# Understanding Bias Terms in Neural Representations

**Weixiang Zhang, Boxi Li, Shuzhao Xie, Chengwei Ren, Yuan Xue, Zhi Wang***
Shenzhen International Graduate School, Tsinghua University
zwx.healthy@gmail.com, libx22@mails.tsinghua.edu.cn
wangzhi@sz.tsinghua.edu.cn

## Abstract

In this paper, we examine the impact and significance of bias terms in Implicit Neural Representations (INRs). While bias terms are known to enhance nonlinear capacity by shifting activations in typical neural networks, we discover their functionality differs markedly in neural representation networks. Our analysis reveals that INR performance neither scales with increased number of bias terms nor shows substantial improvement through bias term gradient propagation. We demonstrate that bias terms in INRs primarily serve to eliminate *spatial aliasing* caused by symmetry from both coordinates and activation functions, with input-layer bias terms yielding the most significant benefits. These findings challenge the conventional practice of implementing full-bias INR architecture. We propose using freezing bias terms exclusively in input layers, which consistently outperforms fully biased networks in signal fitting tasks. Furthermore, we introduce Feature-Biased INRs (Feat-Bias), which initialize input-layer bias with high-level features extracted from pre-trained models. This feature-biasing approach effectively addresses the limited performance in INR post-processing tasks due to neural parameter uninterpretability, achieving superior accuracy while reducing parameter count and improving reconstruction quality. Our code is available at *this link*.

## 1 Introduction

Implicit Neural Representation (INR) [48, 45, 30] represents a novel paradigm in signal representation, utilizing Multilayer Perceptrons (MLPs) to establish continuous mappings from spatial coordinates (e.g., $\langle x, y\rangle$ for images) to their corresponding attributes (e.g., RGB values of pixels). Despite this methodology enables compact and resolution-agnostic representation of diverse natural signals [2, 45, 5, 35, 30, 18], traditional ReLU-based networks suffer from spectral bias [37], which impedes the fitting of high-frequency signal components during INR encoding. Various approaches have been proposed to address this limitation [48, 45, 42, 21, 60, 61], with the SIREN [45] and its variants [27] emerging as the most widely adopted solutions. These networks implement periodic activation functions, such as $\sin(x)$ and $\sin((|x|+1)x)$, aligning with formulation of Fourier series [3] and achieving state-of-the-art performance in various signal reconstruction tasks. Leveraging these design principles, INRs have demonstrated significant potential, particularly in medical imaging [31, 52, 51], data compression [11, 13, 46], and inverse problem solving [30, 25, 40].

Although periodic activation functions in INRs have demonstrated exceptional interpretability and performance, the role of **bias terms** (i.e., trainable constants added to the weighted sum of inputs for a neuro) in such special networks remains unexplored. In traditional deep learning practices, it is acknowledged that bias terms can enhance model performance by enabling activation function shifts [16], and empirical studies indicate that they significantly enhance classification accuracy, often serving as crucial contributors to final logits [49]. Current research in INRs [45, 27, 17, 15] adheres to this conventional design, incorporating bias terms in each MLP layer. However, upon closer

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

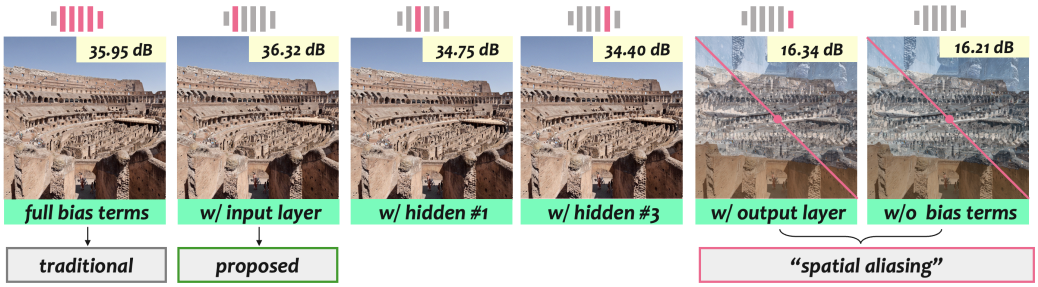


Figure 1: Impact of bias terms in INRs. Pink and Ashen in the top row indicate layers with and without bias terms, respectively. Applying bias terms (without gradient propagation[2]) to the input layer achieves optimal performance, while lacking bias terms or only applying them to output layer results in *"spatial aliasing"* manifested as central symmetry in failed reconstructions.

examination of periodic activation functions, the shifting capacity of bias terms should intuitively be significantly diminished due to the boundedness and periodicity of sine functions, in contrast to the linear growth of ReLU activation. In response to it, we propose to critically reassess bias terms in INRs by examining two questions: (1) *How significant is the impact of bias terms on INRs?* and (2) *What unique functions do bias terms serve in INRs?*

