

Hyb-NeRF: A Multiresolution Hybrid Encoding for Neural Radiance Fields

Yifan Wang
SUSTech

wangyf1998sc@foxmail.com

Yi Gong[†]
SUSTech

gongy@sustech.edu.cn

Yuan Zeng[†]
SUSTech

zengy3@sustech.edu.cn

Abstract

Recent advances in Neural radiance fields (NeRF) have enabled high-fidelity scene reconstruction for novel view synthesis. However, NeRF requires hundreds of network evaluations per pixel to approximate a volume rendering integral, making it slow to train. Caching NeRFs into explicit data structures can effectively enhance rendering speed but at the cost of higher memory usage. To address these issues, we present Hyb-NeRF, a novel neural radiance field with a multi-resolution hybrid encoding that achieves efficient neural modeling and fast rendering, which also allows for high-quality novel view synthesis. The key idea of Hyb-NeRF is to represent the scene using different encoding strategies from coarse-to-fine resolution levels. Hyb-NeRF exploits memory-efficiency learnable positional features at coarse resolutions and the fast optimization speed and local details of hash-based feature grids at fine resolutions. In addition, to further boost performance, we embed cone tracing-based features in our learnable positional encoding that eliminates encoding ambiguity and reduces aliasing artifacts. Extensive experiments on both synthetic and real-world datasets show that Hyb-NeRF achieves faster rendering speed with better rendering quality and even a lower memory footprint in comparison to previous state-of-the-art methods.

1. Introduction

Novel view synthesis targets rendering a scene from a set of images and camera poses obtained from unobserved viewpoints. Synthesizing novel views in real-time at photo-realistic quality is a long-standing problem in computer vision and computer graphics. To address this problem, traditional approaches like rasterization and ray-tracing rely on feature matching and view interpolation, requiring significant manual effort in designing and pre-processing the scene. Recently, Neural Radiance Fields (NeRF) and its variants [2, 6, 20, 23] have shown impressive performance

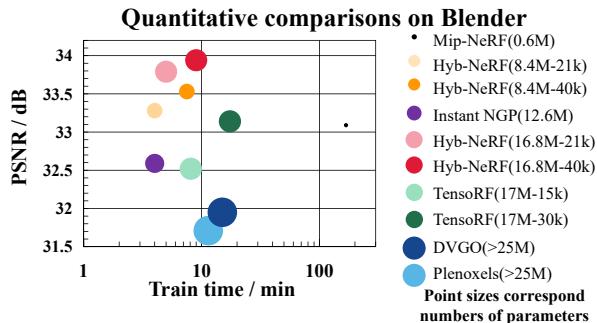
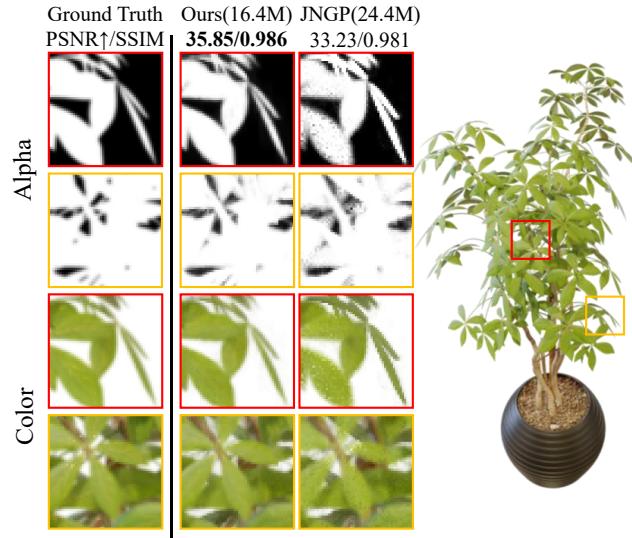


Figure 1. Top: Rendering performance comparison between our Hyb-NeRF and JNGP [41] on the *ficus* scene in Blender. Our Hyb-NeRF can render high-quality color images and synthesize an alpha map with less noise, enabling to accurate reconstruction of the translucency of the leaf edges without jaggies. Bottom: In comparison with previous state-of-the-art fast NeRF training methods, our Hyb-NeRF models can achieve the best rendering quality and memory compactness, while maintaining fast rendering.

on scene representation and novel view synthesis. These approaches obtain high-quality rendering of scenes by implic-

[†] Corresponding authors

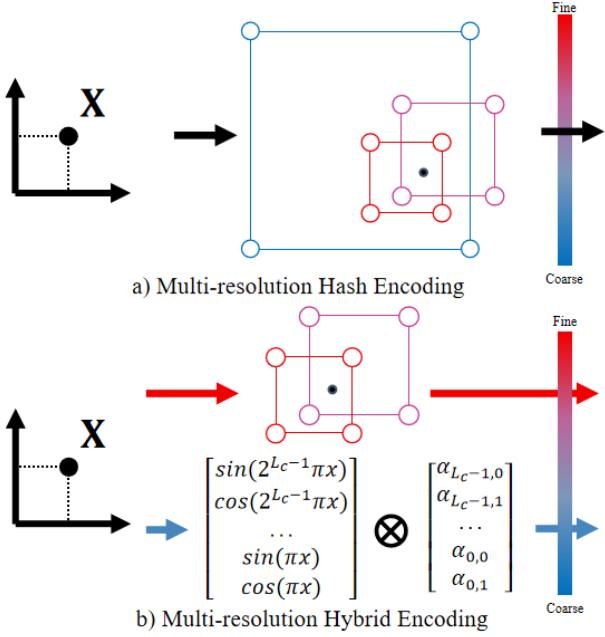


Figure 2. a) Multi-resolution hash encoding [24] maps the input position x to hash-based feature grids at all resolution levels. b) Our multi-resolution hybrid encoding uses the concatenation of coarse-level learnable positional features and fine-level hash-based dense grids of trainable features to represent the position x , resulting in a significantly lower memory footprint and higher quality representation of synthetic and real-world scenes.

itly encoding colors and volume densities using coordinate-based multi-layer perceptrons (MLPs).

