SparseNeRF: Distilling Depth Ranking for Few-shot Novel View Synthesis

Guangcong Wang   Zhaoxi Chen   Chen Change Loy   Ziwei Liu✉

S-Lab, Nanyang Technological University
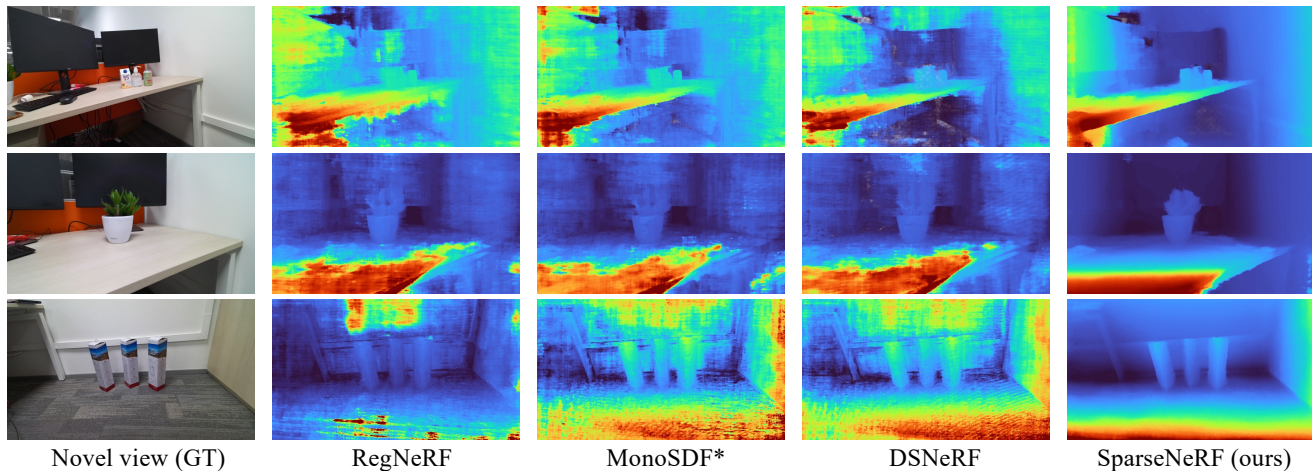
{guangcong.wang,zhaoxi00,ccloy,ziwei.liu}@ntu.edu.sg

Figure 1: Comparing the state-of-the-arts RegNeRF [37], MonoSDF* [65], DSNeRF [17], and our SparseNeRF with three views for training (* denotes a re-implementation for our task). RegNeRF regularizes geometry with sparsity and continuity constraints. MonoSDF, and DSNeRF use scale-invariant depth constraints supervised by coarse depth maps. Our SparseNeRF uses robust depth ranking of coarse depth maps. SparseNeRF can synthesize realistic novel views and coherent geometric depth (Please refer to the supplementary material for rendered videos).

Abstract

Neural Radiance Field (NeRF) significantly degrades when only a limited number of views are available. To complement the lack of 3D information, depth-based models, such as DSNeRF and MonoSDF, explicitly assume the availability of accurate depth maps of multiple views. They linearly scale the accurate depth maps as supervision to guide the predicted depth of few-shot NeRFs. However, accurate depth maps are difficult and expensive to capture due to wide-range depth distances in the wild. This work presents a new Sparse-view NeRF (SparseNeRF) framework that exploits depth priors from real-world inaccurate observations. The inaccurate depth observations are either from pre-trained depth models or coarse depth maps of consumer-level depth sensors. Since coarse depth maps are not strictly scaled to the ground-truth depth maps, we propose a simple yet effective constraint, a local depth ranking method, on NeRFs such that the expected depth ranking of the NeRF is consistent with that of the coarse depth maps in local patches. To preserve the spatial continuity of the esti-

mated depth of NeRF, we further propose a spatial continuity constraint to encourage the consistency of the expected depth continuity of NeRF with coarse depth maps. Surprisingly, with simple depth ranking constraints, SparseNeRF outperforms all state-of-the-art few-shot NeRF methods (including depth-based models) on standard LLFF and DTU datasets. Moreover, we collect a new dataset NVS-RGBD that contains real-world depth maps from Azure Kinect, ZED 2, and iPhone 13 Pro. Extensive experiments on NVS-RGBD dataset also validate the superiority and generalizability of SparseNeRF. Code and dataset are available at https://sparsenerf.github.io/.

1. Introduction

Neural radiance fields (NeRFs) [36, 32, 1, 2, 44] have made tremendous progress in generating photo-realistic novel views of scenes by optimizing implicit function representations given a set of 2D input views. However, in a wide range of real-world scenarios, collecting dense views of a

scene is often expensive and time-consuming [47]. Therefore, it is necessary to develop few-shot NeRF methods that can be learned from sparse views without significant degradation in performance.

Learning a NeRF from sparse views is a challenging problem due to under-constrained reconstruction conditions, especially in textureless areas. Directly applying NeRFs to few-shot scenarios suffers from dramatic degradation [37]. Recently, some methods have greatly improved the performance of few-shot NeRF, which can be categorized into three groups. **1)** The first group [37, 27, 25] is based on geometry constraints (sparsity and continuity regularizations) and high-level semantics. RegNeRF [37] regularized the geometry and appearance of patches rendered from unobserved viewpoints, and annealed the ray sampling space. InfoNeRF [27] imposed an entropy constraint of the density in each ray and a spatial smoothness constraint into the estimated images. However, since a scene often contains multiple layouts (**Figure 1**), sparsity and continuity geometric constraints of a few views cannot guarantee the complete 3D geometric reconstruction. **2)** The second group [64, 5] resorts to pre-training on similar scenes. For example, PixelNeRF [64] proposed to condition a NeRF on convolutional feature maps to learn high-level semantics from other scenes. **3)** The third group [17, 65, 24] exploits depth maps and makes a linearity assumption of depth maps to supervise few-shot NeRFs. For example, DSNeRF [17] exploited sparse 3D points generated by COLMAP [42] or accurate depth maps [13] obtained by high-accuracy depth scanners and the Multi-View Stereo (MVS) algorithm. The depth maps are linearly scaled as supervision to guide the predicted depth of few-shot NeRFs. To use coarse depth maps, MonoSDF [65] uses a local patch-based scale-invariant depth constraint supervised by coarse depth maps instead of global depth maps. However, the scale-invariant depth constraint is strong for real-world coarse depth maps from pre-trained depth models or consumer-level depth sensors due to wide-range depth distances in the wild.

Along the third group, we wish to explore more robust 3D priors from coarse depth maps to complement the under-constrained few-shot NeRF. To address this problem, we present SparseNeRF, a simple yet effective method that distills depth priors from pre-trained depth models [41] or inaccurate depth maps from consumer-level depth sensors (**Figure 3**), which can be easily obtained from real-world scenes. Deriving useful depth cues from such pre-trained models is non-trivial. In particular, although single-view depth estimation methods have achieved good visual performance, thanks to large-scale monocular depth datasets and large ViT models, they cannot yield accurate 3D depth information due to coarse depth annotations, dataset bias, and ill-posed 2D single-view images. The inaccurate depth information contradicts the density prediction of a NeRF

when reconstructing each pixel of a 3D scene based on volume rendering. Directly scaling the coarse depth maps to a NeRF [17, 65] leads to inconsistent geometry against the expected depth of the NeRF.

