NeXT: Towards High Quality Neural Radiance Fields via Multi-Skip Transformer

Yunxiao Wang[1⋆], Yanjie Li[1⋆], Peidong Liu[1], Tao Dai[2⋆⋆], and Shu-Tao Xia[13]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] College of Computer Science and Software Engineering, Shenzhen University
[3] Research Center of Artificial Intelligence, Peng Cheng Laboratory
{wang-yx20,lyj20,lpd19}@mails.tsinghua.edu.cn
daitao.edu@gmail.com, xiast@sz.tsinghua.edu.cn

Abstract. Neural Radiance Fields (NeRF) methods show impressive performance for novel view synthesis by representing a scene via a neural network. However, most existing NeRF based methods, including its variants, treat each sample point individually as input, while ignoring the inherent relationships between adjacent sample points from the corresponding rays, thus hindering the reconstruction performance. To address this issue, we explore a brand new scheme, namely *NeXT*, introducing a multi-skip transformer to capture the rich relationships between various sample points in a ray-level query. Specifically, ray tokenization is proposed to represent each ray as a sequence of point embeddings which is taken as input of our proposed NeXT. In this way, relationships between sample points are captured via the built-in self-attention mechanism to promote the reconstruction. Besides, our proposed NeXT can be easily combined with other NeRF based methods to improve their rendering quality. Extensive experiments conducted on three datasets demonstrate that NeXT significantly outperforms all previous state-of-the-art work by a large margin. In particular, the proposed NeXT surpasses the strong NeRF baseline by **2.74** dB of PSNR on Blender dataset. The code is available at `https://github.com/Crishawy/NeXT`.

Keywords: View Synthesis, Neural Representation, Scene Representation, 3D Deep Learning

1 Introduction

Novel View Synthesis (NVS) aims to render a scene from unobserved viewpoints with a set of images and camera poses as input. Recently, Neural Radiance Field (NeRF) [33] and its variants have demonstrated compelling performance in this task by representing a scene's geometry and appearance with a Multi-Layer Perceptron (MLP). To render each pixel in an output image, NeRF adopts

⋆ Equal contribution.
⋆⋆ Corresponding author.

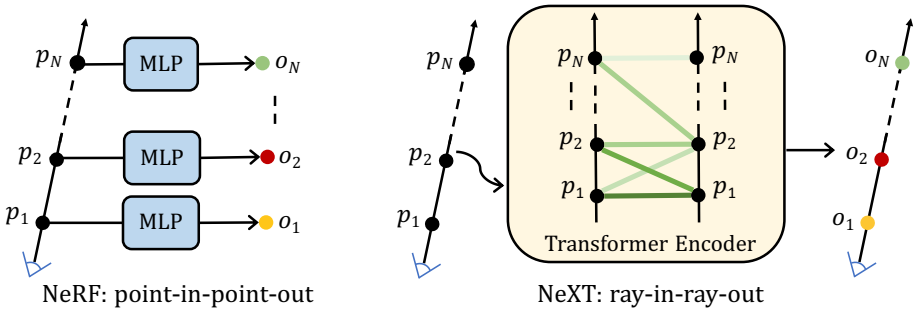

Fig. 1: NeRF (left) renders scenes by individually point-level query of a MLP, *i.e.*, one point in and one point out, which ignores the relations between sample points. NeXT (right) regards the entire ray as network input to make ray-level query, *i.e.*, one ray in and one ray out, to capture the intrinsic dependencies between sample points via self-attention mechanism built-in Transformer encoder.

volume rendering to combine the RGB colors and volume densities from many points sampled along the corresponding 3D ray.

Despite the success of NeRF based methods in novel view synthesis, most existing methods adopt MLP as the function approximator to render scenes in a point-level way. As shown in Fig. 1 (left), NeRF queries the MLP network with one sample point from the corresponding 3D ray. However, such point-level query ignores the *inherent relations* in sample points from the same rays, which deeply imprisons the potential of NeRF.

To exploit such relations among sample points, we attempt to explore a transformer-based paradigm, named NeXT, to achieve ray-level query. The proposed NeXT regards an individual ray, *i.e.*, the whole sampled points set, as the network input, as shown in Fig. 1 (right). In this way, the intrinsic relations between sampled points are captured by self-attention mechanism built in transformer, which help enrich the scene properties information for rendering. To further improve the performance, we propose a multi-skip connection module to better utilize the position information of the sampled points. The network architecture of our proposed NeXT is illustrated in Fig. 2.

In contrast to the previous NeRF-based methods, the proposed NeXT has several benefits for novel view synthesis. **First**, our method renders pixels in a ray-level way to exploit the relations among sample points. **Second**, benefiting from the ray-level query and the captured relationships between sample points, our approach shows much less dependence on the two-stage coarse-to-fine sampling. Even with only one-stage coarse sampling, our approach achieves competitive results compared to the two-stage NeRF. **Third**, the proposed NeXT significantly benefits from scaling up the model capacity, which may provide a promising path towards high quality view synthesis.

The main contributions can be summarized as follows:

– We propose a novel transformer-based paradigm, called NeXT, to realize ray-level query for novel view synthesis. Benefiting from that, inherent relations between sample points along a ray are captured to promote reconstruction.
– We propose a multi-skip connection module to improve the model performance, which enriches the original positional information from sample points.
– Comprehensive experiments conducted over Blender [33], DeepVoxels [48], and multiscale Blender [2] demonstrate that our proposed NeXT outperforms all previous state-of-the-art methods by a large margin.

2 Related Work

Scene Representations for View Synthesis The view synthesis task aims to represent a scene using a set of observed images and camera poses for rendering novel photorealistic images from unobserved viewpoints. With densely captured images, methods based on light field interpolation [11, 24, 20] tackle this task without reconstructing an intermediate representation of the scene. By contrast, when images of the scene are sparsely-captured, explicit representations of the scene's 3D geometry and appearance usually tend to be reconstructed. A line of popular view synthesis methods use mesh-based representations along with either diffuse [59] or view-dependent appearance [6, 12, 62], consisting of classical [6, 12, 59] and learning-based [44, 45] ones. Mesh-based methods demonstrate advantage in storage and compatibility with graphics rendering pipelines. Nevertheless, gradient-based mesh representation methods are typically hard due to local minima or the poor conditioning of loss landscape.