A key commonality of NeRF-like models is to encode the low-dimensional coordinate to a higher-dimensional space that assists the MLPs in learning scene representation more accurately. NeRF and its variants [2, 23] encode the input 5D coordinate as a multi-resolution sequence of Fourier features using fixed positional encodings, allowing the MLPs to capture high-frequency details that are essential for photo-realistic novel view synthesis. Despite the significant progress in representing high-frequency details of scenes, NeRF and its variants still require a large MLP to transform the non-parametric features into color and density, which requires a lengthy time for training and rendering.

To address the computational efficiency issue, caching neural radiance fields into explicit data structures has been considered in recent works [6, 11, 30, 36, 42]. By caching additional trainable parameters in an auxiliary data structure, these approaches can learn color and density with a much shallower MLP, which improves rendering speed at the cost of storing a large set of features in a discrete data structure. Previous methods [11, 42] have shown that high-resolution grids with resolution initialization and progressive interpo-

lation work well on scene representation. Instead of doing progressive interpolation from coarse to fine resolution, Instant NGP [24] employs multi-resolution voxel grids of trainable features that enable end-to-end training. It presents a multi-resolution hash encoding to map grids to fixed-size feature arrays from coarse to fine resolution, see Figure 2, and the feature arrays are cached in a hash table, which represents a scene more compactly without sacrificing rendering quality. By storing the feature grids in memory, the computational burden on the MLP is reduced and training speed is significantly increased. While hash-based multi-resolution grid representation can be fast to train and therefore well suited to fast operation, trainable features at coarse resolution levels have lower resolutions and are less compressed than feature grids at fine resolution levels, resulting in limited representation power and low memory efficiency. In this paper, we aim to move toward developing a multi-resolution hybrid encoding for memory-efficient and high-quality scene representation as well as fast rendering by exploiting the different scene representation properties of positional encodings and parametric grid-based encodings.

We propose Hyb-NeRF, a novel radiance field representation that is end-to-end optimizable for compact and fast reconstruction, and also capable of learning an accurate scene representation for novel view synthesis. Our key ideas are as follows. First, to optimize neural implicit map representations with less memory footprint, we propose coarse-to-fine hybrid features to model the scene geometry details and colors. Instead of using parametric feature grids to represent the scene at coarse levels, we present a learnable positional encoding with much fewer trainable parameters for coarse-level representation. As shown in Figure 2, the learnable positional features are then concatenated with the fine-level multi-resolution feature grids for multi-resolution scene representation, yielding detailed 3D reconstructions and high-quality rendering. The learnable positional encoding can adaptively learn the weights of positional features and improve memory efficiency and rendering quality. Second, we embed cone tracing-based positional features in the learning of positional feature weights, which helps to significantly disambiguate the optimization process and eliminate aliasing artifacts.

Our Hyb-NeRF can effectively reduce memory usage and enable fast high-fidelity novel view synthesis. We extensively evaluate our method with various settings and compare it with several state-of-the-art view synthesis methods in terms of the model size, rendering speed, and quality on three benchmark datasets including both synthetic and real-world scenes. All Hyb-NeRF models can reconstruct high-quality radiance fields in 9 min. Our smallest model with 8.4M trainable parameters takes 4 min to achieve better-rendering quality than state-of-the-art meth-

ods while requiring substantially less memory than previous and concurrent voxel-based methods. Our contributions are summarized as follows:

- We present Hyb-NeRF, a novel multi-resolution hybrid encoding that brings together the benefits of positional features and hash-based feature grids to scene representation, enabling memory-efficient, fast, and high-quality rendering.
- We design a learnable positional encoding that controls positional features with much fewer learnable weights to capture more geometry details at coarse resolution levels and improve rendering quality.
- We introduce cone tracing-based features in the learning of positional feature weights, which enables our encoding to work accurately and robustly at different scales.

2. Related Work

Novel View Synthesis: The task of synthesizing images from novel viewpoints given a set of photographs has been widely studied in the field of computer graphics. Various scene representation methods have been proposed to predict an underlying geometric or image-based representation that enables rendering from novel viewpoints. Light field representations [1, 16, 22] directly synthesize novel views by filtering and interpolating the input images, but require high sampling rates and very dense scene capture. To render novel views from sparsely captured images, some approaches leverage pre-computed proxy geometry of the scene [12, 28]. Mesh-based methods [25, 39] represent the scene with surfaces and allow rendering in real-time. However, it is difficult to optimize a mesh to capture fine geometry and topological information. Volumetric representations, such as voxel grids and multi-plane images, are better suited for gradient-based optimization and can synthesize higher-quality views than mesh-based methods. Recently, convolutional neural networks have been employed to estimate voxel grids [13, 21, 31, 32] and point clouds [19, 34, 40] for inward-facing captures and multi-plane images [8, 9, 45] for forward-facing captures. These discrete representations are effective for view synthesis but do not scale well to higher-resolution imagery. In contrast, recent neural representations for novel view synthesis do not suffer from discretionary artifacts as they encode the scene geometry and appearance as a continuous volume [23, 33].

