NeuRBF: A Neural Fields Representation with Adaptive Radial Basis Functions

Zhang Chen[1†] Zhong Li[1†] Liangchen Song[2] Lele Chen[1]
Jingyi Yu[3] Junsong Yuan[2] Yi Xu[1]

[1] OPPO US Research Center [2] University at Buffalo [3] ShanghaiTech University

{zhang.chen,zhong.li,lele.chen,yi.xu}@oppo.com

{lsong8,jsyuan}@buffalo.edu yujingyi@shanghaitech.edu.cn

https://oppo-us-research.github.io/NeuRBF-website/

Abstract

We present a novel type of neural fields that uses general radial bases for signal representation. State-of-the-art neural fields typically rely on grid-based representations for storing local neural features and N-dimensional linear kernels for interpolating features at continuous query points. The spatial positions of their neural features are fixed on grid nodes and cannot well adapt to target signals. Our method instead builds upon general radial bases with flexible kernel position and shape, which have higher spatial adaptivity and can more closely fit target signals. To further improve the channel-wise capacity of radial basis functions, we propose to compose them with multi-frequency sinusoid functions. This technique extends a radial basis to multiple Fourier radial bases of different frequency bands without requiring extra parameters, facilitating the representation of details. Moreover, by marrying adaptive radial bases with grid-based ones, our hybrid combination inherits both adaptivity and interpolation smoothness. We carefully designed weighting schemes to let radial bases adapt to different types of signals effectively. Our experiments on 2D image and 3D signed distance field representation demonstrate the higher accuracy and compactness of our method than prior arts. When applied to neural radiance field reconstruction, our method achieves state-of-the-art rendering quality, with small model size and comparable training speed.

1. Introduction

Neural fields (also termed implicit neural representation) have gained much popularity in recent years due to their effectiveness in representing 2D images, 3D shapes, radiance fields, *etc*. [50, 44, 15, 70, 62, 47, 48]. Compared to traditional discrete signal representations, neural fields utilize neural networks to establish a mapping from continuous input coordinates to the corresponding output value. Owing to their concise and efficient formulation, neural fields have been applied to various areas ranging from signal compression [66, 22, 12, 86], 3D reconstruction [49, 83, 79], neural rendering [47, 48, 9, 39, 63, 25, 64, 14], medical imaging [24, 81, 75], acoustic synthesis [11] and climate prediction [31].

† Corresponding author.

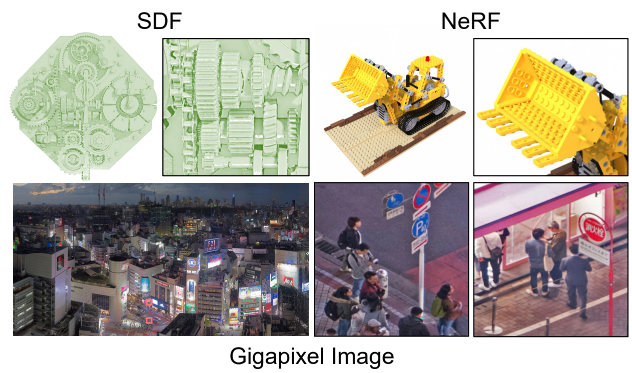

Figure 1. **NeuRBF** provides an accurate and compact neural fields representation for 2D images, 3D SDF, and neural radiance fields.

Early neural fields [50, 44, 15, 47] use neural features that are globally shared in the input domain. Despite the compactness of the models, they have difficulty in representing high-frequency details due to the inductive bias [5, 70] of MLPs. To tackle this problem, local neural fields have been proposed and widely adopted [7, 32, 51, 41, 26, 67, 48, 9], where each local region in the input domain is assigned with different neural features. A common characteristic in this line of work is to use explicit grid-like structures to spatially organize neural features and apply N-dimensional linear interpolation to aggregate local neural features. However, grid-like structures are not adaptive to the target signals and cannot fully exploit the non-uniformity and sparsity in various tasks, leading to potentially sub-optimal accuracy and compactness. While multi-resolution techniques [69, 16, 58, 84, 28] have been explored, it can still be expensive to achieve fine granularity with excessive resolution levels. Some works [47, 70, 62] use frequency encoding to address the low-frequency inductive bias. However, this technique is only applied on either input coordinates or deep features.

In this work, we aim to increase the representation accuracy and compactness of neural fields by equipping the interpolation of basis functions with both spatial adaptivity and frequency extension. We observe that the grid-based linear interpolation, which is the fundamental building block in state-of-the-art local neural fields, is a special case of radial basis function (RBF). While grid-based structures typically grow quadratically or cubically, general RBFs can require fewer parameters (sometimes even constant number) to represent patterns such as lines and ellipsoids. Based upon this observation, we propose NeuRBF, which comprises of a combination of adaptive RBFs and grid-based RBFs. The former uses general anisotropic kernel function with high adaptivity while the latter uses N-dimensional linear kernel function to provide interpolation smoothness.

To further enhance the representation capability of RBFs, we propose to extend them channel-wise and compose them with multi-frequency sinusoid function. This allows each RBF to encode a wider range of frequencies without requiring extra parameters. This multi-frequency encoding technique is also applicable to the features in the MLP, which further improves accuracy and compactness.

To effectively adapt radial bases to target signals, we adopt the weighted variant of K-Means to initialize their kernel parameters, and design a weighting scheme for each of the three tasks (see Fig. 1): 2D image fitting, 3D signed distance field (SDF) reconstruction, and neural radiance field (NeRF) reconstruction. For NeRF, since it involves indirect supervision, traditional K-Means cannot be directly applied. To address this, we further propose a distillation-based approach.

In summary, our work has the following contributions:

- We present a general framework for neural fields based on radial basis functions and propose a hybrid combination of adaptive RBFs and grid-based RBFs.
- We extend radial bases with multi-frequency sinusoidal composition, which substantially enhances their representation ability.
- To effectively adapt RBFs to different target signals, we devise tailored weighting schemes for K-Means and a distillation-based approach.
- Extensive experiments demonstrate that our method achieves state-of-the-art accuracy and compactness on 2D image fitting, 3D signed distance field reconstruction, and neural radiance field reconstruction.

2. Related Work

Global Neural Fields. Early neural fields [50, 44, 15, 77, 45, 21] are global ones and primarily focus on representing the signed distance field (SDF) of 3D shapes. They directly use spatial coordinates as input to multi-layer perceptrons (MLPs) and optionally concatenate a global latent vector for generalized or generative learning. These methods have concise formulation and demonstrate superior flexibility over convolutional neural networks (CNN) and traditional discrete representations in modeling signals in the continuous domain. However, these methods are unable to preserve the high-frequency details in target signals.