Another line of methods consider volumetric representations for view synthesis. In the early stage, volumetric methods directly color voxel grids given some observed images [47]. More recent approaches tend to train deep neural networks for the purpose of predicting voxel representations of scenes [16, 29, 31, 48, 70, 52]. Different from mesh-based methods, adopting gradient-based learning to optimize volumetric approaches is natural and well-suited. In addition, volumetric approaches can realistically represent complex shapes and materials, yield less artifacts, thus become increasing popular. While discrete voxel-based methods have demonstrate impressive performance for novel view synthesis, they are typically restricted at higher resolutions.

A promising trend is to adopt neural function representations to alleviate the limitation of discrete voxel grids [33, 36, 49, 65]. Among those, volumetric NeRF [33] representation has recently raised dramatically increasing attention, which uses a continuous function parameterized by MLP to map 3D coordinates and viewing directions to volumetric densities and color values. NeRF has inspired various subsequent extensions under varying settings, including dynamic scenes [26, 37, 39], limited training views [18, 40, 43, 54, 60, 67, 22, 57], generative modeling [8, 46, 35], non-rigidly deforming objects [17, 38], speed-up [28, 42, 19, 21, 27, 34, 41, 66] and reflectance modeling for relighting [3, 4, 51].

Despite the success of NeRF and its follow-ups, little attention has been paid to exploit the relations between sample points along rays. NeRF renders

a scene by point-level query, which lacks of consideration about the inherent relationships thus leads to suboptimal results. NeXT addresses this issue, enabling ray-level query and points relationships modeling by introducing a novel multi-skip Transformer-based paradigm for novel view synthesis.

Transformers Transformers [58] were first proposed for machine translation, and have since revolutionized many natural language processing tasks [10, 14, 58]. Very recently, Transformer-based methods make impressive strides in computer vision tasks [15, 56, 68, 25, 64, 69, 53, 63, 7, 9, 71], including image classification [15, 56, 68], semantic segmentation [53, 63] and object detection [7, 9, 71].

As far as we know, Transformer in NeRF is explored in NeRF-ID [1] and IBRNet [60]. NeRF-ID aims to learn to propose samples via a differentiable module (e.g., Transformer, Pool, MLPMix [55]), while still remains point-level query. IBRNet [60] focuses on learning a generic view interpolation function that generalizes to novel scenes, where a CNN is critical and color prediction follows point-level query. By contrast, NeXT is a pure Transformer-based paradigm to predict both color and density via ray-level query, proposing multi-skip connection to enrich position information for high quality renderings.

3 Method

Our proposed method is built upon NeRF [33, 13], and can be easily expanded to other follow-ups. In this section, we first revisit the original design of NeRF, and then describe the details of proposed NeXT.

3.1 Background

NeRF [33] represents a scene by an Multilayer Perceptron (MLP), which takes as input a 3D position $\mathbf{x}$ and viewing direction $\mathbf{d}$ and output the corresponding color $\mathbf{c}$ and density σ. To promote the learning of high-frequency details, $\mathbf{x}$ and $\mathbf{d}$ are transformed via a positional encoding γ as the pre-processing.

In NeRF, a pixel is rendered by querying an MLP of n sample points $\mathbf{x}_1, ..., \mathbf{x}_n$ along a ray which connects the camera center with the target pixel. The query process is operated point by point. After that, a set of color values $\mathbf{c}_i$ and density values σ_i is obtained. The final pixel color $\hat{\mathbf{c}}$ can be calculated by:

$$\hat{\mathbf{c}} = \sum_{i=1}^{n} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \text{ where } T_i = \exp(-\sum_{j=1}^{i-1}\sigma_j\delta_j), \tag{1}$$

where $\delta_i = ||\mathbf{x}_{i+1} - \mathbf{x}_i||$ is the distance between adjacent samples and T_i represents the transmittance along the ray.

To improve the sampling efficiency, NeRF propose a coarse-to-fine strategy. In the coarse stage, NeRF obtains N_c evenly-spaced random points with stratified sampling. Given the output of "coarse" network, a piece-constant PDF along the ray is then produced to describe the distribution of the visible scene content. In

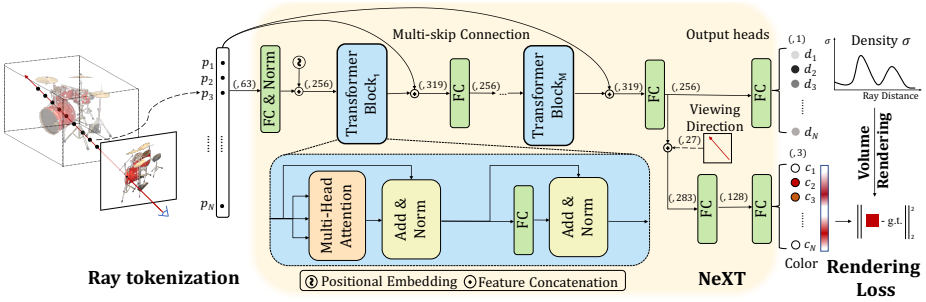

Fig. 2: Overall framework of the proposed NeXT, which consists of three components, *i.e.*, Transformer blocks, multi-skip connection module and output heads for generating density σ and color RGB $\mathbf{c}$. First, NeXT uses *ray tokenization* to generate a set of point embeddings from N sample points along the ray. The obtained embeddings are then fed into the subsequent M *Transformer blocks* to learn relationships between points via self-attention. Note that the *multi-skip connection* is adopted to enrich the original positional information after each Transformer block. Finally, the N output point representations are utilized to yield the scene properties by *output heads* and L_2 reconstruction loss between groundtruth and rendered pixels is adopted for network training.

the fine stage, N_f new points are then obtained based on the PDF using inverse transform sampling. Consequently, the resulting union of these $N_c + N_f$ sample points are sorted and passed to the "fine" network to yield final rendered pixel color. This hierarchical sampling allocates more samples to informative regions.

