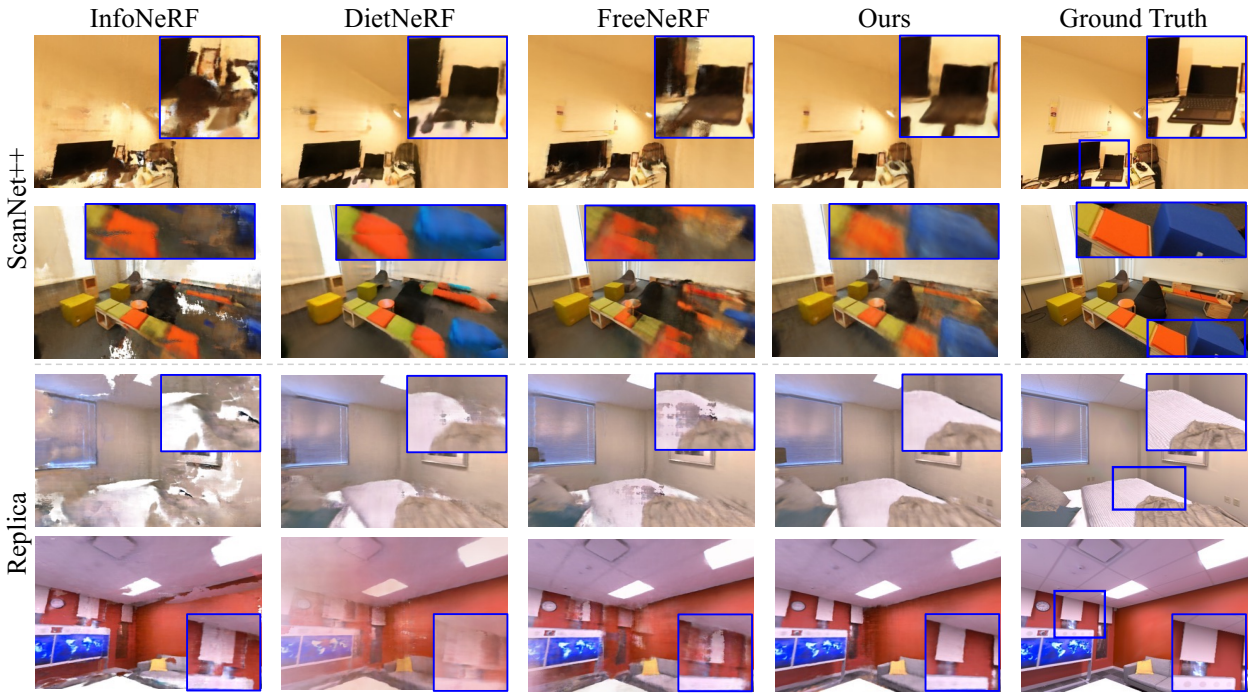

Figure 7: Qualitative comparisons on ScanNet++ and Replica datasets. Our method preserves the global structure more effectively, avoiding distortion and severe floaters, while also capturing finer local details such as edges.

effectiveness of our method as the number of input views increases on the Replica dataset. Table 1 (c) shows that our approach consistently improves the performance with more views. Specifically, our method brings PSNR improvement of over 0.5 for input views of both 12 and 18. This indicates that our method does not suffer over-regularization issues.

Comparison with Current Works

Replica and ScanNet++. In Tab. 2, we compare our S^{3}NeRF with several approaches. Mip-NeRF* applies the monocular depth regularization from (Yu et al. 2022). The result demonstrates that S^{3}NeRF achieves the highest performance, outperforming FreeNeRF by more than 1.5 PSNR on both datasets. Fig. 7 illustrates that S^{3}NeRF shows better global structures and finer details compared with other works. For example, in the first row of examples in Fig. 7, our S^{3}NeRF not only keeps more accurate details of the mouse, but also exhibits significantly fewer artifacts. Our method also shows superior performance over the recent works of sparse-input 3D Gaussian Splatting (3DGS) (Li et al. 2024; Zhu et al. 2025). Note that 3DGS-based methods are optimized with random point cloud initialization, due to failures of applying COLMAP on these benchmarks.

LLFF and DTU. We evaluate our method on standard 3-view benchmarks (Niemeyer et al. 2022) in Tab. 3, deriving semantic labels by combining SAM (Kirillov et al. 2023) and DINO (Oquab et al. 2023) (detailed in the supplement). While most scenes show limited semantic differences, our method achieves competitive performance. Fig. 8 qualitatively shows that our method can keep better global structure. Notably, our method outperforms another self-improved method SE-NeRF (Jung et al. 2023) by over 1.0 PSNR on LLFF, which utilizes rendered RGB from novel views for supervision, validating the superiority of the rendered semantics.

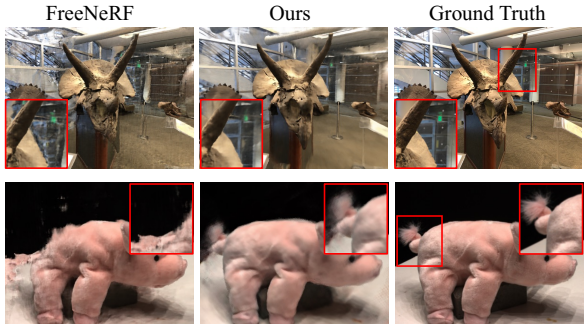

Figure 8: Qualitative comparisons on LLFF and DTU datasets. Our method preserves better structural integrity.

