Split-Layer: Enhancing Implicit Neural Representation by Maximizing the Dimensionality of Feature Space

Zhicheng Cai[1], **Hao Zhu**[1,*], **Linsen Chen**[1], **Qiu Shen**[1,2], **Xun Cao**[1,2]

[1]School of Electronic Science and Engineering, Nanjing University
[2]Key Laboratory of Optoelectronic Devices and Systems with Extreme Performances of MOE, Nanjing University

Abstract

Implicit neural representation (INR) models signals as continuous functions using neural networks, offering efficient and differentiable optimization for inverse problems across diverse disciplines. However, the representational capacity of INR—defined by the range of functions the neural network can characterize—is inherently limited by the low-dimensional feature space in conventional multilayer perceptron (MLP) architectures. While widening the MLP can linearly increase feature space dimensionality, it also leads to a quadratic growth in computational and memory costs. To address this limitation, we propose the split-layer, a novel reformulation of MLP construction. The split-layer divides each layer into multiple parallel branches and integrates their outputs via Hadamard product, effectively constructing a high-degree polynomial space. This approach significantly enhances INR's representational capacity by expanding the feature space dimensionality without incurring prohibitive computational overhead. Extensive experiments demonstrate that the split-layer substantially improves INR performance, surpassing existing methods across multiple tasks, including 2D image fitting, 2D CT reconstruction, 3D shape representation, and 5D novel view synthesis.

Code — https://github.com/Aiolus-X/Split-Layer

Introduction

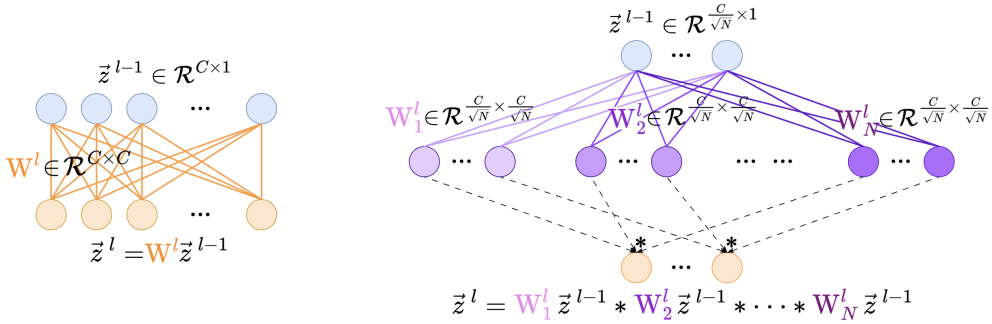
Implicit Neural Representation (INR) (Sitzmann et al. 2020; Liu et al. 2024) is reshaping the foundation of signal processing, garnering significant attention across various fields. Unlike traditional explicit discrete representations, INR leverages neural networks to implicitly model the continuous functional relationship between signal coordinates and their attributes, overcoming the limitations of traditional one in terms of storage efficiency and differentiability (Martel et al. 2021). By integrating seamlessly with physical laws, INR provides a new paradigm for solving inverse problems in a self-supervised manner, thereby reducing the reliance on large-scale paired datasets (Deng et al. 2009) in deep learning and mitigating the challenges of local minima (Bubeck et al. 2015) in classical optimization problems. These unique advantages have positioned INR as a powerful tool with vast application potential, spanning domains such as computational mathematics (Karniadakis et al. 2021; Raissi, Yazdani, and Karniadakis 2020), microscopic imaging (Zhu et al. 2022; Zhou et al. 2023; Zhang et al. 2025; Zhou et al. 2024), medical imaging (Sun et al. 2024; Shen et al. 2025, 2024), and consumer-grade graphics rendering (Mildenhall et al. 2021; Zhu et al. 2023). As a result, INR not only offers new opportunities for research and development but also presents novel challenges that warrant further exploration.

However, the quality of inverse reconstruction is fundamentally determined by the representational capacity of INRs, for which the widely used multilayer perceptron (MLP) network suffers the well-known spectral-bias (Rahaman et al. 2019). To enhance the capacity of INR, two categories of approaches are proposed. The first is mapping the low-dimensional input coordinates to high-dimensional manifolds using embeddings (Tancik et al. 2020; Chen et al. 2023) or hash tables (Müller et al. 2022; Zhu et al. 2024b). Another is activating the neurons with specialized nonlinearity instead of the traditional ReLU (Nair and Hinton 2010), such as periodic/variable-periodic (Sitzmann et al. 2020; Liu et al. 2024) and wavelet (Saragadam et al. 2023) activation functions. However, these approaches primarily *make certain features more easier to be learned by introducing a learning bias* (Yüce et al. 2022) (e.g., Fourier bias and wavelet bias), rather than *expanding the range of features that can be learned*.

Essentially, the representational capacity of INRs is defined by the diversity of features that they can represent, namely, the *feature space* (Scholkopf et al. 1999; Kansizoglou, Bampis, and Gasteratos 2021; Zohdinasab et al. 2023; Ma et al. 2024). Specifically, for a fully connected layer with C neurons, it spans an Euclidean space with C dimensions as illustrated in Sec. . Scaling up the layer width is a common and effective strategy to enhance representational capacity (Hornik, Stinchcombe, and White 1989; Kaplan et al. 2020), as it actually enlarges the feature space's dimensionality, which is proportional to the layer width. However, linearly increasing the dimensionality results in a quadratic growth of parameters due to the formulation of MLPs, which incurs heavy computational budgets.