To render an image with $W \times H$ pixels, the MLP in NeRF is queried $W \times H \times n$ times. The network parameters are optimized by minimizing an L_2 reconstruction loss between the ground-truth and corresponding rendered pixels. For more details, readers may refer to the original NeRF paper [33].

3.2 NeXT

Different from NeRF [33], NeXT aims to capture the inherent relationships between sample points along the same ray and provides a ray-level query paradigm, which is accomplished by the proposed multi-skip Transformer-based network. An overview of our method is shown in Fig. 2.

Ray tokenization. To achieve ray-level query, we first expand the vanilla Transformer to serve as the function approximator instead of MLP used in NeRF. To handle a ray, N points are sampled, resulting in an input sequence length of N for the Transformer. The 3D position information of these points are transformed to a higher dimension space using high frequency functions in [33], and then mapped to D dimensions with a learnable linear projection. In addition, D is also the latent vector sizer of Transformer layer. In this paper, we refer the output of this trainable projection as the point embeddings. Point embeddings along the ray here play the same role as word tokens in NLP community.

Positional embedding. A mapping similar to positional encoding γ is used in the popular Transformer [58] architecture, which is called *positional embedding* to avoid confusion in this paper. Positional embeddings are added to the point embeddings to provide the order information, following the standard Transformer [58]. In this paper, we use 1D sinusoidal positional embeddings by default. We also study the effect of different types of positional embeddings for NeXT later in Table 4a. The generated sequence of embeddings serve as the input to the subsequent Transformer encoder.

Query network. To achieve ray-level rendering, NeXT adopts the Transformer encoder as the query network to represent a scene. Specifically, the encoder learns point feature representation by stacking M blocks, given the 1D point token embeddings sequence as input. Each Transformer block consists of a multi-head self-attention (MSA) module and a multilayer perception (MLP) module. Self-attention is the core mechanism of Transformer and adopted in this work for capturing relationships between sample points along the same ray.

Local-window self-attention. To alleviate the computation cost of self-attention, we divide the point embeddings $\mathbf{X} \in \mathbb{R}^{N \times D}$ into a set of non-overlapping small windows: $\mathbf{X} \rightarrow \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_W\}$. Each window covers L points in order. Then the multi-head self-attention is performed within each window independently. The multi-head self-attention within the i-th window is calculated as:

$$\mathrm{MSA}(\mathbf{X}_i) = \mathrm{SA}(\mathbf{X}_i)_1 \oplus \mathrm{SA}(\mathbf{X}_i)_2 \oplus ... \oplus \mathrm{SA}(\mathbf{X}_i)_H, \tag{2}$$

$$\mathrm{SA}(\mathbf{X}_i)_h = \mathrm{Softmax}[\frac{(\mathbf{X}_i\mathbf{W}_q^h)(\mathbf{X}_i\mathbf{W}_k^h)^{\mathrm{T}}}{\sqrt{D/H}}]\mathbf{X}_i\mathbf{W}_v^h, \tag{3}$$

where $\mathbf{W}_q^h \in \mathbb{R}^{\frac{D}{H} \times D}$, $\mathbf{W}_k^h \in \mathbb{R}^{\frac{D}{H} \times D}$, $\mathbf{W}_v^h \in \mathbb{R}^{\frac{D}{H} \times D}$ for $h \in \{1, ..., H\}$ are learnable parameters of three linear projection layers. H represents the number of heads and $\oplus$ means concatenation. Compared to global self-attention, local-window reduces the computational complexity from $O(N^2)$ to $O(LN)$, which is of linear complexity with the number of sample points.

Multi-skip connection. Vanilla Transformer performs unsatisfactorily for rendering. Inspired by NeRF, we propose a multi-skip connection module to strengthen the utilization of position information from the sample points, which is shown in Fig. 2. Consequently, the input of j-th Transformer Block is obtained by:

$$\mathbf{X}_{in}^j = \mathrm{FC}(\mathbf{X}_{out}^{j-1} \oplus \gamma(\mathbf{X}^0)), \text{ where } j = 2, ..., M, \tag{4}$$

where $\mathbf{X}_{out}^{j-1}$ is the output of $(j-1)$-th block, FC means a fully-connected layer to map the input to D dimensions, $\mathbf{X}^0$ is the original 3D positions of sample points, γ is the positional encoding pre-processing.

Output heads. The output of the Transformer encoder serves as feature representations to yield the final network output, *i.e.*, the scene properties of samples. To output the density σ and RGB color $\mathbf{c}$, a single linear layer and a two-layers MLP are attached as heads, respectively. It's worth noting that the color head takes as input both of feature representations and the viewing direction vectors.

Architecture variants. We introduce our small model with parameters similar to NeRF, denoted as NeXT-S. To achieve high quality rendering results, we further scale up the model and build NeXT-B and NeXT-L, which are variants of about 1.7× and 3.3× model size, respectively. Note that the number of heads is $H = 8$ by default. The architecture hyper-parameters of these three variants are: $D = 192, 256, 256$, $M = 2, 2, 4$ for NeXT-S, NeXT-B and NeXT-L, respectively.

Compared to the MLP in NeRF, the Transformer in NeXT is only queried $W \times H$ times to render an image with $W \times H$ pixels. And L_2 reconstruction loss is also adopted for network training, following NeRF [33].

3.3 Integration with NeRF methods

The proposed NeXT serves as the function approximator to achieve ray-level query and relationships modeling for novel view synthesis task, which can be regarded as a substitute of MLP used by NeRF and most follow-ups. Given its simplicity and effectiveness, it's easy and convenient to improve various existing NeRF methods by our proposed NeXT. In this paper, the original NeRF and the previous state-of-the-art Mip-NeRF [2] are chosen as the examples to show the superiority of proposed NeXT.

