BACON: Band-limited Coordinate Networks for Multiscale Scene Representation

David B. Lindell Dave Van Veen Jeong Joon Park Gordon Wetzstein
Stanford University
http://computationalimaging.org/publications/bacon

Abstract

Coordinate-based networks have emerged as a powerful tool for 3D representation and scene reconstruction. These networks are trained to map continuous input coordinates to the value of a signal at each point. Still, current architectures are black boxes: their spectral characteristics cannot be easily analyzed, and their behavior at unsupervised points is difficult to predict. Moreover, these networks are typically trained to represent a signal at a single scale, so naive downsampling or upsampling results in artifacts. We introduce band-limited coordinate networks (BACON), a network architecture with an analytical Fourier spectrum. BACON has constrained behavior at unsupervised points, can be designed based on the spectral characteristics of the represented signal, and can represent signals at multiple scales without per-scale supervision. We demonstrate BACON for multiscale neural representation of images, radiance fields, and 3D scenes using signed distance functions and show that it outperforms conventional single-scale coordinate networks in terms of interpretability and quality.

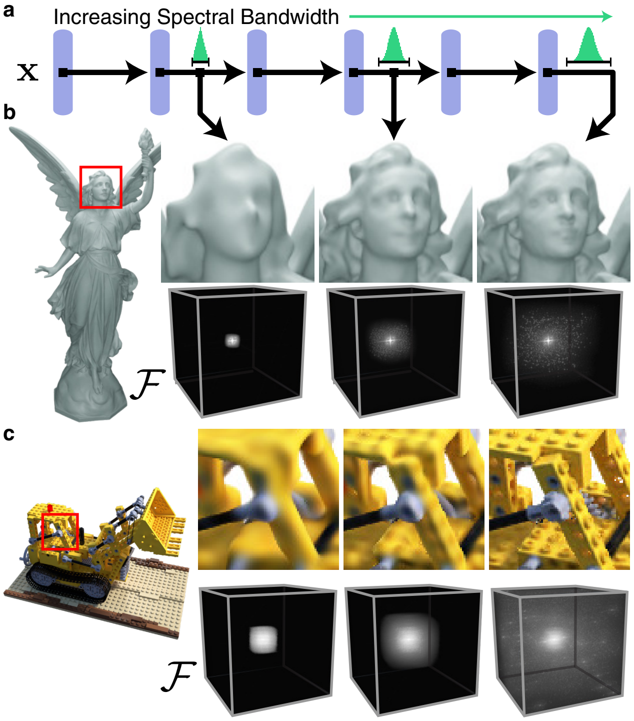

Figure 1. Overview of band-limited coordinate networks (BACON). (a) The proposed architecture produces intermediate outputs with an analytical spectral bandwidth that can be specified at initialization. When supervised on a high-resolution signal, the network learns a multi-resolution decomposition of the output, as shown for fitting 3D shapes via a signed distance function (b) and radiance fields (c). The network is characterized entirely by its Fourier spectrum (see insets) so its behavior is constrained, even at unsupervised locations.

1. Introduction

Coordinate networks are an emerging class of neural networks that can be used to represent or optimize a broad format of signals including images, video, 3D models, audio waveforms, and more [41, 44, 52, 64, 66]. As opposed to storing discrete samples of signals in conventional array- or grid-based formats, neural representations approximate signals using a continuous function that is embedded in the learned weights of a fully-connected neural network. Given an input coordinate, these networks are trained to output the value of a signal at that point. Since even complex or high-dimensional signals can be flexibly optimized using a coordinate network, they have become popular for applications including view synthesis [44], image processing [60], 3D reconstruction [52], and neural rendering [70].

Yet, current coordinate networks are black box models that are designed to represent signals at a single scale. As a result, the behavior of the network at unsupervised coordinates is difficult to predict, with complex dependencies on hyperparameters such as hidden layer size, network depth, or input coordinate encoding. The black box nature of the architecture similarly inhibits multiscale signal representation, since we cannot readily filter or anti-alias these models, and the frequency spectrum of a coordinate network is difficult to analyze. Thus naive downsampling or upsampling by querying the network on a coarser or finer grid of coordinates leads to aliasing or undesired high-frequency artifacts. Ultimately, these characteristics stem from the fact that coordinate networks are not amenable to Fourier analysis and are not designed to be scale aware.

Still, being able to represent and optimize signals at multiple resolutions is an important requirement for many applications. For example in image processing, many techniques rely on image pyramids [62] (e.g., optical flow estimation, compression, filtering, etc.). Representing 3D objects or scenes at multiple levels of detail is useful for speeding up rendering and reducing memory requirements (e.g., mipmapping).

In this work, we introduce band-limited coordinate networks (BACON). The key properties of this architecture are that (1) the maximum frequency at each layer can be manipulated analytically, and (2) the behavior of a trained network is entirely characterized by its Fourier spectrum. BACON is suited to multiscale signal representation because band-limited output layers can be designed with an inductive bias towards a particular resolution or scale.

In addition to introducing BACON, we demonstrate a variety of applications including multiscale representation of images, neural radiance fields, and 3D scenes. Our work takes important steps towards making coordinate-based networks scale aware, and provides a new representation with interpretable behavior. Specifically, we make the following contributions:

- We introduce band-limited coordinate-based networks for representing and optimizing signals.
- We develop methods for spectral analysis of the architecture, and propose a principled, band-limited initialization scheme.
- We demonstrate that our architecture outperforms conventional single-scale coordinate networks for multiscale image fitting, neural rendering, and 3D scene representation.

2. Related Work

Neural Scene Representation and Rendering. Emerging neural scene representations promise 3D-structure-aware, continuous, memory-efficient representations for parts [20, 21], objects [3, 6, 13, 22, 42, 52, 78], or scenes [15, 26, 55, 64, 66]. These can be supervised with 3D data, such as point clouds, and optimized as either signed distance functions [3, 22, 26, 30, 42, 52, 55, 63, 66, 68, 79] or occupancy networks [10, 41]. Using neural rendering [70, 71], representation networks can also be trained using multiview 2D images [4, 19, 27, 33–35, 38, 44, 45, 48–50, 56, 60, 66, 67, 74, 77, 78, 83, 84]. Temporally aware extensions [47] and multimodal variants with part-level semantic segmentation [32] have also been proposed. Recent 2D GANs have analyzed the bandwidth of convolutional layers for image generation [28], and 3D-aware GANs use related ideas but are trained with 2D image collections [7, 8, 14, 46, 51, 61].