Mildenhall *et al*. [47] pioneeringly proposed NeRF, which incorporates neural fields with volumetric rendering for novel view synthesis. They further apply sine transform to the input coordinates (*i.e*., positional encoding), enabling neural fields to better represent high-frequency details. Similar ideas are also adopted in RFF [70] and SIREN [62], which use random Fourier features or periodic activation as frequency encoding. These works also promote neural fields to be a general neural representation applicable to different types of signals and tasks. More recently, other encoding functions or architectures have been explored [23, 72, 40, 60, 74, 73, 19, 36, 87, 52, 53, 18, 85, 57, 80]. For example, MFN [23] replaces MLPs with the multiplication of multiple linear functions of Fourier or Gabor basis functions, and WIRE [57] uses Gabor wavelet as activation function in MLPs. Radial basis functions (RBF) have also been discussed in [52, 53]. However, unlike our work, they only consider simplified forms of RBFs and do not explore spatial adaptivity, leading to nonideal performance.

Local Neural Fields. In parallel to frequency encoding, local neural fields improve representation accuracy by locality. Early attempts [7, 32, 51, 17, 13, 67] uniformly subdivide the input domain into dense grids and assign a neural feature vector to each grid cell. During point querying, these local neural features are aggregated through nearest-neighbor or N-dimensional linear interpolation and then used as input to the following MLPs. Due to feature locality, the depth and width of the MLPs can be largely reduced [67, 26, 33], leading to higher training and inference speed than global neural fields. Apart from neural features, the locality can also be implemented on the network weights and biases [54, 58, 29], where each grid cell has a different MLP. Dense grids can be further combined with RFF [70] or SIREN [62] to improve accuracy on high-frequency details [30, 43]. However, a significant drawback of dense grids is that they are parameter-intensive.

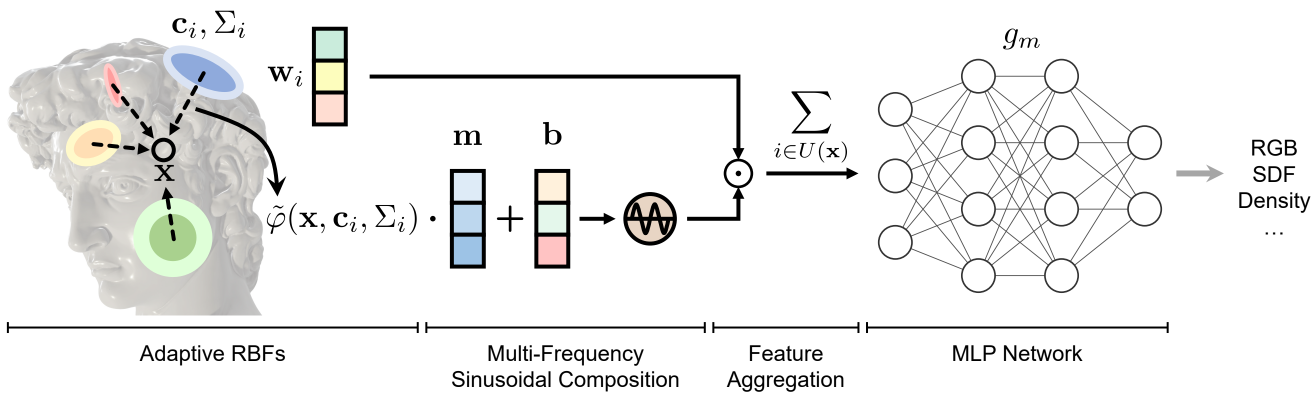

Figure 2. **Illustration of NeuRBF.** Each adaptive RBFs can have different position and shape parameters $\mathbf{c}_i, \Sigma_i$, leading to high spatial adaptivity. With multi-frequency sinusoidal composition, each adaptive RBF is further extended to multiple radial bases with different frequencies, which are then combined with neural features $\mathbf{w}_i$ through Hadamard product. The resulting features are aggregated within the neighborhood $U(\mathbf{x})$ of query point $\mathbf{x}$, and then be mapped to the output domain by the MLP network g_m.

To improve model compactness, numerous techniques have been proposed, such as multi-resolution tree (and/or residual) structures [41, 84, 16, 42, 58, 82, 76, 26], hash grids [48], dictionary optimization [68], permutohedral lattices [56], tensor decomposition [9], orthogonal planes [51, 8, 61, 6, 25], wavelet [55], and multiplicative fields composition [10]. Among them, Instant NGP [48] achieves high accuracy, compactness, and efficiency across different signal types. Despite the additional data structures or operations, these methods still rely on basic grid-based linear interpolation as the building block for neural feature aggregation. Another line of work [27, 38, 78] relaxes the grid structures and allows neural features to be freely positioned in the input domain. However, they use simple interpolation kernel functions, which still have limited spatial adaptivity. Their performance is also inferior to state-of-the-art grid-based ones.

Unlike prior local neural fields, we seek a general framework consisting of hybrid radial bases and enhance their representation capability by simultaneously exploiting spatial adaptivity and frequency extension.

3. Our Method

3.1. Local Neural Fields As Radial Basis Functions

Local neural fields represent a signal f in the form of a function $\hat{f}: \mathbb{R}^D \rightarrow \mathbb{R}^O$, which maps a coordinate $\mathbf{x}$ in the continuous D-dimensional space to an output of O dimensions. The function f can be considered as a composition of two stages, *i.e.*, $f = g_m \circ g_b$, where g_b extracts the local neural features at input location $\mathbf{x}$ from a neural representation (*e.g.*, feature grid), and g_m decodes the resulting feature to the final output. Now we consider grid-based linear interpolation for g_b, which is a common building block in state-of-the-art neural fields. It has the following form: $g_b(\mathbf{x}) = \sum_{i \in U(\mathbf{x})} \varphi(\mathbf{x}, \mathbf{c}_i)\mathbf{w}_i$. $U(\mathbf{x})$ is the set of grid corner nodes that enclose $\mathbf{x}$, $\mathbf{c}_i \in \mathbb{R}^D$ and $\mathbf{w}_i \in \mathbb{R}^F$ are the position and neural feature of node i. $\varphi(\mathbf{x}, \mathbf{c}_i) \in \mathbb{R}$ is the interpolation weight of node i, and is computed as:

$$\varphi(\mathbf{x}, \mathbf{c}_i) = \prod_{j=1}^{D} \max(0, 1 - \frac{|\mathbf{x}_j - \mathbf{c}_{i,j}|}{\sigma}), \tag{1}$$

where σ is the sidelength of each grid cell, and $\mathbf{x}_j, \mathbf{c}_{i,j}$ are the jth element of $\mathbf{x}, \mathbf{c}_i$. Note that Eq. (1) is a special case of radial basis function (RBF) with the form of $\varphi(\mathbf{x}, \mathbf{c}_i, \sigma_i)$, where each RBF has its own position parameter $\mathbf{c_i}$ and shape parameter σ_i. From the perspective of RBF, we use the following formulation for $g_b(\mathbf{x})$:

$$g_b(\mathbf{x}) = \sum_{i \in U(\mathbf{x})} \varphi(\mathbf{x}, \mathbf{c}_i, \sigma_i)\mathbf{w}_i. \tag{2}$$

