Batch Normalization Alleviates the Spectral Bias in Coordinate Networks

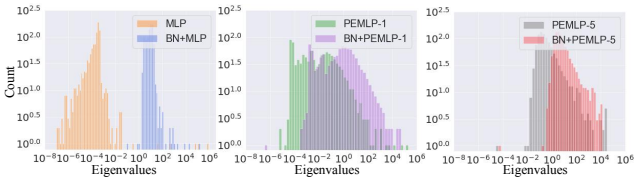
Zhicheng Cai[1], Hao Zhu[1]*, Qiu Shen[1], Xinran Wang[1,2], Xun Cao[1]
[1]School of Electronic Science and Engineering, Nanjing University, Nanjing, China
[2]Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University, Suzhou, China
Corresponding author: zhuhao_photo@nju.edu.cn

Abstract

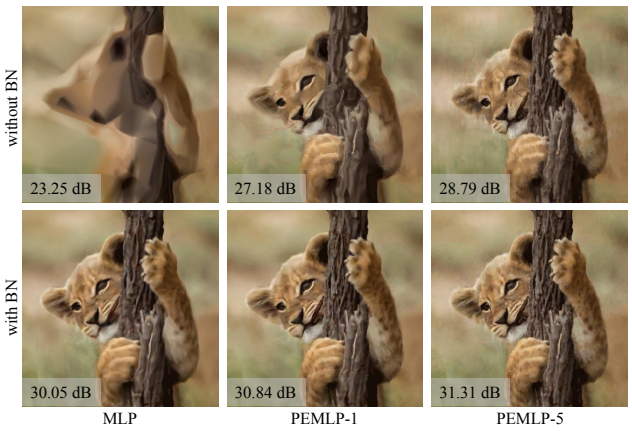
Representing signals using coordinate networks dominates the area of inverse problems recently, and is widely applied in various scientific computing tasks. Still, there exists an issue of spectral bias in coordinate networks, limiting the capacity to learn high-frequency components. This problem is caused by the pathological distribution of the neural tangent kernel's (NTK's) eigenvalues of coordinate networks. We find that, this pathological distribution could be improved using the classical batch normalization (BN), which is a common deep learning technique but rarely used in coordinate networks. BN greatly reduces the maximum and variance of NTK's eigenvalues while slightly modifies the mean value, considering the max eigenvalue is much larger than the most, this variance change results in a shift of eigenvalues' distribution from a lower one to a higher one, therefore the spectral bias could be alleviated (see Fig. 1). This observation is substantiated by the significant improvements of applying BN-based coordinate networks to various tasks, including the image compression, computed tomography reconstruction, shape representation, magnetic resonance imaging and novel view synthesis.

1. Introduction

Coordinate networks, which take the coordinates as inputs and output the signal attributes using multi-layer perceptron (MLP) models, have become a promising framework for solving various inverse problems. Different from the classical convolution-based networks which could only support up to 3D patterns as input [20, 57], the input coordinates are organized as 1D vectors in coordinate networks, enabling the advantage of a general framework for solving inverse problems with any dimensions. Therefore, coordinate networks have been widely applied in different areas of scientific computing [30], such as the hologram/tomography imaging in microscopy [39, 83], 3D reconstruction and free-viewpoint roaming in computer vision/graphics [37, 48], physical simulation in material design and hydrodynamics [8, 59] and medical imaging [67, 68].



(a) Histograms of NTK Eigenvalues



(b) Reconstructed images

Figure 1. Batch normalization significantly alleviates the spectral bias of coordinate networks. (a) Batch normalization shifts the NTK's eigenvalues distribution from a lower one to a higher one, thus (b) the spectral bias is alleviated and better performance is achieved compared with the one without batch normalization (*e.g.*, the texture on the lion's left paw). From left to right, each column refers to the coordinate networks with ReLU activations, and positional encoding [78] with 1 and 5 Fourier bases, respectively.

Yet, due to the spectral bias [58] of ReLU-based MLP, coordinate networks prefer to learn the low-frequency components of the signal, while the high-frequency components are learned at an extremely slow convergence. Several works have been proposed to alleviate the spectral bias, such as the positional encoding [78] or frequency-related activation functions [60, 72]. However, these explorations introduce the 'frequency-specified spectral bias' [81], *i.e.*,

*This work was supported by the National Natural Science Foundation of China under Grants T2221003, 62071216 and the Leading Technology of Jiangsu Basic Research Plan (BK20192003).

only the spectrum components matching the pre-encoded frequencies could be well learned. Therefore, it is essential to encode different frequency bases as many as possible, which not only increases the complexity but also incurs the issue of suppressing lower frequency components [61].

Such a spectral bias problem is related to the training dynamics of MLP. According to recent literature, the training of MLP could be viewed as kernel regression, specifically, the neural tangent kernel (NTK) [25, 78]. Following this perspective, the spectral bias is caused by the pathological distribution of NTK's eigenvalues, that most of the eigenvalues are very small, limiting the convergence speed on high-frequency components. In this paper, we find that this spectral bias could be well alleviated using a classical technique - batch normalization [24], which has been widely applied in convolutional neural networks but is rarely used in the community of coordinate networks.

We theoretically prove that batch normalization technique could significantly reduce the maximum and variance of NTK's eigenvalues while applies almost no changes to the mean value. Considering the fact that the largest eigenvalue is often much larger than most ones in NTK, this variance change could increase most of the eigenvalues. This improvement is not limited to the standard ReLU-based MLP, actually, it also works for positional encoding-based MLPs where the 'frequency-specified spectral bias' could also be alleviated (as shown in Fig. 1).

In summary, we make the following contributions,

1. We leverage mean field theory and conduct simple experiments to theoretically and empirically show that the pathological distribution of NTK's eigenvalues could be significantly improved by using batch normalization, enabling the coordinate networks to learn high-frequency components effectively .
2. We substantiate the significant improvements of BN-based MLP over existing coordinate networks on two representation tasks and three inverse problems, *i.e.*, image compression, shape representation, computed tomography reconstruction, magnetic resonance imaging, and novel view synthesis.

2. Related Work

2.1. Coordinate Networks

Coordinate networks [78] (also termed as *implicit neural representation* or *neural fields*) are gradually replacing traditional discrete representations in computer vision and graphics. Different from classical matrix-based discrete representation, coordinate networks focus on learning a neural mapping function with low-dimensional coordinates inputs and the corresponding signal values outputs, and have demonstrated the advantages of continuous querying and memory-efficient footprint in various signal representation tasks, such as images [13, 14, 73], scenes [26, 38, 46, 71] and videos [6, 82]. Additionally, coordinate networks could be seamlessly combined with different differentiable physical processes, opening a new way for solving various inverse problems, especially the domain-specific tasks where large-scale labelled datasets are unavailable, such as the shape representation [5, 9, 18, 47, 55], computed tomography reconstruction [19, 50, 63, 66, 69] and inverse rendering for novel view synthesis [48, 54, 56, 62, 86].