Instead of directly supervising a NeRF with coarse depth priors, we relax hard depth constraints [17, 65] and distill robust local depth ranking from the coarse depth maps to a NeRF such that the depth ranking of a NeRF is consistent with that of coarse depth. That is, we supervise a NeRF with relative depth instead of absolute depth [17, 65]. To guarantee the spatial continuity of geometry, we further propose a spatial continuity constraint on depth maps such that the NeRF model imitates the spatial continuity of coarse depth maps. The accurate sparse geometry constraints from a limited number of views, combined with relaxed constraints including depth ranking regularization and continuity regularization, finally achieve promising novel view synthesis (**Figure 1**). It is noteworthy that SparseNeRF does not increase the running time during inference as it only exploits depth priors from pre-trained depth models or consumer-level sensors during the training stage (**Figure 2**). In addition, SparseNeRF is a plug-and-play module that can be easily integrated into various NeRFs.

The main contributions of this paper are **1)** SparseNeRF, a simple yet effective method that distills local depth ranking priors from pre-trained depth models. With the help of the local depth ranking constraint, SparseNeRF significantly improves the performance of few-shot novel view synthesis over the state-of-the-art models (including depth-based NeRF methods). To preserve the coherent geometry of a scene, we propose a spatial continuity distillation constraint that encourages the spatial continuity of NeRF to be similar to that of the pre-trained depth model. Both depth ranking prior and spatial continuity distillation are new in the literature on NeRF. **2)** Apart from SparseNeRF, we also contribute a new dataset, NVS-RGBD, which contains coarse depth maps from Azure Kinect, ZED 2, and iPhone 13 Pro. **3)** Extensive experiments on the LLFF, DTU, and NVS-RGBD datasets demonstrate that SparseNeRF achieves a new state-of-the-art performance in few-shot novel view synthesis.

2. Related Work

Neural Radiance Fields. NeRF [36, 48, 8, 39] has made great success in synthesizing novel views of complex scenes due to good representation of neural networks [20, 52, 50, 53, 18, 28]. Block-NeRF [47], CityNeRF [58], and Mega-NeRF [49] scaled the standard NeRF up to city-scale or urban-scale scenes. NeRF−− [56], GNeRF [33], and BARF [31] relaxed the requirements of NeRFs and synthesized novel views without perfect camera poses. Some works attempted to improve NeRFs by considering anti-aliasing [1], sparse 3D grids with spherical harmonics [19].

Some methods [40, 30, 57] extended NeRFs to dynamic scenes. These methods do not focus on generating novel views with a few views. In this paper, we study sparse-view NeRF to reduce dense capture requirements in real-world applications.

Few-shot Novel View Synthesis. There are increasing studies on few-shot novel view synthesis. Basically, the existing few-shot NeRF methods can be categorized into three groups. **First**, some methods exploit continuity constraints on geometry or object semantics. For example, RegNeRF [37] imposed a continuity constraint on geometry and regularized the appearance of patches from unobserved viewpoints with a flow model. InfoNeRF [27] proposed a ray entropy minimization regularization to encourage the density to be as sparse as possible along a ray, and used a ray information gain reduction regularization to constrain the continuous depth of neighbor rays. **Second**, some methods attempt to pe-train a NeRF on other similar scenes and fine-tune the NeRF on the target scene. For example, PixelNeRF [64] conditioned a NeRF on image inputs in a fully convolutional manner, allowing the model to learn scene priors from other scenes and reduce the requirement of dense views. MVSNeRF [5] leveraged plane-swept cost volumes for geometry-aware scene reasoning and combined it with physically based volume rendering. Similar to Pixel-NeRF, MVSNeRF was first trained on other real scenes and was finetuned on target scenes to evaluate its effectiveness and generalizability. **Third**, depth-based models [17, 65] use available depth information to supervise the training of NeRFs. DSNeRF [17] exploited sparse 3D points generated by COLMAP [42] or accurate depth maps [13] obtained by high-accuracy depth scanners. Different from these depth-based models, SparseNeRF distills robust depth ranking from pre-trained depth models or coarse depth maps from consumer-level sensors. As concurrent works, NeRDi [16] and NeuralLift-360 [60] also use ranking-based methods. However, they mainly focus on the single-view setting while SparseNeRF focuses on sparse-view. SparseNeRF uses ranking loss to avoid inconsistent 3D geometry across different views while [16, 60] focuses on a soft geometric regularization on a single view (no cross-view consistency problem during training). Moreover, SparseNeRF introduces a new spatial continuity loss to distill spatial coherence from monocular depth estimators. In addition, some single-view synthesis methods [59] either allow new generative objects from unseen views or focus on specific objects, e.g., face [4] and human [23]. Some NeRF-based methods [22, 11] extended to generation instead of reconstruction. Other AIGC models [21, 51, 54, 67, 10, 9, 63, 29, 62] focus on the 2D generation or category-specific reconstruction [7], which are different.

3. Our Approach

We present SparseNeRF to synthesize novel views given sparse view inputs. Single-view depth estimation is a long-standing computer vision task, aiming to predict a depth map given a single image. In this paper, we are interested in mining the depth priors encapsulated in pre-trained models of single-view depth estimation or coarse depth maps captured by consumer-level depth sensors. However, due to coarse annotations of single-view depth maps, (*e.g.*, user clicks [6], RGB-D [43, 46], and laser/stereo [34]), dataset bias, and imperfect depth estimation models, it is challenging to obtain accurate 3D depth estimation given 2D single-view images. As for consumer-level depth sensors, it still struggles to capture accurate depth maps (**Figure 3**). Driven by these observations, our goal is to make use of the coarse depth maps and distill useful depth priors to guide the learning of a NeRF.

3.1. Preliminary and Problem Formulation

Neural Radiance Fields. Let a NeRF [36] be a mapping function f that maps a 3D spatial location $\mathbf{x}$ and a viewing direction $\mathbf{d}$ into a volume density σ and a color value $\mathbf{c}$. The f is a neural network consisting of eight perceptron (MLP) layers parameterized by θ, which is given by $f_\theta : (\gamma(\mathbf{x}), \gamma(\mathbf{d})) \mapsto (\sigma, \mathbf{c})$, where γ is a positional encoding. For each expected pixel color $\hat{C}(\mathbf{r})$, it is rendered by casting a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f. We evenly partition $[t_n, t_f]$ into N points $(t_1, t_2, ..., t_N)$ along a ray $\mathbf{r}$ and compute expected pixel color $\hat{C}(\mathbf{r})$ by $\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \mathbf{c}_i^*$. The weighted color $\mathbf{c}_i^*$ of a 3D point is computed by $\mathbf{c}_i^* = w_i \mathbf{c}_i$, where $w_i = T_i(1 - \exp(-\sigma_i \delta_i))$, $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ and $\delta_i = t_i - t_{i-1}$. Therefore, the NeRF reconstruction loss can be formulated as

$$\mathcal{L}_{\text{nerf}} = \sum_{\mathbf{r} \in R} ||\hat{C}(\mathbf{r}) - C(\mathbf{r})||^2, \qquad (1)$$

where $\hat{C}(\mathbf{r})$ are rendered color blended by N samples. $C(\mathbf{r})$ is the ground-truth pixel color. We use coarse-to-fine sampling as discussed in the vanilla NeRF [36]. Here we omit the fine rendering for simplification.