3.2. Neural Radial Basis Fields

Compared to grid-based linear interpolation, the advantages of RBFs originate from the additional position and shape parameters $\mathbf{c}_i, \sigma_i$. As illustrated in Fig. 2, our framework makes extensive use of adaptive RBFs. To fully exploit their adaptivity, we propose to use anisotropic shape parameters $\Sigma_i \in \mathbb{R}^{D \times D}$. The first row of Fig. 3 shows that with anisotropic shape parameters, the shape of an RBF's level set can be either circular, elliptical, or even close to a line. This allows an RBF to be more adaptive to target signals. For the kernel function φ, we use the inverse quadratic function as an example, which is computed as:

$$\varphi(\mathbf{x}, \mathbf{c}_i, \Sigma_i) = \frac{1}{1 + (\mathbf{x} - \mathbf{c}_i)^T \Sigma_i^{-1} (\mathbf{x} - \mathbf{c}_i)}. \tag{3}$$

Note that Σ_i is a covariance matrix, which is symmetric. Hence, each Σ_i only has $\frac{D \cdot (D-1)}{2}$ parameters. We can op-

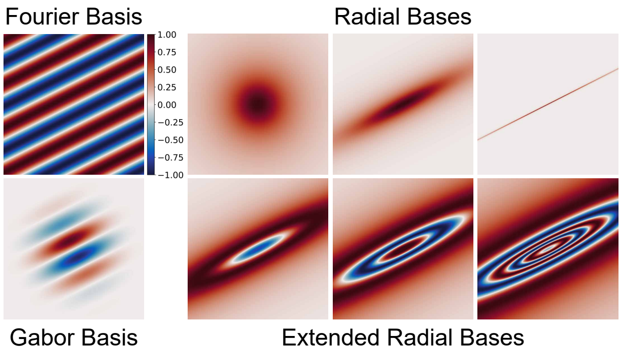

Figure 3. **Comparison of Bases.** For the right 3 columns: the first row shows radial bases with different shape parameters; the bottom row shows extended radial bases with different frequencies.

tionally normalize the radial basis value at each point:

$$\tilde{\varphi}(\mathbf{x}, \mathbf{c}_i, \Sigma_i) = \frac{\varphi(\mathbf{x}, \mathbf{c}_i, \Sigma_i)}{\sum_{k \in U(\mathbf{x})} \varphi(\mathbf{x}, \mathbf{c}_k, \Sigma_k)}. \tag{4}$$

Note that our framework is not limited to a specific function type but supports any others that have the radial basis form. The choice of the function type can thus be finetuned per task.

Sinusoidal Composition on Radial Basis. We notice that while traditional RBF is a scalar function, $\mathbf{w}_i \in \mathbb{R}^F$ is a vector with multiple channels (recall Eq. (2)). Our motivation is to let each channel of $\mathbf{w}_i$ linearly combine with a different variant of the RBF so that the channel-wise capacity of RBF can be exploited. To achieve this, we propose to compose RBF with a multi-frequency sinusoid function, where a radial basis is extended into multiple channels with different frequencies:

$$\boldsymbol{\varphi}(\mathbf{x}, \mathbf{c}_i, \Sigma_i) = \sin(\tilde{\varphi}(\mathbf{x}, \mathbf{c}_i, \Sigma_i) \cdot \mathbf{m} + \mathbf{b}), \tag{5}$$

where $\mathbf{m}, \mathbf{b} \in \mathbb{R}^F$ are the global multiplier and bias applied to $\tilde{\varphi}(\mathbf{x}, \mathbf{c}_i, \Sigma_i)$ before sine transform. The resulting $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{c}_i, \Sigma_i)$ has F channels and is then multiplied with $\mathbf{w}_i$ through Hadamard product. Fig. 2 illustrates this computation process. $g_b(\mathbf{x})$ is thus computed as:

$$g_b(\mathbf{x}) = \sum_{i \in U(\mathbf{x})} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{c}_i, \Sigma_i) \odot \mathbf{w}_i. \tag{6}$$

With Eq. (5), the number of bases encoded by a single pair of $\mathbf{c}_i, \Sigma_i$ is increased from 1 to F, leading to higher representation ability. Note that $\mathbf{m}, \mathbf{b}$ are globally shared across RBFs. We set $\mathbf{b}$ as a learnable parameter and $\mathbf{m}$ as a fixed parameter. We determine the value of $\mathbf{m}$ by specifying the lowest and highest frequencies m_l, m_h. The rest of the elements are obtained by log-linearly dividing the range between m_l and m_h.

Our sinusoidal composition technique differs from positional encoding [47] and random Fourier features [70] in that we apply sine transform to radial bases instead of input coordinates. This allows the composited bases to have elliptical periodic patterns as shown in Fig. 3 second row, while the bases created by [47, 70] are limited to linear periodic patterns. Our technique is also related to the Gabor filter, which combines a Gaussian function and a sinusoidal function using multiplication. Still, the Gabor filter can only produce bases with linear patterns.

Sinusoidal Composition on Feature Vector. We also apply our sinusoidal composition technique to the output features $\mathbf{h}_0$ of the first fully-connected (FC) layer in g_m:

$$\mathbf{f}_0 = \sin(\mathbf{h}_0 \odot \mathbf{m}_0) + \mathbf{h}_0, \tag{7}$$

where $\mathbf{h}_0, \mathbf{m}_0, \mathbf{f}_0 \in \mathbb{R}^{F_0}$ and $\odot$ is Hadamard product. The bias term is omitted since it is already contained in FC layer. The reason to apply this sinusoidal composition to $\mathbf{h}_0$ instead of $g_b(\mathbf{x})$ is to let the network first combines the multi-frequency bases in $g_b(\mathbf{x})$ via an FC layer. Here, we also include a residual connection, which slightly benefits performance. The resulting feature vector $\mathbf{f}_0$ is input to the next layer in g_m. $\mathbf{m}_0$ is set in a similar manner as $\mathbf{m}$ by specifying its lowest and highest frequency m_{l0} and m_{h0}.

Compared to sinusoid activation [62], our multi-frequency approach can produce features of wide frequency range with one sine transform. In addition, it does not require specialized initialization for the FC layers. We experimentally observe that our technique achieves higher performance under radial basis framework. Table 5 shows a quantitative comparison with positional encoding [47] and sinusoid activation [62].

Hybrid Radial Bases. To balance between fitting accuracy and interpolation smoothness, we propose to use a combination of adaptive RBFs and grid-based RBFs. The position and shape parameters of adaptive RBFs can be freely assigned while those of grid-based RBFs are fixed to a grid structure. Adaptive RBFs tend to produce sharp discontinuities when $U(\mathbf{x})$ (the set of neighboring RBFs of the point $\mathbf{x}$) changes. On the other hand, grid-based RBFs do not exhibit such discontinuity and can better preserve function smoothness. Please refer to our supplementary for an illustration. We combine adaptive and grid-based RBFs through feature concatenation, which allows the network to select features accordingly.

3.3. Initialization of Position and Shape Parameters

Motivated by [59], we adapt RBFs to target signals by initializing their position and shape parameters with weighted K-Means clustering. Intuitively, this biases RBF distribution towards data points with higher weights. This technique is simple and effective, and can be applied to different tasks by changing the weighting scheme.

