CVT-*x*RF: Contrastive In-Voxel Transformer for 3D Consistent Radiance Fields from Sparse Inputs

Yingji Zhong[1] Lanqing Hong[2] Zhenguo Li[2] Dan Xu[1]
[1]The Hong Kong University of Science and Technology [2]Huawei Noah's Ark Lab
{yzhongbn,danxu}@cse.ust.hk, {honglanqing,li.zhenguo}@huawei.com

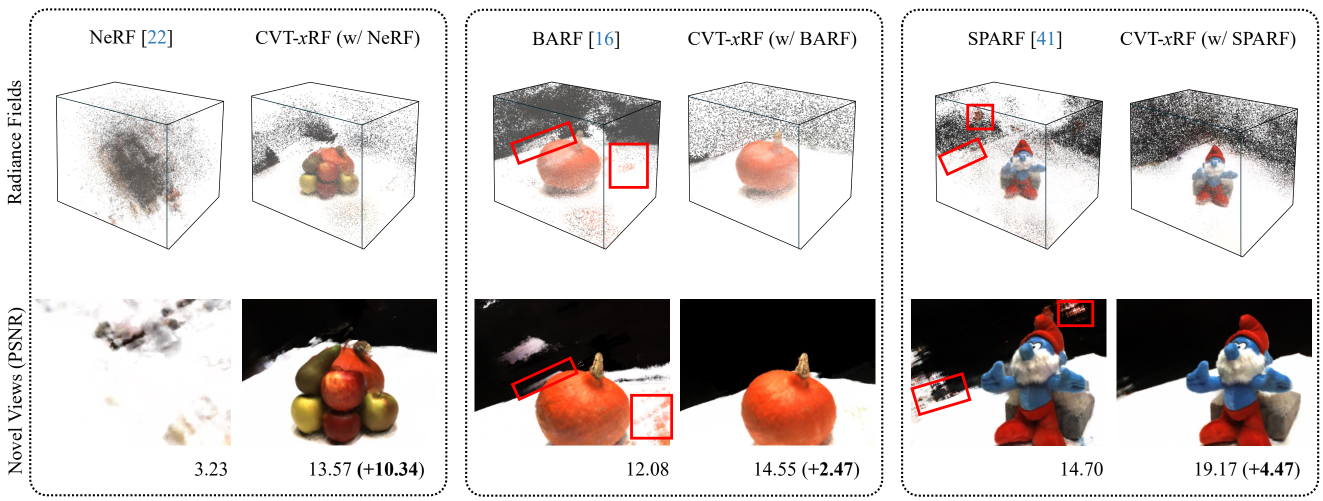

Figure 1. Qualitative illustration of learned 3D radiance fields and rendered images of the proposed CVT (Contrastive In-Voxel Transformer)-*x*RF upon three baselines trained from sparse inputs of three views. Our CVT-*x*RF can significantly improve all the baselines. The radiance fields show different levels of 3D inconsistencies (marked in red boxes for BARF and SPARF), which result in failures or artifacts in rendered images. With CVT, we can obtain radiance fields of better 3D consistency and render images of much higher quality.

Abstract

Neural Radiance Fields (NeRF) have shown impressive capabilities for photorealistic novel view synthesis when trained on dense inputs. However, when trained on sparse inputs, NeRF typically encounters issues of incorrect density or color predictions, mainly due to insufficient coverage of the scene causing partial and sparse supervision, thus leading to significant performance degradation. While existing works mainly consider ray-level consistency to construct 2D learning regularization based on rendered color, depth, or semantics on image planes, in this paper we propose a novel approach that models 3D spatial field consistency to improve NeRF's performance with sparse inputs. Specifically, we first adopt a voxel-based ray sampling strategy to ensure that the sampled rays intersect with a certain voxel in 3D space. We then randomly sample additional points within the voxel and apply a Transformer to infer the properties of other points on each ray, which are then incorporated into the volume rendering. By backpropagating through the rendering loss, we enhance the consistency among neighboring points. Additionally, we propose to use a contrastive loss on the encoder output of the Transformer to further improve consistency within each voxel. Experiments demonstrate that our method yields significant improvement over different radiance fields in the sparse inputs setting, and achieves comparable performance with current works. The project page for this paper is available at https://zhongyingji.github.io/CVT-xRF.

1. Introduction

Representing and modeling 3D properties of scenes is crucial for a wide range of real-world applications, such as autonomous driving, robotic navigation, and 3D content generation. In recent years, implicit neural scene representations [21, 22, 25, 27, 37] have shown impressive abilities to model 3D geometry and appearance in a continuous manner. Among these approaches, Neural Radiance Fields (NeRF) [22] have emerged as a powerful representation for complex scenes. When the NeRF model is optimized with multi-view inputs, high-fidelity images can be synthesized from unseen novel views [1, 2, 18, 23, 43].

Despite the significant progress achieved, NeRF has a notable limitation in that it typically requires dense inputs for training its Multi-Layer Perception (MLP). While if only sparse training inputs are provided, because of missing view supervision, NeRF tends to learn a degenerate scene representation that fails to accurately model the physical properties (*i.e.*, radiance distributions) of the entire scene, thus resulting in large radiance ambiguities [54], as can be observed from Fig. 1. To address this issue, several works have attempted to regularize NeRF during training with different constraints or priors, including sparsity [13] and 2D spatial consistency [26], additional depth supervision [5, 28, 32], and semantic alignment [10] or matching [41] utilizing off-the-shelf pre-trained models. These existing works have achieved important improvements to this problem. However, they primarily focus on ray-level consistency based on the rendered color and depth, or semantics on 2D image planes, while 3D spatial field consistency is not explicitly modeled. The 3D spatial field consistency reflects a natural phenomenon that the radiance field is spatially consistent, *i.e.*, 3D points physically close or semantically related tend to exhibit similar radiance properties. In these existing works, this crucial 3D field consistency can only be *indirectly* regularized through the gradients from the 2D-level regularization onto sampled ray points, making it challenging to effectively model the correlation of radiances among 3D points. As also shown in Fig. 1, the learned radiance fields from sparse inputs of three baselines, *i.e.*, NeRF [22], BARF [16], and SPARF [41], exhibit different levels of inconsistency in 3D space, resulting in failures or artifacts in rendered 2D images.