4 Experiment

Note that the proposed NeXT can be easily incorporated into various NeRF methods to serve as the query network, achieving higher quality rendering results. In this section, we show the examples of integrating NeXT with the original NeRF on Blender as well as DeepVoxels dataset, and Mip-NeRF on multiscale Blender dataset, respectively. The key lies in replacing their query strategy by our proposed NeXT. Our implementation is built based on JAX [5].

4.1 Setup

To verify the effectiveness of proposed method, comprehensive experiments are conducted on three popular datasets, *i.e.*, Blender [33], DeepVoxels [48] and multiscale Blender dataset [2]. We report the average peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [61] metric, which is widely used by NeRF-based methods [33, 2, 42, 1]. Following NeRF [33] and Mip-NeRF [2], our method is trained with batch size 4096 for 1 million iterations. The Adam [23] optimizer with cosine learning rate decay from 5×10^{-4} to 5×10^{-6} is used for optimization. We set $N_c = 128$ and $N_f = 128$ for coarse and fine stage, respectively. We adopt local-window self-attention with $L = 64$ for Blender and multiscale Blender dataset, and global self-attention for DeepVoxels dataset. Besides, we reimplement NeRF based on JAX as a stronger baseline.

Table 1: **PSNR comparisons on Blender dataset.** "*" means adopting center pixel [2] which generates rays through the center of each pixel. NeXT variants surpass previous state-of-the-art methods.

	#Params	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
SRN [50]	-	26.96	17.18	20.73	26.81	20.85	18.09	26.85	20.60	22.26
NV [30]	-	28.33	22.58	24.79	30.71	26.08	24.22	27.78	23.93	26.05
LLFF [32]	-	28.72	21.13	21.79	31.41	24.54	20.72	27.48	23.22	24.88
NSVF [28]	3.2M-16M	33.19	25.18	31.23	37.14	32.29	32.68	34.27	27.93	31.74
NeRF [33]	1,191K	33.00	25.01	30.13	36.18	32.54	29.62	32.91	28.65	31.01
NeRF (JAX) [13]	1,191K	34.06	25.13	30.48	36.87	33.33	29.94	34.66	28.77	31.66
vanilla Trans.	1,889K	33.59	24.97	30.31	36.25	33.38	31.92	33.13	27.99	31.44
NeXT-S	1,232K	33.75	25.34	32.62	37.42	34.52	32.09	33.74	29.25	32.34
NeXT-B	2,152K	34.70	25.79	33.77	38.10	35.67	32.48	34.46	30.07	33.13
NeXT-L	4,062K	36.05	26.32	35.30	38.27	36.78	34.06	35.19	30.35	34.04
NeXT-L*	4,062K	**36.37**	**26.49**	**35.67**	**38.46**	**37.39**	**34.16**	**35.96**	**30.73**	**34.40**

Table 2: **Comparisons on DeepVoxels dataset.** "*" means adopting center pixel [2]. "NA" represents that the results fail to converge after repeating experiments over five times. By contrast, NeXT converges stably and quantitatively outperforms previous state-of-the-art methods over all scenes by a large margin.

	Chair	Pedestal	Cube	Vase	Avg.
	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
DeepVoxels [48]	33.45 / 0.99	32.35 / 0.97	28.42 / 0.97	27.99 / 0.96	30.55 / 0.97
SRN [50]	36.67 / 0.982	35.91 / 0.957	28.74 / 0.944	31.46 / 0.969	33.20 / 0.963
NV [30]	35.15 / 0.980	36.47 / 0.963	26.48 / 0.916	20.39 / 0.857	29.62 / 0.929
LLFF [32]	36.11 / 0.992	35.87 / 0.983	32.58 / 0.983	32.97 / 0.983	34.38 / 0.985
NeRF [33]	42.65 / 0.991	41.44 / 0.986	39.19 / 0.996	37.32 / 0.992	40.15 / 0.991
NeRF (JAX) [13]	44.97 / 0.994	43.74 / 0.992	42.43 / 0.998	NA / NA	NA / NA
NeXT-S	47.53 / 0.995	45.57 / 0.994	47.98 / 0.999	42.72 / 0.997	45.95 / 0.996
NeXT-B	48.20 / 0.996	47.04 / 0.995	48.44 / 0.999	44.61 / 0.998	47.07 / 0.997
NeXT-L	48.73 / 0.997	48.81 / 0.997	49.23 / 0.999	44.98 / 0.998	47.94 / 0.998
NeXT-L*	**50.43** / **0.998**	**50.60** / **0.998**	**51.55** / **0.999**	**46.36** / **0.999**	**49.74** / **0.999**

4.2 Quantitative experiments

Blender dataset. Table 1 shows the PSNR results of our proposed NeXT, and SSIM results are shown in the appendix. Compared to NeRF, our NeXT-S obtains **0.68** PSNR gain with similar parameters. Our methods remarkably benefit from increasing the model capacity. NeXT-L significantly outperforms NeRF by **2.38** PSNR. Especially, when adopting center pixel [2], our NeXT-L* achieves a new state-of-the-art result, crossing **34** average PSNR threshold on Blender dataset for the first time. It's worth noting that vanilla Transformer performs worse ($\downarrow$0.20 PSNR) even with more parameters than NeRF.

DeepVoxels dataset. Table 2 presents the results of our method and other state-of-the-art methods on DeepVoxels dataset, which has about 5$\times$ more number of training images than Blender dataset (479 for training and 1000 for test-

Table 3: **Comparisons on multiscale Blender dataset.** NeXT boosts Mip-NeRF by a clear margin, especially on low resolution scenes.