Architectures for Scene Representation. Neural network architectures for scene representation networks can be roughly classified as feature-based, coordinate-based, or hybrid. Feature-based approaches represent the scene using differentiable feature primitives, such as points [16, 54, 57, 75, 81], surface patches [80], meshes [23, 59, 72, 85], multiplane [18, 43, 86] or multi-sphere [2, 5] images, or using a voxel grid of features [36, 65]. A tradeoff with feature-based representations is that they can be quickly evaluated, but typically have a large memory footprint.

Coordinate-based representations (sometimes called implicit representations or coordinate networks), use a multilayer perceptron (MLP) to map input coordinates to a signal value, for example, the signed distance or occupancy of a 3D scene. These networks can represent signals globally [41, 52, 64, 69] or locally [6, 9, 26, 40, 58]. Some global networks, such as Fourier Features [69] and SIREN [64], have tunable parameters that bias the network to fitting low- or high-frequency signals [79], though without explicit control over the bandwidth.

Hybrid architectures combine feature-based and coordinate representations to achieve best of both worlds [24, 34, 37, 55]. These networks can represent complex, high-dimensional signals continuously across the input domain with a small memory footprint. The proposed method is also a coordinate network, but rather than using an MLP architecture, as with all coordinate networks discussed above, our method builds on recently proposed multiplicative filter networks (MFNs) [17]. We develop the theory of MFNs, with new tools to describe and manipulate the Fourier spectra of these networks, and a new initialization scheme that mitigates vanishing activations in deep networks. These insights enable band-limited coordinate networks, which we demonstrate for multiscale signal representation.

Multiscale Representations. Several existing works have explored multiscale architectures in the context of scene representation networks. For example, proposed methods use an octree [68, 82] to accelerate neural rendering of radiance fields or signed distance functions, or a hierarchy of features [11] to improve 3D shape completion. Multiscale representations can be optimized directly using specialized architectures [37, 79] or progressive training strategies [25, 37]. The closest work to ours in this category is Mip-NeRF [4], which is a coordinate-based network with a scale-dependent positional encoding. After training the network with supervision at multiple scales, the resolution of the network output can be controlled by adjusting the positional encoding. Our work differs in that the bandwidth of

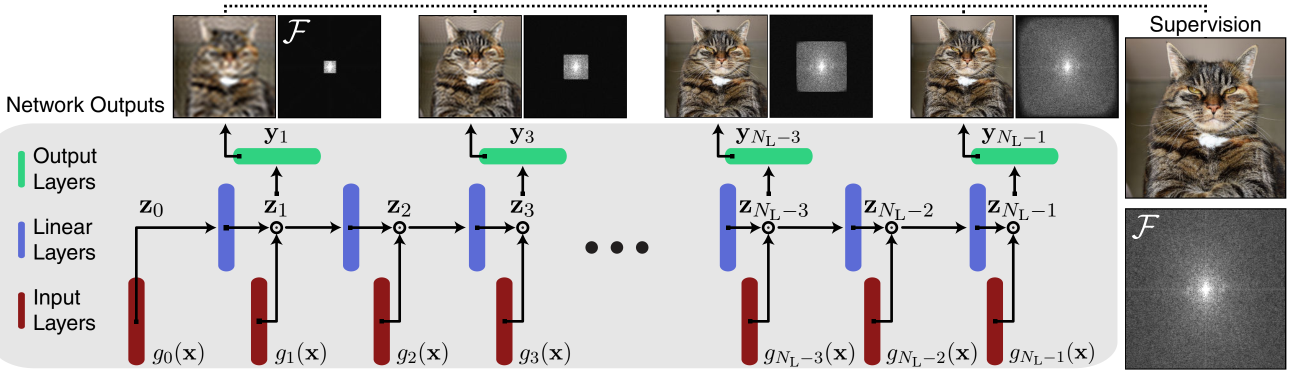

Figure 2. Overview of BACON architecture. We initialize the frequencies of the sine layers of a multiplicative filter network [17] within a limited bandwidth $[-B_i, B_i]$ (bottom row). Then, the bandwidth of each output layer is the sum of the input bandwidths up to that point (top row), allowing the network bandwidth to be explicitly specified. At training time the network can be supervised with a signal at any resolution, and the network learns to fit the signal in a band-limited fashion. Image from DIV2K dataset [1].

the network outputs are constrained by design rather than through training. Thus, our approach learns a band-limited multiscale decomposition of a signal, even without explicit training at multiple scales.

3. Method

This section provides an overview of MFNs and the BACON multiscale architecture, describes the Fourier spectra of these networks, and proposes an initialization scheme for deep networks.

3.1. Band-limited Coordinate Networks

Our approach builds on a recently introduced coordinate-based architecture called Multiplicative Filter Networks (MFNs) [17], which differ from conventional MLPs in that they employ a Hadamard product between linear layers and sine activation functions. While BACON uses an MFN backbone, we significantly extend the theoretical understanding and practicality of these networks by (1) proposing architectural changes to achieve multiscale, band-limited outputs, (2) deriving formulas to quantify the expected frequencies in the representation, and (3) deriving a principled initialization scheme that prevents vanishing activations in deep networks.

In a forward pass through the network, an input coordinate $\mathbf{x} \in \mathbb{R}^{d_\mathrm{in}}$ is first passed through several layers of the form $g_i : \mathbb{R}^{d_\mathrm{in}} \mapsto \mathbb{R}^{d_\mathrm{h}}$, with $g_i(\mathbf{x}) = \sin(\boldsymbol{\omega}_i \mathbf{x} + \boldsymbol{\phi}_i)$, $i = 0, \ldots, N_\mathrm{L} - 1$, and N_L the number of layers in the network. We refer to the intermediate activations as $\mathbf{z}_i \in \mathbb{R}^{d_\mathrm{h}}$, and we allow intermediate outputs of the network $\mathbf{y}_i \in \mathbb{R}^{d_\mathrm{out}}$ at the ith layer, defined as follows (see also Fig. 2).