To explicitly model and learn the aforementioned crucial 3D spatial field consistency, in this paper, we propose a Contrastive In-Voxel Transformer (CVT) structure to implement the 3D field consistency in the sparse inputs setting. As illustrated in Fig. 1, CVT can be flexibly integrated into various baselines, largely boosting the consistency in both 3D radiance fields and 2D rendered images. We denote our method as CVT-xRF, where x indicates that our CVT structure can be plugged into different baseline radiance fields for sparse-view scene modeling. Our proposed CVT-xRF comprises three main components that work seamlessly to achieve this goal. More specifically, (i) the first component is a voxel-based ray sampling strategy. In detail, during training, we first select multiple voxels in 3D space. For each selected voxel, we sample rays that intersect with it, which ensures that the 3D points on the rays within the voxel share similar radiance properties. (ii) The second component of CVT-xRF is a local implicit constraint that is based on an In-Voxel Transformer [42]. Specifically, for each ray, two distinct sets of 3D points are sampled within the same voxel: one set of points is randomly sampled in the 3D voxel, while the other set of points is sampled along the ray. Since both sets of points are within the same voxel, their radiance properties can be closely correlated. We thus leverage the Transformer to implicitly model the correlation of 3D-point radiances. The Transformer's encoder and decoder take the two sets of points as inputs, respectively. The encoder learns representations of neighboring 3D points; the decoder learns the correlation between neighboring points and ray points, and outputs radiances of the ray points for volume rendering of the ray. (iii) The third component of CVT-xRF is a global explicit constraint in the form of a voxel contrastive regularization. During training, multiple voxels in the 3D scene are sampled, and the contrastive regularization is designed to learn field consistency among positive 3D points (within voxels) and negative 3D points (across voxels). CVT-xRF brings significant improvements over different baselines and achieves state-of-the-art performances on multiple challenging benchmarks. In summary, our main contributions are as follows:

- We introduce a novel 3D spatial field consistency mechanism for effectively regularizing the learning of radiance fields from sparse inputs.
- We propose a Contrastive In-Voxel Transformer (CVT) structure to implement 3D field consistency learning, which is constructed with three key components, *i.e.*, voxel-based ray sampling, local implicit constraint, and global explicit constraint. The CVT structure can be flexibly applied to different baselines.
- Our experiments extensively demonstrate that our method brings significant gains over different strong baselines, *e.g.*, on DTU 3-view, our CVT-xRF brings 7.45, 0.95, 1.20 PSNR improvements upon NeRF [22], BARF [16] and SPARF [41], respectively.

2. Related Work

Neural scene representation. Compared to discretized representations [6, 29, 30, 36, 39, 50], neural scene representation [21, 27] excels in modeling the continuous shape and appearance. With differentiable rendering [22, 25, 37], the model can be trained on posed images. Among them, Neural Radiance Fields (NeRF) [22] have gained increasing attention in recent years. It achieves impressive results on novel view synthesis with complex scenes [1, 18, 23, 43]. Besides, NeRF has also shown impressive results on other applications [15, 20, 24, 34, 45, 53].

Novel view synthesis from sparse inputs. One major drawback of NeRF is that it might learn degenerate representations when given sparse inputs [26, 54]. To address this problem, two lines of research have emerged.

The first line of research aims to learn a generalizable radiance field by pre-training the MLP on multi-view datasets and then fine-tuning it with sparse inputs from a target scene. For example, PixelNeRF [52] and IBRNet [46] aug-

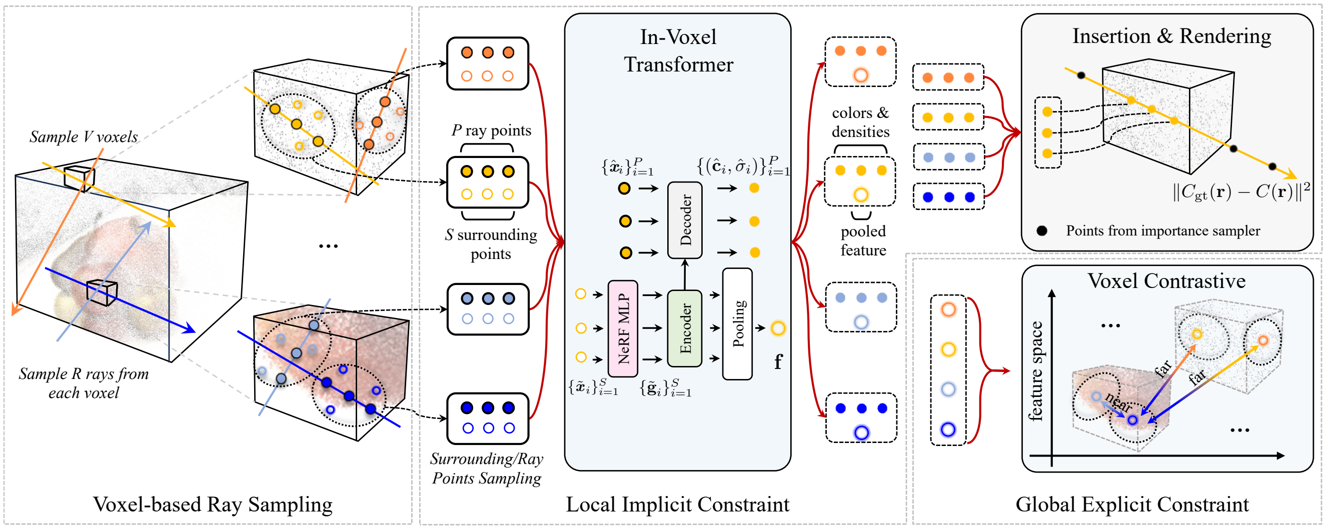

Figure 2. Illustration of the proposed CVT (Contrastive In-Voxel Transformer)-*x*RF for learning radiance fields from sparse inputs. It consists of three parts, *i.e.*, a voxel-based ray sampling strategy, a local implicit constraint module, and a global explicit constraint module. For simplicity, two voxels are shown, along with two rays for each. The local implicit constraint is implemented by a light-weight In-Voxel Transformer which infers colors and densities of ray points by interacting with surrounding 3D points. The ray points are then inserted among the points from the importance sampler for rendering. The global explicit constraint is conducted by a voxel contrastive regularization, which regularizes the radiance properties between points in a voxel to be more similar than that of points across voxels.

ment the input of MLP with features projected from a CNN feature map. MVSNeRF [3] builds a 3D volume by warping CNN features and augments the input of the MLP with features from this volume. Although these methods show promising results, they often require multi-view datasets for pre-training, which are not always available, and their performance may drop when given sufficient inputs due to domain differences. There are also works focusing on training NeRF with a single image [7, 49, 56]. Their methods mainly rely on generative models [8, 33] to synthesize images of novel views. Our setting differs from theirs in that the given images can cover the scene from multiple viewpoints.