Position Initialization. Let $\mathbf{x}_1, ..., \mathbf{x}_m$ be the coordinates of input points and $w_1, ..., w_m$ be the weight of each point (weight calculation will be described later). Given initial cluster centers $\mathbf{c}_1, ..., \mathbf{c}_n$, weighted K-Means optimizes these cluster centers with:

$$\min_{\mathbf{c}_1,...,\mathbf{c}_n} \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} w_j \|\mathbf{x}_j - \mathbf{c}_i\|^2, \tag{8}$$

where a_{ij} is an indicator variable. Following common practice, we solve Eq. (8) with an expectation–maximization (EM)-style algorithm.

Shape Initialization. Inspired by Gaussian mixture model, we initialize the shape parameters Σ_i as the following:

$$\Sigma_i = \frac{\sum_j a_{ij} w_j (\mathbf{x}_j - \mathbf{c}_i)(\mathbf{x}_j - \mathbf{c}_i)^T}{\sum_j a_{ij} w_j}. \tag{9}$$

Weighting Schemes. The weights $w_1, ..., w_m$ control how RBFs will be distributed after initialization. Data points with higher importance should be assigned with higher weights.

For 2D images, we use the spatial gradient norm of pixel value as the weight for each point: $w_j = \|\nabla I(\mathbf{x}_j)\|$.

For 3D signed distance field, we use the inverse of absolute SDF value as point weight: $w_j = 1 \,/\, (|SDF(\mathbf{x}_j)| + 1e-9)$. The inclusion of $1e-9$ is to avoid division by zero.

For neural radiance field, it is a task with indirect supervision. The input signal is a set of multi-view 2D images while the signal to be reconstructed lies in 3D space. Therefore, we cannot directly obtain the weights. To tackle this problem, we propose a distillation method. We first use grid-based neural fields to train a model for $1000 \sim 2000$ training steps. Then we uniformly sample 3D points and use the trained model to predict the density $\sigma(\mathbf{x})$ and color feature vector $\mathbf{f}_c(\mathbf{x})$ at these points. Finally, we convert density to alpha and multiply with the spatial gradient norm of the color feature vector as point weight: $w_j = (1-\exp(-\sigma(\mathbf{x}_j)\delta))\|\nabla \mathbf{f}_c(\mathbf{x}_j)\|$. This weighting scheme takes both density and appearance complexity into account. Compared to 3D Gaussian Splatting [34] and Point-NeRF [78], our approach does not require external structure-from-motion or multi-view stereo methods to reconstruct the point cloud, but distills information from a volumetric model. Hence, our initialization can handle both surface-based objects and volumetric objects.

4. Implementation

In this section, we describe the keypoints of our implementation. More details can be found in our supplementary.

We implement our adaptive RBFs using vanilla PyTorch without custom CUDA kernels. For the grid-based part in our framework, we adopt Instant NGP [48] for 2D image fitting and 3D signed distance field (SDF). We use a PyTorch implementation of Instant NGP from [1]. For neural radiance field (NeRF) reconstruction, we explored TensoRF [9] and K-Planes [25] as the grid-based part. We reduce the spatial resolution and feature channel of the grid-based part, and allocate parameters to the adaptive RBFs accordingly.

For sinusoidal composition, we use $m_l = 2^{-3}, m_h = 2^{12}, m_{l0} = 1, m_{h0} = 1000$ in the image experiments on DIV2K dataset [3, 71], and $m_l = 2^0, m_h = 2^3, m_{l0} = 30, m_{h0} = 300$ in SDF experiments. In NeRF task, we do not apply sinusoidal composition since the improvement is small.

Training is conducted on a single NVIDIA RTX A6000 GPU. We use Adam optimizer [35] where $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 10^{-15}$. The learning rates for neural features are $5\times10^{-3}, 1\times10^{-4}, 2\times10^{-2}$ for image, SDF and NeRF task respectively. In addition, we apply learning rate schedulers that gradually decrease learning rates during training. The position and shape parameters of RBFs can be optionally finetuned via gradient backpropagation. However, we do not observe significant performance gain and therefore fix these parameters during training.

We use L2 loss when fitting 2D images and reconstructing neural radiance field, and use MAPE loss [48] when reconstructing 3D SDF. For SDF task, we use the same point sampling approach as Instant NGP [48]. For NeRF task, we follow the training approaches in TensoRF [9] and K-Planes [25] respectively. In all experiments, both competing methods and our method are trained per scene.

5. Experiment

5.1. 2D Image Fitting

We first evaluate the effectiveness of fitting 2D images. We use the validation split of DIV2K dataset [3, 71] and 6 additional images of ultra-high resolution as evaluation benchmark. DIV2K validation set contains 100 natural images with resolution around 2040×1356. The resolution of the 6 additional images ranges from 6114×3734 to 56718×21450.

We first compare with MINER [58] and Instant NGP ("I-NGP") [48], which exhibit state-of-the-art performance for high-resolution image fitting. We let our method use fewer parameters than the other two. During timing, the time for initializing RBFs is also taken into account.

Table 1 top half shows the comparison on the DIV2K dataset. For our method, we include two additional setups:

	Steps	Time↓	# Tr. Params↓	PSNR↑
		DIV2K		
MINER [58]	35k	16.7m	5.49M	46.92
I-NGP [48]	35k	1.3m	4.91M	47.56
Ours	35k	7.9m	4.31M	**58.56**
Ours$_{3.5k-steps}$	3.5k	**48s**	4.31M	51.53
Ours$_{2.2M}$	35k	7.7m	**2.20M**	49.26
		DIV2K 256×256×3		
BACON [40]	5k	78.2s	268K	38.51
PNF [80]	5k	483.9s	287K	38.99
Ours	5k	**28.5s**	**128K**	**54.84**

Table 1. **2D Image Fitting.** We quantitatively compare our method with MINER [58], Instant NGP ("I-NGP") [48], BACON [40] and PNF [80] on the validation set of DIV2K dataset [3, 71]. "DIV2K": original image resolution; "DIV2K 256×256×3": center cropped and downsampled to 256×256×3.

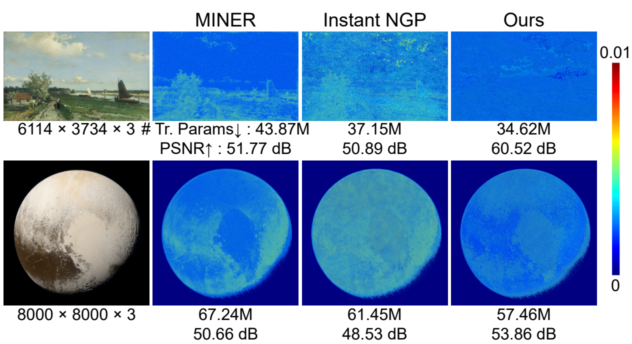

Figure 4. **2D Image Fitting.** Leftmost column shows the fitted images of our method and the resolution of the images. The other columns show the error maps of each method, along with the number of trainable parameters ("# Tr. Params") and PSNR.