$$\begin{aligned} \mathbf{z}_0 &= g_0(\mathbf{x}) \\ \mathbf{z}_i &= g_i(\mathbf{x}) \circ (\mathbf{W}_i \mathbf{z}_{i-1} + \mathbf{b}_i), \quad 0 \le i < N_\mathrm{L} \\ \mathbf{y}_i &= \mathbf{W}_i^\mathrm{out} \mathbf{z}_i + \mathbf{b}_i^\mathrm{out}, \end{aligned} \tag{1}$$

where $\circ$ indicates the Hadamard product. The parameters of the network are $\theta = \{\boldsymbol{\omega}_i \in \mathbb{R}^{d_\mathrm{h} \times d_\mathrm{in}}, \mathbf{b}_i, \boldsymbol{\phi}_i \in \mathbb{R}^{d_\mathrm{h}}, \mathbf{W}_i \in \mathbb{R}^{d_\mathrm{h} \times d_\mathrm{h}}, \mathbf{W}_i^\mathrm{out} \in \mathbb{R}^{d_\mathrm{out} \times d_\mathrm{h}}, b_i^\mathrm{out} \in \mathbb{R}^{d_\mathrm{out}}\}$.

A useful property of this formulation is that the network output can be expressed equivalently as a sum of sines with varying amplitude, frequency, and phase [17].

$$\mathbf{y}_i = \sum_{j=0}^{N_\mathrm{sine}^{(i)}-1} \bar{\alpha}_j \sin(\bar{\boldsymbol{\omega}}_j \mathbf{x} + \bar{\phi}_j), \tag{2}$$

where $\bar{\alpha}_i$, $\bar{\boldsymbol{\omega}}_i$, and $\bar{\phi}_i$ depend on the parameters of the MFN (see supplemental §1.2), and the number of terms in the sum for an N_L layer network is given as (see supplemental §1.1)

$$N_\mathrm{sine}^{(N_\mathrm{L})} = \sum_{i=0}^{N_\mathrm{L}-1} 2^i d_\mathrm{h}^{i+1}. \tag{3}$$

This property stems from the repeated Hadamard product of sines and the trigonometric identity that

$$\sin(a)\sin(b) = \frac{1}{2}\left(\sin(a + b - \pi/2) + \sin(a - b + \pi/2)\right). \tag{4}$$

By applying this identity through the layers of the network, the output can be reduced to a single sum of sines.

3.2. Frequency Spectrum

We exploit the property that MFNs can be expressed as a sum of sines to create band-limited networks. This is achieved by designing the architecture so that the frequency of all represented sines never exceeds a desired threshold.

To this end, we freeze (i.e., do not optimize) the frequencies, or entries of $\boldsymbol{\omega}_i$, and set them to a bandwidth in $[-B_i, B_i]$ using random uniform initialization. Then, since the Hadamard products of sines result in summed frequencies (Eq. 4), the total bandwidth of an output at layer i of the network is less than or equal to $\sum_{j=0}^{i} B_j$ and the maximum bandwidth is $B = \sum_{i=0}^{N_\text{L}-1} B_i$ (see Fig. 2).

When representing signals across a finite input domain, e.g., with input coordinates $\mathbf{x} \in [-0.5, 0.5]^{d_\text{in}}$, it is not necessary to represent all frequencies continuously. Instead, we can assume that the represented signal is periodic, so we are only required to represent discrete frequency values whose spacing is $1/T$, where T is the periodicity or extent of the signal in the primal domain. Moreover, using discrete frequencies allows complete characterization of the network spectrum by applying a fast Fourier transform to a uniformly sampled network output (shown in Fig. 2 for image fitting).

We also analyze the distribution of sine frequencies in the network. Briefly, sines in the network can be associated with one of the N_L terms in the summation of Eq. 3. Then, considering the probability of sines originating from each term results in a compound random variable that gives the overall distribution of frequencies. We provide an extended derivation in the supplemental, showing that the distribution is approximately zero-mean Gaussian with variance

$$\text{Var}(\boldsymbol{\omega}_i) \cdot \sum_{m=0}^{N_\text{L}-1} m \cdot \frac{2^{N_\text{L}-1-m} d_\text{h}^{N_\text{L}-m}}{\sum\limits_{i=0}^{N_\text{L}-1} 2^i d_\text{h}^{i+1}}. \qquad (5)$$

The Gaussian distribution of frequencies results in a greater parameterization of low frequencies in the network; this may be a useful inductive bias since low-frequency Fourier coefficients typically have a greater amplitude than high-frequency coefficients in natural signals [73].

To facilitate representing signals at multiple resolutions, we introduce linear layers at intermediate stages throughout the network to extract band-limited outputs (see Fig. 2). By supervising the outputs of these layers, we can train BACON to fit a signal at multiple scales simultaneously. Interestingly, because the outputs are band-limited, BACON can be trained in a semi-supervised fashion where the bandwidth of the supervisory signal need not match the desired bandwidth of the output of the network, demonstrated in Fig. 2 for image fitting.

3.3. Initialization Scheme

Finally, we derive a principled initialization scheme that ensures the distribution of activation functions at the output of each layer is distributed uniformly at the beginning of training. While the proposed scheme and that of SIREN [64] both involve sine non-linearities, our initialization explicitly accounts for Hadamard products in the architecture and the distribution of inputs to sine layers g_i. We compare our initialization scheme to the initialization proposed by Fathony et al. [17] in Fig. 3. Our proposed scheme resolves a problem with vanishingly small activations for deep networks and results in standard normal distributed activations after each linear layer.

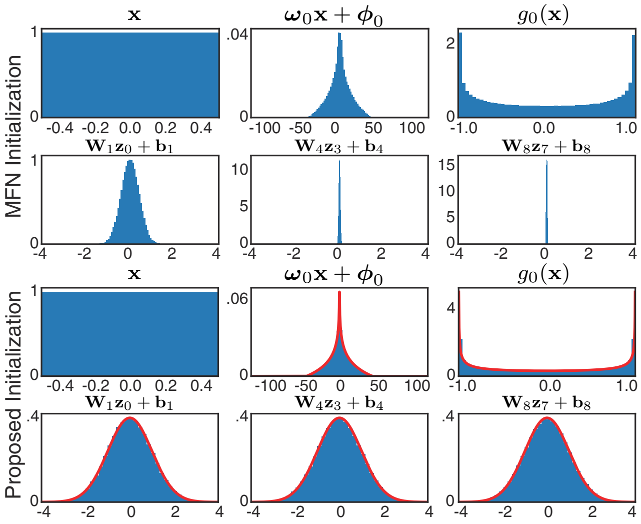

Figure 3. Comparison of distribution of activations at initialization. The initialization scheme proposed for MFNs [17] results in vanishingly small activations for deep networks (shown for layers 0, 1, 4, and 8). The proposed initialization scheme maintains a standard normal distribution after each linear layer (all distributions shown for a network with $d_\text{h} = 1024$), and activations at intermediate outputs closely match our analytical derivations (red lines, see supplemental for details).