Such a problem is closely related to the improper fully

*Corresponding author: zhuhao_photo@nju.edu.cn



(a) Fully Connected Layer (b) Split Layer with N splits

Figure 1: The diagram of split-layer. Solid lines represent learnable weights, and dashed lines represent the Hadamard product.

connected mechanism designed in MLP, where the dimensionality of the feature space has a linear correction with number of neurons. By re-organizing the connection mechanism as a "split"-style (see Fig. 1), the dimensionality of features space could be significantly increased (Fig. 2). Thus, we propose the split-layer, where the original fully connected layer is split into N parts, each with an individual weight matrix $\mathbf{W}^n \in \mathbb{R}^{\frac{C}{\sqrt{N}}\times\frac{C}{\sqrt{N}}}$, and then *integrates* the outputs of these parts though the Hadamard product, thus formulating high-degree polynomials. As theoretically demonstrated in Sec., this modification can effectively expand the dimensionality of the feature space from C to an extremely large dimensionality of $\binom{\frac{C}{\sqrt{N}}+N-1}{N}$ with no additional parameters or computational costs (Fig.2 visualizes the feature space for a simple case), significantly enhancing the INR representational capacity (improving the performance up to 45% in image fitting task).

Specifically, we make the following contributions,

1. We propose the *split-layer*, which is a universal tool for enhancing the representational capacity of INR with different backbones.
2. We theoretically demonstrate the improvements of split-layer on extending the feature space's dimensionality and give the law for deriving the optimal split.
3. We substantiate the significant improvements of split-layer for different INR backbones (MLP, SIREN, Gauss, PEMLP, WIRE, and FINER) and various tasks (image representation, computed tomography reconstruction, shape representation, and novel view synthesis.).

Related Work

Implicit Neural Representations

Implicit Neural Representations (INRs), also referred to as coordinate networks or neural fields (Tancik et al. 2020; Sitzmann et al. 2020; Cai et al. 2024a), are parameterized by multi-layer perceptron (MLP) models to learn mappings from input coordinates to signal attributes, and are gradually emerging as the dominant paradigm for multi-modal signal representation and inverse problem optimization.

Due to the powerful approximation capabilities of neural networks, INRs offer advantages in compact modeling and efficient storage, enabling their successful applications in various signal representation and compression tasks, such as 1D audio (Gao et al. 2022), 2D images (Strümpler et al. 2022), 3D videos/structures/spectrum (Shi et al. 2025; Lindell et al. 2022; Shi et al. 2024), 4D light fields (Sitzmann et al. 2021), and 5D radiance fields (Mildenhall et al. 2021). In addition, due to the continuity and differentiability of neural networks, INRs can seamlessly integrate various physical processes to solve partial differential equations, providing an promising paradigm for addressing a wide range of inverse problems, especially with limited measurements, including inverse rendering for novel view synthesis (Mildenhall et al. 2021; Zhu et al. 2023), phase recovery in lensless imaging (Zhu et al. 2022), 3D static and 4D dynamic heart reconstruction from 2D ultrasound measurements (Shen et al. 2025, 2024) and so on (Park et al. 2019; Chabra et al. 2020; Molaei et al. 2023; Gupta, Colvert, and Contijoch 2022; Reed et al. 2021; Xie et al. 2022).

Improving the Representational Capacity of INR

The representational capacity of INRs plays a critical role in determining the quality of reconstruction and rendering. Existing methods for enhancing the INR representational capacity can be broadly classified into two categories. The first is mapping the low-dimensional input coordinates to high-dimensional manifolds using embeddings or hash tables. PEMLP(Tancik et al. 2020) encodes the coordinates with Fourier basis embeddings. Poly-INR (Singh, Shukla, and Turaga 2023) and PINs (Landgraf, Hornung, and Cabral 2022) propose polynomial basis embeddings. DINER (Zhu et al. 2024b, 2025) maps the coordinates with single scale full-resolution hash table. InstantNGP (Müller et al. 2022) maps with multi-scale pyramid hash tables. PIXEL (Kang et al. 2023) uses multiple shifting hash tables. Another approach involves activating the hidden neurons with specialized nonlinearity instead of the traditional ReLU. SIREN (Sitzmann et al. 2020) activates with periodical sinusoidal function, while GARF (Chng et al. 2022) introduces the non-periodic Guassian activation. FINER (Liu et al. 2024) proposes the variable-periodic activation. WIRE (Saragadam et al. 2023) employs wavelet transformations as activation. MFN (Fathony et al. 2020) and Bacon (Lindell et al. 2022) activate with compositing

multiple filters.

However, current INRs still predominantly rely on the vanilla MLP formulated with fully connected layers, which inherently limits their representational capacity, namely, the dimensionality of feature space is linear to the width of the network. To address this limitation, we propose split-layer, which reformulates the model with high-degree forms and significantly enlarge the feature space, thus enhancing the representational capacity of INRs from model formulation perspective.

Split-Layer for INR

In this section, we first elucidate the formulation of INR and the relationship between the feature space and the representational capacity of INR. Then, the split-layer is proposed and analysed on how it modifies the feature space and INR's representational capacity.

Formulation of INR

Given a signal $\{(\vec{x}_i, \vec{y}_i)\}_{i=1}^K$, INR takes its coordinate $\vec{x}_i = [x_i^1, x_i^2, ..., x_i^{d_{in}}]^\top$ as input and outputs the corresponding attributes $\vec{y}_i = [y_i^1, y_i^2, ..., y_i^{d_{out}}]^\top$, where d_{in} and d_{out} are the dimensions of input coordinate and output attribute, K is the length of the signal. Mostly, INR is built upon the multilayer perceptron (MLP), that

$$\begin{aligned} \vec{z}^0 &= \vec{x} \\ \vec{z}^l &= \mathbf{W}^l \vec{z}^{l-1} + \vec{b}^{l-1}, \;\; l = 1 \dots L \\ \vec{z}^l_{act} &= \rho(\vec{z}^l) \\ f(\vec{x}; \theta) &= \vec{z}^L_{act} \end{aligned} \tag{1}$$

where L is the number of layers, $\vec{z}^{\,l}$ is the linear combination of the $l-1$-th layer, and $\vec{z}^l_{act}$ is the output of l-th layer, ρ is the activation function(Sitzmann et al. 2020; Liu et al. 2024), $\mathbf{W}^l$ is the weight matrix and $\vec{b}^{\,l-1}$ is the bias vector, θ refers to the network parameters to be optimized. For the sake of derivation, the width of each layer is defined as C, as a result, $\mathbf{W}^l \in \mathbb{R}^{C\times C}$ and $\vec{z}^{\,l} = \{z_1^l, z_2^l, ..., z_C^l\}$.