Problem Formulation. Vanilla NeRFs aim to learn a mapping function f_θ with K_d dense views by optimizing a color reconstruction loss in **Eq. (1)**. In this work, we aim to study few-shot NeRF when only K_s sparse views are available ($K_s \ll K_d$). Let H and W denote the height and width of an image, we have HWK_s rays as color reconstruction constraints. Without considering view directions $\mathbf{d}$ and image continuity assumptions (i.e., $\mathbf{c}_i^*$ is independent of $\mathbf{c}_j^*$ when $i \neq j$), suppose we have a $N_c \times N_c \times N_c$ discrete cubic volume to be optimized, which contains N_c^3 weighted color variables $\mathbf{c}_i^*$. We have N_c^3 variables $\mathbf{c}_i^*$ and HWK_s constraints. Ideally, we are able to solve a discrete cube of edge
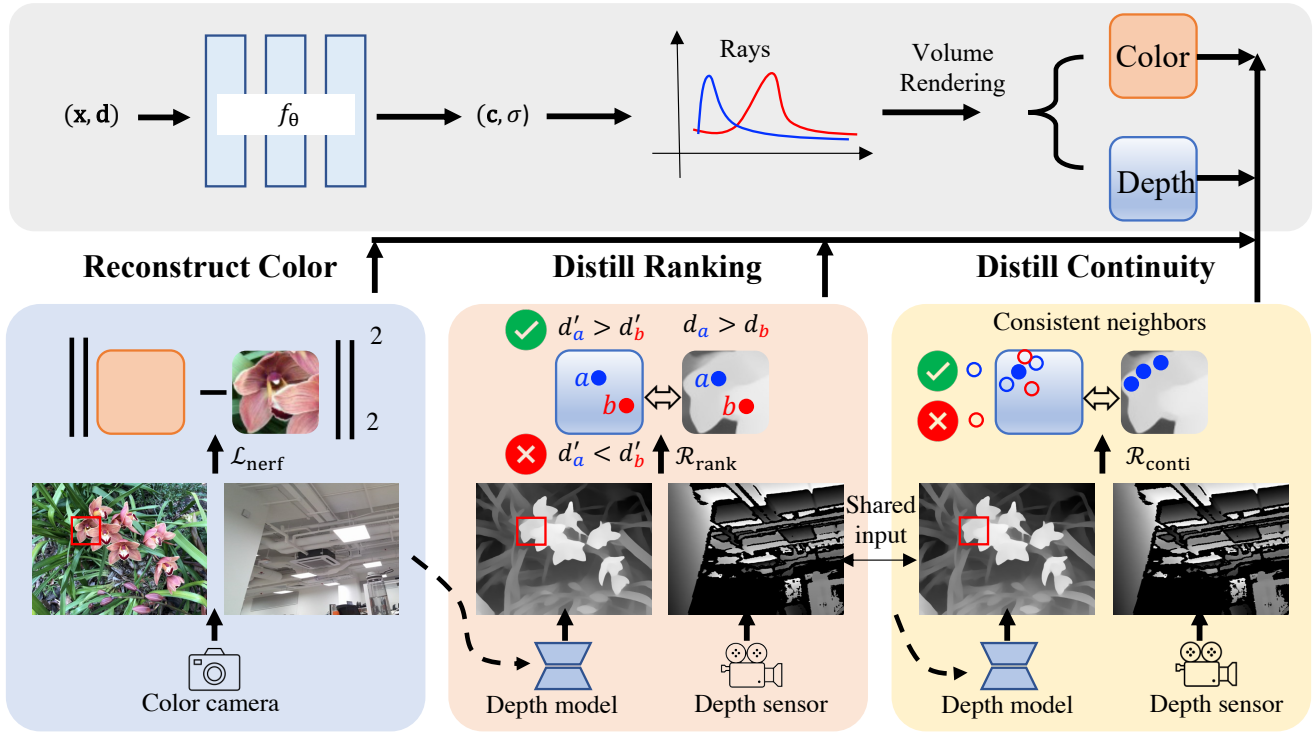
Figure 2: Framework Overview. SparseNeRF consists of two streams, i.e., NeRF and depth prior distillation. As for NeRF, we use Mip-NeRF as the backbone. we use a NeRF reconstruction loss $\mathcal{L}_{\mathrm{nerf}}$. As for depth prior distillation, we distill depth priors from a pre-trained depth model. Specifically, we propose a local depth ranking regularization and a spatial continuity regularization to distill robust depth priors from coarse depth maps.

length $N_c = \sqrt[3]{HWK_s}$. That is, we can sample $\sqrt[3]{HWK_s}$ for each edge of a cubic volume. Take $H = W = 512$ and $K_s = 3$ as an example, we have $N_c \approx 92$, which is far from reconstructing a continuous radiance field. To address this under-constrained optimization problem of few-shot NeRF, an intuitive way is to introduce reasonable regularization terms to constrain few-shot NeRFs conditioned on input $\mathbf{x}$, $\mathbf{d}$, and $C(\mathbf{r})$. Considering these constraints, a general formulation is given by

$$\mathcal{L} = \mathcal{L}_{\mathrm{nerf}} + \lambda \mathcal{R}(\mathbf{x}, \mathbf{d}, C(\mathbf{r})), \qquad (2)$$

where $\mathcal{R}$ is a regularization term.

Remark. RegNeRF [37] and InfoNeRF [27] tackles this problem by introducing regularization terms on continuous depth constraint from unobserved viewpoints, sparsity of density on rays, and patch-based semantic constraint on color appearance. Depth-based models, such as DSNeRF [17] and MonoSDF [65], directly regress depth by leveraging sparse 3D points or deriving ground-truth depth maps with accurate absolute depth maps. Different from them, we propose to use a robust relative depth regularization from coarse depth maps of pre-trained depth models or consumer-level depth sensors.