Neural Radiance Fields: NeRF [23] maps the geometry and appearance of the scene into the weights of MLPs. To assist MLPs in capturing high-frequency variations in geometry and appearance and infer high-quality novel views, NeRF encodes the input coordinate to a higher-dimensional space of multi-resolution Fourier features using positional

encoding [37]. Subsequent efforts have extended NeRF for various applications, e.g., relighting [4, 35, 38], large-scale scene modeling [3, 30, 44], dynamic scene modeling [10, 26, 29], and deformation [27, 43]. However, NeRF has limited reconstruction quality with sampling and aliasing issues. Many recent works have proposed to address these issues [2, 3, 15]. Our method is more closely related to Mip-NeRF [2] that casts cones instead of rays to consider the shape and size of volume viewed by each ray and represents the volume covered by each conical frustum using an integrated positional encoding. While NeRFs are effective for photo-realistic view synthesis, they are often computationally expensive and impractical for fast rendering. In this work, we introduce a learnable positional encoding that assists our model to achieve accurate scene representation and fast rendering with shallow MLPs.

NeRFs with explicit volumetric representations: Recent approaches combine NeRFs with explicit volumetric representations, such as octree [20], voxel grids [36], Tri-Plane [5, 10] and factorization tensors [6, 7], to reduce the size of MLPs and thus the time of training and inference. These approaches store trainable parameters in grids and interpolate these parameters to produce a continuous representation of the scene. Although these approaches use many more parameters than implicit representations, their MLPs are much smaller and can be trained to converge much faster. To represent scenes with a simple grid-based model at high resolution without prohibitive memory requirements, a series of works adopt a multistage training strategy to learn a sparse data structure [11, 42] from coarse to fine. For instance, NSVF [20] learned a sparse voxel structure progressively to encode local properties and achieve efficient and high-quality rendering. DVGO [36] first learned to find a coarse geometry and then a post-activated density voxel grid was used in the second stage for generating fine details. TensoRF [6] decomposed a 4D tensor into low-rank components and applied coarse-to-fine reconstruction to achieve high memory compactness and fast rendering. Unlike updating a data structure periodically, Instant NGP [24] proposed a multi-resolution hash encoding method that stores feature vectors of multi-resolution grids in a compact hash table and enables one-stage end-to-end training with shallow MLPs. While this multi-resolution dense grid-based representation increases rendering speed drastically, it still requires a large memory footprint due to a large number of parameters used in both low and high-resolution grids. In contrast, we design a multi-resolution hybrid encoding that replaces feature grids at coarse resolution levels with parametric positional features and enables one-stage end-to-end training, resulting in more compact modeling and faster reconstruction while achieving even higher-quality rendering.

3. Preliminaries

This section provides the relevant background on neural radiance fields (NeRF)-based volume rendering using positional encodings and a multi-resolution hash encoding.

To represent a 3D scene with implicit fields for novel view synthesis, existing NeRFs map a 3D position \mathbf{x} and a 2D viewing direction \mathbf{d} to the corresponding density σ and 3D color value \mathbf{c} using two MLPs $\mathcal{F}_{\mathbf{w}_\theta}$ and $\mathcal{F}_{\mathbf{w}_\phi}$:

$$\mathcal{F}_{\mathbf{w}_\theta} : \mathbf{x} \rightarrow (\sigma, \mathbf{e}), \quad (1)$$

$$\mathcal{F}_{\mathbf{w}_\phi} : (\mathbf{e}, \mathbf{d}) \rightarrow \mathbf{c}, \quad (2)$$

where \mathbf{w}_θ and \mathbf{w}_ϕ are the weights of $\mathcal{F}_{\mathbf{w}_\theta}$ and $\mathcal{F}_{\mathbf{w}_\phi}$, respectively, and \mathbf{e} is an intermediate embedding to help the MLP $\mathcal{F}_{\mathbf{w}_\phi}$ to predict color \mathbf{c} . To render an image, the predicted color of a pixel $\hat{C}(\mathbf{r})$ is computed by casting a ray $\mathbf{r} = \mathbf{o} + t\mathbf{d}$ (where \mathbf{o} denotes the camera origin and t is the distance from the origin along the ray) into the volume and accumulating the color over N point samples taken along the ray [23]:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad (3)$$

where

$$T_i = \exp \left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right). \quad (4)$$

Where δ_i is the distance between the i -th pair of adjacent samples. To enable the MLPs to capture the high-frequency details from the low-dimensional inputs, i.e., \mathbf{x} and \mathbf{d} , the inputs are projected into a higher-dimensional space by encoding functions.