Feature space and representational capacity of INR

According to the definition in Eqn.1, the i-th element z_i^l output by l-th layer is

$$z_i^l = w_{i1} z_1^{l-1} + w_{i2} z_2^{l-1} + \cdots + w_{iC} z_C^{l-1}, \tag{2}$$

where w_{ij} refers to the i,j-th element of $\mathbf{W}$. Note that the bias term $\vec{b}^{\,l-1}$ is removed here to facilitate derivation. From this equation, the output is a composition of C individual and linearly independent elements $\{z_1^{l-1}, z_2^{l-1}, \cdots, z_C^{l-1}\}$, thus constituting the basis that spans the *feature space*(Bengio, Courville, and Vincent 2013) $\mathcal{F}^C = \mathcal{S}(z_1^{l-1}, z_2^{l-1}, \cdots, z_C^{l-1})$, which is a $C-dimensional\ Euclidean\ space$.

According to recent study (Yüce et al. 2022; Benbarka et al. 2022; Zhu et al. 2024a), the representational capacity of INR could be viewed as a linear combination of Fourier (or Gauss/Wavelet) bases,

$$f(\mathbf{x}; \theta) \in \left\{ \sum_{\omega' \in \mathcal{H}_\omega^C} c_{\omega'} \sin(\omega' \mathbf{x} + \phi_{\omega'}) \mid c_{\omega'}, \phi_{\omega'} \in Q \right\} \tag{3}$$

where Q is the set of rational number, $\mathcal{H}_\omega^C$ is the supported frequency set whose size is proportional to the number of linearly independent elements in feature space.

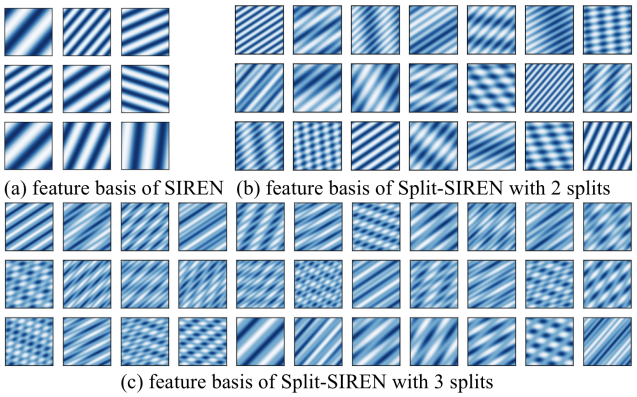
As a result, widening the network layer, for example with a coefficient of 2, will expand the feature space from $\mathcal{F}^C$ to $\mathcal{F}^{2C}$ and thus enhancing the representational capacity of INR, but at the cost of significantly improved computational budget (from C^2 to $4C^2$).

Split-layer reformulates the feature space of INR

To expand the feature space to high dimensionality without increasing the number of parameters and computational budget, we propose the *split-layer* to reformulate the MLP with high-degree forms. As shown in Fig. 1, split-layer splits each layer as several individual parts and then integrates the outputs of these branches through the Hadamard product. Supposing the l-th layer is split into N parts (as shown in Fig. 1b, the neurons of each part is set as $\frac{C}{\sqrt{N}}$ to maintain the total parameters is equivalent to C^2, thus N ranges from 2 to C^2. Correspondingly, the weight matrices of these N branches are denoted as $\mathbf{W}_1^l, \mathbf{W}_2^l, \cdots, \mathbf{W}_N^l \in \mathbb{R}^{\frac{C}{\sqrt{N}} \times \frac{C}{\sqrt{N}}}$, and the input becomes $\vec{z}^{\,l-1} = [z_1^{\,l-1}, z_2^{\,l-1}, \cdots, z_{\frac{C}{\sqrt{N}}}^{\,l-1}]$. In this case, the split-layer with N splits can be formulated as $\vec{z}^{\,l} = \mathbf{W}_1^l \vec{z}^{\,l-1} * \mathbf{W}_2^l \vec{z}^{\,l-1} * \cdots * \mathbf{W}_N^l \vec{z}^{\,l-1}$, where $*$ is the Hadamard product operation, referring to the element-wise product between vectors. In detail, the i-th element z_i^l of $\vec{z}^{\,l}$ could be written as,

$$\begin{aligned} z_i^l &= \sum_{j=1}^{\frac{C}{\sqrt{N}}} w_{ij}^1 z_j^{l-1} * \sum_{j=1}^{\frac{C}{\sqrt{N}}} w_{ij}^2 z_j^{l-1} * \cdots * \sum_{j=1}^{\frac{C}{\sqrt{N}}} w_{ij}^N z_j^{l-1} \\ &= \prod_{n=1}^{N} \left(\sum_{j=1}^{\frac{c}{\sqrt{N}}} w_{ij}^n z_j^{l-1} \right) = \sum_{j_1=1}^{\frac{c}{\sqrt{N}}} \cdots \sum_{j_N=1}^{\frac{c}{\sqrt{N}}} \prod_{n=1}^{N} w_{ij_n}^n \prod_{n=1}^{N} z_{j_n}^{l-1} \\ &= \sum_{(j_1, j_2, \dots, j_N)} \left(\prod_{n=1}^{N} w_{ij_n}^n \right) \left(z_{j_1}^{l-1} z_{j_2}^{l-1} \cdots z_{j_N}^{l-1} \right) \end{aligned} \tag{4}$$

where w_{ij}^n refers to the i,j-th element of the n-th weight matrix $\mathbf{W}_n^l$. Our goal is to calculate the number of different items obtained by the *N-degree symmetric homogeneous polynomials* $z_{j_1}^{l-1} z_{j_2}^{l-1} \cdots z_{j_N}^{l-1}$, which is equivalent to selecting all combinations of N elements from the set $\{x_1, x_2, \cdots, x_{\frac{C}{\sqrt{N}}}\}$ in a reproducible and order-independent manner. As a consequence, the total number of distinct items composing z_i^l is described by the *binomial coefficient* $\binom{\frac{C}{\sqrt{N}}+N-1}{N}$. These homogeneous polynomials are linearly independent to each other, forming the basis that spans a



(a) feature basis of SIREN (b) feature basis of Split-SIREN with 2 splits

(c) feature basis of Split-SIREN with 3 splits

Figure 2: We visualize the features of SIREN with 9 hidden neurons (a), and the corresponding Split-SIREN with 2 splits (b), and corresponding Split-SIREN with 3 splits (c) on a 2D image fitting task. As can be observed, split-layer introduces more diverse feature basis, significantly enlarging the feature space of the original model.