To answer these questions, we conduct empirical studies and make several findings. Consistent with intuition, bias terms $\mathbf{b}$ have significantly smaller effects than weight parameters $\mathbf{W}$, and increasing their magnitude does not consistently enhance performance. However, we unexpectedly find that optimizing bias terms through gradient descent has negligible impact on reconstruction quality. Instead, the mere presence of properly distributed bias terms proves enough critical, particularly benefiting the input layer. Based on these observations, we uncover that the primary role of bias terms in INRs is not to shift activation functions as in traditional MLPs, but rather to ***eliminate spatial aliasing caused by symmetry from both coordinates and activations***. Without bias modulation, input coordinates symmetric about the center (e.g., $\langle m, n\rangle$ and $\langle -m, -n\rangle$) would produce identical outputs, leading to failed reconstructions with central symmetry (see the right part of Fig. 1), a phenomenon we term *spatial aliasing*. Since this symmetry issue originates in coordinate space, bias terms in the input layer yield the most significant benefits. Moreover, given that bias terms have limited impact on shifting activation functions in periodically activated INRs, their presence in non-input layers, along with gradient propagation, proves negligible or even detrimental to INR encoding.

Based on these findings, we challenge the conventional practice of full-bias architecture in INR signal fitting, demonstrating that applying frozen bias terms exclusively to input layers consistently outperforms fully biased networks, achieving optimal performance relative to parameter count. Beyond enhancing INR's expressive capability, our analysis reveals an opportunity to improve *INR post-processing*, i.e., developing neural networks to process INR parameters [62, 23, 19, 15]. While this emerging task has gained importance with the proliferation of INR-represented data (also known as functa [12]), its performance remains limited by the inherent interpretability challenges of neural parameters. For instance, even the recently proposed end-to-end INR classification framework [15] achieves only approximately 60% accuracy on the CIFAR-10 dataset. Unlike existing approaches that focus on parsing intractable neural parameters [62, 23, 19] or designing task-specific end-to-end frameworks [15], we propose Featuring INRs with Bias Terms (Feat-Bias), which directly embeds high-level features from pre-trained encoders (e.g., ViTs [10, 34]) into input-layer bias space, maintaining these values throughout INR encoding. This design is motivated by our key findings that bias term gradient propagation does not affect INR reconstruction quality, thus making bias terms ideal candidates for storing well-established features. Through this approach, we achieve substantial performance improvements using only a lightweight MLP as the neural parameter processor, offering an elegant solution that simultaneously enhances reconstruction quality, reduces model parameters, and improves INR post-processing performance.

[2]Bias terms with gradient propagation yields nearly identical results ( $\pm$0.01dB difference for each case).

## 2 Impact of Bias Terms in INRs

### 2.1 Intuitive Analysis

Given a typical $l$-layer multilayer perceptron, the output value of the $l$-th layer can be formulated as:

$$\mathbf{z}_l = f(\mathbf{z}_{l-1}) = \sigma(\mathbf{W}_l\mathbf{z}_{l-1} + \mathbf{b}_l), \tag{1}$$

where $\mathbf{W}_l$ and $\mathbf{b}_l$ denote the weights and bias at layer $l$, handling linear transformation from the previous layer's output $\mathbf{z}_{l-1}$, and $\sigma(\cdot)$ represents the nonlinear activation function. In common practice, piecewise linear functions such as ReLU are employed as activation functions, owing to their computational efficiency and proven effectiveness across various learning-based models. However, such networks demonstrate limited performance when applied to Implicit Neural Representations (INRs), which aims to represent natural signals through coordinate-based neural networks. Formally, an INR can be defined as $F_\theta : \mathbf{x} \in \mathbb{R}^i \mapsto \mathbf{y} \in \mathbb{R}^o$, mapping an $i$-dimensional coordinate $\mathbf{x}$ to a $o$-dimensional signal $\mathbf{y}$. Given a signal set $\mathcal{S} = \{\mathbf{x_i}, \mathbf{y_i}\}_{i=1}^N$ comprising $N$ pairs of coordinates $\mathbf{x_i}$ and their corresponding signal values $\mathbf{y_i}$, the objective is to fit an $L$-layer MLP $F_\theta(\mathbf{x})$ to the ground truth $\mathbf{y}$ with minimal loss.

In this context, ReLU-based MLPs often yield suboptimal results due to *spectral bias* [37], characterized by preferential fitting of low-frequency over high-frequency components. To address this limitation, periodically activated representation network [45, 27] employ sine-based activation functions, aligning with Fourier-based signal decomposition [3] and demonstrates superior performance across various signal fitting tasks [61]. In these architectures, Eq. 1 can be rewritten as $\mathbf{z}_l = f(\mathbf{z}_{l-1}) = \sin\left(\omega_0\alpha^l(\mathbf{W}_l\mathbf{z}_{l-1} + \mathbf{b}_l)\right)$, where $\omega_0$ controls the network frequency, and $\alpha^l$ determines the activation function's periodicity. SIREN [45] sets $\alpha^l = 1$ for a static period of $2\pi$, while FINER [27] employs $\alpha^l = |\mathbf{W}_l\mathbf{z_{l-1}} + \mathbf{b}_l| + 1$.