3.2. Overview of SparseNeRF

The pipeline of SparseNeRF is illustrated in **Figure 2**. SparseNeRF mainly consists of four components, i.e., a neural radiance field (NeRF), a color reconstruction module, a depth ranking distillation module, and a spatial continuity distillation module. Specifically, we use Mip-NeRF [1] as the backbone and apply an MSE loss for color reconstruction $\mathcal{L}_{\mathrm{nerf}}$. As for depth prior distillation, we either use a pre-trained depth model to estimate depth maps or capture coarse depth maps with consumer-level depth sensors. We use a vision transformer (DPT) [41] that is trained on a large-scale mixed depth dataset (1.4M images with various depth annotations) that covers a wide range of scenes. Therefore, DPT is able to provide general depth priors. Since single-view depth estimation is coarse, we carefully design a local depth ranking regularization and a spatial continuity regularization, which distills robust depth priors from coarse maps to the NeRF.

3.3. Local Depth Ranking Distillation

Single-view depth estimation is a challenging computer vision task, which aims at predicting a depth map of a scene given a single image as input. Due to dataset bias, coarse depth annotations, and imperfect neural models, it is difficult to achieve accurate depth prediction. Depth maps cap-
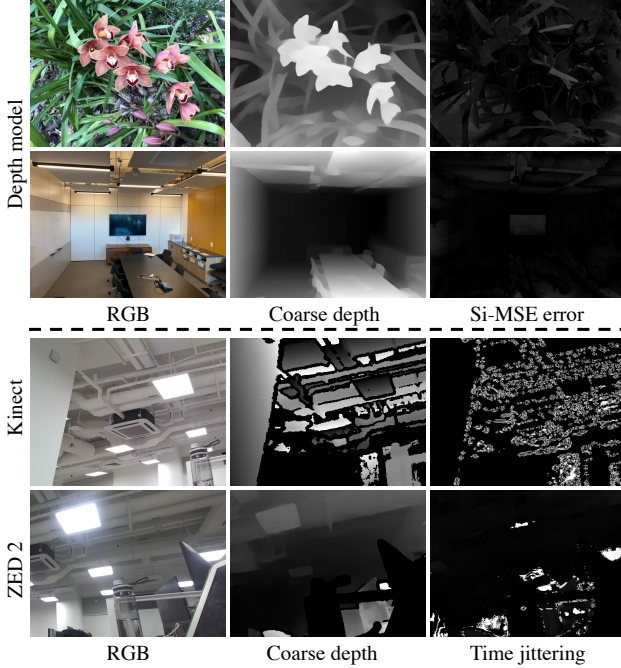
Figure 3: Two types of coarse depth maps. Top: predicted by a pre-trained depth model. Bottom: captured by Azure Kinect and ZED 2 depth sensors. Si-MSE is a scale-invariant depth error. Time jittering refers to the difference between two depth maps captured by a static camera within a small time interval.

tured by consumer-level sensors are also inaccurate due to wide-range depth distances. Directly using coarse depth maps to supervise a NeRF leads to ambiguous rendered novel view synthesis. To avoid the error of coarse depth maps, we relax the depth constraint and exploit the robust depth ranking prior. Given a pair of pixels in a single image, the depth ranking regularization only considers which point is nearer or farther. However, when the scene is complex, even depth ranking is not accurate. As shown in **Figure 4**, it is easy for depth estimation models to compare the depth ranking of white and cyan points, but it is hard to estimate the depth ranking of white and red points. It implies that the depth ranking becomes unreliable as the spatial distance increases.

Motivated by these observations, we propose a local depth ranking distillation method that distills depth ranking priors from coarse depth maps to a NeRF. On one hand, given a local patch P of an RGB image I with a pose p. We compute the depth $d_{\mathbf{r}}$ of the rays that trace from P by $d_{\mathbf{r}} = \sum_{i=1}^{N} w_i t_i$. On the other hand, we use pre-trained depth DPT [41] to estimate the depth of I and crop a local patch d_{dpt} with the same spatial location as $d_{\mathbf{r}}$. We perform the depth ranking distillation by transferring the depth ranking knowledge from d_{dpt} to $d_{\mathbf{r}}$. Let $k1$ and $k2$ be the

Figure 4: Motivation of local depth ranking. It is easy for depth estimation models to tell the white point is nearer than the cyan point, but it is hard to compare while and red points. Best viewed by zooming in.

indices of 2D pixel coordinate of d_{dpt} and $d_{\mathbf{r}}$. The depth ranking regularization is given by

$$\mathcal{R}_{\mathrm{rank}} = \sum_{d_{\mathrm{dpt}}^{k1} \leq d_{\mathrm{dpt}}^{k2}} \max(d_{\mathbf{r}}^{k1} - d_{\mathbf{r}}^{k2} + m, 0), \quad (3)$$

where m is a small margin that allows limited depth ranking errors. In **Eq. (3)**, we randomly sample two depth pixels of d_{dpt}, as denoted as d_{dpt}^{k1} and d_{dpt}^{k2}, $d_{\mathrm{dpt}}^{k1} \leq d_{\mathrm{dpt}}^{k2}$. If the depth rankings of d_{dpt} and $d_{\mathbf{r}}$ are not consistent, i.e., $d_{\mathbf{r}}^{k1} > d_{\mathbf{r}}^{k2}$ and $d_{\mathrm{dpt}}^{k1} \leq d_{\mathrm{dpt}}^{k2}$, we punish the NeRF. We encourage the corresponding depth estimated by NeRF to satisfy the consistent depth ranking of the pre-trained model. We clip the large depth value and normalize the depth map for depth sensors. For depth models, we use relative inverse depth.

3.4. Spatial Continuity Distillation

The local depth ranking constraint guarantees that the predicted depth map estimated by NeRF is consistent with the DPT's depth map. However, it does not constrain the spatial continuity of the depth map. We distill spatial continuity priors from the depth model DPT, which allows large displacement across several depth pixels. If neighbor depth pixels are continuous on the depth map of DPT, we constrain the corresponding depth pixels of NeRF to be continuous. The spatial continuity regularization is given by

$$\mathcal{R}_{\mathrm{conti}} = \sum_{k1} \sum_{d_{\mathrm{dpt}}^{k2} \in \mathrm{KNN}(d_{\mathrm{dpt}}^{k1})} \max(|d_{\mathbf{r}}^{k1} - d_{\mathbf{r}}^{k2}| - m', 0),$$

$$(4)$$

where $\mathrm{KNN}(\cdot)$ returns k-nearest neighbors measured by depth values within a small region, *e.g.*, 6×6 patch. m' is a small margin that allows small depth differences between neighbor pixels.

Full Objective Loss. The full objective loss of SparseNeRF is formulated by

$$\mathcal{L} = \mathcal{L}_{\mathrm{nerf}} + \lambda \mathcal{R}_{\mathrm{rank}} + \gamma \mathcal{R}_{\mathrm{conti}}, \quad (5)$$

where λ and γ control the importance of the regularization terms. In practice, we set $\lambda = 0.2$, $\gamma = 0.02$, $m = m' = 1.0 \times 10^{-4}$.