	#Params	Full Res. PSNR / SSIM	1/2 Res. PSNR / SSIM	1/4 Res. PSNR / SSIM	1/8 Res. PSNR / SSIM
NeRF (JAX) [13]	1,191K	31.20 / 0.950	30.65 / 0.956	26.25 / 0.930	22.53 / 0.871
Mip-NeRF [2]	612K	32.63 / 0.958	34.34 / 0.970	35.47 / 0.979	35.60 / 0.983
NeXT-S	616K	32.18 / 0.954	34.32 / 0.969	36.43 / 0.980	37.57 / 0.987
NeXT-B	1,076K	32.92 / 0.959	35.06 / 0.973	36.99 / 0.982	38.05 / 0.988
NeXT-L	2,031K	**34.38** / **0.968**	**36.47** / **0.979**	**38.19** / **0.986**	**39.29** / **0.991**

ing). As Table 2 shown, all of our NeXT variants outperform previous top-performed methods by a large margin. We also reimplement NeRF based on JAX to serve as a stronger baseline, which brings consistent improvement over the original NeRF for most scenes yet fail to converge for "Vase". By contrast, the proposed NeXT converges stably and achieves best performance on all scenes. In particular, our best model NeXT-L* achieves new state-of-the-art results (↑**9.59** PSNR compared to NeRF) on DeepVoxels dataset.

Multiscale Blender dataset. Multi-scale Blender dataset [2] is designed to better probe accuracy on multi-resolution scenes, which is much more challenging than NeRF's Blender dataset [33]. Mip-NeRF [2] shows impressing superiority over NeRF and previously serves as the state-of-the-art method in this dataset. Hence, we choose Mip-NeRF as the baseline and replace its query network by NeXT. As shown in Table 3, with similar parameters, our NeXT-S boosts Mip-NeRF by **0.62** PSNR gain on average. By further increasing the model capacity, our NeXT-L shows consistent improvement on all resolution scenes and surpasses the previous best-performed Mip-NeRF by a clear margin (↑**2.57** PSNR). Note that the model sizes of our variants are cut in half here by following Mip-NeRF.

4.3 Ablation studies

Model scaling. A natural question is that whether the original NeRF can benefit from model scaling like our proposed NeXT? To answer this question, we conduct comparison experiments between NeRF and NeXT under different model parameters and GFLOPs as shown in Fig. 3. The MLP used in the original NeRF has 8 fully-connected layers with 256 channels per layer. To explore its potential, we increase the model size of MLP by making it deeper and wider. Fig. 3 shows that NeRF does benefit from increasing network capacity yet the gain is limited and overfitting tend to occur. (i) As the MLP goes deeper, the performance of NeRF increases first and then drops when has 20 layers. (ii) As the MLP becomes deeper and wider, it seems that saturation occurs: NeRF-D16-W384 with 4,988K parameters brings no gain than that NeRF-D8-W384 with 2,624K parameters. By contrast, our proposed NeXT variants show strong scalability, *i.e.*, tendency of saturation is not observed by increasing model capacity from NeXT-S to NeXT-L (↑**1.34** PSNR). When increasing the number of

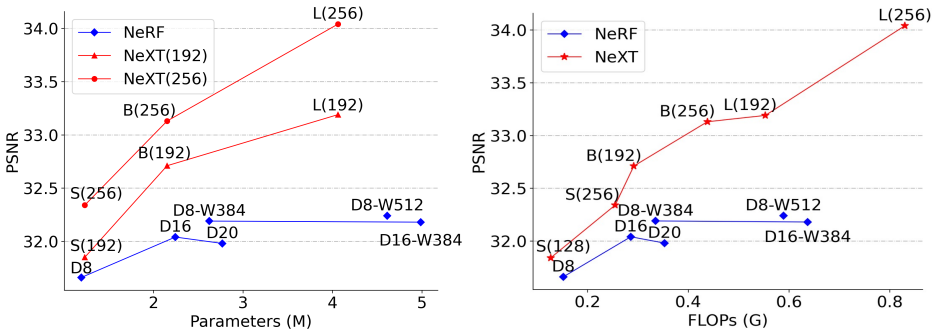

Fig. 3: **Model scaling**. PSNR results are achieved by NeRF and NeXT under different model parameters and FLOPs on Blender dataset. D and W indicate the depth and width of the MLP network in NeRF. The value in the parentheses denotes $N_c + N_f$ and are set to 192 for NeRF by default. NeXT variants show strong scalability of model capacity and significantly benefit from the increase of sample points. Note that FLOPs is calculated to measure the computation cost of rendering a pixel via the query network.

sample points from 192 to 256, NeXT-L achieves further improvement with **0.85** PSNR gain. Besides, our NeXT variants show consistent improvement compared to NeRF with similar FLOPs.

Positional embedding in Transformer. To illustrate the effect of positional embedding, we conduct experiments with different positional embedding types (*i.e.*, no positional embedding, 1D sinusoidal and learnable positional embedding) on Blender dataset. As Table 4a shown, employing 1D sinusoidal positional embedding significantly improves the performance by 0.53 PSNR at most.

Local-window self-attention. The computation cost of global attention increase squarely with enlarging the number of sample points along a ray. Hence, it's of significance to introduce local-window self-attention to alleviate the computation burden. Table 4b shows the results of NeXT with different local window size, which demonstrates that employing local-window self-attention achieves competitive performances compared to using global attention mechanism.

Multi-skip connection. Table 4c shows both NeRF and NeXT benefit from the utilization of multi-skip connection, which demonstrates that the enhancement of position information is vital for novel view synthesis. For example, multi-skip connection brings **0.24** and **1.00** PSNR gains to NeRF and NeXT, respectively.

Hierarchical sampling. We conduct experiments on the hierarchical sampling shown in Table 4d and draw conclusions as follows. First, the proposed NeXT relies much less on the two-stage coarse-to-fine sampling compared to NeRF. When only adopts coarse sampling, *i.e.*, $N_f = 0$, the resulted performance degradation is $\downarrow$0.32 vs. $\downarrow$1.69 PSNR (NeXT vs. NeRF). Second, our approach prominently benefits from more sample points. Increasing the total number of sample points

Table 4: **Ablation studies.** If not otherwise specified, all the ablation studies are performed on NeXT-B on Blender dataset. "LWSA" denotes local-window self-attention. N_c/N_f refer to the number of sample points in coarse/fine stage.