Intuitively, the effect of bias terms differs significantly between ReLU and sine activations. In ReLU-based networks, bias terms can provide more direct and substantial shifting of activated values, thus exerting greater influence on the network's nonlinear behavior. We can define $\delta(x, b) = \sigma(x + b) - \sigma(x)$ as an indicator to represent the effect of bias terms on activation function. For ReLU activation, where $\sigma(x) = \max(0, x)$, we obtain $\delta(x, b) = \max(0, x + b) - \max(0, x)$. Its value is bounded by $b$: $|\delta(x, b)| \leq |b|$, achieving this bound under the condition $x \geq 0 \cap b \geq -x$. In networks with periodic activation, where $\sigma(x) = \sin(\omega\alpha(x))$, we derive:

$$\delta(x, b) = 2\cos(\omega\alpha(x + \frac{2}{b}))\sin(\frac{2\omega\alpha}{b}), \tag{2}$$

with $|\delta(x, b)| \leq 2\sin(\frac{2\omega\alpha}{b}) \leq 2$. Comparing $\delta(x, b)$ between these activations reveals that bias terms provide unbounded control in ReLU networks, whereas in periodically activated networks, their modulation capability is inherently bounded by the sine function, constraining their effect on nonlinear modeling.

Except for unbounded magnitude, bias terms in ReLU activation provide stronger threshold control capabilities, a feature absent in sine activation. With ReLU activation $\sigma(x + b) = \max(0, x + b)$, neurons activate ($\sigma(x + b) > 0$) when $x > -b$, where bias $b$ directly modulates this activation threshold. Minor adjustments in $b$ can toggle neurons between active and inactive states, establishing an effective *gating mechanism*. In contrast, bias terms in sine activation merely induce *phase shifts* rather than implementing hard gating states. Furthermore, the periodic nature of sine functions constrains the effect of bias terms to a $2\pi$ period, further diminishing their contribution to the network's expressiveness.

Based on the preceding analysis, we can hypothesize that the shifting effect of bias terms in INRs is significantly diminished. To validate this hypothesis and understand its implications for signal fitting tasks, comprehensive empirical studies are necessary. Therefore, we propose to critically reassess bias terms in INRs by examining two fundamental questions: (1) *How significant is the impact of bias terms on INRs?* and (2) *What unique functions do bias terms serve in INRs?* These questions will be investigated in Sec. 2.2 and Sec. 2.3, respectively.

### 2.2 Empirical Studies

To validate our analysis from Sec. 2 regarding bias terms' impact in INRs, we conduct empirical studies under various bias-related configurations. Following prior work [41, 27, 61, 48], we implement

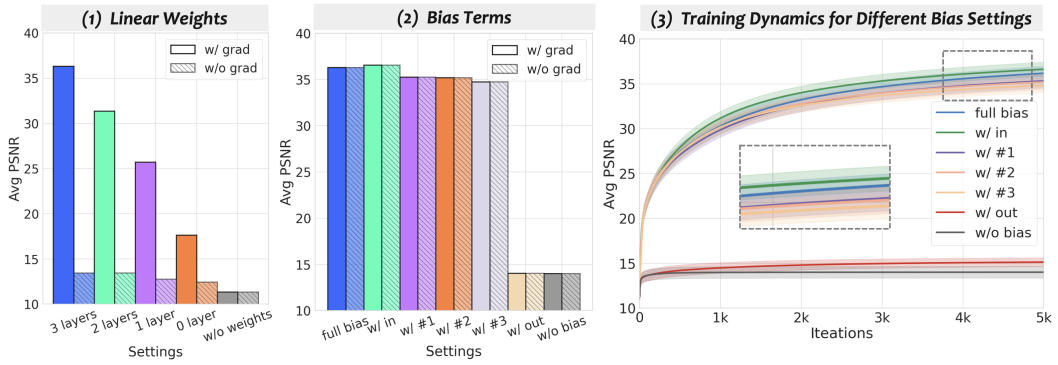


Figure 2: Left: (1) Comparative analysis of INR performance with varying numbers of weights; Middle: (2) Empirical studies on the impact of bias terms in INRs. Right: (3) Training dynamics under different bias configurations. Results demonstrate that applying bias terms exclusively to input layers yields optimal performance.

2D image fitting tasks on the Div2K dataset [1] using a $3 \times 256$ SIREN network (detailed settings and alternative backbones are provided in the appendix).