4. NVS-RGBD Dataset

The popular benchmark LLFF [35] on NeRFs is featured with forward-facing and object-centric. Moreover, we focus on consumer-level depth sensors instead of expensive scanners [14]. We study stand-alone depth cameras instead of a complex calibrated structured multiple camera system [15], and collect from real-world scenes instead of synthesized 3D assets [45]. Few RGBD datasets satisfy all of the requirements. Therefore, we collect a new dataset NVS-RGBD that contains real-world depth maps captured by consumer-level depth sensors, including Azure Kinect, ZED 2, and iPhone 13 Pro. We collect 8 scenes for both Azure Kinect and ZED 2, and 4 scenes for iPhone 13 Pro. The depth maps of Azure Kinect often contain noises around object edges. The depth maps of ZED 2 look smooth but unstable and incorrect when observing time jittering (**Figure 3**). The poses are computed via COLMAP. We collect indoor scenes for ZED 2 and Kinect. For iPhone, we collect both indoor and outdoor scenes. All the scenes are object-centric. Depth maps obtained from sensors have different artifacts coming from the sensor noises. Please refer to the supplementary material for the details.

5. Experiments

Datasets. We conduct experiments on the LLFF [35], DTU [26], and NVS-RGBD datasets. LLFF contains 8 complex forward-facing scenes. Following RegNeRF [37], we use every 8-th image as the held-out test set, and evenly sample sparse views from the remaining images for training. Different from LLFF, DTU is an object-level dataset. Following PixNeRF [64, 37], we use the same 15 scenes in our experiments. On the DTU dataset, the background of these scenes is a white table or a black background, which has few textures. As mentioned in RegNeRF, we mask the background during inference to avoid background bias. We use the same evaluation protocol for a fair comparison. On the NVS-RGBD dataset, for each scene, we use 3 views for training and the rest views for inference. For LLFF and DTU, we use pre-trained depth maps to implement our method. For NVS-RGBD, we use coarse depth maps captured by depth sensors.

Evaluation Metrics. We adopt four evaluation metrics in the experiments, i.e., PSNR, SSIM [55], LPIPS [66], and depth error [17]. Depth error is a scale-invariant MSE derived by minimizing $||w\hat{d} + b - d||$, where $\hat{d}$ and d denote coarse depth maps and pseudo-ground-truth depth maps predicted by dense-view NeRF. Note that we only use dense views to train a good NeRF to generate pseudo-ground-truth depth maps for evaluation.

Implementation Details. We implement SparseNeRF based on the official JAX [3]. We use the Adam optimizer for the learning of SparseNeRF. We use an exponen-

tial learning rate, which decays from 2×10^{-3} to 2×10^{-5}. The batch size is set to 4096. For each scene, we use one GPU-v100 with 32G memory for both training and inference. We also implement the proposed SparseNeRF on GPU-v100 with 16G and RTX 2080 Ti with 11G. We reduce the batch size to 2048 and 1024 for these two types of GPUs, respectively. We use the same backbone and sampled points along rays compared with prior arts.

We train 90k iterations for each scene given three training views, which takes about 2 hours for each scene. For depth maps captured by Kinect and ZED 2, we mask uncertain regions (black regions in depth maps). More precisely, we sample near-far depth pairs in certain regions in local patches to optimize **Eqs. 3** and **4** of the paper.

5.1. Comparisons on LLFF

We compare SparseNeRF with state-of-the-art methods on the LLFF dataset, including SRF [12], PixelNeRF [64], MVSNeRF [5], Mip-NeRF [1], DietNeRF [25], RegNeRF [37], DSNeRF [17] and MonoSDF[65]. Among them, SRF, PixelNeRF, and MVSNeRF are pre-trained on other similar scenes to exploit high-level semantics. Mip-NeRF is the state-of-the-art NeRF designed for dense-view training. InfoNeRF and RegNeRF mainly constrain sparsity and continuity of geometry or natural appearance semantics. More precisely, the compared methods can be classified into four groups. In the first group, SRF, PixelNeRF, and MVSNeRF are pre-trained in the large-scale DTU datasets (88 scenes) and are directly tested on the LLFF dataset. In the second group, SRF, PixelNeRF, and MVSNeRF are pre-trained on DTU and fine-tuned on LLFF per scene. The third group includes Mip-NeRF, DietNeRF, and RegNeRF. They uses geometry continuity and semantic constraints. The fourth group distills knowledge from depth maps or sparse points of COLMAP (DSNeRF). The compared results are shown in **Table 1**. We can see that the proposed SparseNeRF achieves the best performance in PSNR, SSIM, and LPIPS. Note that the first and second groups have to pre-train on lots of other scenes. SparseNeRF achieves better results than DSNeRF and MonoSDF because the depth ranking is more robust than the absolute scale-invariant constraint.

We provide qualitative analysis on LLFF, as shown in **Figure 5**. It is observed that PixelNeRF and MVSNeRF tend to generate ambiguous pixels. The reason is that they integrate CNN features into NeRFs. CNN features can provide high-level semantics to infer under-constrained textures but suffer pixel-level inference. MipNeRF is designed for dense-view synthesis without considering the under-constrained problem, leading to degraded geometry artifacts. Compared with RegNeRF, SparseNeRF uses robust depth priors from pre-trained depth models, which can handle challenging scene geometry with better performance.
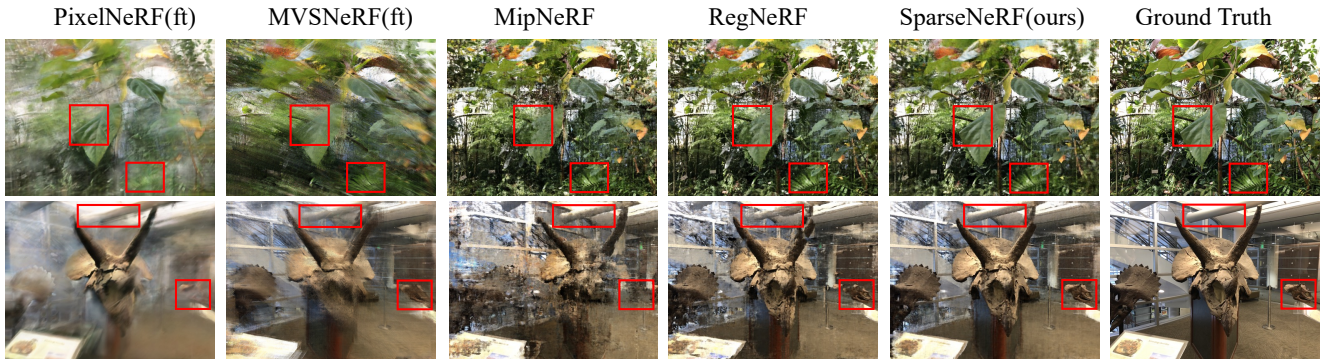
Figure 5: Visual comparisons on the LLFF dataset with three views. Red boxes denote compared regions. SparseNeRF achieves consistent improvement in different scenes.
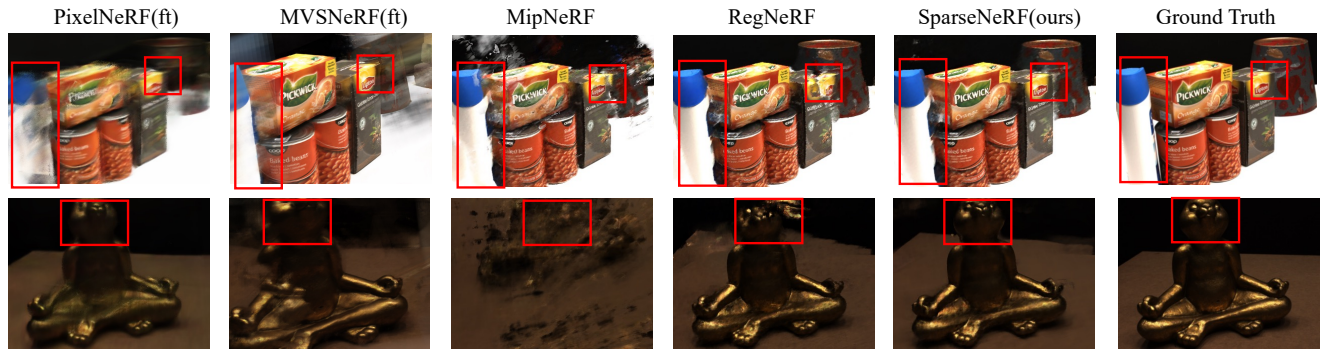