$\binom{\frac{C}{\sqrt{N}}+N-1}{N}-dimensional\ feature\ space\ \mathcal{F}^{\binom{\frac{C}{\sqrt{N}}+N-1}{N}}$. In this way, we obtain an extremely large feature space of $\binom{\frac{C}{\sqrt{N}}+N-1}{N}$ dimensions with the same computations.

Discussion

Feature Comparisons between MLP with Different Splits. As analyzed mentioned above, split-layer reformulates the construction of MLP and extends the feature space without adding network parameters. To verify this issue, we apply the split-layer to the widely used INR architecture, *i.e.*, the SIREN network (Sitzmann et al. 2020). Fig. 2 compares the features output by the first layer of the SIREN. In Fig. 2(a), the original SIREN only provides 9 features where each feature contains only one frequency. By applying the split-layer to the SIREN, it is noticed that more features with complex distributions are produced (as shown in Fig. 2(b) and (c)), verifying the effectiveness of the proposed split-layer in extending the feature space.

Neural Tangent Kernel Perspective. Neural tangent kernel (NTK) (Jacot, Gabriel, and Hongler 2018; Tancik et al. 2020), 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. Previous studies (Tancik et al. 2020; Cai et al. 2024a,b; Liu et al. 2024) theoretically demonstrate that more larger eigenvalues of NTK leads to faster convergence for learning high-frequency components. Fig. 3 left visualizes the distributions of the NTK's eigenvalues of MLP and Split-MLP (with 2 splits) for learning a 1D signal. As can be observed, the NTK's eigenvalues of MLP exhibit the pathological distribution (Cai et al. 2024a,b), mostly concentrated around 10^{-3} and close to 0, with almost no large eigenvalues, thereby limiting the convergence for the high-frequency components. In contrast, the Split-MLP shows a more even distribution of NTK's eigenvalues, significantly expanding the distribution of NTK eigenvalues to the range of $[10^{-2}, 10^{2}]$, offering better convergence performance than the original MLP, thus validating the effectiveness of split-layer form the neural tangent kernel perspective.

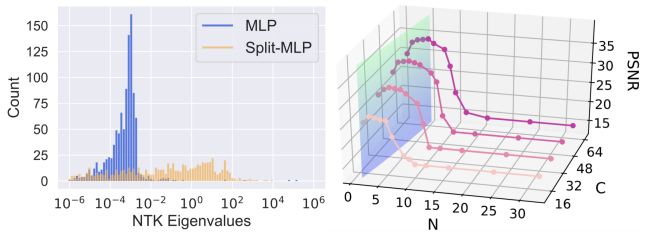


Figure 3: Left: Comparisons of the distribution of NTK's eigenvalues. Most of the MLP's eigenvalues are smaller than 1, while the eigenvalues of Split-MLP are increased to the range of $[10^{-2}, 10^{2}]$, meaning better performance of Split-MLP for representing a signal with high-frequency components. Right: Verification of the Eqn. 5 for obtaining the optimal split. The curved surface visualizes the Eqn. 5 and four curves plots the quality of image fitting with different splits. It is observed that the optimal result of each curve appears near the surface, verifying the robustness of Eqn. 5.

Optimal Split. According to the derivation of the feature space, improving the split N will increase the dimensionality of the feature space at the cost of reducing the size of each weight matrix $\{\mathbf{W}_i^l\}_{i=1}^N$. On the contrary, a smaller weight matrix has less freedom to explore different combinations of characteristics and hierarchical structures (Unterthiner et al. 2020; Navon et al. 2023; Denil et al. 2013). We empirically find that the optimal split could be written as,

$$N^* \approx (0.17C)^{\frac{2}{3}} \tag{5}$$

To verify the robustness of the selection of N^*, we conduct the image fitting task with different settings of network width C. In Fig. 3 right, the curved surface indicates Eqn. 5 and the four curves indicate results with different widths. It is observed that the optimal node of each curve always appears near the surface, verifying the robustness of the Eqn. 5.

Experiments

Tasks. We validate the effectiveness of applying split-layer to INRs on four separate tasks, *i.e.*, 2D image fitting, 2D computed tomography (CT) reconstruction, 3D shape representation, and 5D novel view synthesis. Note that the first 3 tasks are the ones widely used in INR research to evaluate performance, assessing the fundamental models in current INR studies, while the last task represents the main practical uses of INR and needs to be compared with a range of state-of-the-art methods.