We first examine whether model performance scales with the number of bias terms. For comparison, we conduct parallel studies with varying configurations of weights $\mathbf{W}_l$ across different layers. For weights, we systematically retain different numbers of layers to investigate their effects (Fig. 2-(1)). For bias, we selectively preserve bias in specific layers while eliminating it in others for finer-grained control (Fig.2-(2)). Results shown in Fig. 2 demonstrate that weights are significantly more crucial for network performance, exhibiting a strong linear relationship between parameter count and reconstruction quality. In contrast, bias terms show minimal impact; even reducing bias terms to approximately 25% of their original number (by retaining only one layer's bias) maintains nearly consistent reconstruction quality, supporting our intuitive analysis:

**(Observation #1)** *Compared to weights, bias terms exhibit substantially smaller effects on INRs' performance, and increasing their magnitude does not consistently lead to performance improvements.*

To investigate bias effects further, we conduct experiments by blocking gradient backpropagation. Fig. 2-(1) shows that weight optimization is essential for signal reconstruction, as networks fail without their gradients. In contrast, we observe an unexpected phenomenon for bias terms:

**(Observation #2)** *The gradient optimization of bias has negligible impact on INRs.*

Our observations reveal that the impact of bias terms on INRs is even more limited than initially hypothesized, with their gradient updates having minimal effect on reconstruction quality. This suggests structural redundancy in current INR architectures, where computational resources are inefficiently allocated to storing and updating largely ineffective bias parameters. However, completely eliminating bias terms or restricting them to the output layer significantly degrades performance, indicating that strategic bias placement is more fundamental than optimization. Through comprehensive analysis of bias configurations, we discovered that bias terms in the input layer achieve optimal performance relative to parameter count. Notably, architectures with frozen input-layer bias consistently outperform fully biased configurations. In summary:

**(Observation #3)** *The existence of bias terms plays a more essential role than their optimization, with neurons in the input layer benefiting most significantly from this property.*

### 2.3 Bias can Eliminate *Spatial Aliasing*

Although empirical studies in Sec. 2.2 demonstrate that bias terms have minimal yet indispensable effects on INRs, the underlying mechanism remains unclear: *What unique functions do bias terms serve in INRs?* As revealed in Observation #2, the negligible impact of bias term optimization suggests that their conventional role in enhancing nonlinear representational capacity through activation value shifting, fundamental to ReLU-based networks, no longer applies to INRs. Upon examining reconstruction failures without bias modulation, we observe a consistent phenomenon:

Figure 3: Visualization for *spatial aliasing*: reconstructed signals demonstrate central symmetry, manifesting as aliasing artifacts between distinct regions of the input. Green and Pink denote the ground truth and reconstructed signals with *spatial aliasing*, respectively.

pixel intensities in the reconstructed image exhibit symmetry around the image center, resulting in aliasing effects between different regions of the input image, which we term *spatial aliasing*. We propose that the primary impact of bias terms in INRs is to eliminate this *spatial aliasing*.

**Why does *spatial aliasing* happen in bias-free INRs?** We attribute this phenomenon to the combined effect of activation function symmetry and input coordinate space structure. In INR training, it is common practice to rescale the coordinate $\mathbf{x}$ to $\tilde{\mathbf{x}}$ as: $\tilde{\mathbf{x}} = \frac{2\mathbf{x}}{\max(\mathbf{x})-\min(\mathbf{x})} - \frac{1}{2}$, which normalizes $\tilde{\mathbf{x}}$ to $[-1, 1]$. This normalization has been proven optimal for INR overfitting [45, 60]. Under this configuration, coordinates that are symmetric about the image center generate values of equal magnitude but opposite signs, which, in the absence of bias modulation, directly induces spatial aliasing. Specifically, consider the input layer of a bias-free INR with centrally symmetric coordinates $\langle m, n\rangle$ and $\langle -m, -n\rangle$, where $m, n \in [-1, 1] \setminus \{0\}$. The activated value of the $i$-th neuron can be formalized as

$$z_0^i = f(m, n) = \sin(\omega_0 \alpha^0 (W_i^0 m + W_i^1 n)). \tag{3}$$

This function exhibits central symmetry, satisfying $f(-m, -n) = -f(m, n)$, a property that extends to all neurons in the bias-free network. While removing symmetry from either the coordinate space or activation function would eliminate spatial aliasing, it would significantly degrade reconstruction quality compared to the default symmetric configuration (see Sec. 4.3). In summary:

**Proposition 1** *The primary role of bias in periodically activated INRs serves as to eliminate spatial aliasing resulted from the the symmtery of both coordinate space and activation function, with neglectable effect on enhance the nonlinear expressiveness of model with activation shifting.*

**Discussion.** The distinct role of bias terms in INRs, as established in Proposition 1, aligns with both our intuitive analysis (Sec. 2) and empirical observations(Sec. 2.2). This finding provides additional explanations for the training dynamics under various bias configurations in Fig. 2-(3): **(1)** Bias terms achieve maximal effectiveness in the input layer due to direct coordinate processing. While hidden layers can also mitigate spatial aliasing, their efficacy gradually diminishes with increasing depth from the input layer. **(2)** The output layer's limited neurons (matching the output signal dimension, three for images) provide only marginal improvement over bias-free networks. This minimal bias modulation proves insufficient to fully address spatial aliasing due to the extremely constrained number of bias. **(3)** More bias terms do not necessarily enhance performance: INRs with bias terms exclusively in the input layer outperform fully-biased configurations. This suggests that while input layer bias effectively eliminates spatial aliasing, bias terms in deeper layers provide negligible activation shifting and potentially act as low-magnitude noise, slightly compromising network overfitting capacity.