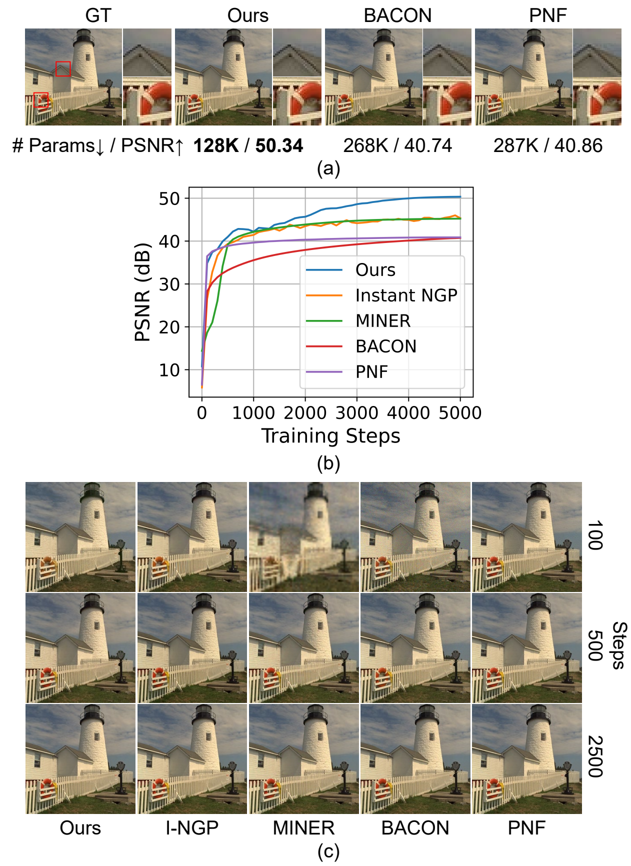

Figure 5. **2D Image Fitting** on an image from Kodak dataset [20]. (a) Final results after 5k training steps. (b) Training curves. (c) Intermediate results.

one using fewer training steps and one using fewer trainable parameters. When using the same number of training steps, our method outperforms the other two by over 10 dB in Peak Signal-to-Noise Ratio (PSNR) with less trainable parameters. Although Instant NGP has faster training speed due to their heavily-optimized CUDA implementation, our method is implemented with vanilla PyTorch and is easily extensible. In addition, with only 3.5k training steps ($1/10$ of that of Instant NGP), our method already reaches a PSNR of 51.53 dB, which is almost 4 dB higher than Instant NGP. Meanwhile, the training time is only 48s and even faster than Instant NGP. The time for RBF initialization is around 2s. "Ours$_{2.2M}$" additionally demonstrates the high compactness of our method. After reducing trainable parameters to be over 50% fewer than the competing methods, our approach still retains a higher fitting accuracy.

In Fig. 4, we show the fitting results on 2 ultra-high resolution images. Besides achieving higher PSNR than the other two, our method also has a more uniform error distribution. This reflects the adaptivity of RBFs, which allows a more accurate representation of details. Results on other images can be found in our supplementary material.

We additionally compare with BACON [40] and PNF [80] on the 100 images in DIV2K validation set. In this experiment, the images are center cropped and downsampled to 256×256×3 following the practice of BACON [40]. We use their official codes and settings for BACON and PNF, and let our method use the same batch size (65,536) and training steps (5k) as them. The results are shown in Table 1 bottom half. We further conduct comparisons on a sample image from Kodak dataset [20], and show the qualitative results and training curves in Fig. 5. The image is similarly center cropped and resized to 256×256×3. The results show that our method has both fast convergence and high fitting accuracy. Higher PSNR demonstrates the ability to more precisely represent target signals, and implies fewer parameters and training steps to reach a specified PSNR. For the image in Fig. 5, Instant NGP and MINER reach 45.34 dB and 45.23 dB PSNR with 140K parameters and 5k steps. Our method instead can reach 45.59 dB PSNR with only 72K parameters and 3.5k steps.

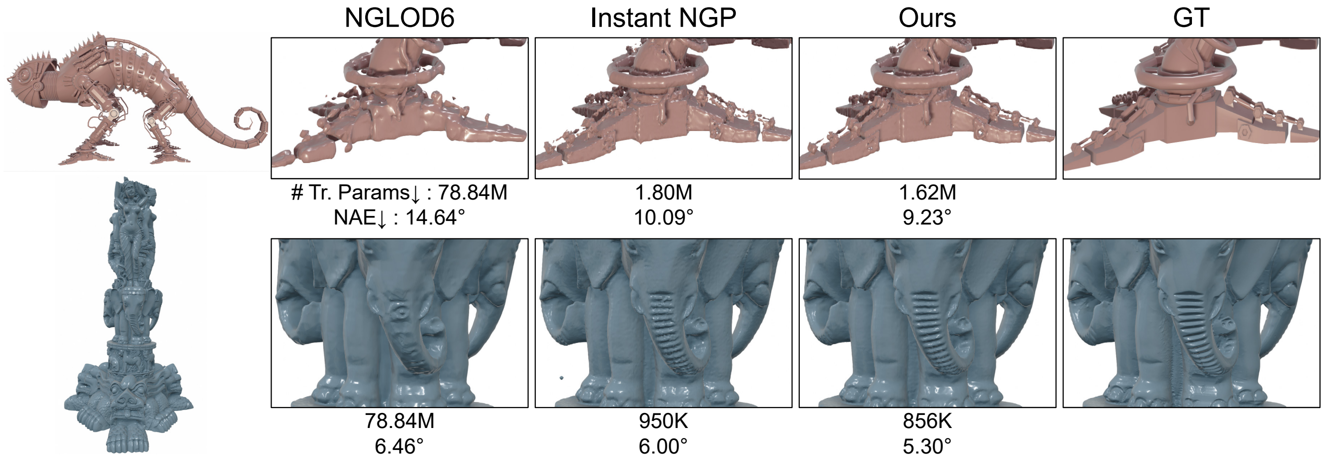

Figure 6. **3D Signed Distance Field Reconstruction.** Leftmost column shows the reconstructed geometry of our method. The other columns show qualitative and quantitative comparisons of reconstruction results. "# Tr. Params" is the number of trainable parameters and "NAE" is the normal angular error.

	Steps	# Tr. Params↓	IoU↑	NAE↓
NGLOD5 [69]	245k	10.15M	0.9962	6.58
NGLOD6 [69]	245k	78.84M	0.9963	6.14
I-NGP [48]	20k	950K	0.9994	5.70
Ours	20k	856K	**0.9995**	**4.93**
I-NGP$_{400K}$ [48]	20k	498K	0.9992	6.39
Ours$_{400K}$	20k	**448K**	0.9994	5.53

Table 2. **3D Signed Distance Field Reconstruction.** We quantitatively compare our method with NGLOD [69] and Instant NGP ("I-NGP") [48].

5.2. 3D Signed Distance Field Reconstruction

We use 10 3D models from the Stanford 3D Scanning Repository [65], the Digital Michelangelo Project [37], and TurboSquid [2] as benchmark data. These models contain delicate geometric details and challenging topologies. We compare our method with NGLOD [69] and Instant NGP [48]. For evaluation metrics, we use Intersection over Union (IoU) and normal angular error (NAE). NAE measures the face normal difference of corresponding points and can better reflect the accuracy of reconstructed surface than IoU.