Figure 6: Visual comparisons on the DTU dataset with three views. Red boxes denote compared regions. SparseNeRF achieves consistent improvement in different scenes.

Table 1: Quantitative Comparison on LLFF with three views. There are four groups. The first group denotes models that are pre-trained on other scenes and tested on a target scene (P). The second group requires further finetuning on a target scene (P&FT). The third group uses geometry and semantic constraints (geo. & sem.). The fourth group distills depth knowledge (depth KD).

	Setting	PSNR ↑	SSIM ↑	LPIPS ↓
SRF [12]	P	12.34	0.250	0.591
PixelNeRF [64]		7.93	0.272	0.682
MVSNeRF [5]		17.25	0.557	0.356
SRF ft [12]	P&FT	17.07	0.436	0.529
PixelNeRF ft [64]		16.17	0.438	0.512
MVSNeRF ft [5]		17.88	0.584	0.327
Mip-NeRF [1]	geo. & sem.	14.62	0.351	0.495
DietNeRF [25]		14.94	0.370	0.496
RegNeRF [37]		19.08	0.587	0.336
MonoSDF*[65]	depth KD	18.45	0.565	0.388
DSNeRF[17]		18.94	0.582	0.362
SparseNeRF (Ours)		**19.86**	**0.624**	**0.328**

Table 2: Quantitative Comparison on DTU with three views. The first group is pre-trained on other scenes and tested on a target scene (P). The second group requires further finetuning on a target scene (P&FT). The third group uses geometry and semantic constraints (geo. & sem.). The fourth group distills depth knowledge (depth KD).

	Setting	PSNR ↑	SSIM ↑	LPIPS ↓
SRF [12]	P	15.32	0.671	0.304
PixelNeRF [64]		16.82	0.695	0.270
MVSNeRF [5]		18.63	0.769	0.197
SRF ft [12]	P&FT	15.68	0.698	0.281
PixelNeRF ft [64]		18.95	0.710	0.269
MVSNeRF ft [5]		18.54	0.769	0.197
Mip-NeRF [1]	geo. & sema.	8.68	0.571	0.353
DietNeRF [25]		11.85	0.633	0.314
RegNeRF [37]		18.89	0.745	**0.190**
MonoSDF* [65]	depth KD	18.92	0.748	0.237
DSNeRF [17]		16.90	0.570	0.450
SparseNeRF(Ours)		**19.55**	**0.769**	0.201

5.2. Comparisons on DTU

Similar to the LLFF dataset, we conduct four-group comparisons on the DTU dataset, as shown in **Table 2**. In this first group, given a target scene, SRF, PixelNeRF, and MVSNeRF are pre-trained on other DTU scenes and are tested on the target scene. In the second group, these three methods are further fine-tuned on the target scene. In the third group, we compare Mip-NeRF, DietNeRF, and RegNeRF. Note that for the pre-trained models, they split DTU as a training set (88 scenes) and a test set (15 scenes). Since

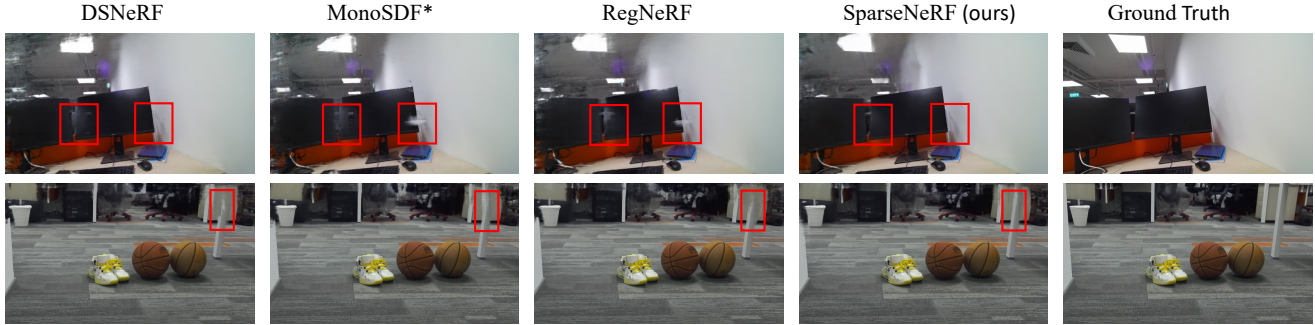| DSNeRF | MonoSDF* | RegNeRF | SparseNeRF (ours) | Ground Truth |

Figure 7: Visual comparisons on the NVS-RGBD dataset with three views. Red boxes denote compared regions. SparseNeRF achieves consistent improvement in different scenes.
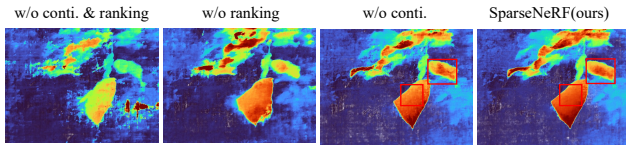

| w/o conti. & ranking | w/o ranking | w/o conti. | SparseNeRF(ours) |

Figure 8: Ablation study on visual effect. The three columns denote "w/o continuity and ranking distillation", "w/o continuity distillation", and "w/ continuity and ranking distillation" (SparseNeRF, ours), respectively.

Table 3: Comparison with State of the Arts on NVS-RGBD. * denotes a new implementation from related tasks.