## 3 Application: Featuring INRs with Bias Terms

In this section, we introduce Featuring INRs with Bias (Feat-Bias), a method that leverages our analysis of bias terms to enhance INR post-processing tasks, which aim to develop neural networks for processing INR parameters (e.g., INR classification task) [62, 23, 19, 15]. While this emerging field has gained significance with the proliferation of INR-represented data (also known as functa [12]), performance remains limited due to the inherent interpretability challenges of neural parameters. For instance, even the recently proposed end-to-end INR classification framework [15] achieves only approximately 60% accuracy on the CIFAR-10 dataset. Current methods aiming on INR post-processing broadly fall into two categories: symmetry-equivariant parsers for neural parameters and task-specific end-to-end frameworks. The former focuses on designing parameter parsers that

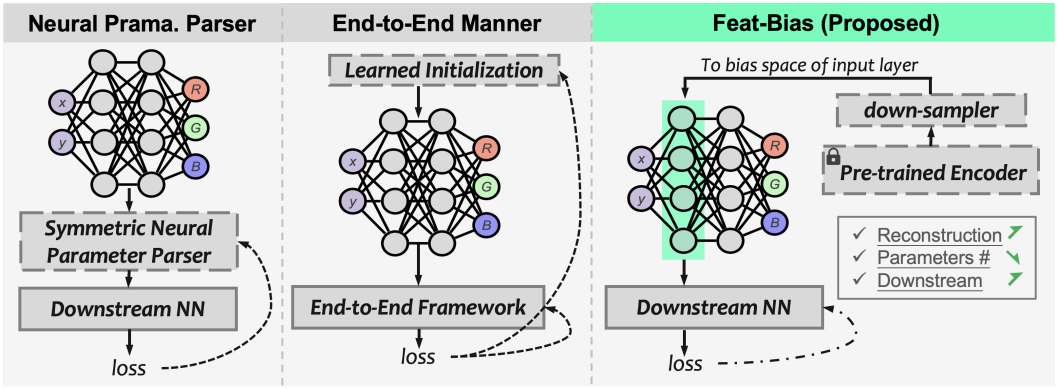


Figure 4: Comparison of mechanisms in INR post-postprocessing. Left: Neural parameter parsers that address the inherent permutation symmetries of neurons. Middle: End-to-end frameworks that integrate INR encoding directly into task-specific networks. Right: Feat-Bias (proposed) that embeds high-level features into input-layer bias initialization. Our method leverages the understanding of bias terms in INRs, addressing neural parameter complexity through a straightforward approach that surpasses existing methods.

consider the inherent permutation symmetries of neurons, while the latter integrates INR encoding directly into task-specific networks, demonstrating rather promising results.

Our Feat-Bias method differs by directly embedding high-level features into neural parameters (specifically bias terms of the input layer) at the initialization step, maintaining these values fixed throughout overfitting. This design is motivated by our empirical observations and Proposition 1: since the backpropagation of bias terms in INRs does not affect reconstruction, these values remain constant throughout the INR encoding process. This stability makes bias terms ideal storage spaces for high-level features extracted from well-established pre-trained encoders (e.g., ViTs [10, 34]). By utilizing a lightweight MLP to process these bias values, we achieve excellent performance while circumventing the challenging process of extracting features from intractable weights.

Specifically, we find that feature vectors extracted from pre-trained encoders approximately follow a uniform distribution. By rescaling these features to $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$ (where $n$ denotes the number of input neurons) to match default bias initialization, we maintain reconstruction performance while significantly enhancing INRs' feature representation capabilities. Moreover, while INR input layer dimensions can be flexibly adjusted, pre-trained models produce fixed-dimension features (e.g., 768 dimensions for ViT-Base). This dimensional mismatch can be efficiently resolved through linear interpolation. Due to the high expressiveness of the encoded features, we achieve state-of-the-art performance in INR post-processing tasks even with significant dimensionality reduction. While Feat-Bias offers limited insights into permutation-symmetric neural parameter processing, it demonstrates a practical application of our understanding of bias terms in neural representations, providing an efficient solution for INR post-processing.