The second line of research utilizes regularization techniques during training. DS-NeRF [5] aligns the density distribution of each ray with the depth supervision which is available from structure-from-motion. A similar method is also applied in DDP [32], DINER [28] and SparseNeRF [44]. DietNeRF [10] regularizes the semantic consistency among images from arbitrary views in the embedding space of CLIP [31]. InfoNeRF [13] applies sparsity regularization by minimizing the entropy of each ray density. RegNeRF [26] leverages a 2D consistency loss on depths and colors of image patches to impose that neighboring pixels have similar geometry and appearance. However, in this paper, we explore modeling 3D local and global spatial consistency for optimizing NeRFs from sparse inputs. Compared with the 2D consistency utilized in RegNeRF [26], 3D consistency is a stronger regularization which can directly regularizes 3D spatial neighboring regions to learn consistent physical radiance properties. The most relevant method with ours is Nerfbusters [48], which refines local regions to ensure consistency by a data-driven diffusion prior. In contrast, our method applies a contrastive in-voxel Transformer structure to implement 3D consistency from both local and global perspectives without using any off-the-shelf pre-trained models and external priors.

3. The Proposed CVT-*x*RF

In this work, we propose to use 3D spatial field consistency to regularize radiance fields when training from sparse inputs. Because of the sparse supervision, it is critically important to handle the 3D field consistency in radiance field learning, *i.e.*, neighboring regions in 3D space having similar physical properties, *e.g.*, density and color. Our proposed CVT-*x*RF for implementing 3D field consistency learning is illustrated in Fig. 2. Our CVT-*x*RF comprises three major components, which are a voxel-based ray sampling strategy (Sec. 3.2), a local implicit constraint module based on a designed light-weight In-Voxel Transformer (Sec. 3.3), and a global explicit constraint module based on a voxel contrastive regularization (Sec. 3.4). We elaborate on them after a brief review of NeRF in Sec. 3.1 as follows.

3.1. Preliminary

Neural Radiance Field (NeRF) adopts an MLP network to represent a scene, which maps the 3D coordinate $\boldsymbol{x} = (x, y, z)$ and its 2D view direction $\mathbf{d} = (\theta, \phi)$ to a pair of radiance property values, *i.e.*, $(\mathbf{c}, \sigma)$, where $\mathbf{c}$ and σ represent the color and the density, respectively. To facilitate subsequent discussions, we additionally extract a feature vector $\mathbf{g}$ from the layer of the MLP that predicts the density. The aforementioned process can be formulated as:

$$(\mathbf{c}, \sigma, \mathbf{g}) = \mathrm{MLP}(\gamma(\boldsymbol{x}), \gamma(\mathbf{d})), \tag{1}$$

where $\gamma(\cdot)$ refers to a positional encoding. The color of each ray is calculated through volumetric rendering, which accumulates colors of N points sampled from a uniform or an importance sampler. This process can be formulated as: $C(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_{\mathrm{i}}\delta_{\mathrm{i}}))\mathbf{c}_{\mathrm{i}}$, where δ_i refers to the distance between two adjacent samples and T_i is the accumulated transmittance calculated by: $T_i = \exp(-\sum_{\mathrm{j}=1}^{\mathrm{i}-1} \sigma_{\mathrm{j}}\delta_{\mathrm{j}})$. During training, random rays are sampled along with their corresponding ground truth colors. Their colors are rendered by volume rendering and the MLP is learned via supervision from a mean squared error as:

$$\mathcal{L}_{\mathrm{mse}} = \sum_{\mathbf{r}} \|C_{\mathrm{gt}}(\mathbf{r}) - C(\mathbf{r})\|^2, \tag{2}$$

where $C_{\mathrm{gt}}(\mathbf{r})$ denotes the ground truth color. Though NeRF achieves impressive results given dense inputs, it cannot recover correct geometry and appearance from sparse inputs and fails to synthesize high-quality images of novel views.

3.2. Voxel-based Ray Sampling Strategy

Due to the sparse-view supervision, 3D spatial field consistency is not guaranteed in the learned NeRF representation. The consistency indicates that neighboring regions in 3D space have similar radiance properties. However, defining appropriate neighboring regions remains a challenge. We propose a reasonable hypothesis that regions within a small voxel in 3D space are likely to present similar properties, and our experiments in supplementary materials demonstrate that this hypothesis leads to significant gains across a wide range of voxel sizes. To implement this hypothesis, we uniformly divide the scene into voxels with an equal size. Since we have access to training images and their corresponding camera parameters, we can record the voxels that each ray intersects with. Each voxel can then store multiple rays that intersect with it. We introduce a voxel-based ray sampling strategy as shown in Fig. 2, which supports the local and global field consistency learning proposed in Sec. 3.3 and Sec. 3.4. Our strategy starts with a sampling of V voxels, denoted as $\{\mathbf{V}_i\}_{i=1}^{V}$. It then samples R rays from each voxel and returns $\{\mathbf{R}_i\}_{i=1}^{V}$, where each $\mathbf{R}_i$ refers to a set of R rays sampled from a voxel $\mathbf{V}_i$. Using this strategy, we can rewrite the supervision of Eq. (2) as:

$$\mathcal{L}_{\mathrm{mse}} = \sum_{i=1}^{V} \sum_{\mathbf{r}\in\mathbf{R}_i} \|C_{\mathrm{gt}}(\mathbf{r}) - C(\mathbf{r})\|^2, \tag{3}$$

and the batch size of training rays is thus $V \times R$.