	Metrics	PSNR↑	SSIM↑	LPIPS↓	depth err↓
Kinect	RegNeRF [37]	25.78	0.840	0.242	5.9×10^{-3}
	DSNeRF [17]	25.91	0.838	0.238	4.8×10^{-3}
	MonoSDF* [65]	25.50	0.838	0.239	5.5×10^{-3}
	SparseNeRF(ours)	**26.28**	**0.850**	**0.232**	**3.9×10^{-3}**
ZED 2	RegNeRF [37]	24.98	0.781	0.271	3.5×10^{-3}
	DSNeRF [17]	24.97	0.789	**0.262**	3.1×10^{-3}
	MonoSDF* [65]	24.54	0.784	0.275	3.2×10^{-3}
	SparseNeRF(ours)	**26.22**	**0.804**	**0.262**	**2.4×10^{-3}**

the training set and the test set contains similar scenes, a test scene shares a similar distribution (e.g., similar backgrounds) with training scenes. Pre-training on other scenes greatly benefits a target scene. Therefore, SRF, Pixel-NeRF, and MVSNeRF also greatly achieve promising results. Finally, we compare depth-based DSNeRF [17] and MonoSDF [65]. DSNeRF studies two kinds of depth information, i.e., sparse 3D points generated by COLMAP [42] and ground-truth depth maps obtained by high-accuracy depth scanners and the Multi-View Stereo (MVS) algorithm. The depth maps of NVS-RGBD are captured by consumer-level sensors and contain lots of noise, which is hard to be applied to DSNeRF. Therefore, we adopt sparse 3D points for DSNeRF. Compared with these methods, SparseNeRF still achieves the best performance without pre-training on other scenes. We provide a qualitative analysis, as shown in **Figure 6**. SparseNeRF achieves better visual results than previous state-of-the-art methods.

5.3. Comparisons on NVS-RGBD

We then implement three state-of-the-art methods on our new NVS-RGBD dataset, including RegNeRF [37], DSNeRF [17], and MonoSDF [65]. All of them are optimized per scene without pre-training on the other scenes. RegNeRF adopted continuous geometry assumptions from unobserved viewpoints. DSNeRF uses a global scale-invariant depth constraint to supervise NeRFs. MonoSDF mainly focuses on implicit surface reconstruction. It is as-

sumed that local patches of depth maps from monocular cameras satisfy $d' = wd + b$ where d is predicted by the pre-trained model and d' is then computed by the NeRF. In **Table 3**, we observe that SparseNeRF achieves better performance compared with these prior arts. We provide a qualitative analysis in **Figure 7**, demonstrating the superiority of SparseNeRF over previous state-of-the-art methods.

Table 4: Comparison of Depth Ranking and Depth scaling.

LLFF	PSNR↑	SSIM↑	LPIPS↓	depth err↓
baseline	19.08	0.587	0.336	10.1×10^{-3}
w/ depth scaling	18.45	0.565	0.388	8.9×10^{-3}
w/ depth ranking	**19.86**	**0.624**	**0.328**	**6.3×10^{-3}**

5.4. Ablation Studies and Further Analyses

Effectiveness of Depth Distillation. To validate the effectiveness of the depth distillation, we isolate the depth constraint and keep other modules unchanged. As shown in the first and third rows of **Table 4**, it is observed that without the local depth ranking distillation, SparseNeRF degrades on PSNR, SSIM, LPIPS, and depth error, which demonstrates the effectiveness of the depth distillation. We further study the depth maps from unobserved views. The ablation study of visual effect is shown in **Figure 8**. It shows that depth ranking greatly improves performance in geometry. Spatial continuity regularization further improves the detailed coherence of scenes.

Table 5: Impact of different pre-trained depth models.

LLFF	PSNR↑	SSIM↑	LPIPS↓	depth err↓
Baseline	19.08	0.587	0.336	10.1×10^{-3}
MiDaS small	19.35	0.592	0.333	6.9×10^{-3}
DPT Hybrid	19.86	0.625	0.326	6.3×10^{-3}
DPT Large	19.86	0.624	0.328	6.3×10^{-3}

Further Analyses on Depth Ranking and Depth Scaling. To validate the robustness of local depth ranking, we compare depth ranking regularization and depth scaling regularization on LLFF. We use the idea from MonoSDF [65], which mainly focuses on implicit surface reconstruction. With the specific design for surface shapes, MonoSDF slightly degrades the performance of RGB view synthesis. We re-implement it as depth scaling. Since the single-view depth estimation is not accurate, the linear depth scaling constraint is too strong such that depth distillation misleads the learning of the NeRF. **Figure 3** shows two cases of scale-invariant errors. Instead, depth ranking relaxes the constraint and only focuses on depth comparison, which is more robust than depth scaling. As shown in **Table 4**, depth ranking performs better than depth scaling on LLFF.

Impact of Different Pre-trained Depth Models. We test our method on three depth estimation models, i.e., MiDaS small, DPT Hybrid, and DPT Large. As shown in **Table 5**, all of the depth models are better than the baseline. DPT Hybrid and DPT Large achieve comparable results, which are better than MiDaS small.

FreeNeRF [61] potentially complements SparseNeRF. FreeNeRF and SparseNeRF improve few-shot NeRF in different ways. FreeNeRF introduces a frequency regularization while SparseNeRF distills depth priors from pre-trained general depth estimators. We implement SparseNeRF on FreeNeRF and the results significantly improve, as shown in **Table 6**.

Table 6: FreeNeRF potentially complement SparseNeRF.

LLFF	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	19.60	0.614	0.302
FreeNeRF+our depth distillation	**20.26**	**0.646**	**0.292**

User Study. To show that the proposed spatial continuity regularization improves the 3D consistency in 3D space, we conducted a user study. For each scene, we rendered videos for both with and without spatial continuity regularization terms. Users are asked to select the better-quality video according to 3D consistency/coherence and completeness (good geometry, few floaters). We evaluate 8 scenes on LLFF and there are 27 participants. The result shows 88.9% of users prefer "w/ conti" and 11.1% prefer "w/o conti". To evaluate the advantages of SparseNeRF on 3D consistency and shape completeness, we conduct another user study. Users are asked to evaluate "baseline+RegNeRF's continu-

ity term" and "baseline+our continuity term". The 27 users participated in the study and 8 scenes on LLFF are evaluated. The result shows 87.5% of users prefer "baseline+our continuity term" and 12.5% prefer "baseline+RegNeRF's continuity term". RegNeRF encourages the overall patch to be as smooth as possible without considering the sharp edges of objects, which conflicts with the reconstruction from sparse views. Our continuity regulation distills the neighbor relationship from depth models is more accurate. We show two examples of depth maps in **Figure 9**.
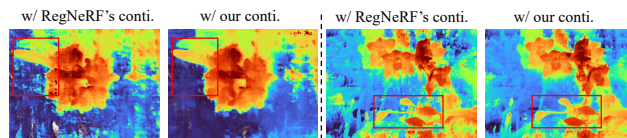

Figure 9: Visual comparison on continuity regulation.