# 4 Experiments

## 4.1 INR Representation Task

**Experimental Settings.** In this section, we demonstrate how our findings regarding bias terms can enhance INR representation performance,serving as a detailed extension of our empirical studies (Sec. 2.2). Following prior work [41, 27, 61, 48], we conduct 2D image fitting tasks on the DIV2K dataset [1] (additional datasets are detailed in supplementary materials). We employ a $3\times256$ MLP with SIREN [45] and FINER [27] architectures, setting the total number of iterations $T$ to 5000. The reconstruction quality is evaluated using PSNR, SSIM [50], and LPIPS [58] metrics. All experiments in this section were performed using the PyTorch framework [36] on NVIDIA RTX 3090 GPU with 24.58 GB VRAM.

**Quantitative Results.** The quantitative results are presented in Tab. 1. Building upon our findings in Sec. 2.2, we challenge the traditional INR architecture that applies bias terms in all layers (Standard) by proposing a model that utilizes bias terms solely in the input layer (Input Bias). We evaluate this against alternative bias configurations: a network without any bias terms (Bias Free), and networks with bias terms only in the first hidden layer ($H_1$ Bias) or output layer (Output Bias). Except for the Standard configuration, all variants operate without bias term backpropagation, as gradient propagation through bias terms yields negligible differences ($\pm$ 0.01dB for each case). As shown in Tab. 1, our proposed Input Bias consistently enhances reconstruction quality across both SIREN and FINER architectures, achieving maximum PSNR improvements of 0.88 dB and 1.13 dB, respectively. The Bias Free and Output Bias configurations exhibit spatial aliasing artifacts due to the absence of appropriate bias terms, confirming our analysis in Sec. 2.3. Furthermore, our mechanism not only improves performance but also reduces the parameter count of each INR by approximately $\frac{(l-1)n}{ln^2+ln}$, where $l$ denotes the number of hidden layers and $n$ represents the neuron count per hidden layer.

Table 1: Results of 2D image fitting tasks with varying bias configurations[3].

| Settings | 1k Iterations PSNR ↑ | 1k Iterations SSIM↑ | 3k Iterations PSNR↑ | 3k Iterations SSIM↑ | 5k Iterations PSNR↑ | 5k Iterations SSIM↑ | 5k Iterations LPIPS↓ |
|---|---|---|---|---|---|---|---|
| Standard | $30.30_{\pm 0.05}$ | $0.881_{\pm 0.002}$ | $34.69_{\pm 0.07}$ | $0.948_{\pm 0.001}$ | $36.19_{\pm 0.05}$ | $0.959_{\pm 0.001}$ | $0.023_{\pm 0.001}$ |
| Bias Free$^{\oslash}$ | $13.98_{\pm 0.00}$ | $0.556_{\pm 0.000}$ | $14.00_{\pm 0.00}$ | $0.574_{\pm 0.001}$ | $14.03_{\pm 0.01}$ | $0.577_{\pm 0.000}$ | $0.415_{\pm 0.000}$ |
| $H_1$ Bias | $29.87_{\pm 0.02}$ | $0.870_{\pm 0.001}$ | $34.00_{\pm 0.03}$ | $0.938_{\pm 0.001}$ | $35.34_{\pm 0.04}$ | $0.950_{\pm 0.001}$ | $0.035_{\pm 0.001}$ |
| Output Bias$^{\oslash}$ | $14.49_{\pm 0.01}$ | $0.562_{\pm 0.001}$ | $14.96_{\pm 0.01}$ | $0.583_{\pm 0.000}$ | $15.13_{\pm 0.01}$ | $0.588_{\pm 0.000}$ | $0.407_{\pm 0.002}$ |
| Input Bias | $31.18_{\pm 0.01}$ | $0.902_{\pm 0.001}$ | $35.32_{\pm 0.01}$ | $0.953_{\pm 0.002}$ | $36.63_{\pm 0.01}$ | $0.962_{\pm 0.00}$ | $0.019_{\pm 0.001}$ |
| Standard | $32.25_{\pm 0.09}$ | $0.916_{\pm 0.002}$ | $36.83_{\pm 0.07}$ | $0.962_{\pm 0.001}$ | $38.47_{\pm 0.05}$ | $0.971_{\pm 0.001}$ | $0.014_{\pm 0.001}$ |
| Bias Free$^{\oslash}$ | $13.99_{\pm 0.01}$ | $0.568_{\pm 0.000}$ | $14.01_{\pm 0.00}$ | $0.581_{\pm 0.003}$ | $14.03_{\pm 0.01}$ | $0.583_{\pm 0.001}$ | $0.415_{\pm 0.000}$ |
| $H_1$ Bias | $31.08_{\pm 0.03}$ | $0.894_{\pm 0.001}$ | $35.23_{\pm 0.01}$ | $0.949_{\pm 0.000}$ | $36.62_{\pm 0.02}$ | $0.958_{\pm 0.001}$ | $0.026_{\pm 0.001}$ |
| Output Bias$^{\oslash}$ | $14.51_{\pm 0.01}$ | $0.573_{\pm 0.000}$ | $14.98_{\pm 0.01}$ | $0.590_{\pm 0.000}$ | $15.14_{\pm 0.00}$ | $0.594_{\pm 0.000}$ | $0.406_{\pm 0.001}$ |
| Input Bias | $33.38_{\pm 0.06}$ | $0.932_{\pm 0.001}$ | $37.90_{\pm 0.03}$ | $0.968_{\pm 0.001}$ | $39.29_{\pm 0.02}$ | $0.974_{\pm 0.000}$ | $0.010_{\pm 0.000}$ |