2.2. Overcoming the Spectral Bias

MLPs with conventional activations such as ReLU encounter the issue of *spectral bias* [2, 58, 78], which restricts their ability to capture the high-frequency components present in visual signals. To overcome this limitation, two types of methods are raised in literature, namely, the function expansion and the signal re-organization. The first type of methods treats the coordinate networks as a function expansion process, *e.g.*, the Fourier, Taylor, Wavelet and Gaussian expansions, as a result, more pre-encoded bases could improve the expressive power. Following this idea, various bases have been introduced to encode the input coordinates or activate the hidden neurons, such as the Fourier/polynomial encodings [34, 70, 78], periodic/non-periodic/wavelet activation functions [42, 60, 66, 72], and compositing multiple filters [15, 38]. However, these methods are sensitive to the training configurations.

Another type of methods focuses on mapping the input complex signal to another one which is composed of low-frequency components [5, 26, 27, 52, 75, 85], thus the original signal could be well learned. This mapping function is often implemented by introducing learnable hash tables between the input coordinate and the subsequent neural network, such as the single scale full-resolution hash table used in DINER [85], multi-scale pyramid hash tables in Instant-NGP [52] and multiple shifting hash tables in PIXEL [27]. These methods achieve high performance for representing complex signals at the cost of losing ability for interpolation, often requiring additional regularizations [84].

Different from previous methods, this paper raises a novel way to alleviate the spectral bias by introducing normalization methodology to the coordinate networks area.

2.3. Normalization for Deep Learning

Normalization [1, 24, 49, 64, 79, 80] has been an indispensable methodology for deep learning [4, 22, 40, 41]. As a milestone technique, batch normalization (BN) [24] is raised to solve the issue of *internal covariate shift*, which improves training stability, optimization efficiency and generalization ability [32, 51]. It has been a fundamental component for modern visual models and successfully applied to a wide range of computer vision tasks [7, 76, 77]. Furthermore, many variants of BN are proposed, such as Conditional BN [11] and Decorrelated BN [21]. In the realm of theoretical analysis, [23] highlights that BN tends to re-

duce the dependency on weight initialization in optimization trajectories. [12, 65] prove that BN improves optimization by mitigating the pathological curvature and smoothing the loss landscape. [3, 44] prove that BN permits using a larger learning rate according to the gradient dynamics and random matrix theory. This paper provides a theoretical analysis and empirical evidence demonstrating that normalization with full-batch dataset can effectively overcome the spectral issues. Furthermore, it firstly introduces the normalization methodology into coordinate networks, which is experimentally validated to enhance the model representational capacity more significantly.

3. Methodology

To lay the foundation for our theoretical analysis, we first give the background of coordinate networks including the formulation and the neural tangent kernel. Then, the eigenvalues of NTK matrices with and without batch normalization are analysed statistically by leveraging the mean field theory, revealing an alleviated spectral bias in coordinate networks after batch normalization.

3.1. Background

3.1.1 Coordinate Networks

Given a signal $\{\vec{x}_i, \vec{y}_i\}_{i=1}^T$, coordinate networks learn a function that maps input coordinates to the output corresponding signal values. It is often parameterized by a fully connected neural network consisting of one input layer with N^0 neurons and L hidden layers (including one output layer) with N^l neurons per hidden layer ($l = 1, 2, ..., L$). Without a loss of generality, we focus on a one-dimensional signal with one attribute (*i.e.*, $N^0 = N^L = 1$), additionally all other layers have same N neurons (*i.e.*, $N^1 = ... = N^{L-1} = N$). Thus we can describe the network $f(\mathbf{X}; \theta)$ as,

$$\begin{aligned} \mathbf{H}^0 &= \mathbf{X} = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_T] \\ \mathbf{H}^l &= \phi(\mathbf{W}^l \mathbf{H}^{l-1} + \text{Tile}(\vec{b}^{\,l})) \\ f(\mathbf{X}; \theta) &= \mathbf{H}^L = \mathbf{W}^L \mathbf{H}^{L-1} + \text{Tile}(\vec{b}^{\,L}), \end{aligned} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N^0 \times T}$ and T refer to the matrix of all coordinates and number of coordinates in the training dataset, respectively. $\theta = \{\mathbf{W}^l, \vec{b}^{\,l} \mid l = 1, ..., L\}$ is the network parameters in $f(\mathbf{X}; \theta)$. $\mathbf{W}^l \in \mathbb{R}^{N^l \times N^{l-1}}$ and $\vec{b}^{\,l} \in \mathbb{R}^{N^l}$ are the weight matrix and bias vector of the l-th layer, respectively, and are randomly initiated with Gaussian distribution. $\phi(\cdot)$ is the nonlinear activation function. Tile$(\cdot)$ operator repeats the input vector T times and forms a matrix with T columns.

3.1.2 Neural Tangent Kernel and Spectral Bias

Neural tangent kernel (NTK) [25, 78], which approximates the training of neural network as kernel regression, has become a popular lens for monitoring the dynamic behaviors and convergence of a neural network. Given a neural network $f(\mathbf{X}; \theta)$, its NTK is defined as,

$$\begin{aligned} \mathbf{K} &= (\nabla_\theta f(\mathbf{X}; \theta))^\top \nabla_\theta f(\mathbf{X}; \theta) \\ &= (\nabla_\theta \mathbf{H}^L)^\top \nabla_\theta \mathbf{H}^L. \end{aligned} \tag{2}$$

When the network $f(\mathbf{X}; \theta)$ is trained following an L_2 loss function, SGD optimizer and a learning rate η, the network output after n training iterations can be approximated by the NTK as [36, 78]:

$$\mathbf{Y}^{(n)} \approx \left(\mathbf{I} - e^{-\eta \mathbf{K} n}\right) \mathbf{Y}, \tag{3}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{Y}$ refers to signal values of all points in training dataset. Since the NTK matrix $\mathbf{K}$ is a positive semi-definite matrix, it could be decomposed using SVD as $\mathbf{K} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q}$ and $\mathbf{\Lambda}$ are the eigenvector matrix and diagonal matrix full of eigenvalues, respectively. Thus, the training error could be modelled as,

$$\begin{aligned} |\mathbf{Y}^{(n)} - \mathbf{Y}| &\approx e^{\eta \mathbf{K} t} = \mathbf{Q} e^{-\eta \mathbf{\Lambda} t} \mathbf{Q}^\top \\ \Rightarrow \mathbf{Q}^\top |\mathbf{Y}^{(n)} - \mathbf{Y}| &\approx e^{-\eta \mathbf{\Lambda} t} \mathbf{Q}^\top. \end{aligned} \tag{4}$$

It could be noticed that the training error is determined by the eigenvalues in $\mathbf{\Lambda}$. The network $f(\mathbf{X}; \theta)$ could learn the components with a large eigenvalue rapidly, and has a slow convergence to small eigenvalues which often refer to the high frequency components of the signal to be learned [78]. This phenomenon is termed as 'spectral bias'.