Encodings: Given a position \mathbf{x} and a viewing direction \mathbf{d} , NeRF [23] first transforms each from \mathbb{R} into a higher-dimensional space \mathbb{R}^{2L} using positional encoding-based Fourier features [23, 37]. Instead of casting a single infinitesimally narrow ray per point, integrated positional encoding [2] casts a cone from each point and controls the decay of the high-frequency Fourier features by approximating the conical frustums as a Gaussian distribution and embedding them into the positional encoding. The choice of the 3D conical frustum can significantly reduce aliasing artifacts. The fixed positional encodings are subsequently consumed by two large MLPs to estimate color and density. To reduce the size of the MLPs and save rendering time, hash encoding [24] maps a cascade of grids to features through a spatial hash function [24] and trilinear interpolation. Since the features are stored as trainable parameters, the size of the MLPs can be significantly reduced and both the training and rendering times can be saved when compared to functional encoding-based representations [2, 23].

4. Method

In this section, we describe Hyb-NeRF in detail. Given a set of scenes with a collection of images and their camera parameters, Hyb-NeRF learns a neural rendering model that enables photo-realistic novel view synthesis. Hyb-NeRF encodes the position \mathbf{x} from L multiple resolution levels with trainable encoding parameters. At the coarse levels, a learnable positional encoding is designed to adaptively map the position \mathbf{x} from low-dimension space \mathbb{R} to a higher-dimension space \mathbb{R}^{2L} , resulting an embedding vector $\gamma^{coarse}(\mathbf{x}; \alpha)$ with trainable parameters α . At the fine levels, we model the high-frequency details with multi-resolution parametric feature grids $\gamma^{fine}(\mathbf{x}; \vartheta)$ with trainable parameters ϑ . The multi-resolution hybrid encoding of the position \mathbf{x} is the concatenation of the coarse-level encoding and the fine-level encoding, i.e., $\gamma^{hyb}(\mathbf{x}; \alpha, \vartheta) = [\gamma^{coarse}(\mathbf{x}; \alpha), \gamma^{fine}(\mathbf{x}; \vartheta)]$. The direction \mathbf{d} is transformed into the spherical harmonic features $\xi(\mathbf{d})$ using the spherical harmonic basis. The encoding results of the position \mathbf{x} and direction \mathbf{d} are then fed into two concatenated shallow MLPs $\mathcal{F}_{\mathbf{w}_\theta}$ and $\mathcal{F}_{\mathbf{w}_\phi}$ to produce implicit fields. Later, the implicit fields with the estimated densities and colors are used for volume rendering. The overview of our Hyb-NeRF is illustrated in Figure 3.

Fine-level Encoding: The goal of fine-level encoding is to capture the high-frequency geometric details in a scene. To realize it, we adopt multi-resolution hash-based feature grids $\gamma^{fine}(\mathbf{x}; \vartheta) = \{\gamma_l^{fine}(\mathbf{x}; \vartheta_l)\}_{l=1}^{L_f}$. The spatial resolution of each level $\gamma_l^{fine}(\mathbf{x}; \vartheta_l)$ is set between the coarsest N_{min} and the finest resolution N_{max} :

$$N_l := \lfloor N_{min} b^l \rfloor, \quad (5)$$

where

$$b := \exp \left(\frac{\ln N_{max} - \ln N_{min}}{L_f - 1} \right). \quad (6)$$

Coarse-level Encoding: At coarse resolution levels, the encoding function aims to model the coarse scene geometry and scene layout. While the high-frequency geometric details can be obtained by the multi-resolution hash encoding at fine resolution levels, coarse-level feature grids with trainable parameters have limited representation power. The positional encoding can map the low-dimensional position \mathbf{x} into a sparser, higher-dimensional space without any trainable parameters. Combined with a large MLP, positional encodings achieve high-quality scene representation with slow training and rendering speed. To bring together the benefits of both feature grids and positional encodings to the fast high-quality novel view synthesis, we propose a learnable positional encoding for coarse-level scene representation. Specifically, we transform the input position \mathbf{x} into sinusoidal activations across L_c different frequency levels using a fixed positional encoding [23, 37]:

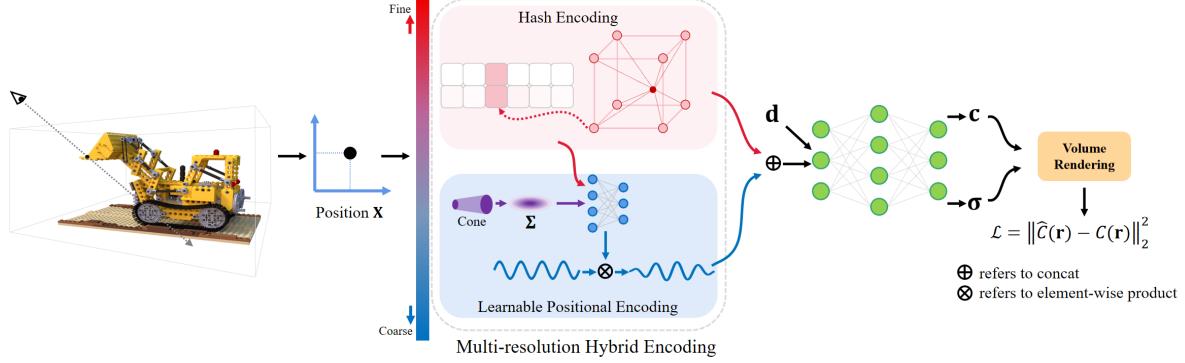


Figure 3. Illustration of our Hyb-NeRF. Given an input 3D position \mathbf{x} , we encode \mathbf{x} to hybrid features from coarse-to-fine resolution levels. For coarse levels, we design a learnable positional encoding to map the position \mathbf{x} into parametric Fourier features. For fine levels, we map \mathbf{x} to parametric features using a multi-resolution hash encoding function. The concatenated feature vectors from all levels are used to predict color and density by two shallow MLPs.