Compared methods. For these three tasks, a total of twelve methods are compared, including the MLP with traditional ReLU (marked as ReLU) (Nair and Hinton 2010), MLP with Fourier positional encoding (PEMLP) (Tancik et al. 2020), SIREN (Sitzmann et al. 2020), MLP with Gaussian activation (Gauss) (Ramasinghe and Lucey 2022), WIRE (Saragadam et al. 2023) and FINER (Liu et al. 2024) . As a common practice, the encoding scale of PEMLP is set

Task	Method	ReLU	SIREN	Gauss	PEMLP	WIRE	FINER
Image Fitting PSNR (dB↑)	Baseline	21.24	38.52	31.74	29.60	31.31	38.72
	Split	30.89	39.25	40.84	40.78	35.39	40.09
	Gain	↑45.43%	↑1.90%	↑28.67%	↑37.77%	↑13.03%	↑3.54%
CT Reconstruction PSNR (dB↑)	Baseline	26.78	18.32	27.44	28.11	28.26	25.10
	Split	30.19	29.11	28.72	32.29	31.67	29.58
	Gain	↑12.73%	↑58.90%	↑4.66%	↑14.87%	↑12.07%	↑17.84%
Shape Representation Chamfer distance (↓)	Baseline	1.00e-4	2.44e-5	2.19e-5	9.76e-6	1.38e-5	4.89e-5
	Split	2.01e-5	1.21e-5	5.33e-6	4.59e-6	9.41e-6	1.82e-5
	Gain	↑79.90%	↑50.41%	↑75.66%	↑52.97%	↑31.81%	↑62.78%

Table 1: Results of different INRs on image fitting task (measured in PSNR), CT reconstruction task (measured in PSNR), and shape representation task (measured in chamfer distance error, lower chamfer distance means better performance).

Figure 4: Comparisons of different methods for representing the 2D Image Santorini.

as 10 (Mildenhall et al. 2021), the frequency parameter ω of SIREN is set as 30 (Sitzmann et al. 2020). We replace all the original hidden fully connected layers of these six methods with split-layers to test the effectiveness of split-layer. To facilitate deployment and more effectively illustrate the efficacy of our method, we consistently set the split number as 2, which still yields excellent results. The weights of all the networks are randomly initialized. For SIREN and FINER, we use the specific weight initialization schemes as raised in (Sitzmann et al. 2020; Liu et al. 2024). For the left methods, we use the default LeCun initialization (LeCun et al. 2002).

2D Image Fitting

Configurations. We first use an image representation task to evaluate the performance of applying split-layer to different implicit neural representations. The INR aims at learning a mapping from 2D pixel location to 3D RGB color. We perform experiments on the *Natural* image dataset (Tancik et al. 2020) consisting of 16 512×512 RGB images. We use networks with 4 hidden layers and a width C of 256, which is sufficiently wide. 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 (Kingma and Ba 2014).

Results. Tab. 1 exhibits the average PSNR (Peak Signal-to-Noise Ratio) of these methods. As can be observed, split-layer significantly enhances the performance of ReLU by 9.65dB, achieving a substantial improvement up to 45.4%. Split-layer also significantly enhances the performance of PEMLP and Gauss by 11.18dB and 9.10, up to a remarkable improvement of 37.77% and 28.67%, respectively. The experimental results validate that split-layer can effectively improve the representational capacity of different implicit neural representation models.

Visualization. We visualize the reconstructed images by these methods in Fig. 4. As can be observed, the reconstruction of ReLU is over-smoothed and poor-quality. For PEMLP, the reconstructed image is blurry and fails to restore finer details. For Gauss and SIREN, the reconstructed images exhibit obvious noise. In contrast, applying split-layer

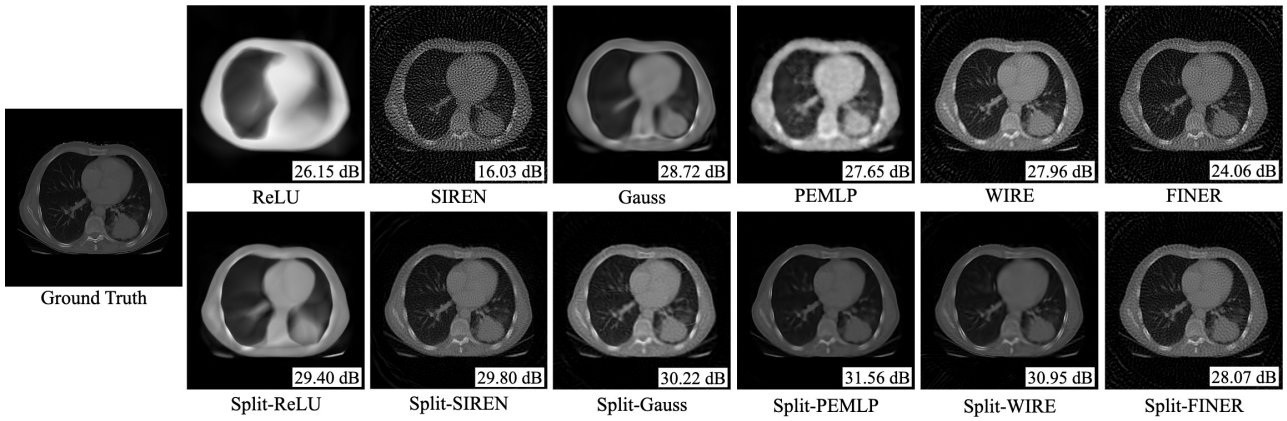


Figure 5: Comparisons of different methods for CT reconstruction. The corresponding error maps are also visualized.

significantly improves the reconstruction quality, with less noise and finer details clearly preserved (as shown by the characters and stages in the zoomed-in read box). Notably, Split-PEMLP achieves a high PSNR of 45.08dB, demonstrating excellent reconstruction. These results underscore the superiority of split-layer over existing methods

2D Computed Tomography

Configurations. For CT reconstruction task, we observe integral projections of a density field instead of direct supervisions. We train a network that takes in 2D pixel coordinates and predicts the corresponding volume density. We conduct the experiments on the x-ray colorectal dataset (Saragadam et al. 2023; Clark et al. 2013), each image has a resolution of 512×512 and is emulated with 100 CT measurements. We use networks with 2 hidden layers and a width C of 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 5,000 iterations using Adam optimizer.

Results. Tab. 1 exhibits the average PSNR of these methods. As can be observed, split-layer significantly enhances the performance of ReLU by 3.41dB, delivering an improvement up to 12.73%. Split-layer also significantly boosts the performance of PEMLP and Gauss by 4.18dB and 1.28dB, achieving an improvement up to 14.87% and 4.66%, respectively. Notably, the performance of original SIREN is very poor, which achieves a PSNR of only 18.32dB. When applied with split-layer, the performance of SIREN is significantly enhanced up to 29.11dB, bringing a remarkable improvement of 58.90%. The experimental results effectively validate that split-layer can generally improve the representational capacity of implicit neural representations.