3.2. Batch Normalization Alleviates Spectral Bias

3.2.1 Batch normalization

As the name says, batch normalization aims at normalizing the network output at different layers along the batch dimension. Without a loss of generality, we starts from the simplest case, *i.e.*, applying BN to the last layer of $f(\mathbf{X}; \theta)$, *i.e.*,

$$\begin{aligned} f^{BN}(\mathbf{X}; \theta) &= \frac{\mathbf{H}^L - \mu}{\sigma} \gamma + \beta \\ \mu = \mathbb{E}[\mathbf{H}^L], &\quad \sigma = \sqrt{\mathbb{E}[(\mathbf{H}^L)^2] - (\mathbb{E}[\mathbf{H}^L])^2}, \end{aligned} \tag{5}$$

where the scale and shift parameters γ and β are initialized as 1 and 0, respectively.

In this case, the NTK changes and could be derived by substituting the Eqn. 5 into Eqn. 2. The modified NTK $\mathbf{K}^{BN}$ could be written as

$$\begin{aligned} \mathbf{K}^{BN} &= (\nabla_\theta f^{BN}(\mathbf{X}; \theta))^\top \nabla_\theta f^{BN}(\mathbf{X}; \theta) \\ &= \frac{(\nabla_\theta \mathbf{H})^\top \nabla_\theta \mathbf{H}}{\sigma^2} - \frac{\mathbf{H}^\top \mathbf{H} (\nabla_\theta \mathbf{H})^\top \nabla_\theta \mathbf{H}}{T \sigma^4}, \end{aligned} \tag{6}$$

where $\mathbf{H} = \mathbf{H}^L - \mu$ is introduced for the sake of derivation. Please refer the supplemental material for details of the derivation.

3.2.2 Decomposition of NTK Matrices

According to the mean field theory [28, 29], the $T \times T$ NTK matrix of the coordinate network $\mathbf{K} = (\nabla_\theta \mathbf{H}^L)^\top \nabla_\theta \mathbf{H}^L$ can be calculated and re-written in the explicit numerical form as:

$$\mathbf{K} = (L-1)N\mathbf{M} + \mathcal{O}(\sqrt{N}) \tag{7}$$

where $\mathbf{M}$ is a $T \times T$ matrix:

$$\mathbf{M} = \begin{bmatrix} \kappa_1 & \kappa_2 & \cdots & \kappa_2 \\ \kappa_2 & \kappa_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \kappa_2 \\ \kappa_2 & \cdots & \kappa_2 & \kappa_1 \end{bmatrix} \tag{8}$$
$$\kappa_1 = \frac{1}{L-1}\sum_{l=1}^{L} p_t^{l-1} q_t^l, \quad \kappa_2 = \frac{1}{L-1}\sum_{l=1}^{L} p_{st}^{l-1} q_{st}^l$$

κ_1 and κ_2 are two constant values corresponding to the order parameters of the mean field theory $p_t^l, q_t^l, p_{st}^l, q_{st}^l$ [29].

Furthermore, the NTK matrix with batch normalization (*i.e.*, Eqn. 6) could be rewritten as,

$$\begin{aligned} &\mathbf{K}^{BN} = \mathbf{A}(\nabla_\theta \mathbf{H}^L - \nabla_\theta \mu)^\top (\nabla_\theta \mathbf{H}^L - \nabla_\theta \mu), \\ &\text{where } \mathbf{A} = \frac{1}{\sigma^2}\left(\mathbf{I} - \frac{1}{T}\frac{\mathbf{H}^\top \mathbf{H}}{\sigma^2}\right). \end{aligned} \tag{9}$$

$\mathbf{A}$ is a $T \times T$ matrix corresponding to the variance division. Here we define the $T \times T$ projector matrix $\mathbf{G}$ corresponding to the mean subtraction:

$$\mathbf{G} = \mathbf{I} - \frac{1}{T} = \begin{bmatrix} \frac{T-1}{T} & -\frac{1}{T} & \cdots & -\frac{1}{T} \\ -\frac{1}{T} & \frac{T-1}{T} & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{1}{T} \\ -\frac{1}{T} & \cdots & -\frac{1}{T} & \frac{T-1}{T} \end{bmatrix}. \tag{10}$$

$\mathbf{G}$ satisfies $\mathbf{G}^\top \mathbf{G} = \mathbf{G}$ and $\nabla_\theta \mathbf{H}^L \mathbf{G} = (\nabla_\theta \mathbf{H}^L - \nabla_\theta \mu)$. Thus Eqn. 9 becomes:

$$\begin{aligned} \mathbf{K}^{BN} &= \mathbf{A}\mathbf{G}^\top (\nabla_\theta \mathbf{H}^L)^\top \nabla_\theta \mathbf{H}^L \mathbf{G} \\ &= \mathbf{A}\mathbf{G}^\top \mathbf{K}\mathbf{G} \\ &= (L-1)N\mathbf{A}\mathbf{M}^{BN} + \mathcal{O}(\sqrt{N}) \\ &\text{where } \mathbf{M}^{BN} = \mathbf{G}^\top \mathbf{M}\mathbf{G} \\ &= \begin{bmatrix} \frac{(\kappa_1-\kappa_2)(T-1)}{T} & \frac{\kappa_2-\kappa_1}{T} & \cdots & \frac{\kappa_2-\kappa_1}{T} \\ \frac{\kappa_2-\kappa_1}{T} & \frac{(\kappa_1-\kappa_2)(T-1)}{T} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{\kappa_2-\kappa_1}{T} \\ \frac{\kappa_2-\kappa_1}{T} & \cdots & \frac{\kappa_2-\kappa_1}{T} & \frac{(\kappa_1-\kappa_2)(T-1)}{T} \end{bmatrix}, \end{aligned} \tag{11}$$

thus we have[1]:

$$\begin{aligned} \mathbf{A}\mathbf{M}^{BN} &= \frac{\kappa_1-\kappa_2}{\sigma^2}\left(\mathbf{I} - \frac{1}{T\sigma^2}(1-\frac{1}{T})\mathbf{H}^\top \mathbf{H}\right) \\ &\approx \frac{\kappa_1-\kappa_2}{\sigma^2}\mathbf{I} \\ \mathbf{K}^{BN} &\approx (L-1)N\frac{\kappa_1-\kappa_2}{\sigma^2}\mathbf{I} + \mathcal{O}(\sqrt{N}). \end{aligned} \tag{12}$$