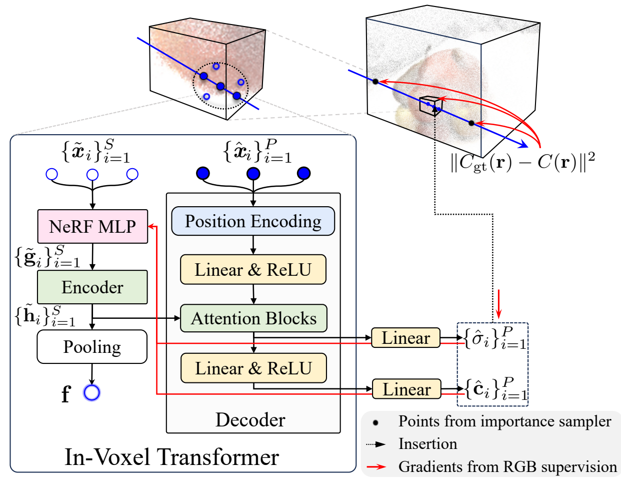

Figure 3. The proposed local implicit constraint with a light-weight In-Voxel Transformer for 3D field consistency learning. The colors and densities of ray points $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{P}$ are predicted by surrounding points $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^{S}$ through the Transformer. The predicted colors and densities are inserted into the ray for rendering.

3.3. Local Implicit Constraint

Now we start to introduce the proposed local implicit constraint for 3D field consistency learning. Considering a voxel $\mathbf{V}$, the physical radiance properties of two small regions in this voxel should be similar, which means we can infer the properties of one region through an interaction with the other. In the following, we represent these two regions in the voxel $\mathbf{V}$ by two point sets, which contain the surrounding points and the ray points, respectively. For conciseness, we drop subscripts and only consider a ray $\mathbf{r}$ sampled from the voxel $\mathbf{V}$.

Surrounding and ray points sampling. As shown in Fig. 2, in voxel $\mathbf{V}$, we sample S surrounding points $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^{S}$ for each ray $\mathbf{r}$. Concretely, we firstly obtain two points that the ray intersects $\mathbf{V}$ with, *i.e.*, $\boldsymbol{x}_{\mathrm{in}}$ and $\boldsymbol{x}_{\mathrm{out}}$. We then perform sphere sampling using $\mathcal{F}_{\mathrm{sphere\_sample}}$ with the center located at the midpoint of the two intersecting points as:

$$\{\tilde{\boldsymbol{x}}_i\}_{i=1}^{S} = \mathcal{F}_{\mathrm{sphere\_sample}}((\boldsymbol{x}_{\mathrm{in}} + \boldsymbol{x}_{\mathrm{out}})/2, \mathrm{radius}, S), \tag{4}$$

where radius is set to $1/n$ of the voxel size. According to the 3D field consistency discussed, the radiance properties (*i.e.*, colors and densities) of the surrounding points are highly beneficial for inferring those of the ray points. As shown in Fig. 2, we sample P ray points $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{P}$ along the ray $\mathbf{r}$ with $\mathcal{F}_{\mathrm{line\_sample}}$. Specifically, we randomly sample points along the line segment connecting $\boldsymbol{x}_{\mathrm{in}}$ and $\boldsymbol{x}_{\mathrm{out}}$ as:

$$\{\hat{\boldsymbol{x}}_i\}_{i=1}^{P} = \mathcal{F}_{\mathrm{line\_sample}}(\boldsymbol{x}_{\mathrm{in}}, \boldsymbol{x}_{\mathrm{out}}, P). \tag{5}$$

Prediction by a light-weight In-Voxel Transformer. After obtaining the surrounding points and ray points, we introduce a light-weight Transformer structure to perform inference of the radiance properties of the ray points based on the surrounding points. The proposed Transformer structure consists of an encoder and a decoder. The encoder is designed to encode the properties of the region containing the surrounding points, while the decoder is responsible for decoding the radiances of the ray points based on the encoded information from the surrounding points.

Fig. 3 illustrates the details of the Transformer structure. The input to the encoder consists of the point features $\mathbf{g}$ obtained from the MLP, as depicted in Eq. 1. Concretely, We forward the coordinates of the surrounding points into MLP and obtain their corresponding features, *i.e.*, $\{\tilde{\mathbf{g}}_i\}_{i=1}^S = \mathrm{MLP}(\{\tilde{\boldsymbol{x}}_\mathrm{i}\}_{\mathrm{i}=1}^\mathrm{S}, \mathbf{d})$. The encoding procedure is as follows:

$$\{\tilde{\mathbf{h}}_i\}_{i=1}^S = \mathrm{Encoder}(\{\tilde{\mathbf{g}}_i\}_{i=1}^S), \tag{6}$$

where $\mathrm{Encoder}$ consists of self-attention blocks and outputs updated features $\{\tilde{\mathbf{h}}_i\}_{i=1}^S$. To achieve a compact representation, we aggregate $\{\tilde{\mathbf{h}}_i\}_{i=1}^S$ by a pooling operation as:

$$\mathbf{f} = \mathrm{Pool}(\{\tilde{\mathbf{h}}_i\}_{i=1}^S), \tag{7}$$

where Pool is implemented with a max pooling in practice. $\mathbf{f}$ is now the feature that represents a specific 3D region containing the surrounding points in the voxel. Then, the decoder aims at inferring colors and densities of the ray points from the encoded representation $\{\tilde{\mathbf{h}}_i\}_{i=1}^S$ of surrounding points. Different from the common practice of NeRF that predicts the densities and colors of points in isolation, the decoder performs the prediction based on the encoded information from the neighboring points. The detailed architecture of the decoder is illustrated in Fig. 3. For P ray points, the decoder firstly transforms their coordinates by position encoding, which is followed by a non-linearity. The self-attention blocks receive both the non-linear output and the encoder output $\{\tilde{\mathbf{h}}_i\}_{i=1}^S$, to decode the radiances for the P ray points. The outputs of the attention blocks are directly utilized to predict the densities of the ray points, *i.e.*, $\{\hat{\sigma}_i\}_{i=1}^P$. After applying a non-linearity to the outputs of the attention blocks, the colors, denoted as $\{\hat{\mathbf{c}}_i\}_{i=1}^P$, are predicted. The decoding procedure is formulated as:

$$\{(\hat{\mathbf{c}}_i, \hat{\sigma}_i)\}_{i=1}^P = \mathrm{Decoder}(\{\hat{\boldsymbol{x}}_i\}_{i=1}^P, \{\tilde{\mathbf{h}}_i\}_{i=1}^S). \tag{8}$$