In the supplemental, we provide an extended derivation, which we summarize as follows. Assume the input to the network is uniformly distributed $\mathbf{x} \sim \mathcal{U}(-0.5, 0.5)$, with $\boldsymbol{\omega}_i \sim \mathcal{U}(-B_i, B_i)$ and $\boldsymbol{\phi}_i \sim \mathcal{U}(-\pi, \pi)$, where we describe the distribution of each element of the matrix or vector. Then, $\boldsymbol{\omega}_i\mathbf{x} + \boldsymbol{\phi}_i$ is distributed as

$$\begin{cases} 1/B_i \log\left(B_i/\min(|2x|, B_i)\right), & -B/2 \le x \le B/2 \\ 0 & \text{else} \end{cases}$$

and $g_i(\mathbf{x}) = \sin(\boldsymbol{\omega}_i\mathbf{x} + \boldsymbol{\phi}_i)$ is approximately arcsine distributed with variance 0.5 (see supplemental, red plots in Fig. 3). Now, let $\mathbf{W}_i \sim \mathcal{U}[-\sqrt{6/d_\text{h}}, \sqrt{6/d_\text{h}}]$. Then we have that $\mathbf{W}_1 g_0(\mathbf{x}) + \mathbf{b}_1$ converges to the standard normal distribution with increasing d_h (see supplemental). Finally, the Hadamard product $g_1(\mathbf{x}) \circ (\mathbf{W}_1\mathbf{z}_0 + \mathbf{b}_1)$ is the product of arcsine distributed and standard normal random variables which again has a variance of 0.5. Applying the next linear layer results in another standard normal distribution, which is also the case after all subsequent linear layers (see red plots of Fig. 3).

4. Experiments

We demonstrate BACON on three separate tasks: image fitting, view synthesis using neural radiance fields, and 3D shape fitting using signed distance functions.

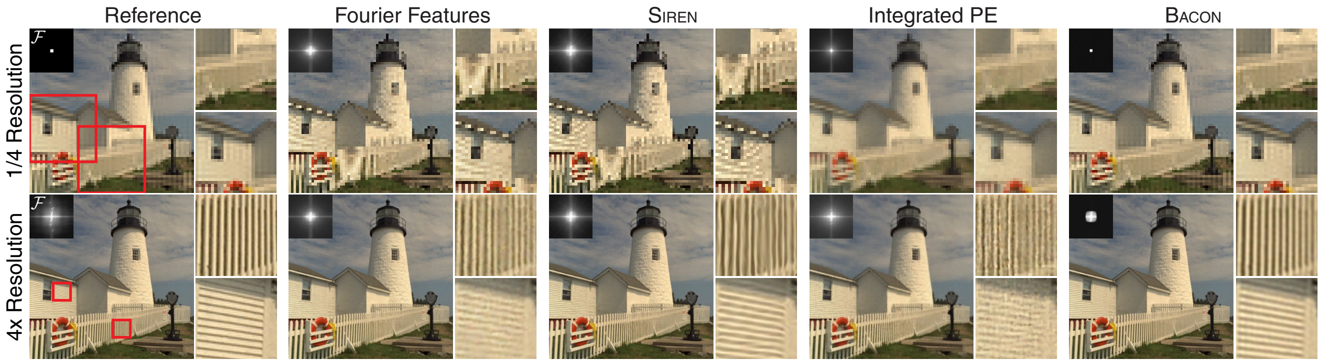

Figure 4. Image fitting results. We train networks using Fourier Features [69], SIREN [64], and integrated positional encoding (PE) [4] to fit an image at 256×256 (1×) resolution. We show network outputs at 1/4 and 4× resolution. Fourier Features and SIREN fit to a single scale and show aliasing when subsampled. Integrated PE is explicitly supervised at 1/4 and 1× resolution and learns reasonable anti-aliasing; however, all methods except BACON show high-frequency artifacts at 4× resolution (insets). BACON is supervised at a single scale and approximates low-pass filtered and high-resolution reference images (left column and Fourier spectra insets).

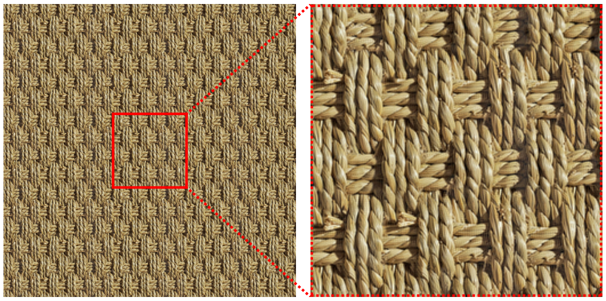

Figure 5. BACON periodic extrapolation behavior[1].

4.1. Image Fitting

We use an image fitting task to evaluate the performance of BACON and to demonstrate its band-limited behavior. BACON is compared to three other baselines: a network with Gaussian Fourier Features positional encoding [69], SIREN [64], and the integrated positional encoding of Mip-NeRF [4], which is scale-dependent.

We initialize all networks with 4 hidden layers, 256 hidden features, and we train on the 256×256 resolution image for 5000 iterations using PyTorch [53] and Adam [31]. The batch size is equal to the number of image pixels. For Fourier Features and integrated positional encoding, we use encoding scales of 6 and 10, respectively, to balance between image quality and high-frequency overfitting. For SIREN, we initialize the frequency parameter to $\omega_0 = 30$.

Fourier Features and SIREN are trained to minimize the loss $\mathcal{L}_{\text{img}} = \|\mathbf{y} - \mathbf{y}_{\text{GT}}\|_2^2$, where $\mathbf{y}$ is the network output and $\mathbf{y}_{\text{GT}}$ are the image pixel values. For BACON, we sum this loss over all network outputs, with explicit supervision at all scales on the full-resolution image. The integrated positional encoding network is supervised explicitly on anti-aliased image pixels at 1/4, 1/2, and full resolution, following Barron et al. [4]. Finally, we initialize BACON to have a maximum bandwidth B of 0.5 cycles/pixel, which is the Nyquist limit for the image. The frequencies $\boldsymbol{\omega}_i$ are initialized so that the outputs $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_4$ are constrained to quarter, half, or full bandwidth. That is, $B_0 = B_1 = B/8$, and $B_2 = B_3 = B_4 = B/4$ such that $\sum_i B_i = B$.