Fig. 6 demonstrates example results on 3 objects. Our method produces more accurate geometry, with sharp edges and smooth surfaces. Comparatively, the results of NGLOD are overly smooth while those of Instant NGP contain noises.

In Table 2, we compare the performance under different numbers of trainable parameters. Our approach consistently has higher IoU and lower NAE. The advantages of our method are larger when using fewer parameters, which is also demonstrated in Fig. 7.

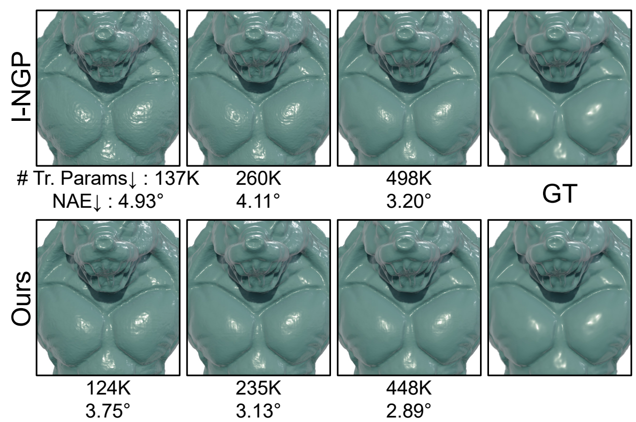

Figure 7. **3D Signed Distance Field Reconstruction.** We compare the reconstruction accuracy of Instant NGP ("I-NGP") [48] and ours under different parameter count.

5.3. Neural Radiance Field Reconstruction

We evaluate our approach on both 360° scenes and forward-facing scenes. Metrics of the comparison methods are taken from their paper whenever available. Full per-scene results are available in our supplementary material.

360° Scenes. We use the Synthetic NeRF dataset [47] which is a widely adopted benchmark for neural radiance field reconstruction. We utilize TensoRF [9] as the grid-based part in this experiment. We compare with numerous representative methods in this area, as listed in Table 3. Among them, Instant NGP [48] and TensoRF [9] represent state-of-the-art performance while Factor Fields [10] is concurrent to our work. For Point-NeRF [78], their SSIM metrics are recomputed with a consistent SSIM implementation as other work.

	Batch Size	Steps	Time↓	# Params↓	PSNR↑	SSIM↑	LPIPS_{VGG} ↓	LPIPS_{Alex} ↓
NeRF [47]	4096	300k	∼ 35h	1.25M ●	31.01	0.947	0.081	-
Mip-NeRF 360 [4]	16384	250k	∼ 3.4h	3.23M ●	33.25	0.962	0.039 ●	-
Point-NeRF [78]	-	200k	∼ 4.5h	-	33.31 ●	0.962	0.050	0.028
Plenoxels [26]	5000	128k	11.4m ●	194.5M	31.71	0.958	0.049	-
Instant NGP [48]	262144	35k	3.8m ●	12.21M	33.18	0.963 ●	0.051	0.028
TensoRF [9]	4096	30k	17.4m	17.95M	33.14	0.963 ●	0.047 ●	0.027 ●
Factor Fields [10]	4096	30k	12.2m ●	5.10M	33.14	0.961	-	-
K-Planes [25]	4096	30k	38m	33M	32.36	0.962	0.048	0.031
Ours	4096	30k	33.6m	17.74M	34.62 ●	0.975 ●	0.034 ●	0.018 ●
$\text{Ours}_{3.66M}$	4096	30k	29.3m	3.66M ●	33.97 ●	0.971 ●	0.039 ●	0.022 ●

Table 3. **Neural Radiance Field Reconstruction.** We quantitatively compare our method with numerous state-of-the-art methods on the Synthetic NeRF dataset [47]. Best 3 scores in each metric are marked with gold ●, silver ● and bronze ●. "-" denotes the information is unavailable in the respective paper.

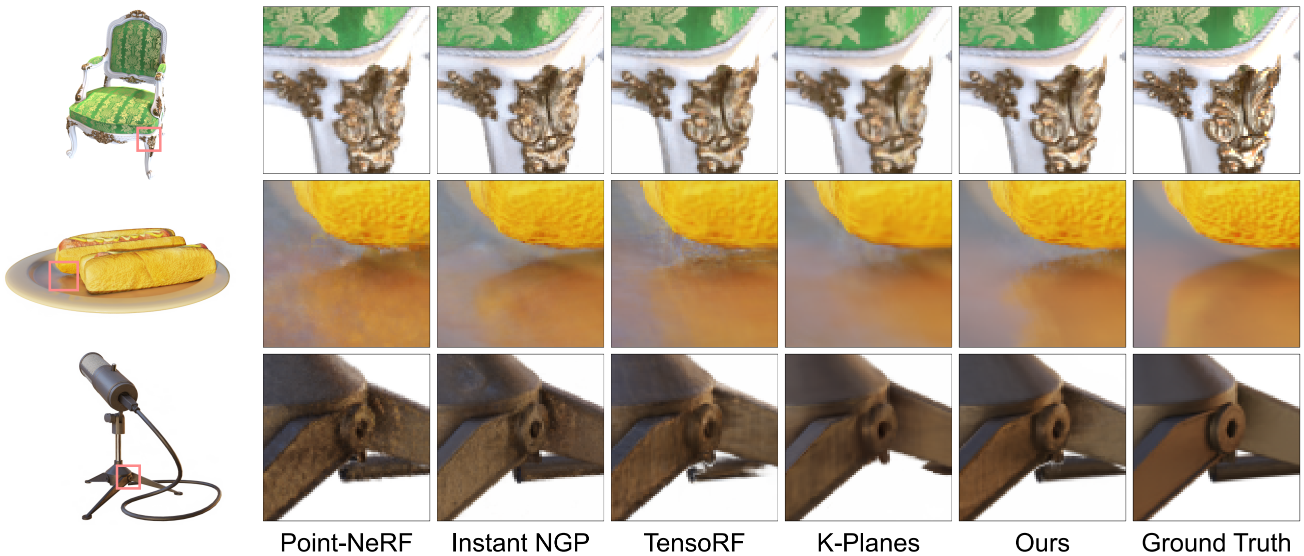

Figure 8. **Neural Radiance Field Reconstruction.** Qualitative comparisons on the Synthetic NeRF Dataset [47]. Leftmost column shows the full-image results of our method.

Table 3 comprehensively compares training time, number of parameters and novel view rendering metrics. Our method surpasses competing methods by a noticeable margin in rendering accuracy. Fig. 8 reflects the higher quality of our results, which contain more accurate details and fewer artifacts. Meanwhile, our method retains a moderate model size (same as TensoRF [9]) and comparable training time. After reducing to 3.66M parameters, our model still achieves high rendering accuracy and outperforms other methods that use more parameters (Plenoxels [26], Instant NGP [48], TensoRF [9], Factor Fields [10], K-Planes [25]). Fig. 9 compares the novel view synthesis accuracy with representative methods (Instant NGP [48], TensoRF [9]) under similar parameter count. Our method consistently performs better than the other two and also achieves higher PSNR than vanilla NeRF [47] when using the same number of parameters.