Insertion and rendering. After we have obtained the colors and densities of P ray points, we then insert them into the ray. Along with the original N points that are sampled from the importance sampler, we combine the radiances of the $N + P$ points for volume rendering as illustrated in Fig. 3. During training, the gradients from the color rendering loss (Eq. 3) are backpropagated to the points along the ray, including the ray points. As shown in Fig. 3, the gradients can flow back to the encoder, thus updating the parameters of the MLP. Through the interaction between the surrounding points and the ray points in the neighboring regions, our local implicit constraint can largely enhance the 3D field consistency during training.

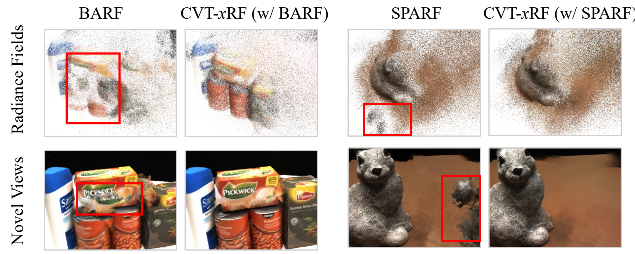

Figure 4. Visualization of learned radiance fields (in 3D) and the corresponding rendering results of two baselines, *i.e.*, BARF [16] and SPARF [41], with/without the proposed CVT.

3.4. Global Explicit Constraint

The local implicit constraint enhances the 3D field consistency of neighboring regions by the interaction between surrounding points and ray points. In this section, we propose a global explicit constraint, which directly enforces the similarity between features of neighboring regions for 3D field consistency. After applying the local implicit constraint, we can obtain a pooled feature $\mathbf{f}$ from the features of a set of surrounding points as in Eq. (7). Thus, each $\mathbf{f}$ can represent a specific region depicted by the set of surrounding points within a voxel, we denote $\mathbf{f}$ as a region feature. Following the voxel-based ray sampling strategy in Sec. 3.2, we sample V voxels, with R rays from each voxel during training. We can then obtain a set of features from the encoder output of the Transformer, *i.e.*, $\{\mathbf{f}_i^{(j)}\}_{i=1:V, j=1:R}$.

Inspired by contrastive learning schemes [4, 38] that learn discriminative features via optimizing the distance between feature pairs, we propose to use a voxel contrastive loss to further enhance the 3D field consistency. Specifically, for each region feature (an anchor), its distance to other neighboring features from the same voxel (positive pairs) should be smaller than the distance to region features from other voxels (negative pairs). Following Chen *et al.* [4], for each anchor region feature, we only select a positive region feature from the same voxel to construct a positive pair, and select all negative region features in other voxels to build its negative pairs. The positive/negative pair selection is illustrated in Fig. 5. The contrastive loss $\mathcal{L}_\text{contrast}$ regarding $V \times R$ region features can be formulated as:

$$\mathcal{L}_\text{contrast} = -\sum_{i=1}^{V}\sum_{j=1}^{R}\Bigg(\frac{\langle \mathbf{f}_i^{(j)}, \mathbf{f}_i^{(pos)}\rangle}{\tau} - \log\Big(\exp\big(\frac{\langle \mathbf{f}_i^{(j)}, \mathbf{f}_i^{(pos)}\rangle}{\tau}\big) + \sum_{\substack{n=1\\ n\neq i}}^{V}\sum_{k=1}^{R}\exp\big(\frac{\langle \mathbf{f}_i^{(j)}, \mathbf{f}_n^{(k)}\rangle}{\tau}\big)\Big)\Bigg),$$

where $\langle\cdot\rangle$ and τ denote the cosine similarity and the temperature. $\mathbf{f}^{(pos)}$ denotes a positive pair of the anchor, which is randomly selected from the same voxel with the anchor. During training, the gradients from the contrastive loss can flow back to the MLP (see Fig. 5) to update its parameters,

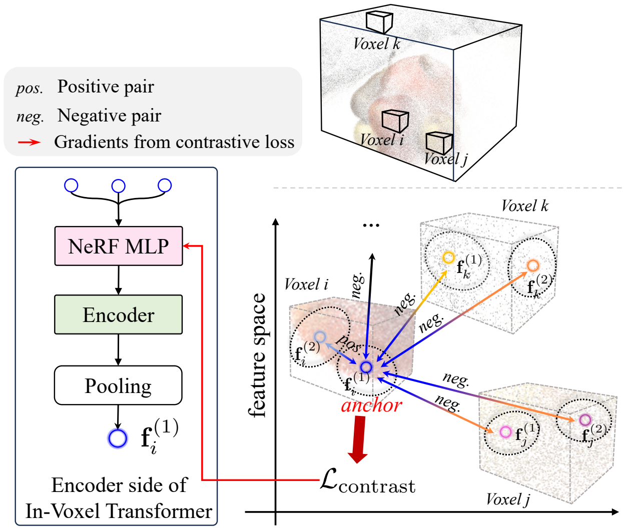

Figure 5. The proposed global explicit constraint for field consistency learning is implemented by a contrastive loss. For each anchor, we select its positive pair in the same voxel, while its negative pairs in other voxels.

which makes the MLP network learn more consistent region features, leading to better 3D field consistency.

3.5. Overall Objective

Based on the voxel-based ray sampling strategy, as well as the local implicit and the global explicit constraints, the overall training loss of our CVT-*x*RF can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{mse}} + \lambda \mathcal{L}_{\text{contrast}}, \tag{9}$$

where λ is a balancing parameter. NeRF commonly uses a coarse-level and a fine-level MLP, which apply a uniform sampling and an importance sampling, respectively. Our proposed CVT-*x*RF is only applied on the fine-level MLP.