Conclusion

This paper introduces an observation that rendered semantics from dense novel views is a more effective form of augmented data than rendered RGB, based on which we propose a self-improved S^{3}NeRF for sparse-input NeRF. S^{3}NeRF trains a student NeRF with the semantic guidance of dense novel views rendered from a teacher NeRF. The student NeRF is guided by the rendered semantics from supervision and feature levels. We also introduce an indoor benchmark with as few as 6 input images. Experiments validate our approach and it achieves competitive performance.

Acknowledgements

The work is supported in part by the Early Career Scheme of the Research Grants Council of the Hong Kong SAR under grant No. 26202321, ITF PRP/046/24FX, SAIL Research Project, and HKUST-Zeekr Collaborative Research Fund.

References

Bai, J.; Huang, L.; Gong, W.; Guo, J.; and Guo, Y. 2023. Self-NeRF: A Self-Training Pipeline for Few-Shot Neural Radiance Fields. *arXiv preprint arXiv:2303.05775*.

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2023. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *ICCV*.

Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; and Su, H. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.

Deng, K.; Liu, A.; Zhu, J.-Y.; and Ramanan, D. 2022. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*.

Jain, A.; Tancik, M.; and Abbeel, P. 2021. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*.

Jensen, R.; Dahl, A.; Vogiatzis, G.; Tola, E.; and Aanæs, H. 2014. Large scale multi-view stereopsis evaluation. In *CVPR*.

Jung, J.; Han, J.; Kang, J.; Kim, S.; Kwak, M.-S.; and Kim, S. 2023. Self-Evolving Neural Radiance Fields. *arXiv preprint arXiv:2312.01003*.

Kim, M.; Seo, S.; and Han, B. 2022. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *ICCV*.

Kundu, A.; Genova, K.; Yin, X.; Fathi, A.; Pantofaru, C.; Guibas, L. J.; Tagliasacchi, A.; Dellaert, F.; and Funkhouser, T. 2022. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR*.

Kwak, M.-S.; Song, J.; and Kim, S. 2023. Geconerf: Few-shot neural radiance fields via geometric consistency. *arXiv preprint arXiv:2301.10941*.

Li, J.; Zhang, J.; Bai, X.; Zheng, J.; Ning, X.; Zhou, J.; and Gu, L. 2024. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *CVPR*.

Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. In *NeurIPS*.

Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM TOG*, 38(4): 1–14.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4): 1–15.

Niemeyer, M.; Barron, J. T.; Mildenhall, B.; Sajjadi, M. S.; Geiger, A.; and Radwan, N. 2022. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Roessle, B.; Barron, J. T.; Mildenhall, B.; Srinivasan, P. P.; and Nießner, M. 2022. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*.

Siddiqui, Y.; Porzi, L.; Bulò, S. R.; Müller, N.; Nießner, M.; Dai, A.; and Kontschieder, P. 2023. Panoptic lifting for 3d scene understanding with neural fields. In *CVPR*.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

Truong, P.; Rakotosaona, M.-J.; Manhardt, F.; and Tombari, F. 2023. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*.

Wang, G.; Chen, Z.; Loy, C. C.; and Liu, Z. 2023. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *ICCV*.

Wang, Q.; Wang, Z.; Genova, K.; Srinivasan, P. P.; Zhou, H.; Barron, J. T.; Martin-Brualla, R.; Snavely, N.; and Funkhouser, T. 2021. Ibrnet: Learning multi-view image-based rendering. In *CVPR*.

Wynn, J.; and Turmukhambetov, D. 2023. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *CVPR*.

Yang, J.; Pavone, M.; and Wang, Y. 2023. FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization. In *CVPR*.

Yeshwanth, C.; Liu, Y.-C.; Nießner, M.; and Dai, A. 2023. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*.

Yin, F.; Liu, W.; Huang, Z.; Cheng, P.; Chen, T.; and Yu, G. 2022. Coordinates Are NOT Lonely-Codebook Prior Helps Implicit Neural 3D Representations. In *NeurIPS*.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021. pixelnerf: Neural radiance fields from one or few images. In *CVPR*.

Yu, Z.; Peng, S.; Niemeyer, M.; Sattler, T.; and Geiger, A. 2022. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.

Zhang, M.; Zheng, S.; Bao, Z.; Hebert, M.; and Wang, Y.-X. 2023. Beyond RGB: Scene-Property Synthesis with Neural Radiance Fields. In *WACV*.

Zhi, S.; Laidlow, T.; Leutenegger, S.; and Davison, A. J. 2021. In-place scene labelling and understanding with implicit scene representation. In *ICCV*.

Zhong, Y.; Hong, L.; Li, Z.; and Xu, D. 2024. CVT-xRF: Contrastive In-Voxel Transformer for 3D Consistent Radiance Fields from Sparse Inputs. In *CVPR*.

Zhu, Z.; Fan, Z.; Jiang, Y.; and Wang, Z. 2025. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *ECCV*.