$$\gamma_p(\mathbf{x}) = [\sin(\mathbf{x}), \cos(\mathbf{x}), \dots, \sin(2^{L_c-1}\mathbf{x}), \cos(2^{L_c-1}\mathbf{x})]. \quad (7)$$

$\gamma_p(\mathbf{x})$ is a continuous, multi-scale, periodic representation of \mathbf{x} along each coordinate. To smoothly represent the scene from coarse-to-fine levels, the setting of the level of the positional encoding L_c is based on the coarsest resolution N_{min} as:

$$N_{min} \leq 2^{L_c}. \quad (8)$$

Since the frequency distribution of the local signal may vary with coordinates and shallow MLPs will cause a performance bottleneck of the fixed positional encoding, we introduce learnable weights α to adaptively control the Fourier features to adapt the variation and improve the representation power. We use a single-layer MLP $\mathcal{F}_{\mathbf{w}_p}$ with trainable parameters \mathbf{w}_p and a tanh active function to learn the weights of the Fourier features. Inspired by recent work of embedding cones into the fixed positional encoding in integrated positional encoding (IPE) [2], we embed the cone tracing-based features into the learning of weights α that assists the coarse-level encoding in representing texture with less aliasing artifacts. In addition, to adaptively capture local details according to the position, we also embed the fine-level multi-resolution feature grids into the learning of the weights α . To this end, the MLP $\mathcal{F}_{\mathbf{w}_p}$ takes the concatenation of the multi-resolution hash-based feature grids $\gamma^{fine}(\mathbf{x}; \vartheta)$ and the cone tracing-based Fourier features $\gamma_p(f(\mathbf{x}))$ as input. The weights α can be learned as:

$$\mathcal{F}_{\mathbf{w}_p} : (\gamma^{fine}(\mathbf{x}; \vartheta), \gamma_p(f(\mathbf{x}))) \rightarrow \alpha, \quad (9)$$

where $f(\mathbf{x})$ is the cone tracing-based transformation of the point \mathbf{x} . We adopt cone tracing-based features by casting a

cone from each pixel and approximating a conical frustum with a multivariate Gaussian [2]. The covariance of the final multivariate Gaussian Σ is given by

$$\Sigma = \sigma_t^2(\mathbf{d}\mathbf{d}^T) + \sigma_r^2 \left(\mathbf{I} - \frac{\mathbf{d}\mathbf{d}^T}{\|\mathbf{d}\|_2^2} \right), \quad (10)$$

where σ_t^2 and σ_r^2 are the variances along the ray and the perpendicular direction of the ray, respectively. Instead of computing the diagonal of the covariance matrix and integrating only the axis-aligned Fourier features within the Gaussian cone, here we compute the full upper triangular elements of the covariance matrix:

$$f(\mathbf{x}) = \text{triu}(\Sigma). \quad (11)$$

This remains the non-axis-aligned/non-diagonal components and improves the representation quality of the MLPs. We map the input $f(\mathbf{x})$ into a higher-dimensional space using the fixed positional encoding in Eq.(7) and obtain our final cone tracing-based Fourier features $\gamma_p(f(\mathbf{x}))$. The coarse-level encoding can be obtained by weighting the fixed positional encoding as:

$$\gamma^{coarse}(\mathbf{x}, \alpha) = \gamma_p(\mathbf{x}) \otimes \alpha \quad (12)$$

where \otimes is element-wise multiplication.

Model Architecture: We employ two concatenated MLPs $\mathcal{F}_{\mathbf{w}_\theta}$ and $\mathcal{F}_{\mathbf{w}_\phi}$ to map each encoded position and view direction into its corresponding volume density σ and color \mathbf{c} :

$$\mathcal{F}_{\mathbf{w}_\theta} : \gamma^{hyb}(\mathbf{x}; \alpha, \vartheta) \rightarrow (\sigma, \mathbf{e}), \quad (13)$$

$$\mathcal{F}_{\mathbf{w}_\phi} : (\mathbf{e}, \xi(\mathbf{d})) \rightarrow \mathbf{c}. \quad (14)$$

The estimated density and color are then used to predict pixel color $\hat{C}(\mathbf{r})$ as in (3). Our model is optimized end-to-end by minimizing the L_2 reconstruction loss between the rendered pixel color $\hat{C}(\mathbf{r})$ and the ground truth color $C(\mathbf{r})$:

$$\mathcal{L} = \left\| C(\mathbf{r}) - \hat{C}(\mathbf{r}) \right\|_2^2. \quad (15)$$

5. Experiments

This section evaluates the proposed Hyb-NeRF on rendering of synthetic and realistic scenes. We compare our method with previous state-of-the-art methods quantitatively and qualitatively and provide extensive ablation studies to validate different options in encoding designs. In addition, we also provide the rendering performance of our model with different training times and amounts of parameters.