4. Experiments

4.1. Datasets and Evaluation

Datasets. We evaluate our proposed method on multi-view DTU dataset [11] and Synthetic dataset [22]. We report results of 3, 6, and 9 input views for the DTU dataset, while 3 and 8 input views for the Synthetic dataset. For more details about the scenes and view selection, we refer readers to the supplementary material.

Evaluation. For quantitative comparison of synthesis results, we report the mean of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [47] over different scenes. We refer readers to supplementary material for LPIPS perceptual metric [55], geometric mean [26], and the consideration of evaluation on DTU dataset.

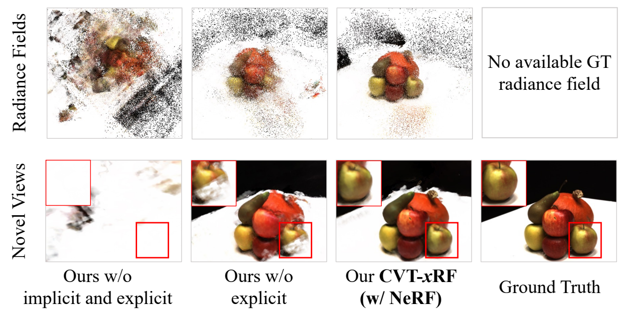

Figure 6. Efficacy of local implicit and global explicit constraints.

Methods	DTU 3-view PSNR	DTU 3-view SSIM	Synthetic 3-view PSNR	Synthetic 3-view SSIM
NeRF [22]	6.68	0.249	11.41	0.724
CVT-*x*RF (w/ NeRF)	**14.13**	**0.518**	**19.28**	**0.815**
w/o implicit and explicit	7.23	0.264	15.66	0.795
w/o explicit	11.85	0.458	18.26	0.806

Table 1. Effectiveness of different parts of the proposed CVT-*x*RF over vanilla NeRF. Implicit and explicit refer to the local implicit constraint and global explicit constraint, respectively.

4.2. Ablation Studies

Our CVT-*x*RF consists of three parts, which are voxel-based ray sampling, local implicit constraint, and global explicit constraint. We analyze the effect of these components on the DTU and the Synthetic dataset of 3 input views.

Effect of voxel-based ray sampling. The proposed sampling strategy is a prerequisite of the local implicit and the global explicit constraints. To study its effect, we apply our sampling strategy on NeRF [22], denoted as 'w/o implicit and explicit' in Tab. 1. It can be observed that it brings improvements over the random ray sampling strategy applied by NeRF. The results demonstrate that the proposed sampling strategy is more effective for the sparse-input setting.

Effect of local implicit constraint. Tab. 1 validates that, using the local implicit constraint, denoted as 'w/o explicit', brings a large improvement upon the voxel-based ray sampling strategy. The local implicit constraint is implemented by a Transformer architecture, which infers the radiances of the ray points from the interaction with surrounding points. Thus, the 3D field consistency of the region that surrounding points are distributed in is enhanced as illustrated in Fig. 6. With the implicit constraint, the radiance field distribution is learned with better 3D consistency.

Effect of global explicit constraint. As shown in Tab. 1, the global explicit constraint also increases the performance. On DTU dataset, the PSNR is improved from 11.85 to 14.13, and the SSIM is also boosted from 0.458 to 0.518. These improvements verify that, by considering the negative pairs in the $\mathcal{L}_{\text{contrast}}$, our method can effectively facilitate the learning of the 3D field consistency. This can also be confirmed from Fig. 6: certain amounts of artifacts are

Methods	*Pre.*	Setting	Full-image PSNR			Full-image SSIM			Object PSNR			Object SSIM		
			3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view
PixelNeRF [52]	×	Trained on DTU and Finetuned per Scene	**18.74**	21.02	22.23	0.618	0.684	0.714	16.82	19.11	20.40	0.695	0.745	0.768
MVSNeRF [3]			16.33	18.26	20.32	0.602	0.602	0.735	18.63	20.70	22.40	0.769	0.823	0.853
PixelNeRF ft [52]			17.38	21.52	21.67	0.548	0.670	0.680	18.95	20.56	21.82	0.710	0.753	0.781
MVSNeRF ft [3]			16.26	18.22	20.32	0.601	0.601	0.736	18.54	20.49	22.22	0.769	0.822	0.853
NeRF [22]	×	Optimized per Scene	6.68	15.32	16.29	0.249	0.626	0.693	7.79	18.23	18.80	0.595	0.758	0.801
mip-NeRF [1]			7.64	14.33	20.71	0.227	0.568	0.799	8.68	16.54	23.58	0.571	0.741	0.879
RegNeRF [26]			15.33	19.10	22.30	0.621	0.757	0.823	18.89	22.20	24.93	0.745	0.854	0.884
FlipNeRF [35]			-	-	-	-	-	-	19.55	22.45	25.12	0.767	0.839	0.882
FreeNeRF [51]			18.02	22.39	24.20	0.680	0.779	0.833	19.92	23.25	25.38	0.787	0.841	0.888
CVT-*x*RF (w/ BARF)			18.06	**23.40**	**26.56**	**0.762**	**0.872**	**0.910**	**21.33**	**25.50**	**27.68**	**0.844**	**0.911**	**0.938**
DietNeRF [10]	✓	Optimized per Scene	10.01	18.70	22.16	0.354	0.668	0.740	11.85	20.63	23.83	0.633	0.778	0.823
SPARF [41]			18.30	23.24	25.75	0.780	0.870	0.910	21.01	**25.76**	27.30	0.870	0.920	0.940
SPARF*			18.32	23.43	25.75	0.784	0.879	0.910	21.26	25.07	27.30	0.873	**0.921**	0.940
CVT-*x*RF (w/ SPARF)			**18.98**	**24.51**	**27.04**	**0.801**	**0.884**	**0.919**	**21.51**	25.14	**27.63**	**0.874**	0.920	**0.945**

Table 2. Comparison on DTU dataset. We present the performances of both full images and foreground objects. We organize the comparisons into two categories according to whether the methods use off-the-shelf models pre-trained on other datasets (indicated by *Pre.*) or not. The improvement brought by the pre-trained model in RegNeRF is limited, so we place it into the first category. * means that we rerun the experiments with their official code. The best, second-best and third-best entries of the first category of comparison are marked in **red**, blue and orange, respectively. For the second category of comparison, we mark the best entries with **bold**.