Results of image fitting on a test scene from the Kodak dataset [12] are shown in Fig. 4. Since Fourier Features and SIREN only represent the signal at the trained resolution, sampling the network at 1/4 resolution results in aliasing. We show the interpolation performance of these networks by evaluating a 4× upsampled grid of 1024×1024 pixels. When upsampled, BACON does not synthesize spurious high frequencies and has a band-limited output. All other methods have non-zero high-frequency spectra and exhibit artifacts in the reconstruction. We show additional image fitting experiments in the supplemental, including evaluation of deep 8- and 16-layer BACONs and MFNs.

Periodic Extrapolation. Since BACON uses discrete frequencies at each sine layer $g_i(\mathbf{x})$, the representation is periodic. We demonstrate this by fitting a seamless texture using coordinates $\mathbf{x} \in [-0.5, 0.5]$ (red square of Fig. 5) and querying the network output for $\mathbf{x} \in [-2, 2]$.

Scale Interpolation. Although BACON outputs at discrete scales, we can interpolate between multiscale outputs, similar to the trilinear filtering used to render from mipmaps [76]. See supplemental for additional details and results.

4.2. Neural Radiance Fields

Neural radiance fields (NeRF) [44] have become a popular method for view synthesis and neural rendering. The method operates on a dataset of multiview images with known camera positions, where each image pixel is associated with a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ that extends from the camera center of projection $\mathbf{o}$ in the direction $\mathbf{d}$ passing through the pixel. A pixel color $\mathbf{C}(\mathbf{r})$ is predicted using the volume

[1] *Image:* https://www.sketchuptextureclub.com/

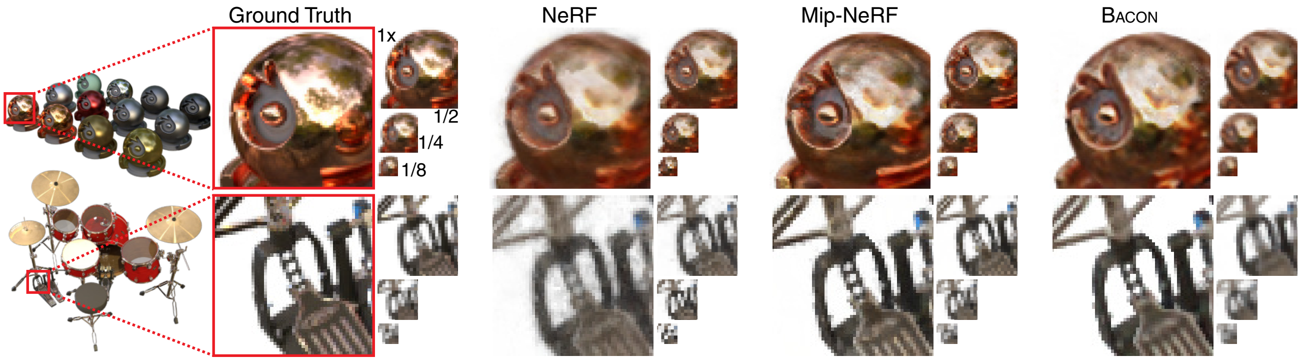

Figure 6. Neural rendering results. We compare NeRF [44], Mip-NeRF [4], and BACON supervised on a multiscale synthetic dataset [4]. BACON captures higher frequency details better than NeRF while requiring fewer parameters to render at 1/2, 1/4, and 1/8 resolution.

	PSNR ↑					# Params.			
	1×	1/2	1/4	1/8	Avg.	1×	1/2	1/4	1/8
NeRF	26.734	28.941	29.297	26.464	27.859	511K			
Mip-NeRF	29.874	31.307	32.093	32.832	**31.526**	511K			
BACON	27.430	28.066	28.520	28.475	<u>28.123</u>	531K	398K	266K	133K

Table 1. Performance of NeRF, Mip-NeRF, and BACON averaged across the multiscale Blender dataset. BACON achieves better average performance than NeRF while requiring fewer parameters to render the lower resolution images.

rendering equation to integrate predicted intermediate values of color $\mathbf{c}$ and opacity σ along the ray [4]. In practice, a neural network is queried to evaluate samples of $\mathbf{c}$ and σ along each ray $\mathbf{r}(t)$, and the volume rendering integral is evaluated using quadrature as [39, 43]

$$\mathbf{C}(\mathbf{r},\mathbf{t}) = \sum_j T_j(1 - \exp(-\sigma_j(t_{j+1} - t_j)))\,\mathbf{c}_j, \\ \text{with} \quad T_j - \exp\left(-\sum_{i'<i} \sigma_{i'}(t_{i'+1} - t_{i'})\right), \tag{6}$$

where T_j represents the transmittance or visibility of a point on the ray, and the values $w_j = T_j(1-\exp(-\sigma_j(t_{i+1}-t_j)))$ can be interpreted as alpha compositing weights applied to the predicted colors $\mathbf{c}_j$. After training, novel views can be rendered by simply evaluating the corresponding rays.

We evaluate BACON for this task and compare to NeRF and Mip-NeRF baselines trained on a multiscale Blender dataset [4] with images at full (512×512), 1/2, 1/4, and 1/8 resolution. For the baselines, we use the implementations of Barron et al.[2] [4]. All networks are trained according to the procedure of Mip-NeRF; we use the Adam optimizer with a batch size of 4096 rays and 1e6 training iterations. The learning rate is annealed logarithmically from 1e-3 to 5e-6 for BACON and 5e-4 to 5e-6 for the baselines. All networks are composed of 8 hidden layers with 256 hidden features.