3.2.3 Statistical Characteristics of NTK's Eigenvalues

Since it is difficult to obtain all eigenvalues $\{\lambda_i\}_{i=1}^T$ of the NTK, we focus on analysing its statistic characteristics, *i.e.*, mean value m_λ, variance v_λ and the maximum value λ_{max}. According to the matrix theory [16], these values could be calculated or estimated by,

$$\begin{aligned} &m_\lambda = \frac{1}{T}Trace(\mathbf{K}), \quad v_\lambda = s_\lambda - m_\lambda^2 \\ &\frac{s_\lambda}{m_\lambda} = \frac{\sum_{i=1}^T \lambda_i^2}{\sum_{i=1}^T \lambda_i} \le \lambda_{max} \le \sqrt{\sum_{i=1}^T \lambda_i^2} = \sqrt{T s_\lambda}, \end{aligned} \tag{13}$$

where $s_\lambda = \frac{1}{T}Trace(\mathbf{K}\mathbf{K}^\top)$ is the second momentum. According to the decomposition of the NTK matrices in the above section, these values could be obtained directly,

Standard MLP:

$$\begin{aligned} &m_\lambda = \frac{(L-1)NT\kappa_1 + \mathcal{O}(\sqrt{N})T}{T} \sim \mathcal{O}(N) \\ &v_\lambda \sim \mathcal{O}(N^3) - \mathcal{O}(N^2) \sim \mathcal{O}(N^3) \\ &\mathcal{O}(N^2) \le \lambda_{max} \le \mathcal{O}(N^2), \ \ \lambda_{max} \sim \mathcal{O}(N^2). \end{aligned}$$

BN-based MLP:

$$\begin{aligned} &m_\lambda = \frac{(L-1)}{\sigma^2}\frac{NT(\kappa_1-\kappa_2)}{T} + \frac{\mathcal{O}(\sqrt{N})T}{T} \sim \mathcal{O}(N) \\ &v_\lambda \sim \mathcal{O}(N^2) + \mathcal{O}(N^{\frac{3}{2}}) - \mathcal{O}(N^2) \sim \mathcal{O}(N^{\frac{3}{2}}) \\ &\mathcal{O}(N) \le \lambda_{max} \le \mathcal{O}(N^{\frac{3}{2}}). \end{aligned} \tag{14}$$

Comparing these statistical values without and with batch normalization, it is observed that the mean values are slighted changed and have the same scale of $\mathcal{O}(N)$. However, the variances are significantly reduced ($\mathcal{O}(N^3)$ vs $\mathcal{O}(N^{\frac{3}{2}})$), considering the largest eigenvalue is also largely reduced ($\mathcal{O}(N^2)$ vs $\mathcal{O}(N^{\frac{3}{2}})$), this variance change indicates a shift of eigenvalues' distribution from a lower one to a higher one (as shown in Fig. 1a). As a result, the spectral bias could be alleviated. Although the analysis mentioned above is built upon the assumption of applying BN to the last layer, it could be expanded to a general case by modeling deep BN-based coordinate networks as a stacking process layer by layer.

Additionally, BN also works for the successful positional encoding-based MLP (PEMLP) [78]. According to NTK

[1] Please refer the supplemental material for details of derivation.

2D Image

Model	SIREN	ReLU	BN	MFN	WIRE	PE	BN+PE
5×20	23.18	21.21	23.39	22.03	21.95	22.10	24.26
5×30	24.57	22.57	24.39	23.34	23.10	23.47	25.54
10×28	25.52	23.25	26.27	25.25	25.05	24.63	26.85
10×40	28.32	23.56	27.95	26.92	27.34	26.66	28.71
13×49	29.42	24.33	29.08	28.14	29.02	28.23	30.84

2D CT

Model	SIREN	ReLU	BN	MFN	WIRE	PE	BN+PE
2×128	28.30	26.31	28.50	24.82	28.30	28.16	30.14
2×256	16.16	26.78	28.41	27.97	28.26	28.11	31.60
4×128	18.31	28.21	30.42	29.15	28.31	30.42	32.45
4×256	18.32	28.30	30.48	30.92	28.70	30.00	32.64
8×256	18.37	29.93	31.24	31.18	18.61	30.79	33.64

3D Shape

Model	SIREN	ReLU	BN	MFN	WIRE	PE	BN+PE
2×128	0.954	0.905	0.952	0.886	0.956	0.957	0.983
2×256	0.964	0.904	0.958	0.972	0.964	0.968	0.992

3D MRI

Model	SIREN	ReLU	BN	MFN	WIRE	PE	BN+PE
2×256	26.04	24.52	28.97	27.24	25.31	30.17	32.42
4×256	27.89	26.11	30.62	29.74	30.24	32.12	34.56

5D NeRF

Metrics	Plenoxels	NeRF	BN+NeRF	DVGO	DINER	BN+DINER
PSNR↑	30.99	31.04	31.37	33.23	33.11	33.31
SSIM↑	0.956	0.953	0.956	0.966	0.967	0.968
LPIPS↓	0.050	0.054	0.050	0.034	0.034	0.033

Table 1. Results of different coordinate networks on different tasks. The values in the 2D Image, 2D CT, and 3D MRI tasks refer to the PSNR values, while the values in the 3D Shape task refer to the IoU values. We color code each cell as best, second best, and third best.

analysis in [78], the role of Fourier-based positional encoding could be viewed as adding an additional term h_γ to the original NTK, *i.e.*, a composed NTK function $\mathbf{K} \circ h_\gamma$, where h_γ is a linear combination of the used Fourier bases and is independent of the subsequent network. As a result, the mean and variance of the PEMLP's NTK will also change following the rule summarized in Eqn. 14, and the 'frequency-specified spectral bias' in PEMLP could also be alleviated by batch normalization.