Discussion. SparseNeRF could have a wide range of applications due to its promising results. We provide two applications on iPhone 13 Pro. Please refer to the project page for more experiments. Like UNISURF [38, 65], we also evaluate the SparseNeRF using Chamfer distance, showing the competitive results with MonoSDF and UNISURF, especially for complex scenes.

6. Conclusion

In this paper, we propose a SparseNeRF framework that synthesizes novel views with sparse view inputs. To tackle the under-constrained few-shot NeRF problem, we propose a local depth ranking regularization that distills the depth ranking prior from coarse depth maps. To preserve the spatial continuity of geometry, we propose a spatial continuity regularization that distills the depth continuity priors. With the proposed depth priors, SparseNeRF achieves a new state-of-the-art performance on three datasets.

Limitation. SparseNeRF significantly improves the performance of few-shot NeRF, but cannot be generalized to occluded views that are unobserved in the training views.

Potential Negative Impact. SparseNeRF focuses on scene synthesis. It might be misused to create misleading content.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srini-

vasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis.*, 2021. 1, 2, 4, 6, 7

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018. 6

[4] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3D generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 3

[5] 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 *Int. Conf. Comput. Vis.*, 2021. 2, 3, 6, 7

[6] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Adv. Neural Inform. Process. Syst.*, 2016. 3

[7] Xipeng Chen, Guangrun Wang, Dizhong Zhu, Xiaodan Liang, Philip Torr, and Liang Lin. Structure-preserving 3D garment modeling with neural sewing machines. *Adv. Neural Inform. Process. Syst.*, 2022. 3

[8] Zixuan Chen, Jianhuang Lai, Lingxiao Yang, and Xiaohua Xie. CuNeRF: Cube-based neural radiance field for zero-shot medical image arbitrary-scale super resolution. *Int. Conf. Comput. Vis.*, 2023. 2

[9] Zhaoxi Chen and Ziwei Liu. Relighting4D: Neural re-lightable human from videos. In *Eur. Conf. Comput. Vis.*, 2022. 3

[10] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Trans. Graph.*, 41(6):1–16, 2022. 3

[11] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Scene-dreamer: Unbounded 3D scene generation from 2D image collections. In *arXiv*, 2023. 3

[12] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 6, 7

[13] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016. 2, 3

[14] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 6

[15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 6

[16] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al.

[17] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 3, 4, 6, 7, 8

NerDI: Single-view NeRF synthesis with language-guided diffusion as general image priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[18] Chenhe Dong, Guangrun Wang, Hang Xu, Jiefeng Peng, Xiaozhe Ren, and Xiaodan Liang. Efficientbert: Progressively searching multilayer perceptron via warm-up knowledge distillation. *arXiv preprint arXiv:2109.07222*, 2021. 2

[19] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2022. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 2020. 3

[22] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. EVA3D: Compositional 3D human generation from 2D image collections. *arXiv preprint arXiv:2210.04888*, 2022. 3

[23] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. Sherf: Generalizable human NeRF from a single image. *arXiv preprint arXiv:2303.12791*, 2023. 3

[24] 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. 2

[25] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Int. Conf. Comput. Vis.*, 2021. 2, 6, 7

[26] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 6

[27] Mijeong Kim, Seonguk Seo, and Bohyung Han. InfoNeRF: Ray entropy minimization for few-shot neural volume rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2, 3, 4

[28] Changlin Li, Bohan Zhuang, Guangrun Wang, Xiaodan Liang, Xiaojun Chang, and Yi Yang. Automated progressive learning for efficient training of vision transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[29] Xiaoming Li, Wangmeng Zuo, and Chen Change Loy. Learning generative structure prior for blind text image super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[30] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 3

[31] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Int. Conf. Comput. Vis.*, 2021. 2

[32] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Adv. Neural Inform. Process. Syst.*, 2020. 1

[33] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. GNeRF: GAN-based neural radiance field without posed camera. In *Int. Conf. Comput. Vis.*, 2021. 2

[34] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 3

[35] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):1–14, 2019. 6

[36] 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 *Eur. Conf. Comput. Vis.* Springer, 2020. 1, 2, 3

[37] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Reg-NeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 3, 4, 6, 7, 8

[38] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Int. Conf. Comput. Vis.*, 2021. 9

[39] Jiefeng Peng, Jiqi Zhang, Changlin Li, Guangrun Wang, Xiaodan Liang, and Liang Lin. Pi-nas: Improving neural architecture search by reducing supernet training consistency shift. In *Int. Conf. Comput. Vis.*, 2021. 2

[40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 3

[41] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Int. Conf. Comput. Vis.*, 2021. 2, 4, 5

[42] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2, 3, 8

[43] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Eur. Conf. Comput. Vis.* Springer, 2012. 3

[44] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Adv. Neural Inform. Process. Syst.*, 2021. 1

[45] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 6

[46] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012. 3

[47] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[48] Tang Tao, Longfei Gao, Guangrun Wang, Peng Chen, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. LiDAR-NeRF: Novel lidar view synthesis via neural radiance fields. *arXiv preprint arXiv:2304.10406*, 2023. 2

[49] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable construction of large-scale NeRFs for virtual fly-throughs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[50] Guangrun Wang, Ping Luo, Xinjiang Wang, Liang Lin, et al. Kalman normalization: Normalizing internal representations across network layers. *Adv. Neural Inform. Process. Syst.*, 2018. 2

[51] Guangrun Wang and Philip HS Torr. Traditional classification neural networks are good generators: They are competitive with DDPMs and GANs. *arXiv preprint arXiv:2211.14794*, 2022. 3

[52] Guangrun Wang, Guangcong Wang, Keze Wang, Xiaodan Liang, and Liang Lin. Grammatically recognizing images with tree convolution. 2020. 2

[53] Guangrun Wang, Keze Wang, and Liang Lin. Adaptively connected neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[54] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. Stylelight: Hdr panorama generation for lighting estimation and editing. In *Eur. Conf. Comput. Vis.*, 2022. 3

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

[56] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF––: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[57] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 3

[58] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. CityNeRF: Building nerf at city scale. *arXiv preprint arXiv:2112.05504*, 2021. 2

[59] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. SinNeRF: Training neural radiance fields on complex scenes from a single image. *arXiv preprint arXiv:2204.00928*, 2022. 3

[60] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild

2D photo to a 3D object with 360deg views. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[61] Jiawei Yang, Marco Pavone, and Yue Wang. FreeNeRF: Improving few-shot neural rendering with free frequency regularization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 9

[62] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Vtoonify: Controllable high-resolution portrait video style transfer. *ACM Trans. Graph.*, 41(6):1–15, 2022. 3

[63] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Gp-unit: Generative prior for versatile unsupervised image-to-image translation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 3

[64] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2, 3, 6, 7

[65] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022. 1, 2, 3, 4, 6, 7, 8, 9

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 6

[67] Long Zhuo, Guangcong Wang, Shikai Li, Wanye Wu, and Ziwei Liu. Fast-vid2vid: Spatial-temporal compression for video-to-video synthesis. In *Eur. Conf. Comput. Vis.*, 2022. 3