### 4.2 INR Classification Task

**Experimental Settings.** In this section, we evaluate the effectiveness of our proposed Feat-Bias (Sec. 3) on INR downstream tasks. Following prior work [62, 19, 15], we focus on INR classification as the primary task. We compare Feat-Bias against state-of-the-art methods including NFN [62], ScaleGMN [19], $\text{Zoo}_{[\text{ViT}]}$ [29], and MWT [15], encompassing both symmetry-equivariant neural parameter parsers and end-to-end approaches. Due to code availability constraints, results marked with † are quoted from original papers. The downstream network of Feat-Bias consists of a lightweight $3 \times 256$ MLP classifier, trained for 1000 iterations using a cosine scheduler with a learning rate of $1\text{e}{-3}$. All experiments are repeated three times, reporting both mean and standard deviation. Additional experiments regarding this section can be found in supplementary material.

**Dataset of Neural Representations.** Beyond standard image datasets, establishing a large-scale dataset of neural representations is crucial. While Implicit-Zoo [29] provides CIFAR-10 in neural representation form, their architecture is incompatible with Feat-Bias, which requires specific bias initialization in the input layer. Therefore, we re-train the CIFAR-10 neural representation dataset using a $1 \times 64$ SIREN network trained for 1000 epochs, maintaining alignment with Implicit-Zoo [29] parameters except for the bias initialization scheme. Our implementation achieves an average reconstruction PSNR of 38dB across the CIFAR-10 dataset, with an average training time of 3.11 seconds per image versus Implicit-Zoo's reported 10 seconds.

**Quantitative Results.** Tab. 2 shows the classification results for CIFAR-10 dataset [24], where we adopt Implicit-Zoo's default configuration with a $1 \times 64$ SIREN network to ensure consistent experimental conditions across all comparisons. For all baselines except end-to-end MWT [15] and our method, we directly implement the released checkpoints from Implicit-Zoo as input to the neural parameter parser network. As evidenced by the results, our method achieves superior performance across accuracy, precision, and F1 score metrics, demonstrating Feat-Bias's effectiveness in INR post-processing tasks. While our reported performance is lower than the original MWT paper due to

[3]Settings without underlines represent standard SIREN implementations, while underlined settings correspond to FINER; $\oslash$ represents reconstructions exhibiting *spatial aliasing*.

utilizing smaller network architecture ($1 \times 64$ versus their $3 \times 256$), it is noteworthy that even their state-of-the-art results on CIFAR-10 classification peaked at 60% accuracy, substantially below the performance of our method. Additionally, our approach maintains high reconstruction quality without the accuracy-reconstruction trade-off commonly observed in end-to-end methods. Furthermore, by implementing a lightweight MLP processor, we significantly reduce both computational overhead and parameter count, bringing execution time from minutes (min) to seconds (sec) [4].