Simple examples for fitting a 1D signal and a 2D image. Fig. 1a compares the NTK's eigenvalues without and with batch normalization for learning a 1D signal. It is observed that most of NTK's eigenvalues are increased when batch normalization is added, *e.g.*, most of eigenvalues are increased from $\sim 10^{-4}$ to $\sim 10^{1}$, from $\sim 10^{-3}$ to $\sim 10^{1}$, from $\sim 10^{-1}$ to $\sim 10^{1}$ in standard MLP, PEMLP with 1 and 5 Fourier bases, respectively. Additionally, the largest eigenvalues are all reduced when BN is applied, verifying the analysis in Eqn. 14. As a result, more high-frequency details are provided when applying the BN for fitting a 2D image compared with the one without batch normalization, as shown in Fig. 1b. In summary,

Proposition 1 *Batch normalization alleviates the spectral bias by making a shift of NTK's eigenvalues distribution from a lower one to a higher one.*

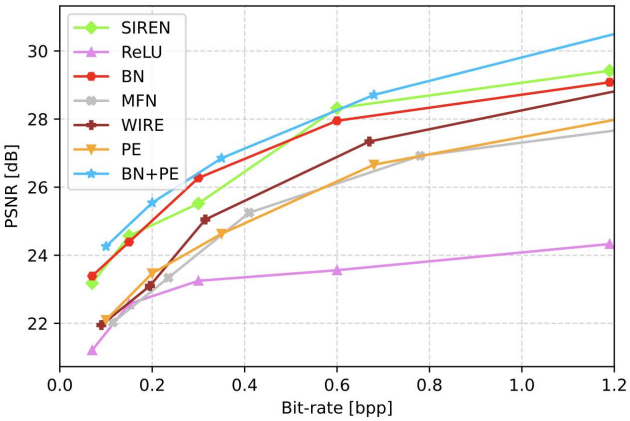


Figure 2. Rate-distortion curves of various coordinate networks under different bpps trained on the Kodak dataset.

4. Experiments

Tasks. We validate the effectiveness of applying batch normalization to coordinate networks on five separate tasks, *i.e.*, 2D image representation and compression, 2D computed tomography (CT) reconstruction, 3D shape representation, 3D magnetic resonance imaging (MRI) reconstruction, and 5D novel view synthesis.

Compared methods. For the first four tasks, a total of seven methods are compared, including the network with traditional activation ReLU (ReLU) [53], ReLU based network with Fourier features positional encoding (PE) [78], network with sinusoidal nonlinearity (SIREN) [72], multiplicative filter network (MFN) [15] and complex Gabor wavelet nonlinearity (WIRE) [66]. Because the BN is a universal tool, the results of batch normalization on the ReLU (BN) and positional encoding ReLU (BN+PE) are both compared. For BN and BN+PE, we add one batch normalization layer immediately after the ReLU activation of each fully-connected layer. As a common practice, the encoding scale of PE is set as 10 [48], the frequency parameter ω_0 of SIREN is set as 30 [72], the frequency parameter ω and the spread parameter s of WIRE are respectively set as 20 and 10 [66]. The weights of all the networks are randomly initialized. For SIREN and MFN, we utilize the specific weight initialization schemes as raised in [72] and [15], respectively. For the left five methods, we utilize the default LeCun random initialization [35].

4.1. 2D Image Representation and Compression

Configurations. We first use an image representation task to evaluate the performance of applying batch normalization to the coordinate networks. We perform experiments on the Kodak image dataset [13] consisting of 24 RGB images with a high resolution of 768×512. To utterly explore the representational capacity of various methods, we

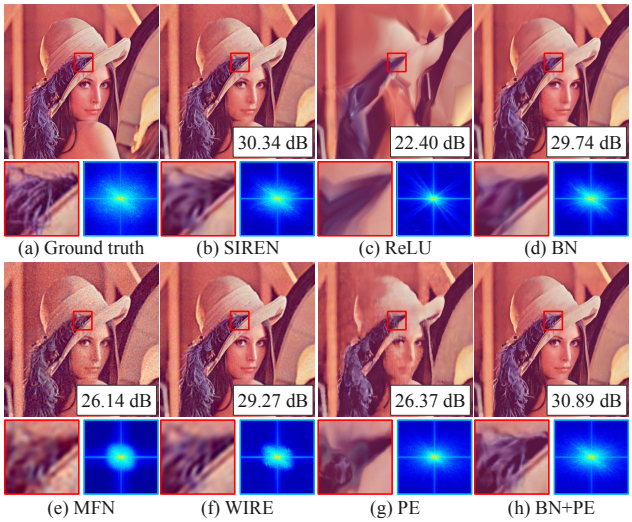


Figure 3. Comparisons of different methods for representing the Lena. The corresponding Fourier spectra are also visualized.

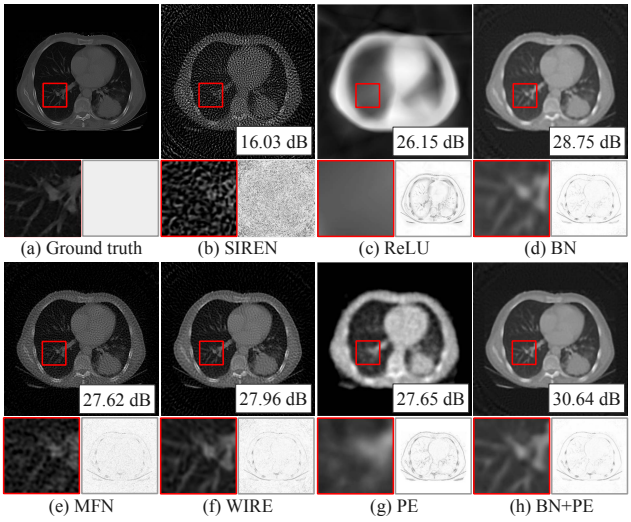


Figure 4. Comparisons of different coordinate networks for CT reconstruction. The corresponding error maps are also visualized.

use networks with limited parameters as introduced in [13], namely, networks with the configurations of (in the format of [$hidden\ layers \times hidden\ features$]) [5×20], [5×30], [10×28], [10×40], and [13×49]. Thus it can be regarded as the image compression task using coordinate networks. We use the L_2 distance between the network output and the ground truth as the loss function. All the models are trained for 50,000 iterations using Adam optimizer [31]. The initial learning rate is set as $2e{-}4$, except for BN and BN+PE, which use a larger initial learning rate of $1e-2$ thanks to the insensitivity to the learning rate of batch normalization. The batch size is equal to the number of image pixels.