Forward-Facing Scenes. We use the LLFF dataset [46] which contains 8 real unbounded forward-facing scenes. In this experiment, we explore using K-Planes [25] as the grid-based part . As shown in Table 4, our approach achieves the highest PSNR and second-best SSIM. Although Mip-NeRF 360 has a higher score in SSIM, its training time is 7 times longer than ours. Compared to Plenoxels and TensoRF, our method has higher rendering accuracy, fewer parameters and comparable training speed. Fig. 10 shows example novel view synthesis results, where ours contain fewer visual artifacts.

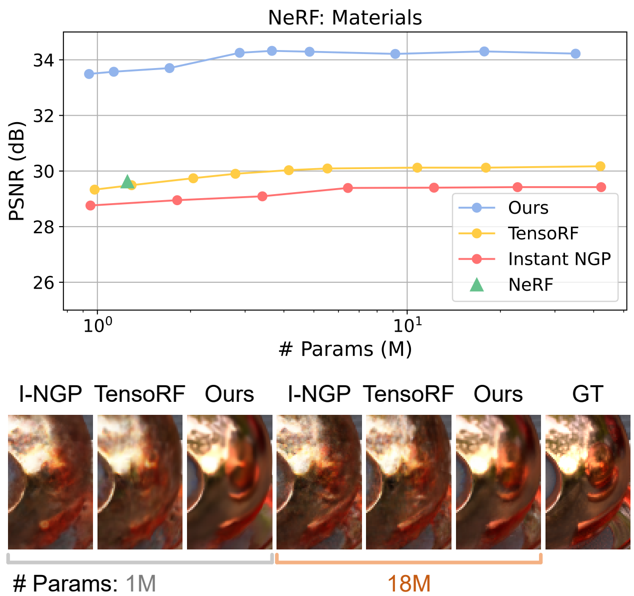

Figure 9. **Neural Radiance Field Reconstruction.** We compare the novel view synthesis quality under different parameter count on the "Materials" scene. Top is a quantitative comparison of rendering PSNR. Bottom is a qualitative comparison between Instant NGP ("I-NGP") [48], TensoRF [9] and ours at 1M and 18M parameters.

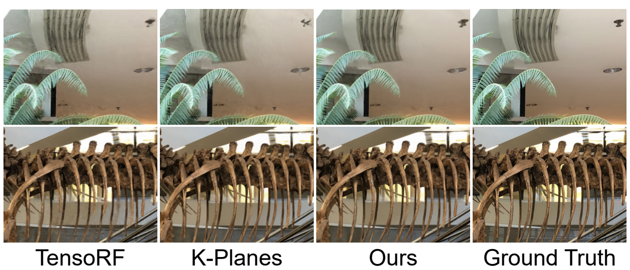

Figure 10. **Neural Radiance Field Reconstruction.** Qualitative comparisons on the LLFF Dataset [46].

	Time↓	# Params↓	PSNR↑	SSIM↑
NeRF [47]	36h	1.25M	26.50	0.811
Mip-NeRF 360 [4]	3.8h	3.23M	26.86	0.858
Plenoxels [26]	24m	$\sim$ 500M	26.29	0.839
TensoRF [9]	25m	45M	26.73	0.839
K-Planes [25]	33m	18.7M	26.92	0.847
Ours	31m	18.7M	27.05	0.849

Table 4. **Neural Radiance Field Reconstruction.** Quantitative comparisons on the LLFF Dataset [46].

5.4. Ablation Study

In Table 5, we conduct ablation study on adaptive RBFs (A-RBF) and multi-frequency sinusoidal composition (MSC) using the DIV2K validation set [3, 71] and the 3D shapes in Sec. 5.2. All image models are trained for 3500 steps and all SDF models are trained for 20000 steps. To demonstrate the effectiveness of sinusoidal composition in our framework, we further include variants that replace it with positional encoding [47] (Ours-PE) and sinusoid activation [62] (Ours-SIREN). For Ours-PE, we apply positional encoding [47] (PE) on input coordinate $\mathbf{x}$ and concatenate the features with $g_b(\mathbf{x})$ before input to the decoder network g_m. For Ours-SIREN, we apply sinusoidal activation [62] to the hidden layers in g_m, and use the method in [62] to initialize fully-connected layers. As shown in Table 5, without adaptive RBFs and sinusoidal composition, there is a noticeable drop in accuracy. Compared to PE and SIREN, our multi-frequency sinusoidal composition technique achieves higher performance.

	2D Images		3D SDF	
	PSNR↑	SSIM↑	IoU↑	NAE↓
No A-RBF	42.37	0.9918	0.9994	5.70
No MSC on RBF	48.19	0.9940	**0.9995**	5.04
No MSC on Feat.	48.46	0.9935	**0.9995**	5.09
No MSC on Both	43.81	0.9870	**0.9995**	5.16
Ours Full	**51.53**	**0.9961**	**0.9995**	**4.93**
Ours-PE	43.72	0.9870	0.9994	5.46
Ours-SIREN	45.98	0.9920	0.9994	5.69

Table 5. **Ablation Study.** We ablate on the adaptive RBFs (A-RBF) and multi-frequency sinusoidal composition (MSC). "Ours-PE" replaces MSC with positional encoding [47]. "Ours-SIREN" replaces MSC with sinusoid activation [62].

6. Conclusion

We have proposed NeuRBF, which provides accurate and compact neural representations for signals. We demonstrate that by simultaneously exploiting the spatial adaptivity and frequency extension of radial basis functions, the representation ability of neural fields can be greatly enhanced. To effectively adapt radial basis functions to target signals, we further devise tailored weighting schemes. Our method achieves higher accuracy than state-of-the-arts on 2D shape fitting, 3D signed distance field reconstruction, and neural radiance field reconstruction, while using same or fewer parameters. We believe our framework is a valuable step towards more expressive neural representations.

By far, we have not explored generalized learning, which would be a promising extension of our framework. Another future direction would be incorporating dictionary learning to further increase model compactness.

Acknowledgements

The authors thank the anonymous reviewers for their valuable feedback, and Anpei Chen and Zexiang Xu for helpful discussions.