For BACON, we adapt the training procedure and architecture as follows. Rays within the multiscale Blender dataset fall within an 8 by 8 unit volume ($\mathbf{r}(t) \in [-4, 4]^3$), and we find that setting the maximum bandwidth B to 64 cycles per unit interval allows fitting high frequency image details. To simplify the training procedure, we evaluate all methods without the viewing direction input originally used for NeRF. This also enables visualization of the BACON Fourier spectrum (see Fig. 1). Thus the input to all networks is a 3D coordinate corresponding to the position along the ray $\mathbf{r}(t)$. BACON produces four outputs, one for each scale of the dataset: $\mathbf{y}_i$, $i \in [2, 4, 6, 8]$. The B_i constrain each output to 1/8, 1/4, 1/2, and full resolution, with $\sum_{i=0}^{2} B_i = B/8$, $\sum_{i=0}^{4} B_i = B/4$, and so on. We also adapt the hierarchical sampling procedure of NeRF [44], wherein the alpha compositing weights w_j from an initial forward pass are used to resample the ray in regions of non-zero opacity. To improve efficiency, we use the lowest-resolution output of the network for this initial forward pass and apply the following loss function on pixels rendered using the resampled rays with 256 samples.

$$\mathcal{L}_{\text{BACON}} = \sum_{i,j,k} \|(\mathbf{C}_k(\mathbf{r}_i, \mathbf{t}_j) - \mathbf{C}_{\text{GT},k}(\mathbf{r}_i)\|_2^2, \tag{7}$$

where i, j, and k index rays, ray positions, and dataset scales, respectively. For quantitative evaluation, we use per-scale supervision so the BACON outputs are directly comparable to the multiscale ground truth images. Finally, we adopt the regularization strategy of Hedman et al. [24] to penalize non-zero off-surface opacity (see supplemental for results without per-scale supervision and an ablation study).

Qualitative and quantitative evaluations of BACON for neural rendering, are shown in Fig. 6 and Table 1. BACON achieves better performance than NeRF trained on the multiscale dataset at 1/8 and 1× resolution. We report PSNR at each scale, averaged over all scenes in the multiscale Blender dataset in Table 1. In Fig. 6, we observe that BACON recovers higher frequency details compared to NeRF on the *Materials* and *Drums* scenes. Mip-NeRF incorporates an additional mechanism which changes the positional encoding along each ray to account for the expansion of the viewing frustum, and achieves the best performance. Still, we find that BACON produces high-quality results with a fraction of the parameters at low resolution.

[2] https://github.com/google/mipnerf

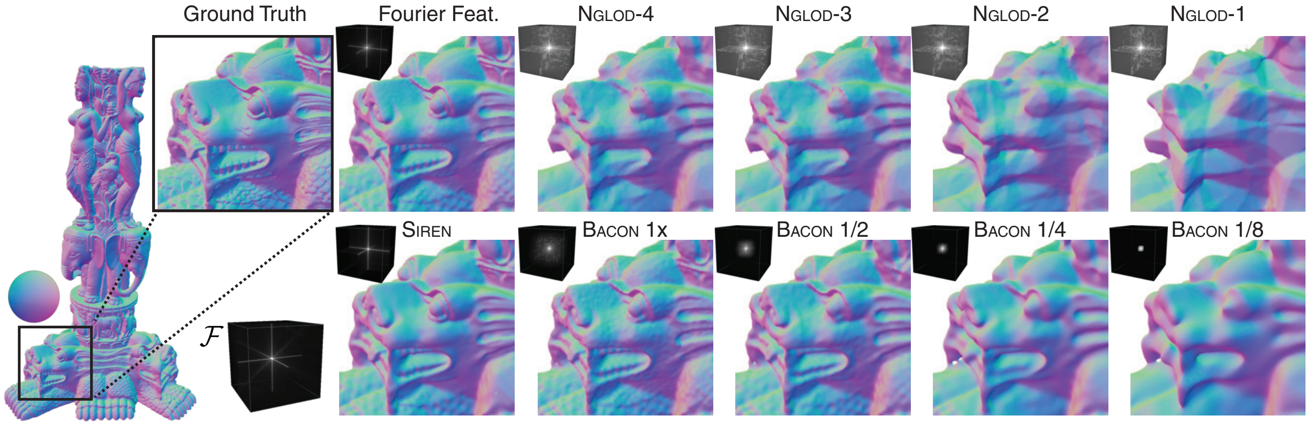

Figure 7. Shape fitting results. Results on the Thai Statue from the Stanford 3D Scanning Repository are shown for levels-of-detail 1–4 of Neural Geometric Level of Detail (NGLOD) [68], Fourier Features [69], SIREN [64], and BACON. All methods perform similarly at their highest detail output (see Table 2), but BACON learns a smooth multiscale decomposition of the shape. Insets show the spectra of the extracted signed-distance functions, revealing the band-limited output of BACON. Additional results included in the supplemental.

	FF	SIREN	NGLOD-4	NGLOD-5	BACON 1×
# Params.	527K	528K	1.35M	10.1M	531K
Chamfer↓	**2.166e-6**	2.780e-6	8.358e-6	2.422e-6	2.198e-6
IOU ↑	**9.841e-1**	9.751e-1	9.479e-1	9.811e-1	9.833e-1

Table 2. Shape fitting performance of Fourier Features [69], SIREN [64], Neural Geometric Level of Detail (NGLOD) [68], and BACON averaged across 5 test scenes (detailed in main text). All methods achieve roughly comparable performance, including BACON despite simultaneously representing multiple scales. Multiple levels of detail are shown for NGLOD, which requires more parameters to populate the explicit feature grids.

Additionally, we can use BACON to learn semi-supervised multiscale decompositions of the neural radiance fields. In this case, we train each output scale at the full resolution, and BACON automatically learns band-limited representations at the intermediate output layers. We show an example of this for the *Lego* scene in Fig. 1. Additional results for BACON in the explicitly supervised and semi-supervised cases are shown in the supplemental.

4.3. 3D Shape Representation

Neural representation networks have shown promise for representing and manipulating 3D shapes. BACON is well-suited for this task, and we evaluate its performance on a range of shapes from the Stanford 3D scanning repository[3].

We compare BACON to Fourier Features [69], SIREN [64], and Neural Geometric Level of Detail (NGLOD) [68], a representation which optimizes explicit features stored on a sparse voxel octree. We do not compare to Mip-NeRF since it requires per-scale supervision. All networks are trained to directly fit a signed distance function (SDF) estimated from a ground truth mesh.