(a) **Positional embedding in Transformer.** The 1D sinusoidal positional embedding strategy performs better.

Positional Embed.	PSNR	SSIM
✗	32.19	0.955
Learnable	32.44	0.956
Sinusoid	**32.72**	**0.958**

(b) **Attention design.** NeXT with LWSA reduces computational complexity and maintains competitive performance.

Method	Window Size	PSNR	SSIM
w/o LWSA	-	32.25	0.955
w/ LWSA	64	**32.34**	**0.955**
	32	32.26	0.954

(c) **Multi-skip connection.** Multi-skip connection boosts both the proposed NeXT and original NeRF.

Method	Skip Layer	PSNR	SSIM
NeRF (JAX)	4	31.66	0.953
NeRF (JAX)	2,4,6	31.90	0.955
NeXT	-	31.44	0.949
NeXT	1	31.83	0.951
NeXT	1,2	**32.44**	**0.956**

(d) **Hierarchical sampling.** NeXT benefits from more sample points and shows less dependence on coarse-to-fine sampling.

Method	N_c	N_f	PSNR	SSIM
NeRF (JAX)	128	128	31.76	0.954
NeRF (JAX)	64	128	31.66	0.953
NeRF (JAX)	192	0	29.97	0.938
NeXT	128	128	**32.94**	**0.959**
NeXT	64	128	32.44	0.956
NeXT	192	0	32.12	0.953

from $N_c + N_f = 192$ to 256 brings ↑0.50 PSNR gains for NeXT yet only ↑0.10 PSNR gains for NeRF.

4.4 Visualization

What is learnt? Sample points along a ray contribute differently to the final rendered pixel. For example, points in the free space or occluded regions barely affect the rendered image. In the light of that the proposed NeXT renders a pixel via a ray-level query, it's expected that NeXT can learn the relative importance of each point via the built-in self-attention mechanism. To investigate what is learnt in NeXT, we first refer to the average received attention scores of each point as "attention weights". Given the multi-head attention scores matrix $\mathbf{AW} \in \mathbb{R}^{H\times N\times N}$, the average attention weight $\hat{\mathbf{AW}} \in \mathbb{R}^{N}$ is calculated by:

$$\hat{\mathbf{AW}} = \text{Softmax}(\frac{1}{H}\sum_{h}^{H}\sum_{i}^{N}\mathbf{AW}_{h,i,:}), \tag{5}$$

where H and N is the number of attention head and sample points respectively.

To demonstrate the effectiveness of NeXT, we visualize the average attention weight in the last NeXT-B block and alpha values of all corresponding sample

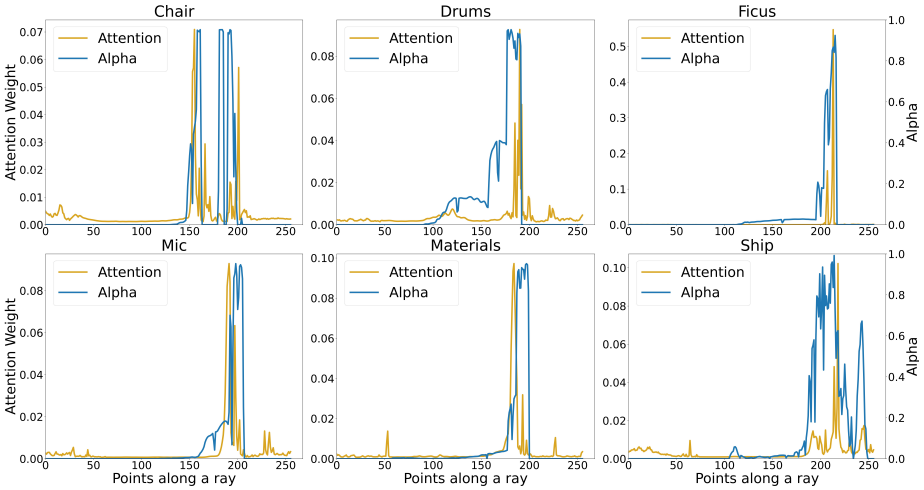

Fig. 4: Visualization of average attention weight and alpha values of sample points for rendering scenes on Blender dataset. The attention weight curve show some similar trends to alpha curve.

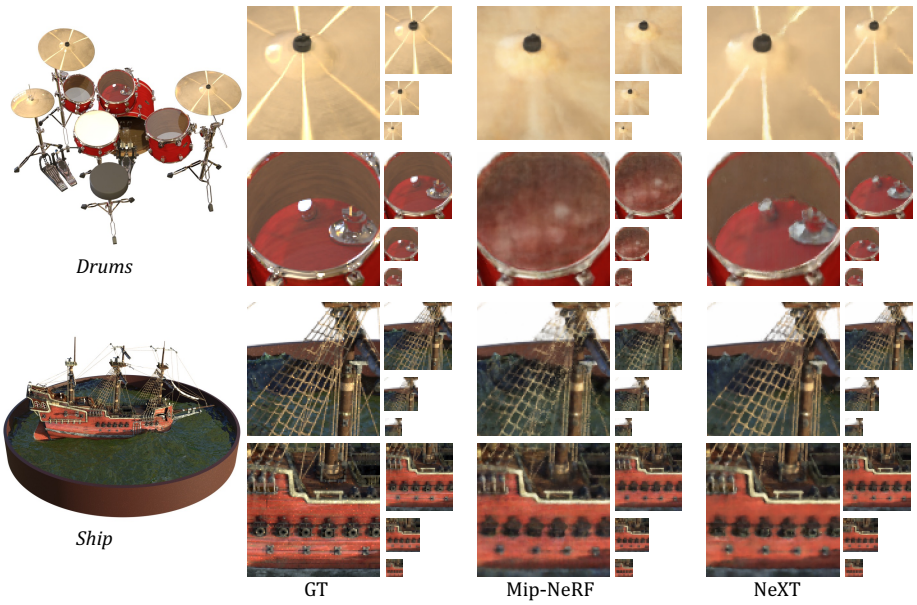

Fig. 5: Visualization of renderings of NeXT-L compared to the groundtruth and Mip-NeRF on two scenes of multiscale Blender dataset. We visualize the cropped regions in four different scales. NeXT qualitatively outperforms Mip-NeRF with more fine details (*e.g.*, the gloss on the drum).