References

[1] https://github.com/ashawkey/torch-ngp. 5

[2] https://www.turbosquid.com. 7

[3] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 5, 6, 9

[4] 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. 8, 9

[5] Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[6] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 3

[7] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 608–625. Springer, 2020. 1, 2

[8] 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

[9] 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, 3, 5, 7, 8, 9

[10] 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, 7, 8

[11] Changan Chen, Alexander Richard, Roman Shapovalov, Vamsi Krishna Ithapu, Natalia Neverova, Kristen Grauman, and Andrea Vedaldi. Novel-view acoustic synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6419, 2023. 1

[12] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 1

[13] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 2

[14] Zhang Chen, Anpei Chen, Guli Zhang, Chengyuan Wang, Yu Ji, Kiriakos N Kutulakos, and Jingyi Yu. A neural rendering framework for free-viewpoint relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5599–5610, 2020. 1

[15] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 2

[16] Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Häne, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3d shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13087–13096, 2021. 2, 3

[17] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6970–6981, 2020. 2

[18] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 264–280. Springer, 2022. 2

[19] Junwoo Cho, Seungtae Nam, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Streamable neural fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*, pages 595–612. Springer, 2022. 2

[20] Eastman Kodak Company. Kodak lossless true color image suite. https://r0k.us/graphics/kodak/. 6

[21] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsdf. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 51–67. Springer, 2020. 2

[22] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Golinski, Y Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *Transactions on Machine Learning Research*, 2022(11), 2022. 1

[23] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2021. 2

[24] Jie Feng, Ruimin Feng, Qing Wu, Zhiyong Zhang, Yuyao Zhang, and Hongjiang Wei. Spatiotemporal implicit neural representation for unsupervised dynamic mri reconstruction. *arXiv preprint arXiv:2301.00127*, 2022. 1

[25] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 1, 3, 5, 8, 9

[26] 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. 1, 2, 3, 8, 9

[27] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 3

[28] Kang Han and Wei Xiang. Multiscale tensor decomposition and rendering equation encoding for view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4232–4241, 2023. 2

[29] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Implicit neural representations with levels-of-experts. In *Advances in Neural Information Processing Systems*, 2022. 2

[30] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Sape: Spatially-adaptive progressive encoding for neural optimization. *Advances in Neural Information Processing Systems*, 34:8820–8832, 2021. 2

[31] Langwen Huang and Torsten Hoefler. Compressing multidimensional weather and climate data into neural networks. *arXiv preprint arXiv:2210.12538*, 2022. 1

[32] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 1, 2

[33] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. Relu fields: The little non-linearity that could. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2

[34] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 5

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[36] Zoe Landgraf, Alexander Sorkine Hornung, and Ricardo Silveira Cabral. Pins: progressive implicit networks for multiscale neural representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 11969–11984, 2022. 2

[37] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000. 7

[38] Tianyang Li, Xin Wen, Yu-Shen Liu, Hua Su, and Zhizhong Han. Learning deep implicit functions for 3d shapes with dynamic code clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12840–12850, 2022. 3

[39] Zhong Li, Liangchen Song, Celong Liu, Junsong Yuan, and Yi Xu. Neulf: Efficient novel view synthesis with neural 4d light field. In *Eurographics Symposium on Rendering*, 2022. 1

[40] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16252–16262, 2022. 2, 6

[41] 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

[42] Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG)*, 40:1 – 13, 2021. 3

[43] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021. 2

[44] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 1, 2

[45] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019. 2

[46] 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. 8, 9

[47] 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, 4, 7, 8, 9

[48] 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. 1, 3, 5, 6, 7, 8, 9

[49] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 1

[50] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1, 2

[51] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European*

Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 523–540. Springer, 2020. 1, 2, 3

[52] Sameera Ramasinghe and Simon Lucey. Learning positional embeddings for coordinate-mlps. *arXiv preprint arXiv:2112.11577*, 2021. 2

[53] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: towards a unifying framework for activations in coordinate-mlps. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 142–158. Springer, 2022. 2

[54] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 2

[55] Daniel Rho, Byeonghyeon Lee, Seungtae Nam, Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Masked wavelet representation for compact neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20680–20690, 2023. 3

[56] Radu Alexandru Rosu and Sven Behnke. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8475, 2023. 3

[57] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 2

[58] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 318–333. Springer, 2022. 2, 3, 5, 6

[59] Friedhelm Schwenker, Hans A Kestler, and Günther Palm. Three learning phases for radial-basis-function networks. *Neural networks*, 14(4-5):439–458, 2001. 4

[60] Shayan Shekarforoush, David Lindell, David J Fleet, and Marcus A Brubaker. Residual multiplicative filter networks for multiscale reconstruction. *Advances in Neural Information Processing Systems*, 35:8550–8563, 2022. 2

[61] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 3

[62] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 1, 2, 4, 9

[63] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 1

[64] Liangchen Song, Zhong Li, Xuan Gong, Lele Chen, Zhang Chen, Yi Xu, and Junsong Yuan. Harnessing low-frequency neural fields for few-shot view synthesis. *arXiv preprint arXiv:2303.08370*, 2023. 1

[65] Stanford University. The Stanford 3d scanning repository. `https://graphics.stanford.edu/data/3Dscanrep`. 7

[66] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 74–91. Springer, 2022. 1

[67] 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. 1, 2

[68] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 3

[69] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 2, 7

[70] 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. 1, 2, 4

[71] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 5, 6, 9

[72] Peng-Shuai Wang, Yang Liu, Yu-Qi Yang, and Xin Tong. Spline positional encoding for learning 3d implicit signed distance fields. 2021. 2

[73] Francis Williams, Zan Gojcic, Sameh Khamis, Denis Zorin, Joan Bruna, Sanja Fidler, and Or Litany. Neural fields as learnable kernels for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18500–18510, 2022. 2

[74] Francis Williams, Matthew Trager, Joan Bruna, and Denis Zorin. Neural splines: Fitting 3d surfaces with infinitely-wide neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9949–9958, 2021. 2

[75] Jelmer M Wolterink, Jesse C Zwienenberg, and Christoph Brune. Implicit neural representations for deformable image registration. In *International Conference on Medical Imaging with Deep Learning*, pages 1349–1359. PMLR, 2022. 1

[76] Zhijie Wu, Yuhe Jin, and Kwang Moo Yi. Neural fourier filter bank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14153–14163, 2023. 3

[77] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in neural information processing systems*, 32, 2019. 2

[78] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 3, 5, 7, 8

[79] Xiangyu Xu, Lichang Chen, Changjiang Cai, Huangying Zhan, Qingan Yan, Pan Ji, Junsong Yuan, Heng Huang, and Yi Xu. Dynamic voxel grid optimization for high-fidelity rgb-d supervised surface reconstruction. *arXiv preprint arXiv:2304.06178*, 2023. 1

[80] Guandao Yang, Sagie Benaim, Varun Jampani, Kyle Genova, Jonathan Barron, Thomas Funkhouser, Bharath Hariharan, and Serge Belongie. Polynomial neural fields for subband decomposition and manipulation. *Advances in Neural Information Processing Systems*, 35:4401–4415, 2022. 2, 6

[81] Runzhao Yang, Tingxiong Xiao, Yuxiao Cheng, Qianni Cao, Jinyuan Qu, Jinli Suo, and Qionghai Dai. Sci: A spectrum concentrated implicit neural compression for biomedical data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4774–4782, 2023. 1

[82] Runzhao Yang, Tingxiong Xiao, Yuxiao Cheng, Jinli Suo, and Qionghai Dai. Tinc: Tree-structured implicit neural compression. pages 18517–18526, 2023. 3

[83] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1

[84] 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

[85] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19228–19238, 2022. 2

[86] Yunfan Zhang, Ties van Rozendaal, Johann Brehmer, Markus Nagel, and Taco Cohen. Implicit neural video compression. *arXiv preprint arXiv:2112.11312*, 2021. 1

[87] Jianqiao Zheng, Sameera Ramasinghe, and Simon Lucey. Rethinking positional encoding. *arXiv preprint arXiv:2107.02561*, 2021. 2