For BACON, Fourier Features, and SIREN, we use networks with 8 hidden layers and 256 hidden features. For Fourier Features (Gaussian encoding), we set the encoding scale to 8 and for SIREN, we set $\omega_0 = 30$. We train on locations sampled from the zero level set and add Laplacian noise; this results in an exponential decay in the number of samples off the zero level, as proposed by Davies et al. [13]. We find that the width of the Laplacian distribution has a large impact on performance. Setting the variance σ_L^2 too small results in poor off-surface fitting, but setting the variance too high reduces the number of samples on the zero level set, degrading the appearance of the surface. Thus, we introduce a coarse and fine sampling procedure wherein we produce "fine" samples using a small variance of $\sigma_\text{L}^2 = 2\text{e-}6$ and "coarse" samples with $\sigma_\text{L}^2 = 2\text{e-}2$. Samples are drawn in the domain $[-0.5, 0.5]^3$, and we initialize the frequencies of BACON similar to the NeRF experiments (additional details in supplemental). We train using a loss function

$$\mathcal{L}_\text{SDF} = \lambda_\text{SDF} \|\mathbf{y}^c - \mathbf{y}_\text{GT}^c\|_2^2 + \|\mathbf{y}^f - \mathbf{y}_\text{GT}^f\|_2^2, \quad (8)$$

where $\mathbf{y}$ is the network output, $\mathbf{y}_\text{GT}$ represents the ground truth SDF values, the f and c superscripts indicate fine and coarse samples, and we set λ_SDF to 0.01 for all experiments. For BACON we compute this loss at all output scales.

We train Fourier Features, SIREN, and BACON on each dataset for 200,000 iterations with a batch size of 5,000 coarse and 5,000 fine SDF samples. Models are optimized using Adam [31], and we logarithmically anneal the learning rate of each method from 1e-2 (BACON), 1e-3 (Fourier Features), and 1e-4 (SIREN) to a final value of 1e-4 during the course of training. For NGLOD, we use the default training settings in the authors' code[4], which samples 500,000 points at each training epoch, uses a batch size of 512, and trains for 250 epochs. We train NGLOD models with a maximum of 4 or 5 levels of detail. Additional levels of detail result in improved performance, but require more memory.

[3] http://graphics.stanford.edu/data/3Dscanrep/

[4] https://github.com/nv-tlabs/nglod

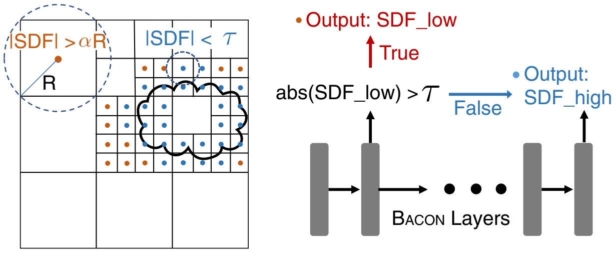

Figure 8. Adaptive-frequency multiscale SDF evaluation for fast mesh extraction. We propose a multi-scale evaluation which subdivides non-empty cells, i.e., when the SDF is smaller than the cell radius. We further accelerate the evaluation by using fewer layers (red) for regions that do not require high-frequency details. We evaluate the full network (blue) only when query locations are sufficiently close ($|\text{SDF}| < \tau$) to the surface.

	Dense Grid (512^3)	Adaptive-Frequency	Adaptive + Multiscale (Proposed)
Time (s)	17.91	5.50	**0.222**

Table 3. SDF evaluation time. The proposed Adaptive + Multiscale method achieves a roughly 80× speedup over naive evaluation of the SDF on a dense grid (averaged over 5 test scenes).

We fit each method to four scenes from the Stanford 3D Scanning Repository (*Armadillo*, *Dragon*, *Lucy*, and *Thai Statue*), as well as a simple sphere baseline (all objects are shown in the supplemental). The models are extracted at 512^3 resolution using marching cubes and evaluated using Chamfer distance and intersection over union (IOU), and we report these numbers averaged over the 5 scenes in Table 2. The highest resolution outputs of all methods achieve comparable performance, though note that NGLOD-5 requires over an order of magnitude more parameters than the other representations, and BACON achieves this despite representing all scales simultaneously.

We see similar qualitative trends in Fig. 7, with Fourier Features, SIREN, NGLOD-4, and BACON all producing detailed reconstructions of the *Thai Statue* scene. BACON produces a smooth reconstruction at multiple scales because of its band-limited output layers. This can be compared to the low-resolution outputs of NGLOD, which show fewer finer details, but also have coarse, angular artifacts from the ReLU non-linearity used in the network. This follows from the NGLOD frequency spectrum (see Fig. 7), which is non-zero for high frequencies, including at the coarsest scale.

Accelerated Marching Cubes. We observe that the band-limited, multi-output nature of our network allows efficient allocation of resources when evaluating SDFs on a dense grid for mesh extraction. The key idea is to use the lower-layer output of BACON when a cell is far away from the surface (Fig. 8). That is, we early-stop the computation within the network when $|\text{SDF}| < \tau$, where we set τ to 0.7× the finest voxel size. This adaptive computation significantly reduces the mesh extraction time (Table 3).

Moreover, we propose a multiscale approach for further acceleration. As SDF values indicate the distance to the closest surface, we can consider a cell to be empty when $|\text{SDF}| > \alpha\text{R}$, for circumsphere radius R and some $\alpha > 1$ that improves robustness to imperfect SDFs (we use $\alpha = 2$). Starting from the coarsest resolution grid, we subdivide a cell only when $|\text{SDF}| < \alpha\text{R}$ to prune empty space. Multiscale extraction approaches have been proposed for extracting occupancy fields from coordinate networks [41], but using an SDF facilitates pruning since each sample reveals a region of empty space.

We combine the two strategies by applying the adaptive-frequency evaluation on each level of the multiscale grids, leading to roughly 80× faster SDF evaluation than the naive approach as shown in Table 3 (see supplemental results, all timings evaluated on an NVIDIA RTX A6000 GPU).

5. Conclusion

In this work, we take steps towards making coordinate networks interpretable and scale aware. Our approach enables analyzing and controlling the spectral bandwidth of the network at intermediate layers, allowing multiscale signal representation, even without explicit supervision. Since we can characterize the bandwidth of the network using Fourier analysis, its behavior is provably constrained, even at unsupervised locations. Moreover, BACON's intermediate outputs help to improve inference times via adaptive frequency evaluation. We show that BACON outperforms other single-scale coordinate networks for multiscale image fitting, neural rendering, and 3D scene representation.