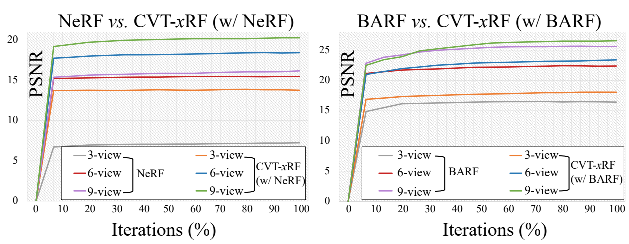

Figure 7. Performances of NeRF and BARF throughout the training process with/without the proposed CVT.

removed with the global explicit constraint.

4.3. Performance on Different Baselines

We validate the efficacy of our method on different baselines, which are NeRF [22], BARF [16], and SPARF [41]. BARF and SPARF include pose optimization modules in their methods. We switch them off for fair comparisons.

Methods	Mem / Time (50k)	3-view		6-view		9-view	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NeRF [22]	4.4G / 1.6h	6.68	0.249	15.32	0.626	16.29	0.693
CVT-*x*RF (w/ NeRF)	6.4G / 2.8h	**14.13**	**0.518**	**18.43**	**0.713**	**20.28**	**0.754**
BARF [16]	4.4G / 1.6h	16.43	0.703	22.26	0.863	25.54	0.908
CVT-*x*RF (w/ BARF)	6.4G / 2.8h	**18.06**	**0.762**	**23.40**	**0.872**	**26.56**	**0.910**
SPARF [41]	*20G / 11.0h	18.32	0.784	23.43	0.879	25.75	0.910
CVT-*x*RF (w/ SPARF)	*22G / 13.0h	**18.98**	**0.801**	**24.51**	**0.884**	**27.04**	**0.919**

Table 3. Improvements and extra training overheads over different baselines on DTU dataset of different input views. * refers to peak memory consumption. The overheads are measured on scan40.

Performance and training overhead. Tab. 3 shows that the proposed CVT-*x*RF brings considerable improvement on three baselines in all the settings. CVT-*x*RF brings more than 7.0, 3.0, 3.0 PSNR improvement over NeRF given 3, 6, 9 input views, respectively. For the baselines that already achieve high performance such as BARF and SPARF, our approach can still bring a large gain regarding the different number of input views. It should be noted that SPARF is currently one of the most effective methods of learning radiance fields from sparse inputs. Tab. 3 also shows that, our CVT-*x*RF does not bring heavy overhead on different baselines, in terms of the GPU memory and the training time. We visualize the radiance fields learned by BARF and SPARF with and without CVT in Fig. 4. It is obvious that CVT improves the 3D field consistency and effectively removes the floating artifacts in the rendered images.

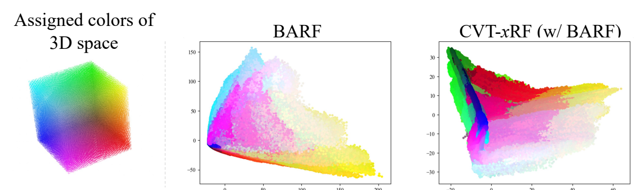

Figure 8. The feature distribution of DTU scan45 after PCA, with and without the proposed CVT. We uniformly sample the points in 3D space and assign distinct colors to them. It is observed that With CVT, the features demonstrate greater diversity compared to the homogeneity observed with BARF.

Convergence speed. The convergence speeds of models with and without the proposed CVT are illustrated in Fig. 7. In the majority of cases, the baseline models that incorporate CVT can consistently outperform the baselines throughout the entire training process. This indicates that our CVT-*x*RF exhibit the capability to converge rapidly to a relatively high performance level from start of the training.

Distribution of the learned MLP features. Fig. 8 presents the distribution of features **g** from the MLP learned on DTU scan45. Uniform sampling of points in 3D space is conducted, and their corresponding MLP features are produced. PCA is then applied to visualize the point features. The figure demonstrates that the proposed CVT-*x*RF can learn more discriminative features across different regions. This

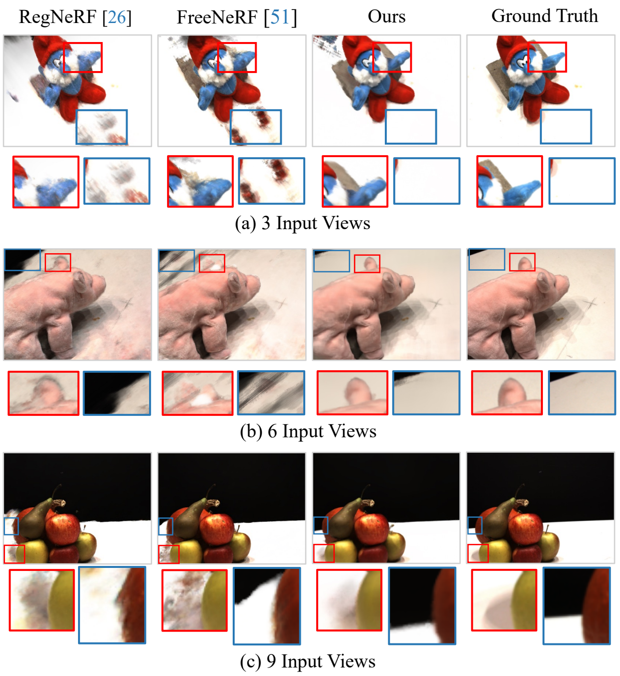

Figure 9. Qualitative comparisons on DTU dataset. For 3, 6, 9 input views, our method clearly preserves better consistency and exhibits significantly fewer artifacts.

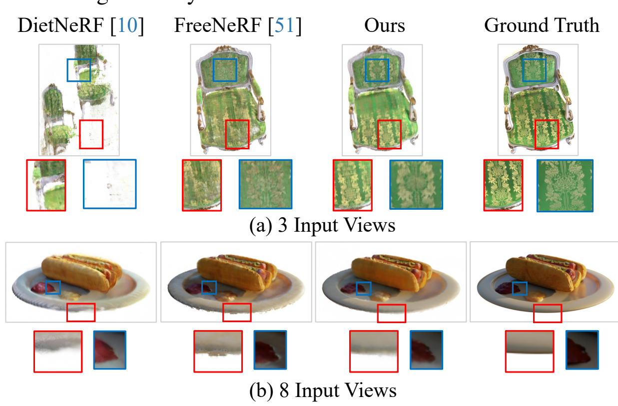

Figure 10. Qualitative comparisons on Synthetic dataset. For 3 input views, our method preserves more details. For 8 input views, our method shows more accurate colors and keeps sharper edges.