Datasets: We conduct our experiments on three datasets including Blender [23], Synthetic-NSVF [20] and Tanks&Temples [18, 20]. 1) Blender: It consists of 8 synthetic scenes and each scene has an object (*chair, drums, ficus, hotdog, lego, materials, mic, and ship*) with 400 synthesized images and their corresponding camera parameters. 2) Synthetic-NSVF: Similar to Blender, Synthetic-NSVF contains synthetic scenes with more complex physical structures. We use a subset of five scenes (*bike, palace, robot, toad, and wineholder*) in our experiments. Each scene has a set of synthesized images of an object and camera poses. For both Blender and Synthetic-NSVF, the image resolution is 800×800 pixels, and we follow the setups in NeRF and NSVF to use 100 views for training and 200 for testing. 3) Tanks&Temples: It is a benchmark real-world dataset for image-based 3D reconstruction. We use a subset of the provided scenes (*Ignatius, Truck, Barn, Caterpillar, and Family*), each containing views captured by an inward-facing camera circling. We follow the default split to produce training and testing views and the resolution of each view is 1920×1080 pixels.

Baselines: We compare our method to the following baseline methods: NeRF [23], NSVF [20], Mip-NeRF [2], DVGO [36], TensorRF [6], Instant NGP [24]. Since we implement our Hyb-NeRF on top of a Jittor [14] reimplementations of Instant NGP, we also include this implementation as our baseline and refer to it as JNGP [41].

Implementation Details: We use $L_c = 8$ frequency bands for our learnable positional encoding $\gamma^{coarse}(\mathbf{x})$ at coarse levels. For the hash encoding performed at high resolutions, the setting of the hash table length T and the dimension of the feature vectors F at each level are the same as Instant NGP ($T = 2^{19}$ and $F = 2$). The density MLP \mathcal{F}_{w_θ} is a single-hidden-layer network and the color MLP \mathcal{F}_{w_ϕ} is a two-hidden-layer network. For both, each layer contains 64 channels. We set the coarsest resolution $N_{min} = 180$ and use two different setups for L_f , i.e., $L_f = 8$ and $L_f = 16$,

resulting in 8.4 million (M) and 16.8M trainable parameters of our model, respectively. We also include the rendering results of JNGP [41] with two different L_f , i.e., $L_f = 16$ and $L_f = 32$, which produce 12.6M and 24.4M parameters, respectively. It is valuable to mention that the coarsest resolution N_{min} in Instant NGP is set to 16, which is much smaller than ours. Since our model uses a larger N_{min} and a smaller L_f , it contains much fewer trainable parameters than Instant NGP. We use a batch size of 256K samples and the Adam optimizer [17] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We train our model with a learning rate of 5×10^{-3} and the same learning rate decay schedule as Instant NGP. Our experiments are run on a PC with a single NVIDIA Geforce RTX3090 GPU (24GB).

5.1. Comparisons

Quantitative Comparison: We report our quantitative comparison results in Table 1 in terms of the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) on the three datasets. To evaluate the effect of training time and model size on rendering accuracy, we also report corresponding optimization iterations (the optimization iterations of NeRF, NSVF, and MipNeRF are not present, as this comparison would not be particularly meaningful), training time as well as the number of trainable parameters. We provide rendering results of our method with different numbers of parameters and training times, including two early-stop models and two fully-trained models. We observe that our early-stop model with 8.4M parameters can obtain comparable numerical performance to most of the baselines. Increasing the number of resolution level L_f from 8 to 16 further improves the reconstruction accuracy of our method. Our early-stop model with 16.8M parameters performs better than other methods on synthetic datasets and performs nearly on par with the state-of-the-art neural implicit model on realistic scenes while using less training time. In addition, our model trained with 16.8M parameters in 9 minutes outperforms other methods with fewer errors on both synthetic and real-world scenes.

Qualitative Comparison: Figure 4 shows the visual comparison results on synthetic scenes in Blender. It shows that our model can recover the finer appearance and geometric details, e.g., *lego*'s holes, reflections on the surface of *materials* and *drums*, *chair*'s color and texture. The visual comparison results on real-world scenes in Tanks&Temples are shown in Figure 5. We observe that TensorRF synthesizes reasonable views but still contains blurry textures. JNGP synthesizes views with more texture details but struggles to handle the varying exposure and inconsistent masks, resulting in chromatic aberrations. Since our model embeds conical frustums, it obtains more visually realistic colors while reconstructing surface detail well.

Method	Blender						Synthetic-NSVF		Tanks&Temples	
	#Params↓	Time↓	Iters↓	PSNR(dB)↑	SSIM↑	LIPIS↓	PSNR(dB)↑	SSIM↑	PSNR(dB)↑	SSIM↑
NeRF [23]	1191k	3h	-	31.01	0.947	0.081	29.97	0.944	25.78	0.864
NSVF [20]	3-16M	>48h	-	31.75	0.953	0.047	34.47	0.976	28.48	0.901
MipNeRF [2]	612K	2.8h	-	33.09	0.947	0.043	-	-	-	-
DVGO [36]	>25M	15m	30k	31.95	0.957	0.053	34.51	0.972	28.41	0.911
TensoRF [6]	17M	17m	30k	33.14	0.963	0.047	36.24	0.981	28.56	0.920
Instant NGP [24]	12.6M	4min	30k	32.59	0.960	0.053	-	-	-	-
JNGP [41]	12.6M	5min	40k	32.67	0.959	0.054	34.91	0.976	27.95	0.916
	24.4M	7.5min	40k	32.96	0.963	0.051	35.71	0.983	28.11	0.921
Hyb-NeRF (early-stop)	8.4M	4min	21k	33.28	0.960	0.055	35.68	0.981	28.34	0.909
	16.8M	5min	21k	33.79	0.964	0.049	36.72	0.984	28.58	0.915
Hyb-NeRF (fully-trained)	8.4M	7.5min	40k	33.49	0.961	0.053	36.27	0.982	28.70	0.915
	16.8M	9min	40k	33.94	0.964	0.047	37.14	0.985	29.04	0.922

Table 1. Quantitative comparison on Blender, Synthetic-NSVF, and Tanks&Temples. We also report comparison results of the average training times, the amounts of parameters, and iteration steps for the Blender dataset. Our method achieves the best rendering quality with efficient memory use.