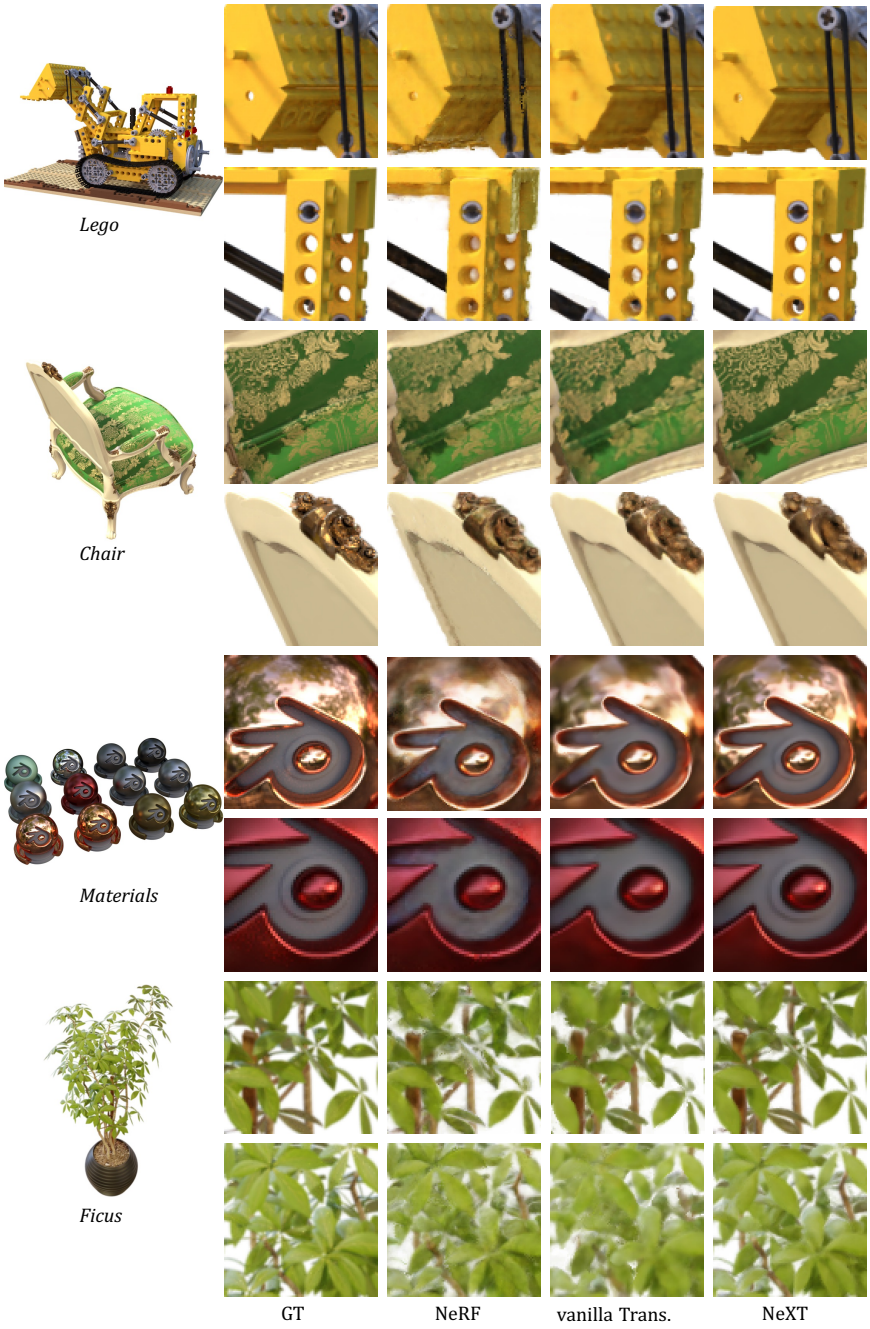

Fig. 6: Visualization of synthesized views of NeXT-L* versus the groundtruth, NeRF and vanilla Transformer. Cropped regions on four scenes of Blender dataset are presented. NeXT remarkably outperforms NeRF, particularly on objects with rich texture details such as *Chair's* patterns and *Ficus's* leaves.

points along a ray when rendering scenes in Blender dataset, as shown in Fig. 4. The alpha values is obtained by $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$, where σ_i is the volume density. Fig. 4 shows that the attention weight curve has some similar trends to alpha curve (*e.g.*, the peaks of the attention weight curve and the alpha curve tend to appear at close points). NeXT tends to capture the scene properties from object space (*e.g.*, $\alpha > 0$) while ignoring those of empty space (*e.g.*, $\alpha = 0$).
Qualitative results. As shown in Fig. 5, we visualize the cropped regions at four different scales on two scenes from the test set of multiscale Blender. NeXT remarkably outperforms Mip-NeRF with more fine details such as *Drums*'s gloss and *Ship*'s nets. In addition, Fig. 6 shows four synthesized views on Blender dataset of NeXT versus groundtruth, NeRF and vanilla Transformer. We observe that NeXT qualitatively outperforms prior work with smooth and fine details such as *Lego*'s ropes, *Chair*'s patterns, *Materials*'s gloss and *Ficus*'s leaves.

5 Limitation and Future Work

In this section, we present some promising directions for future work as follows:
- **Lightweight design.** We report average runtime of three runs measured on 8 NVIDIA V100 GPUs for fair comparisons on Multi-scale Blender dataset. Total training time: 17.07/43.31 hours (Mip-NeRF/NeXT-B). The average inference time for rendering an image: 2.45/4.01 seconds (Mip-NeRF/NeXT-B). Compared to NeRF methods, NeXT has higher runtime due to the interaction of points. Speeding up NeXT is important and promising future work.
- **Real Forward-Facing Scenes.** We also work on verifying the effectiveness of NeXT on other datasets. The results will be presented at the project site[4].
- **Knowledge distillation.** NeXT boosts renderings via increasing model capacity and modeling interdependencies between sampled points, however, brings about challenges in real-time use. Hence, the exploration of transferring the knowledge encoded in NeXT for improving existing faster methods is expected.