Results. Tab. 1 exhibits the average PSNR (Peak Signal-to-Noise Ratio) of these methods. Accordingly, we plot the rate-distortion curves of these methods under various bpps (bits-per-pixel = number-of-parameters×bits-per-parameter / number-of-pixels) as shown in Fig. 2. As can be observed, ReLU consistently presents the worst performance due to the spectral bias, while applying batch normalization can significantly enhance the performance of networks with ReLU. For example, BN improves the PSNR of ReLU by up to 4.39dB and 4.75dB with the network of [10×40], and [13×49], respectively. BN also exhibits competitive performance compared to other existing method for that it consistently obtains higher PSNR than PE (up to 1.64dB with the network of [10×28]), WIRE (up to 1.44dB with the network of [5×20]) and MFN (up to 1.36dB with the network of [5×20]), and BN also surpasses SIREN with the network of [5×20], and [10×28]. Moreover, BN can also significantly improve the performance of PE. It is worth emphasizing that BN+PE consistently achieves the highest PSNR among all the seven methods, which further validates the effectiveness of BN.

Visualization. We visualize the reconstructed images and corresponding Fourier frequency spectra of various methods in Fig. 3. The target visual signal is 2D image Lena with a resolution of 512 × 512, and the network is of [10×28]. As can be obviously observed in Fig. 3c, the reconstructed image of ReLU is over-smoothed and poor-quality, failing to present the details of the target image. For PE, the quality of the reconstructed image (Fig. 3g) is poor for that the overall image is blurry and the finer details are failed to be restored. Moreover, both low- and high-frequencies are learned limitedly. However, after applying batch normalization to the networks, the quality of the reconstructed images (Fig. 3d, h) is significantly improved, and the details are clearly represented. Furthermore, as clearly shown by the Fourier spectra in Fig. 3d and Fig. 3h, more low- and high-frequency components are learned effectively with the application of BN, illustrating the effectiveness of our method on alleviating the spectral bias.

For SIREN (Fig. 3b), MFN (Fig. 3e) and WIRE (Fig. 3f), the presence of artifacts is clearly observable, resulting from over-fitting to high-frequency components. This, in turn, leads to noticeable noise and compromises the quality of reconstruction. This issue is particularly severe for MFN and WIRE, as indicated by their corresponding Fourier spectra, both methods tend to emphasize high frequencies while neglecting low-frequency components. In contrast, the coordinate networks with BN produce clean background reconstructions with minimal artifacts and noise. This phenomenon further demonstrates the robustness of BN to noise and its superior capability in representing the target visual signal.

4.2. 2D Computed Tomography

Configurations. In the CT reconstruction task, we observe integral projections of a density field instead of direct supervisions. In our experiments, we train a network that takes

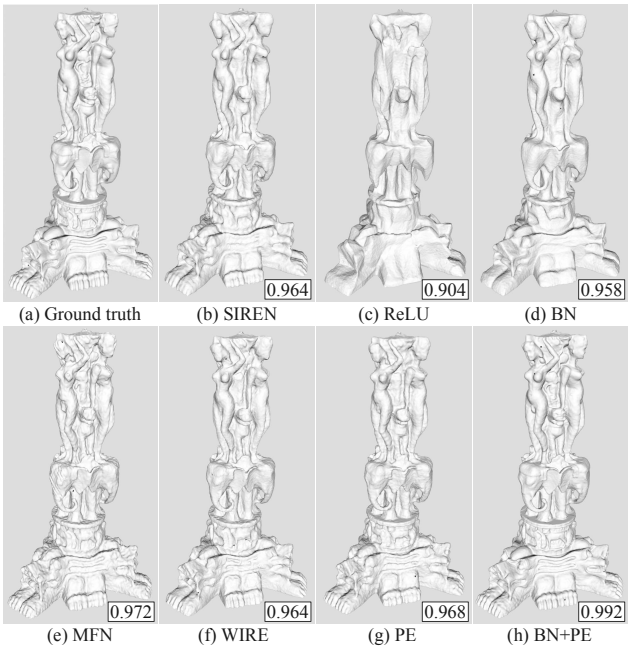


Figure 5. Meshes generated with occupancy volumes by various coordinate networks.

in 2D pixel coordinates and predicts the corresponding volume density at each location. We conduct the experiments on the x-ray colorectal dataset [10, 66], each image has a resolution of 512×512 and is emulated with 100 CT measurements. We use networks with five architecture configurations, namely, [2×128], [2×256], [4×128], [4×256], and [8×256]. To solve the inverse problem, the network is indirectly supervised by the MSE loss between a sparse set of ground-truth integral projections and integral projections computed from the network's output. All the models are trained for 20,000 iterations using Adam optimizer with a initial learning rate of $5e-3$ and a minimal learning rate of $5e-4$. The batch size is the total number of coordinates.

Results. Tab. 1 exhibits the results of various methods, measured in PSNR. As can be observed, BN+PE consistently achieves the highest PSNR and SSIM with all the network configurations. Furthermore, simple applying BN to the network surpasses all the other existing methods except for MFN with the network of [4×256], which has a slightly higher PSNR of 0.44dB. For WIRE, the performance degrades significantly when the network is deepened to 8 layers. While SIREN can only achieve moderate performance with the network of [2×128], for other network configurations, SIREN fails to reconstruct the target signal effectively. For other methods, the performances are enhanced when the network is deepened.

Visualization. Fig. 4 visualizes the CT images reconstructed by different methods with the network of [2×256], the corresponding error maps are also visualized in the bottom right corner of each method. As can be observed,

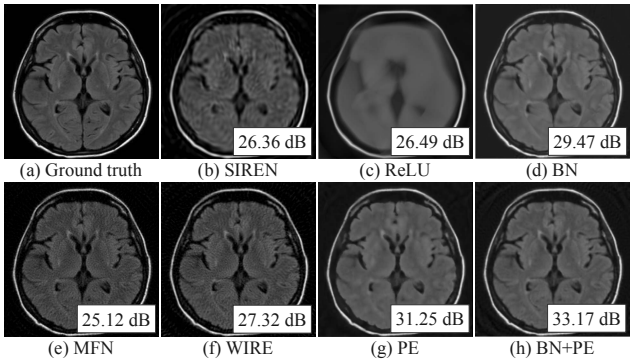


Figure 6. MRI reconstruction from various coordinate networks.

ReLU (Fig. 4c) leads to excessively smooth results, only displaying blurry patterns. The result of PE (Fig. 4g) is noisy, failing to exhibit the precise details. When applying batch normalization to ReLU and PE, the reconstruction quality (Fig. 4d and Fig. 4h) is significantly improved as the finer details are effectively represented. The other methods of novel nonlinear activations all results in obvious artifacts, especially for SIREN, which achieves a extremely low PSNR of only 16dB. This phenomenon may be caused by the over-fitting of these methods to the high frequencies, thus introducing tremendous noise.