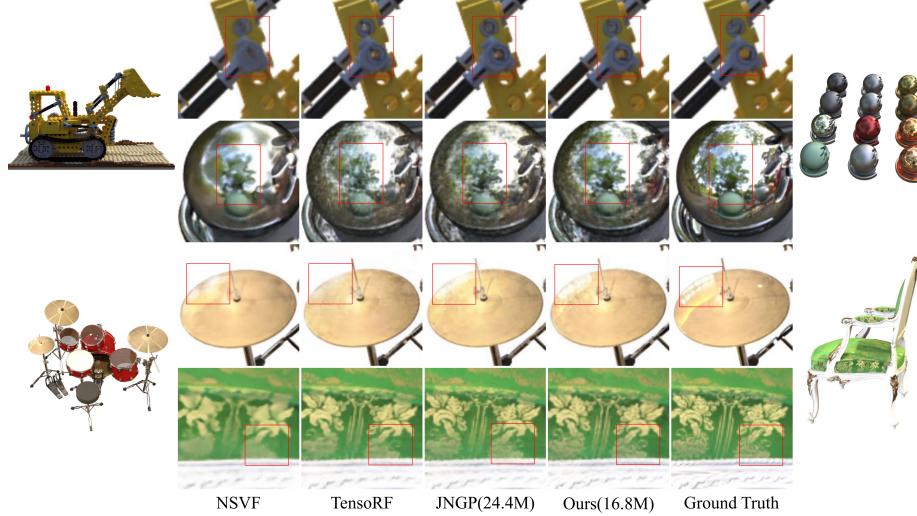


Figure 4. Visual comparison between our fully-trained model with NSVF [20], TensoRF [6] and JNGP [41] on four synthetic scenes in the Blender dataset. Our model synthesizes the most photo-realistic novel views with finer detail.

5.2. Ablation Studies

Effect of the learnable positional encoding: We perform ablation evaluations on the Blender dataset to validate the effectiveness of design choice on learning positional features and the concatenation of the learnable positional features and dense feature grids from coarse-to-fine resolution levels. First, we compare the rendering quality from the baseline JNGP model and two variants of Hyb-NeRF: one is directly concatenating the original fixed positional encoding at coarse levels (**Hyb-NeRF, fixed PE**) and another is concatenating the learnable positional encoding

that only uses the fine-level hash-based feature grids for parameter learning (**Hyb-NeRF, learnable PE w hash encoding**). Quantitative results in Table 2 show that the model with fixed PE (**Hyb-NeRF, fixed PE**) outperforms JNGP, but achieves lower rendering quality than all models with learnable positional encoding (**Hyb-NeRF, learnable PE w hash encoding; Hyb-NeRF, learnable PE w cone; Hyb-NeRF**). This is because fixed positional features at coarse levels guarantee the convergence of **Hyb-NeRF, fixed PE**, and all trainable feature grids at fine levels help to capture better geometry details. **Hyb-NeRF, learnable PE w hash encoding** achieves better results than **Hyb-NeRF, fixed PE**,

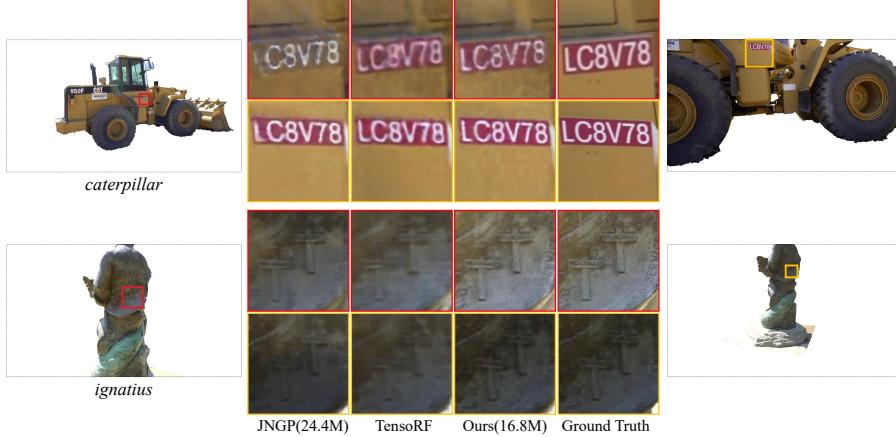


Figure 5. Qualitative comparison results between our fully-trained model, JNGP [41] and TensoRF [6] on *caterpillar* and *ignatius* scenes in the Tanks&Temples dataset. Our model reconstructs better physical details and colors at different scales.

since the position-related fine-resolution feature grids allow the learnable positional encoding to adaptively capture local details, enabling accurate scene representation with the shallow MLPs.