6 Conclusion

In this paper, we explore a Transformer-based query network for NVS task, namely NeXT, achieving ray-level query by ray tokenization. NeXT captures relationships between samples via self-attention mechanism, and proposes a multi-skip module to further adapt Transformer-based query network for NVS task. The proposed NeXT shows new state-of-the-art results on three popular datasets, outperforming previous best methods by a large margin. We hope that the general query network presented in this paper will be valuable to other researchers and provide a potential path towards high quality renderings.
Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant 62171248, and the PCNL KEY project (PCL2021A07).

[4] https://github.com/Crishawy/NeXT

References

1. Arandjelović, R., Zisserman, A.: Nerf in detail: Learning to sample for view synthesis. arXiv preprint arXiv:2106.05264 (2021)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
3. Bi, S., Xu, Z., Srinivasan, P., Mildenhall, B., Sunkavalli, K., Hašan, M., Hold-Geoffroy, Y., Kriegman, D., Ramamoorthi, R.: Neural reflectance fields for appearance acquisition. arXiv preprint arXiv:2008.03824 (2020)
4. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: Nerd: Neural reflectance decomposition from image collections. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12684–12694 (2021)
5. Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Wanderman-Milne, S.: Jax: composable transformations of python+ numpy programs, 2018. URL http://github. com/google/jax **4**, 16 (2020)
6. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 425–432 (2001)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
8. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5799–5809 (2021)
9. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1601–1610 (2021)
10. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
11. Davis, A., Levoy, M., Durand, F.: Unstructured light fields. In: Computer Graphics Forum. vol. 31, pp. 305–314. Wiley Online Library (2012)
12. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 11–20 (1996)
13. Deng, B., Barron, J.T., Srinivasan, P.P.: Jaxnerf: an efficient jax implementation of nerf (2020)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

16. Flynn, J., Broxton, M., Debevec, P., DuVall, M., Fyffe, G., Overbeck, R., Snavely, N., Tucker, R.: Deepview: View synthesis with learned gradient descent. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2367–2376 (2019)
17. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8649–8658 (2021)
18. Gao, C., Shih, Y., Lai, W.S., Liang, C.K., Huang, J.B.: Portrait neural radiance fields from a single image. arXiv preprint arXiv:2012.05903 (2020)
19. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14346–14355 (2021)
20. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 43–54 (1996)
21. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5875–5884 (2021)
22. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5885–5894 (2021)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 31–42 (1996)
25. Li, Y., Zhang, S., Wang, Z., Yang, S., Yang, W., Xia, S.T., Zhou, E.: Tokenpose: Learning keypoint tokens for human pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11313–11322 (2021)
26. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6498–6508 (2021)
27. Lindell, D.B., Martel, J.N., Wetzstein, G.: Autoint: Automatic integration for fast neural volume rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14556–14565 (2021)
28. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)
29. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. arXiv preprint arXiv:1906.07751 (2019)
30. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. arXiv preprint arXiv:1906.07751 (2019)
31. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019)
32. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019)

33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
34. Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Alla Chaitanya, C.R., Kaplanyan, A., Steinberger, M.: Donerf: Towards real-time rendering of neural radiance fields using depth oracle networks. arXiv e-prints pp. arXiv–2103 (2021)
35. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11453–11464 (2021)
36. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
37. Ost, J., Mannan, F., Thuerey, N., Knodt, J., Heide, F.: Neural scene graphs for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2856–2865 (2021)
38. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5865–5874 (2021)
39. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10318–10327 (2021)
40. Raj, A., Zollhoefer, M., Simon, T., Saragih, J., Saito, S., Hays, J., Lombardi, S.: Pva: Pixel-aligned volumetric avatars. arXiv preprint arXiv:2101.02697 (2021)
41. Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K.M., Tagliasacchi, A.: Derf: Decomposed radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14153–14161 (2021)
42. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14335–14345 (2021)
43. Rematas, K., Martin-Brualla, R., Ferrari, V.: Sharf: Shape-conditioned radiance fields from a single view. arXiv preprint arXiv:2102.08860 (2021)
44. Riegler, G., Koltun, V.: Free view synthesis. In: European Conference on Computer Vision. pp. 623–640. Springer (2020)
45. Riegler, G., Koltun, V.: Stable view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12216–12225 (2021)
46. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. Advances in Neural Information Processing Systems **33**, 20154–20166 (2020)
47. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. International Journal of Computer Vision **35**(2), 151–173 (1999)
48. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2437–2446 (2019)
49. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019)
50. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019)

51. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7495–7504 (2021)
52. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 175–184 (2019)
53. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7262–7272 (2021)
54. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: Learned initializations for optimizing coordinate-based neural representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2846–2855 (2021)
55. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al.: Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems **34** (2021)
56. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)
57. Trevithick, A., Yang, B.: Grf: Learning a general radiance field for 3d representation and rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15182–15192 (2021)
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
59. Waechter, M., Moehrle, N., Goesele, M.: Let there be color! large-scale texturing of 3d reconstructions. In: European conference on computer vision. pp. 836–850. Springer (2014)
60. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2021)
61. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
62. Wood, D.N., Azuma, D.I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D.H., Stuetzle, W.: Surface light fields for 3d photography. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 287–296 (2000)
63. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34** (2021)
64. Yang, S., Quan, Z., Nie, M., Yang, W.: Transpose: Towards explainable human pose estimation by transformer. arXiv e-prints pp. arXiv–2012 (2020)
65. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems **33**, 2492–2502 (2020)

66. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
67. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021)
68. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 558–567 (2021)
69. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution transformer for dense prediction. arXiv preprint arXiv:2110.09408 (2021)
70. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
71. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)