4.3. 3D Shape Representation

Configurations. In this section, we demonstrate the representational capacity of batch normalization for representing 3D shapes as occupancy networks. To be specific, the input data is a mesh grid with 512^3 resolution, where the voxels inside the volume are assigned as 1, and the voxels outside the volume are assigned as 0. Then we use the occupancy network to implicitly represent a 3D shape as the "decision boundary" of coordinate networks, which is trained to output 0 for points outside the shape and 1 for points inside the shape. Test error is calculated using cross-entropy loss between the network output and the ground truth points. We conduct the experiments on the data from Stanford 3D Scanning Repository [33]. We use networks of two architecture configurations, namely, [2×128] and [2×256]. In the experiment, all the models are trained for 200 epochs using Adam optimizer with a initial learning rate of $5e-3$ and a minimum learning rate of $5e-4$. 200,000 points are randomly sampled in each iteration during the training process. The network outputs are extracted as a 512^3 grid using marching cubes [43] with a threshold of 0.5 for evaluation and visualization.

Results. Tab. 1 exhibits the experimental results evaluated by IoU (Intersection over Union). As can be observed, BN significantly enhances the performance of simple ReLU, up to 5.19% and 5.97% higher IoU with the network of [2×128] and [2×256], respectively. Moreover, BN+PE consistently achieves the best performance with every network configurations.

Visualization. Fig. 5 visualizes the grids of thai-statue scene represented by these methods with the network of [2×256]. As shown in Fig. 5, ReLU makes the surface over-smooth, while BN significantly improves such phenomenon allowing finer details to emerge. Besides, MFN introduces too many undesired textures and fluctuations on the surface, indicating over-fitting to high-frequencies and noise. Furthermore, BN+PE achieves the best representational result, without incurring artifacts or noise.

4.4. 3D Magnetic Resonance Imaging

Configurations. For the 3D MRI task, we observe measurements which are the Fourier transform coefficients of the atomic response to radio waves under a magnetic field. We train an MLP that takes in 3D voxel coordinates and predicts the corresponding intensity at each location with an indirect supervision. We conduct experiments on the ATLAS brain dataset [78], each sample has a volume resolution of 96^3. We use networks of two architecture configurations, that is, [2×256] and [4×256]. In the experiment, all the models are trained for 1,000 iterations using Adam optimizer with an initial learning rate $2e-3$.

Results. Tab. 1 exhibits the experimental results of various methods evaluated by PSNR. As can be observed, BN significantly enhances the performance of simple ReLU, up to 4.45dB and 4.51dB higher PSNR with the network of [2×256] and [4×256], respectively. BN also significantly surpasses the methods with frequency-related activation functions (*e.g.*, SIREN, MFN and WIRE). Moreover, BN+PE consistently achieves the highest PSNR among all the seven methods for both network configurations.

Visualization. Fig. 6 visualizes one MRI slice reconstructed by these methods with the network of [2×256]. For the methods with frequency-related activation functions, the reconstructed slices are full of unpredictable textures, indicating the over-fitting to the noise. While networks with BN have distinct details and clear background, indicating the superior representational capacity for the target signal and robustness to the noise.

4.5. 5D Novel View Synthesis

Configurations. In this section, we demonstrate the improvements of BN on novel view synthesis using the neural radiance fields (NeRF) [48]. NeRF models the 3D world as a 5D radiance fields using coordinate networks, where the input contains the 3D position and 2D viewing direction of a point and the output attributes include the RGB color and point density. Then the color of each pixel is calculated by querying the above attributes along the ray defined by the pixel position and camera's parameters and applying the volume rendering techniques [45]. Finally, the radiance field is optimized by supervising rendered color with the ground truth one. Once the radiance field is convergent, the image from any view could be synthesized by following the second step mentioned above.

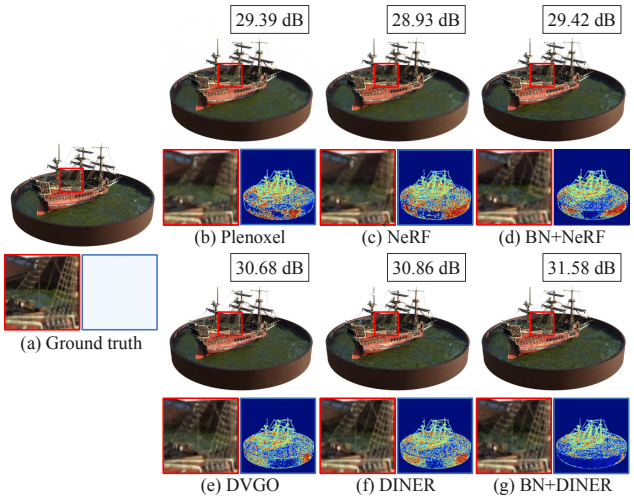


Figure 7. Neural radiance fields optimization with various methods. The corresponding error maps are also visualized.

To better verify the effectiveness, we apply BN to two popular NeRF baselines, *i.e.*, the original NeRF [48] which models the radiance field as a continuous function, and the DINER [85] which adopts a discrete form. We compare our results with four baselines, namely, NeRF, DINER, Plenoxels [17], and DVGO [74]. In the experiment, we follow all authors' default configurations.

Results and visualization. Tab. 1 lists the quantitative comparisons on the down-scaled Blender dataset [48] (resolution of 400×400). Compared with original results (NeRF and DINER), applying BN improves the PSNR up to 0.33dB and 0.20dB, respectively. In addition, BN-enhanced DINER achieves the best results among all the methods. Fig. 7 qualitatively compares the reconstructed details on the 'Ship' dataset. The corresponding error maps are also visualized. It is noticed that additional BN significantly reduce the errors (BN+NeRF vs NeRF and BN+DINER vs DINER), and more details are reconstructed such as the zoomed-in cable wind rope (BN+NeRF vs NeRF).