Effect of the cone-tracing embedding: We also evaluate the effect of the cone-tracing embedding. We compare the rendering quality from Hyb-NeRF with and without embedding the cone tracing-based features in the learning of positional feature weights (**Hyb-NeRF, learnable PE w cone**; **Hyb-NeRF, learnable PE w hash encoding**). As shown in Table 2, the embedding of the cone tracing-based features in $\gamma^{coarse}(x; \alpha)$ indeed eliminates aliasing artifacts and improves rendering quality. Meanwhile, we show that a direct concatenation of the multi-resolution hash encoding and IPE at all resolution levels (**JNGP, cat. IPE**) provides better rendering quality than JNGP, but lower rendering quality than our Hyb-NeRF models. This is because IPE as an additional input encoding of the multi-resolution hash encoding can assist the MLPs in capturing more geometry details. However, this performance boost is limited, since the shallow MLPs use IPE with untrainable features and have limited representation power. Our cone-tracing embedding strategy (**Hyb-NeRF, learnable PE w cone**) provides a significant rendering quality improvement over other embedding design choices.

6. Conclusion

We proposed Hyb-NeRF, a novel neural scene representation that is end-to-end optimizable for high-quality and fast rendering. Hyb-NeRF maps the input coordinate to a hybrid encoding that includes parametric positional features at coarse resolution levels and hash-based feature grids at fine resolution levels. The parametric positional features use much fewer trainable parameters for accurate coordi-

Method	PSNR(dB)↑	SSIM↑
JNGP	32.67	0.959
JNGP, cat. IPE	33.09	0.960
Hyb-NeRF, fixed PE	32.83	0.960
Hyb-NeRF, learnable PE w hash encoding	33.19	0.961
Hyb-NeRF, learnable PE w cone	33.53	0.962
Hyb-NeRF	33.94	0.964

Table 2. Quantitative ablation study results on Blender. We compare our model with variants in terms of PSNR and SSIM. All models use 16 hash encoding levels.

nate representation at coarse levels, resulting in a significantly lower memory footprint and higher rendering quality. In addition, we embed the cone tracing-based features in the learning of modulating positional features, leading to better reconstruction quality and photo-realistic novel view synthesis. We show that using a learnable multi-resolution hybrid encoding with tiny MLPs as scene representation enables our Hyb-NeRF to provide favorably against the state-of-the-art in synthesizing more realistic rendering results and efficient memory use.

7. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants 62106095 and 62071212, Shenzhen Science and Technology Program under Grant KCXFZ20211020174802004.

References

- [1] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning neural light fields with

- ray-space embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19819–19829, 2022. 3
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1, 2, 3, 4, 5, 6, 7
- [3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 3
- [4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 3
- [5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 3
- [6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 1, 2, 3, 6, 7, 8
- [7] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor fields: A unified framework for neural fields and beyond. *arXiv preprint arXiv:2302.01226*, 2023. 3
- [8] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7781–7790, 2019. 3
- [9] John Flynn, Michael Broxton, Paul Debevec, Matthew Du-Vall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 3
- [10] Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. *arXiv preprint arXiv:2301.10241*, 2023. 3
- [11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2, 3
- [12] Michael Goesele, Jens Ackermann, Simon Fuhrmann, Carsten Haubold, Ronny Klowsky, Drew Steedly, and Richard Szeliski. Ambient point clouds for view interpolation. In *ACM SIGGRAPH 2010 papers*, pages 1–6. 2010. 3
- [13] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Deepvoxels++: Enhancing the fidelity of novel view synthesis from 3d voxel embeddings. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3
- [14] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences*, 63:1–21, 2020. 6
- [15] Brian KS Isaac-Medina, Chris G Willcocks, and Toby P Breckon. Exact-nerf: An exploration of a precise volumetric parameterization for neural radiance fields. *arXiv preprint arXiv:2211.12285*, 2022. 3
- [16] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016. 3
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 6
- [19] Hoang-An Le, Thomas Mensink, Partha Das, and Theo Gevers. Novel view synthesis from single images via point cloud transformation. *arXiv preprint arXiv:2009.08321*, 2020. 3
- [20] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1, 3, 6, 7
- [21] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3
- [22] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3
- [23] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 4, 6, 7
- [24] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 3, 4, 6, 7
- [25] Richard A Newcombe and Andrew J Davison. Live dense reconstruction with a single moving camera. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1498–1505. IEEE, 2010. 3
- [26] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 3

- [27] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. In *Advances in Neural Information Processing Systems*. 3
- [28] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017. 3
- [29] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 3
- [30] Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *arXiv preprint arXiv:2302.12249*, 2023. 2, 3
- [31] Konstantinos Rematas and Vittorio Ferrari. Neural voxel renderer: Learning an accurate and controllable rendering tool. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5417–5427, 2020. 3
- [32] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 3
- [33] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 3
- [34] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep novel view synthesis from colored 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 1–17. Springer, 2020. 3
- [35] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 3
- [36] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2, 3, 6, 7
- [37] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 3, 4
- [38] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 3
- [39] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 836–850. Springer, 2014. 3
- [40] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 3
- [41] Guo-Wei Yang, Zheng-Ning Liu, Dong-Yang Li, and Hao-Yang Peng. Jnerf: An efficient heterogeneous nerf model zoo based on jittor. *Computational Visual Media*, 9(2):401–404, 2023. 1, 6, 7, 8
- [42] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2, 3
- [43] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 3
- [44] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 3
- [45] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 3