Visualization We visualize the reconstructed CT images by these eight methods in Fig. 5, the corresponding error maps are also visualized in the bottom right corner of each method. As can be observed, ReLU leads to excessively smooth results, displaying only blurry patterns. The result of PEMLP is noisy, failing to exhibit the precise details. However, when split-layer is applied to ReLU and PEMLP, the reconstruction quality improves significantly, with finer details effectively represented. Additionally, SIREN yields an extremely low PSNR of only 16dB, likely due to the pre-encoded frequencies that introduce a bias inconsistent with the target, resulting in substantial noise. After applying the split-layer to SIREN, the reconstruction quality improves markedly, with the PSNR increasing to 29.80dB.

3D Shape Representation

Configurations. In this section, we demonstrate the representational capacity of INRs 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 INR, 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 Stanford 3D Scanning Repository (Laboratory 2014). We use networks with 2 hidden layers and a width C of 256. All the models are trained for 200 epochs using Adam optimizer. 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 (Lorensen and Cline 1998) with a threshold of 0.5 for evaluation.

Results. Tab. 1 exhibits the experimental results evaluated chamfer distance (lower chamfer distance means better performance). As observed, split-layer markedly improves the performance of ReLU, SIREN, Gauss, and PEMLP by 79.90%, 50.41%, 75.66%, 52.97% on average, demonstrating the general effectiveness of split-layer. Moreover, Split-PEMLP obtains the best performance, achieving a exceedingly low CD of $4.59e-6$ (close to the optimal value 0). The experimental results validate the general effectiveness of split-layer.

Visualization. Fig. 6 visualizes the meshes of thai-statue scene represented by these eight methods. As observed, ReLU makes the surface over-smooth, while split-layer significantly improves such phenomenon allowing finer details to emerge. Besides, SIREN and Gauss introduces too many undesired textures and fluctuations on the surface, indicat-

Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Average
NeRF	31.37	24.50	28.90	34.94	30.71	28.60	28.99	27.27	29.41
Split-NeRF	31.78	24.81	29.34	35.33	31.76	28.87	31.85	27.83	30.20
DVGO	34.07	25.30	32.59	36.75	34.65	29.59	33.12	29.00	31.88
Split-DVGO	34.28	25.49	32.92	36.77	34.97	29.76	33.35	29.27	32.10
DINER	34.49	25.43	33.28	36.45	34.82	29.58	33.43	29.25	32.09
Split-DINER	34.85	25.47	33.39	36.92	35.14	29.59	34.01	29.49	32.36

Table 2: Results of different methods on inverse rendering for novel view synthesis, measured in PSNR. We color code each cell as best, second best, and third best.

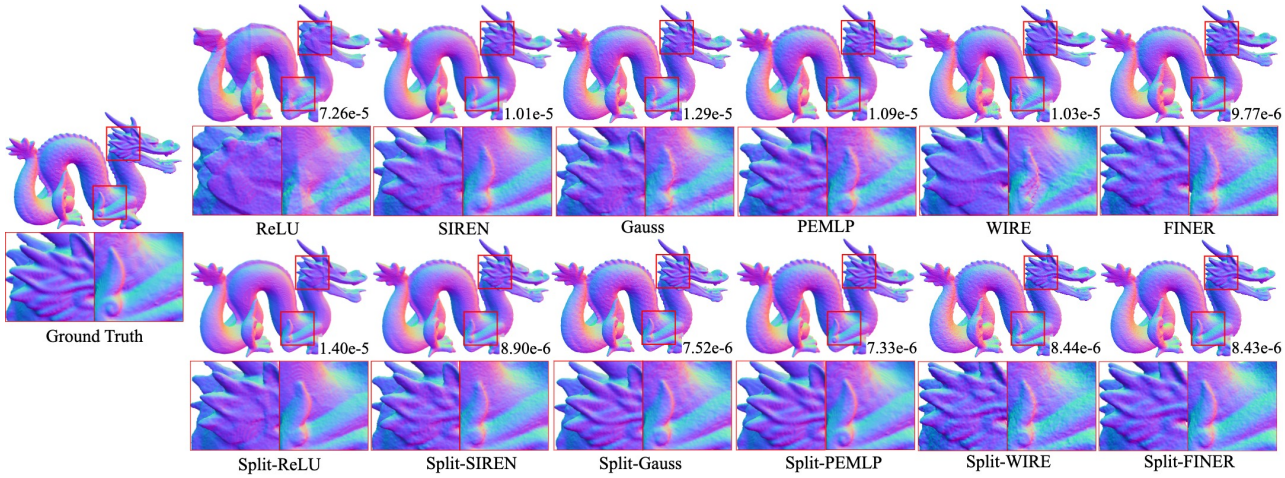


Figure 6: Meshes of Dragon generated with occupancy volumes by various methods.

ing over-fitting to high-frequencies and noise. Furthermore, Split-PEMLP achieves the best representational result, providing clear details without incurring artifacts or noise.

Inverse Rendering for Novel View Synthesis

Configurations. We further demonstrate the effectiveness of split-layer on inverse rendering for novel view synthesis using the neural radiance fields (NeRF) (Mildenhall et al. 2021). NeRF models 3D world as 5D radiance fields using INRs, where the input contains the 3D position and 2D viewing direction of a point and the output attributes, namely, the RGB color and point density. Then the color of each pixel is calculated by querying these attributes along the ray defined by the pixel position and camera's parameters and applying the volume rendering techniques (Max 1995). Finally, the radiance field is optimized by supervising rendered color with the ground truth. Once the radiance field is convergent, the images from any view could be synthesized by following the second step mentioned above. To better verify the effectiveness, we apply split-layer to three popular NeRF baselines, *i.e.*, the original NeRF (Mildenhall et al. 2021) which models the radiance field as a continuous function, the DVGO (Sun, Sun, and Chen 2022) which adopts a discrete form of direct voxel grid optimization, and DINER (Zhu et al. 2024b) optimizes with a disorder-invariant hash-table. Experiments are conduct on the *nerf synthetic* dataset (Mildenhall et al. 2021) with a full resolution of 800×800, following all authors' configurations.