Limitations. We also highlight a few limitations of BACON and promising future directions. While we demonstrated fitting two- and three-dimensional signals, fitting signals in higher dimensions may require more parameters to achieve dense spectral coverage due to the curse of dimensionality. Still, it may be possible to optimize the initialization of frequencies in a way that maximizes spectral coverage and mitigates this challenge. Also, our current work is limited to single scene overfitting. However, many generative models work by increasing the frequency of the output using successive upsampling layers [29], which is similar in spirit to our method. Recent work on band-limited models for image synthesis has shown great promise [28], so applying BACON for generative modeling is an exciting area for future research.

Societal Impact. We condemn the misuse of scene representation networks, including BACON, for malicious deepfakes or spreading misinformation, and we emphasize the importance of research to thwart such efforts (see, e.g., Tewari et al. [70] for a discussion of related strategies).

Acknowledgments. This project was supported in part by a PECASE by the ARO, NSF award 1839974, Samsung GRO, and Stanford HAI.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, 2017. 3

[2] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *Proc. ECCV*, 2020. 2

[3] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proc. CVPR*, 2020. 2

[4] 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 *Proc. ICCV*, 2021. 2, 5, 6

[5] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Trans. Graph. (SIGGRAPH)*, 39(4), 2020. 2

[6] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *Proc. ECCV*, 2020. 2

[7] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *Proc. CVPR*, 2022. 2

[8] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *Proc. CVPR*, 2021. 2

[9] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proc. CVPR*, 2021. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. CVPR*, 2019. 2

[11] Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Hane, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3D shape representation. In *Proc. ICCV*, 2021. 2

[12] Eastman Kodak Company. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/. 5

[13] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3D shapes. *arXiv preprint arXiv:2009.09808*, 2020. 2, 7

[14] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. GRAM: Generative radiance manifolds for 3d-aware image generation. In *Proc. CVPR*, 2022. 2

[15] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 2

[16] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proc. CVPR*, 2017. 2

[17] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *Proc. ICLR*, 2020. 2, 3, 4

[18] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proc. CVPR*, 2019. 2

[19] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200fps. In *Proc. ICCV*, 2021. 2

[20] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *Proc. CVPR*, 2020. 2

[21] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proc. ICCV*, 2019. 2

[22] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proc. ICML*, 2020. 2

[23] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph. (SIGGRAPH Asia)*, 37(6), 2018. 2

[24] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021. 2, 6

[25] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. SAPE: Spatially-adaptive progressive encoding for neural optimization. In *Proc. NeurIPS*, 2021. 2

[26] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3D scenes. In *Proc. CVPR*, 2020. 2

[27] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *Proc. CVPR*, 2020. 2

[28] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 2, 8

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, 2019. 8

[30] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *CVPR*, 2021. 2

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, 2014. 5, 7

[32] Amit Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. *Proc. 3DV*, 2020. 2

[33] David B. Lindell, Julien N. P. Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *Proc. CVPR*, 2021. 2

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

[35] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. CVPR*, 2020. 2

[36] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph. (SIGGRAPH)*, 38(4), 2019. 2

[37] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. ACORN: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph. (SIGGRAPH)*, 40(4), 2021. 2

[38] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proc. CVPR*, 2021. 2

[39] Nelson Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph*, 1(2):99–108, 1995. 6

[40] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proc. ICCV*, 2021. 2

[41] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. CVPR*, 2019. 1, 2, 8

[42] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proc. ICCV*, 2019. 2

[43] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph. (SIGGRAPH)*, 38(4), 2019. 2, 6

[44] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, 2020. 1, 2, 5, 6

[45] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum*, 40(4), 2021. 2

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

[47] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4D reconstruction by learning particle dynamics. In *Proc. ICCV*, 2019. 2

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

[49] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proc. ICCV*, 2019. 2

[50] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proc. ICCV*, 2021. 2

[51] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-resolution 3D-consistent image and geometry generation. In *Proc. CVPR*, 2022. 2

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

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proc. NeurIPS*, 2019. 5

[54] Songyou Peng, Chiyu Max Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable Poisson solver. In *Proc. NeurIPS*, 2021. 2

[55] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. ECCV*, 2020. 2

[56] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proc. CVPR*, 2021. 2

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

[58] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny mlps. In *Proc. ICCV*, 2021. 2

[59] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Proc. ECCV*, 2020. 2

[60] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. ICCV*, 2019. 1, 2

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

[62] Eero P. Simoncelli and William T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proc. ICIP*, 1995. 2

[63] Vincent Sitzmann, Eric R. Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. MetaSDF: Meta-learning signed distance functions. In *Proc. NeurIPS*, 2020. 2

[64] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 1, 2, 4, 5, 7

[65] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. DeepVoxels: Learning persistent 3D feature embeddings. In *Proc. CVPR*, 2019. 2

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

[67] 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 *CVPR*, 2021. 2

[68] 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 *Proc. CVPR*, 2021. 2, 7

[69] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proc. NeurIPS*, 2020. 2, 5, 7

[70] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. *Computer Graphics Forum*, 39(2):701–727, 2020. 1, 2, 8

[71] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. *arXiv preprint arXiv:2111.05849*, 2021. 2

[72] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph. (SIGGRAPH)*, 38(4):1–12, 2019. 2

[73] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in neural systems*, 14(3):391, 2003. 4

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

[75] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):1–12, 2019. 2

[76] Lance Williams. Pyramidal parametrics. *Computer Graphics (Proc. SIGGRAPH)*, 17(3):1–11, 1983. 5

[77] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Proc. NeurIPS*, 2021. 2

[78] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Proc. NeurIPS*, 2020. 2

[79] Wang Yifan, Lukas Rahmann, and Olga Sorkine-Hornung. Geometry-consistent neural shape representation with implicit displacement fields. In *Proc. ICLR*, 2022. 2

[80] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Trans. Graph.*, 38(6):1–14, 2019. 2

[81] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-Points: Optimizing neural implicit surfaces with hybrid representations. In *Proc. CVPR*, 2021. 2

[82] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2

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

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

[85] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, Jonathan T. Barron, Ravi Ramamoorthi, and William T. Freeman. Neural light transport for relighting and view synthesis. *ACM Trans. Graph.*, 40(1), 2021. 2

[86] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (SIGGRAPH)*, 37(4), 2018. 2