5. Conclusion

In this paper, we leverage the mean field theory to prove that batch normalization can optimize the NTK's eigenvalues distribution, enabling the coordinate networks overcome the spectral bias, namely, better suited for learning the high-frequency components. We also empirically show that adding batch normalization transits the NTK's eigenvalues distribution towards larger values for both standard MLP and position encoding-enhanced version. Extensive experiments substantiate that BN-based network significantly outperforms existing methods on various representation and inverse optimization tasks. In the future, we will explore the evolution of NTK's properties when other normalization techniques are applied to the coordinate networks.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[2] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International Conference on Machine Learning*, pages 685–694. PMLR, 2020. 2

[3] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018. 3

[4] Zhicheng Cai and Qiu Shen. Falconnet: Factorization for the light-weight convnets. *arXiv preprint arXiv:2306.06365*, 2023. 2

[5] 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. 2

[6] 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. 2

[7] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. *arXiv preprint arXiv:2303.03667*, 2023. 2

[8] Yuyao Chen, Lu Lu, George Em Karniadakis, and Luca Dal Negro. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express*, 28(8):11618–11633, 2020. 1

[9] 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. 2

[10] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26:1045–1057, 2013. 7

[11] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[12] Yonatan Dukler, Quanquan Gu, and Guido Montúfar. Optimization theory for relu neural networks trained with normalization layers. In *International conference on machine learning*, pages 2751–2760. PMLR, 2020. 3

[13] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021. 2, 5, 6

[14] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*, 2022. 2

[15] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2020. 2, 5

[16] Joel N Franklin. *Matrix theory*. Courier Corporation, 2012. 4

[17] 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. 8

[18] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019. 2

[19] Kunal Gupta, Brendan Colvert, and Francisco Contijoch. Neural computed tomography. *arXiv preprint arXiv:2201.06574*, 2022. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[21] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018. 2

[22] Lei Huang, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[23] Daniel Jiwoong Im, Michael Tao, and Kristin Branson. An empirical analysis of deep network loss surfaces. *arXiv*, 2016. 2

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 2

[25] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018. 2, 3

[26] 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. 2

[27] Namgyu Kang, Byeonghyeon Lee, Youngjoon Hong, Seok-Bae Yun, and Eunbyung Park. Pixel: Physics-informed cell representations for fast and accurate pde solvers. *arXiv preprint arXiv:2207.12800*, 2022. 2

[28] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. The normalization method for alleviating pathological sharpness in wide neural networks. *Advances in neural information processing systems*, 32, 2019. 4

[29] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1032–1041. PMLR, 2019. 4

[30] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. 1

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[32] Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Ming Zhou, Klaus Neymeyr, and Thomas Hofmann. Towards a theoretical understanding of batch normalization. *stat*, 1050: 27, 2018. 2

[33] Stanford Computer Graphics Laboratory. The stanford 3d scanning repository, 2014. http://graphics.stanford.edu/data/3Dscanrep/. 7

[34] Zoe Landgraf, Alexander Sorkine Hornung, and Ricardo Silveira Cabral. Pins: progressive implicit networks for multi-scale neural representations. *arXiv preprint arXiv:2202.04713*, 2022. 2

[35] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002. 5

[36] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019. 3

[37] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8456–8465, 2023. 1

[38] 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

[39] Renhao Liu, Yu Sun, Jiabei Zhu, Lei Tian, and Ulugbek S Kamilov. Recovery of continuous 3d refractive index maps from discrete intensity-only measurements using neural fields. *Nature Machine Intelligence*, 4(9):781–791, 2022. 1

[40] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. 2

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 2

[42] Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[43] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353, 1998. 7

[44] Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*, 2018. 3

[45] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 8

[46] 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. 2

[47] 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

[48] 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, 5, 8

[49] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2

[50] Amirali Molaei, Amirhossein Aminimehr, Armin Tavakoli, Amirhossein Kazerouni, Bobby Azad, Reza Azad, and Dorit Merhof. Implicit neural representation in medical imaging: A comparative survey. *arXiv preprint arXiv:2307.16142*, 2023. 2

[51] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018. 2

[52] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[53] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 5

[54] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 2

[55] 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. 2

[56] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[57] Adnan Qayyum, Inaam Ilahi, Fahad Shamshad, Farid Boussaid, Mohammed Bennamoun, and Junaid Qadir. Untrained neural network priors for inverse imaging problems: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6511–6536, 2023. 1

[58] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 1, 2

[59] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481): 1026–1030, 2020. 1

[60] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pages 142–158. Springer, 2022. 1, 2

[61] Sameera Ramasinghe, Lachlan E MacDonald, and Simon Lucey. On the frequency-bias of coordinate-mlps. *Advances in Neural Information Processing Systems*, 35:796–809, 2022. 2

[62] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14153–14161, 2021. 2

[63] Albert W Reed, Hyojin Kim, Rushil Anirudh, K Aditya Mohan, Kyle Champley, Jingu Kang, and Suren Jayasuriya. Dynamic ct reconstruction from limited views with implicit neural representations and parametric motion fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2258–2268, 2021. 2

[64] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016. 2

[65] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018. 3

[66] 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, 5, 7

[67] Chengkang Shen, Hao Zhu, You Zhou, Yu Liu, Lili Dong, Weipeng Zhao, David Brady, Xun Cao, Zhan Ma, and Yi Lin. Cardiacfield: Computational echocardiography for universal screening. *Research Square*, 2023. 1

[68] Chengkang Shen, Hao Zhu, You Zhou, Yu Liu, Lili Dong, Weipeng Zhao, David Brady, Xun Cao, Zhan Ma, and Yi Lin. Tracking anything in heart all at once. *arXiv preprint arXiv:2310.02792*, 2023. 1

[69] Liyue Shen, John Pauly, and Lei Xing. Nerp: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[70] Rajhans Singh, Ankita Shukla, and Pavan Turaga. Polynomial implicit neural representations for large diverse datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2051, 2023. 2

[71] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[72] 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, 5

[73] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022. 2

[74] 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. 8

[75] 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

[76] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[77] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 2

[78] 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, 3, 4, 5, 8

[79] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2

[80] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[81] 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. 1

[82] Yunfan Zhang, Ties van Rozendaal, Johann Brehmer, Markus Nagel, and Taco Cohen. Implicit neural video compression. *arXiv preprint arXiv:2112.11312*, 2021. 2

[83] Hao Zhu, Zhen Liu, You Zhou, Zhan Ma, and Xun Cao. DNF: Diffractive neural field for lensless microscopic imaging. *Optics Express*, 30(11):18168–18178, 2022. 1

[84] Hao Zhu, Fengyi Liu, Qi Zhang, Xun Cao, and Zhan Ma. Rhino: Regularizing the hash-based implicit neural representation. *arXiv preprint arXiv:2309.12642*, 2023. 2

[85] Hao Zhu, Shaowen Xie, Zhen Liu, Fengyi Liu, Qi Zhang, You Zhou, Yi Lin, Zhan Ma, and Xun Cao. Disorder-invariant implicit neural representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2024. 2, 8

[86] Junyu Zhu, Hao Zhu, Qi Zhang, Fang Zhu, Zhan Ma, and Xun Cao. Pyramid nerf: Frequency guided fast radiance field optimization. *International Journal of Computer Vision*, pages 1–16, 2023. 2