Methods	3-view PSNR	3-view SSIM
MVSNeRF [3]	15.12	0.820
GeoNeRF [12]	17.67	0.730
ENeRF [17]	18.14	0.830
mip-NeRF [1]	16.52	0.800
DSNeRF [5]	15.13	0.820
DietNeRF [10]	17.55	0.770
RegNeRF [26]	17.39	0.820
FreeNeRF [51]	20.75	0.842
ConsistentNeRF [9]	19.63	0.830
CVT-*x*RF (w/ NeRF)	19.28	0.815
CVT-*x*RF (w/ BARF)	**21.58**	**0.851**

(a) Synthetic 3-view setting.

Methods	8-view PSNR	8-view SSIM
NeRF [22]	14.94	0.687
NV [19]	17.86	0.741
Simplified NeRF [10]	20.09	0.822
DietNeRF$_{50k}$ [10]	23.15	0.867
DietNeRF$_{200k}$ [10]	23.59	0.874
FreeNeRF [51]	24.26	**0.883**
CVT-*x*RF (w/ BARF)	23.33	0.874
CVT-*x*RF (w/ FreeNeRF)	**24.56**	**0.883**

(b) Synthetic 8-view setting.

Table 4. Comparison on Synthetic dataset. The first block lists the methods that require pre-training on other datasets. The best, second-best and third-best entries are marked in **red**, blue and orange, respectively. No public results for methods that require pre-training are available for 8-view.

verifies the superior capability of the proposed method in modeling the scene's radiance characteristics.

4.4. State-of-the-art Comparison

In the following, unless specifically mentioned, we mainly consider BARF [16] as our baseline and use CVT-*x*RF (w/ BARF) to compare with the state-of-the-art methods.

DTU dataset. Tab. 2 presents the results obtained on the DTU dataset. The comparison methods are organized into two categories based on whether the methods use off-the-shelf pre-trained models on other datasets or not. For instance, SPARF [41] and DietNeRF [10] employ a matching network [40] and CLIP [31], respectively. In the first category of comparisons, our method consistently achieves the highest performance across most cases. The qualitative results can be observed in Fig. 9. Notably, our CVT-*x*RF (w/ BARF) even achieves comparable performance to SPARF in the 6/9-view settings, while requiring significantly lower training overhead, as shown in Tab. 3. Regarding the second category of comparisons, CVT-*x*RF (w/ SPARF) surpasses SPARF, particularly in terms of the full-image performance. This indicates that the proposed CVT-*x*RF complements the matching mechanism of SPARF that does not explicitly model the 3D field consistency.

Synthetic dataset. Tab. 4 shows the comparisons on the Synthetic dataset. For 3-view, our approach achieves the best results compared with other methods, and also achieves higher performance compared to the works that require pre-training [3, 12, 17]. Fig. 10 (a) shows that, our method can recover better object details compared to FreeNeRF [51]. For 8-view, CVT-*x*RF (w/ BARF) achieves lower performance. We observe that the occlusion regularization in FreeNeRF is crucial in this setting and we combine the CVT structure with it, denoted as CVT-*x*RF (w/ FreeNeRF). We not only achieve a 0.3 PSNR improvement over the best-performing method but also significantly enhance the radiance distribution. However, these improvements may not be evident from the evaluation metrics alone. Please refer to the supplementary material for more details.

5. Conclusion

In this paper, we introduce a novel approach for learning 3D spatial field consistency to regularize NeRF when training from sparse inputs. The field inconsistency is typically caused by the lack of supervision across scene views. We propose a novel Contrastive In-Voxel Transformer structure to learn the 3D spatial field consistency, which is composed of a voxel-based ray sampling strategy, a local implicit constraint, and a global explicit constraint. Our experiments demonstrate that our method outperforms various NeRF baselines in terms of 2D rendering quality, and exhibits better 3D field consistency.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 2, 7, 8

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[3] 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*, 2021. 3, 7, 8

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 5

[5] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 2, 3, 8

[6] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018. 2

[7] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *ICML*, 2023. 3

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[9] Shoukang Hu, Kaichen Zhou, Kaiyu Li, Longhui Yu, Lanqing Hong, Tianyang Hu, Zhenguo Li, Gim Hee Lee, and Ziwei Liu. Consistentnerf: Enhancing neural radiance fields with 3d consistency for sparse view synthesis. *arXiv preprint arXiv:2305.11031*, 2023. 8

[10] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 2, 3, 7, 8

[11] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 6

[12] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *CVPR*, 2022. 8

[13] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 2022. 2, 3

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *ICML*, 2021. 2

[16] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2, 5, 7, 8

[17] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH*, 2022. 8

[18] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 2

[19] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 8

[20] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, 2022. 2

[21] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 1, 2

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 6, 7, 8

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

[24] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[25] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 1, 2

[26] 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*, 2022. 2, 3, 6, 7, 8

[27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 1, 2

[28] Malte Prinzler, Otmar Hilliges, and Justus Thies. Diner: Depth-aware image-based neural radiance fields. In *CVPR*, 2023. 2, 3

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3, 8

[32] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. 2, 3

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[34] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 2

[35] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *ICCV*, 2023. 7

[36] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, 2019. 2

[37] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 1, 2

[38] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016. 5

[39] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017. 2

[40] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021. 8

[41] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023. 2, 5, 7, 8

[42] 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*, 2017. 2

[43] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 2

[44] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *ICCV*, 2023. 3

[45] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 2

[46] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2

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

[48] Frederik Warburg, Ethan Weber, Matthew Tancik, Aleksander Holynski, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. *arXiv preprint arXiv:2304.10532*, 2023. 3

[49] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 3

[50] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2

[51] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, 2023. 7, 8

[52] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 7

[53] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 2

[54] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[56] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023. 3