Results. Tab. 2 lists the quantitative comparisons of these six methods. Compared with original results, applying split-layer averagely improves the PSNR of NeRF, DVGO, and DINER up to 0.79dB, 0.22dB, and 0.27dB, respectively. In addition, Split-enhanced DINER achieves the best results among all the methods. These experimental results further verify the generic effectiveness of split-layer.

Conclusion

We propose *split-layer*, which reformulates the fully-connected layer with high-degree forms. We theoretically demonstrate that split-layer can expand the feature space to exceedingly high dimensionality, thus significantly enhancing the INR representational capacity without extra parameters. Extensive experiments validate our theoretical analysis, demonstrating that split-layer can significantly enhance model performance of various INR backbones across various representation and inverse optimization tasks. In the future, we will explore other methods to reformulate the neural network and further enlarge the feature space, such as Hilbert kernel or Gaussian kernel.

Acknowledgment

The work was supported in part by the National Natural Science Foundation of China (Grants No. 62473193, 62231002).

References

Benbarka, N.; Höfer, T.; Zell, A.; et al. 2022. Seeing implicit neural representations as fourier series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2041–2050.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828.

Bubeck, S.; et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4): 231–357.

Cai, Z.; Zhu, H.; Shen, Q.; Wang, X.; and Cao, X. 2024a. Batch Normalization Alleviates the Spectral Bias in Coordinate Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25160–25171.

Cai, Z.; Zhu, H.; Shen, Q.; Wang, X.; and Cao, X. 2024b. Towards the Spectral bias Alleviation by Normalizations in Coordinate Networks. *arXiv preprint arXiv:2407.17834*.

Chabra, R.; Lenssen, J. E.; Ilg, E.; Schmidt, T.; Straub, J.; Lovegrove, S.; and Newcombe, R. 2020. 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*, 608–625. Springer.

Chen, A.; Xu, Z.; Wei, X.; Tang, S.; Su, H.; and Geiger, A. 2023. Dictionary Fields: Learning a Neural Basis Decomposition. *ACM Trans. Graph.*, 40(4): 1–12.

Chng, S.-F.; Ramasinghe, S.; Sherrah, J.; and Lucey, S. 2022. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *European Conference on Computer Vision*, 264–280. Springer.

Clark, K.; Vendt, B.; Smith, K.; Freymann, J.; Kirby, J.; Koppel, P.; Moore, S.; Phillips, S.; Maffitt, D.; Pringle, M.; et al. 2013. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *Journal of digital imaging*, 26: 1045–1057.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Denil, M.; Shakibi, B.; Dinh, L.; Ranzato, M.; and De Freitas, N. 2013. Predicting parameters in deep learning. *Advances in neural information processing systems*, 26.

Fathony, R.; Sahu, A. K.; Willmott, D.; and Kolter, J. Z. 2020. Multiplicative filter networks. In *International Conference on Learning Representations*.

Gao, R.; Si, Z.; Chang, Y.-Y.; Clarke, S.; Bohg, J.; Fei-Fei, L.; Yuan, W.; and Wu, J. 2022. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10598–10608.

Gupta, K.; Colvert, B.; and Contijoch, F. 2022. Neural Computed Tomography. *arXiv preprint arXiv:2201.06574*.

Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5): 359–366.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Kang, N.; Lee, B.; Hong, Y.; Yun, S.-B.; and Park, E. 2023. Pixel: Physics-informed cell representations for fast and accurate pde solvers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8186–8194.

Kansizoglou, I.; Bampis, L.; and Gasteratos, A. 2021. Deep feature space: A geometrical perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6823–6838.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Karniadakis, G. E.; Kevrekidis, I. G.; Lu, L.; Perdikaris, P.; Wang, S.; and Yang, L. 2021. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Laboratory, S. C. G. 2014. The Stanford 3D Scanning Repository. Http://graphics.stanford.edu/data/3Dscanrep/.

Landgraf, Z.; Hornung, A. S.; and Cabral, R. S. 2022. PINs: Progressive Implicit Networks for Multi-Scale Neural Representations. In *International Conference on Machine Learning*, 11969–11984. PMLR.

LeCun, Y.; Bottou, L.; Orr, G. B.; and Müller, K.-R. 2002. Efficient backprop. In *Neural networks: Tricks of the trade*, 9–50. Springer.

Lindell, D. B.; Van Veen, D.; Park, J. J.; and Wetzstein, G. 2022. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16252–16262.

Liu, Z.; Zhu, H.; Zhang, Q.; Fu, J.; Deng, W.; Ma, Z.; Guo, Y.; and Cao, X. 2024. 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*.

Lorensen, W. E.; and Cline, H. E. 1998. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, 347–353.

Ma, X.; Dai, X.; Bai, Y.; Wang, Y.; and Fu, Y. 2024. Rewrite the Stars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5694–5703.

Martel, J. N.; Lindell, D. B.; Lin, C. Z.; Chan, E. R.; Monteiro, M.; and Wetzstein, G. 2021. Acorn: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG)*, 40(4): 1–13.

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

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Molaei, A.; Aminimehr, A.; Tavakoli, A.; Kazerouni, A.; Azad, B.; Azad, R.; and Merhof, D. 2023. Implicit neural representation in medical imaging: A comparative survey. *arXiv preprint arXiv:2307.16142*.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4): 1–15.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Navon, A.; Shamsian, A.; Achituve, I.; Fetaya, E.; Chechik, G.; and Maron, H. 2023. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning*, 25790–25816. PMLR.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174.

Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; and Courville, A. 2019. On the spectral bias of neural networks. In *International Conference on Machine Learning*, 5301–5310. PMLR.

Raissi, M.; Yazdani, A.; and Karniadakis, G. E. 2020. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481): 1026–1030.

Ramasinghe, S.; and Lucey, S. 2022. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, 142–158. Springer.

Reed, A. W.; Kim, H.; Anirudh, R.; Mohan, K. A.; Champley, K.; Kang, J.; and Jayasuriya, S. 2021. 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*, 2258–2268.

Saragadam, V.; LeJeune, D.; Tan, J.; Balakrishnan, G.; Veeraraghavan, A.; and Baraniuk, R. G. 2023. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18507–18516.

Scholkopf, B.; Mika, S.; Burges, C. J.; Knirsch, P.; Muller, K.-R.; Ratsch, G.; and Smola, A. J. 1999. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5): 1000–1017.

Shen, C.; Zhu, H.; Zhou, Y.; Liu, Y.; Yi, S.; Dong, L.; Zhao, W.; Brady, D. J.; Cao, X.; Ma, Z.; et al. 2024. Continuous 3D Myocardial Motion Tracking via Echocardiography. *IEEE Transactions on Medical Imaging*.

Shen, C.; Zhu, H.; Zhou, Y.; Liu, Y.; Yi, S.; Dong, L.; Zhao, W.; Brady, D. J.; Cao, X.; Ma, Z.; et al. 2025. CardiacField: computational echocardiography for automated heart function estimation using two-dimensional echocardiography probes. *European Heart Journal-Digital Health*, 6(1): 137–146.

Shi, J.; Chen, Z.; Li, H.; Zhao, Q.; Lu, M.; Chen, T.; and Ma, Z. 2025. On Quantizing Neural Representation for Variable-Rate Video Coding. In *The Thirteenth International Conference on Learning Representations*.

Shi, J.; Jiang, M.; Lu, M.; Chen, T.; Cao, X.; and Ma, Z. 2024. HINER: Neural Representation for Hyperspectral Image. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9837–9846.

Singh, R.; Shukla, A.; and Turaga, P. 2023. Polynomial Implicit Neural Representations for Large Diverse Datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2041–2051.

Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473.

Sitzmann, V.; Rezchikov, S.; Freeman, B.; Tenenbaum, J.; and Durand, F. 2021. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34: 19313–19325.

Strümpler, Y.; Postels, J.; Yang, R.; Gool, L. V.; and Tombari, F. 2022. Implicit neural representations for image compression. In *European Conference on Computer Vision*, 74–91. Springer.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5459–5469.

Sun, S.; Han, K.; You, C.; Tang, H.; Kong, D.; Naushad, J.; Yan, X.; Ma, H.; Khosravi, P.; Duncan, J. S.; et al. 2024. Medical image registration via neural fields. *Medical Image Analysis*, 97: 103249.

Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; and Ng, R. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547.

Unterthiner, T.; Keysers, D.; Gelly, S.; Bousquet, O.; and Tolstikhin, I. 2020. Predicting neural network accuracy from weights. *arXiv preprint arXiv:2002.11448*.

Xie, Y.; Takikawa, T.; Saito, S.; Litany, O.; Yan, S.; Khan, N.; Tombari, F.; Tompkin, J.; Sitzmann, V.; and Sridhar, S. 2022. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, 641–676. Wiley Online Library.

Yüce, G.; Ortiz-Jiménez, G.; Besbinar, B.; and Frossard, P. 2022. A Structured Dictionary Perspective on Implicit Neural Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19228–19238.

Zhang, O.; Zhou, H.; Feng, B. Y.; Larsson, E. M.; Alcalde, R. E.; Yin, S.; Deng, C.; and Yang, C. 2025. Single-shot volumetric fluorescence imaging with neural fields. *Advanced Photonics*, 7(2): 026001–026001.

Zhou, H.; Feng, B. Y.; Guo, H.; Lin, S.; Liang, M.; Metzler, C. A.; and Yang, C. 2023. Fourier ptychographic microscopy image stack reconstruction using implicit neural representations. *Optica*, 10(12): 1679–1687.

Zhou, Y.; Xu, C.; Jin, Z.; Chen, Y.; Zheng, B.; Wang, M.; Xiong, B.; Cao, X.; and Gu, N. 2024. Physics-Informed Ellipsoidal Coordinate Encoding Implicit Neural Representation for high-resolution volumetric wide-field microscopy. *bioRxiv*, 2024–10.

Zhu, H.; Liu, F.; Zhang, Q.; Cao, X.; and Ma, Z. 2025. RHINO: Regularizing the Hash-based Implicit Neural Representation. *Science China: Information Sciences*, 1–17.

Zhu, H.; Liu, Z.; Zhang, Q.; Fu, J.; Deng, W.; Ma, Z.; Guo, Y.; and Cao, X. 2024a. FINER++: Building a Family of Variable-periodic Functions for Activating Implicit Neural Representation. *arXiv preprint arXiv:2407.19434*.

Zhu, H.; Liu, Z.; Zhou, Y.; Ma, Z.; and Cao, X. 2022. DNF: Diffractive neural field for lensless microscopic imaging. *Optics Express*, 30(11): 18168–18178.

Zhu, H.; Xie, S.; Liu, Z.; Liu, F.; Zhang, Q.; Zhou, Y.; Lin, Y.; Ma, Z.; and Cao, X. 2024b. Disorder-Invariant Implicit Neural Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5463–5478.

Zhu, J.; Zhu, H.; Zhang, Q.; Zhu, F.; Ma, Z.; and Cao, X. 2023. Pyramid NeRF: Frequency Guided Fast Radiance Field Optimization. *International Journal of Computer Vision*, 1–16.

Zohdinasab, T.; Riccio, V.; Gambi, A.; and Tonella, P. 2023. Efficient and effective feature space exploration for testing deep learning systems. *ACM Transactions on Software Engineering and Methodology*, 32(2): 1–38.