Table 2: INR classification on CIFAR-10 Datasets

| Method | *Classification Task* Accuracy (%) ↑ | Precision (%) ↑ | F1 ↑ | Time ↓ | Params (kilo #) ↓ | *INRs* PSNR (dB) ↑ | SSIM ↑ |
|---|---|---|---|---|---|---|---|
| $\text{NFN}_{\text{NP}}$ [62] | $26.19_{\pm 0.06}$ | $24.62_{\pm 1.83}$ | $23.30_{\pm 1.39}$ | $25.02_{\pm 0.98}$ (min) | 544 | 34.59 | 0.968 |
| $\text{NFN}_{\text{HNP}}$ [62] | $25.61_{\pm 0.67}$ | $28.10_{\pm 1.39}$ | $23.89_{\pm 0.65}$ | $23.01_{\pm 0.94}$ (min) | 1672 | 34.59 | 0.968 |
| ScaleGMN [19] | $55.31_{\pm 0.64}$ | $56.10_{\pm 1.33}$ | $55.25_{\pm 0.95}$ | $215.86_{\pm 0.28}$ (min) | 397 | 34.59 | 0.968 |
| ScaleGMN-B [19] | $56.14_{\pm 0.80}$ | $57.67_{\pm 0.53}$ | $56.12_{\pm 0.79}$ | $256.32_{\pm 0.99}$ (min) | 492 | 34.59 | 0.968 |
| WT [15] | $39.34_{\pm 2.13}$ | $38.55_{\pm 2.05}$ | $38.47_{\pm 2.15}$ | $71.75_{\pm 1.25}$ (min) | 261 | 29.08 | 0.887 |
| $\text{MWT}_{\text{Mid-Task}}{}^{\ddagger}$ [15] | $43.38_{\pm 1.35}$ | $42.90_{\pm 1.43}$ | $42.95_{\pm 1.42}$ | $79.74_{\pm 0.98}$ (min) | 261 | 27.53 | 0.868 |
| MWT [15] | $46.94_{\pm 0.37}$ | $46.48_{\pm 0.50}$ | $46.51_{\pm 0.46}$ | $79.49_{\pm 2.44}$ (min) | 261 | 23.23 | 0.695 |
| $\text{Zoo}_{\text{[ViT]}}$ [29]$^{\dagger}$ | $80.82_{\pm 0.86}$ | $80.76_{\pm 0.87}$ | $80.75_{\pm 0.86}$ | / | ∼5k | 34.59 | 0.968 |
| $\text{Zoo}_{\text{[ViT]}}$ [29]$^{\dagger}$ + S | $80.24_{\pm 0.47}$ | $80.49_{\pm 0.63}$ | $80.44_{\pm 0.57}$ | / | ∼5k | 34.59 | 0.968 |
| $\text{Zoo}_{\text{[ViT]}}$ [29]$^{\dagger}$ + LC | $81.33_{\pm 0.23}$ | $81.29_{\pm 0.22}$ | $81.30_{\pm 0.23}$ | / | ∼5k | 34.59 | 0.968 |
| $\text{Zoo}_{\text{[ViT]}}$ [29]$^{\dagger}$ + LP | $79.51_{\pm 0.23}$ | $79.37_{\pm 0.34}$ | $79.37_{\pm 0.35}$ | / | ∼5k | 34.59 | 0.968 |
| $\text{Zoo}_{\text{[ViT]}}$ [29]$^{\dagger}$ + LP$^{*}$ | $81.57_{\pm 0.29}$ | $81.53_{\pm 0.30}$ | $81.51_{\pm 0.30}$ | / | ∼5k | 34.59 | 0.968 |
| $\text{Feat-Bias}_{\text{[ViT]}}$ | $91.68_{\pm 0.19}$ | $91.70_{\pm 0.19}$ | $91.68_{\pm 0.19}$ | $64.96_{\pm 11.34}$ (sec) | 151 | 37.99 | 0.991 |
| $\text{Feat-Bias}_{\text{[DINOV2]}}$ | $93.78_{\pm 0.21}$ | $93.80_{\pm 0.22}$ | $93.78_{\pm 0.22}$ | $64.35_{\pm 10.91}$ (sec) | 151 | 38.02 | 0.991 |

‡ denotes $\omega_{task}$ reported in MWT [15].

### 4.3 Ablation Study: Additional Verification for *Spatial Aliasing*

**Motivations and Settings.** In this section, we further investigate the source of spatial aliasing and elucidate why bias terms cannot be entirely eliminated from INR architectures. Consistent with the settings in Sec. 4.1, we conduct an ablation study examining the symmetry of coordinate space and activation functions to explore their impact on the reconstruction quality of INRs. We denote the elimination of coordinate space symmetry as w/o coord. sym., achieved by rescaling coordinates to $[0, 1]$ rather than $[-1, 1]$. Similarly, w/o act. sym. ($\pm 1$) indicates the removal of activation function symmetry by modifying the activation function from $\sin(\omega_0 \alpha(\cdot))$ to $\sin(\omega_0 \alpha(\cdot)) \pm 1$.

**Results and Analysis.** The results are presented in Tab. 3, where each block reports reconstruction metrics (PSNR / SSIM / LPIPS). Our experiments demonstrate that spatial aliasing occurs exclusively in bias-free networks that maintain symmetry in both coordinate space and activation functions, providing strong evidence that this source of spatial aliasing is unique to INR architectures. While removing symmetry from either component eliminates spatial aliasing, it significantly compromises reconstruction quality, yielding results substantially inferior to conventional practice. Therefore, employing bias terms solely in the input layer can effectively eliminate spatial aliasing while maintaining the symmetrical properties of INR, thus achieving optimal reconstruction quality.

Table 3: Ablation study for *spatial aliasing*

| Settings | standard | w/o coord. sym. | w/o act. sym. ($+1$) | w/o act. sym. ($-1$) |
|---|---|---|---|---|
| Full-Bias | 36.19 / 0.962 / 0.023 | 33.09 / 0.929 / 0.069 | 30.86 / 0.892 / 0.133 | 30.85 / 0.891 / 0.134 |
| Bias-Free | **14.03 / 0.577 / 0.415** | 33.01 / 0.928 / 0.073 | 30.51 / 0.882 / 0.148 | 30.48 / 0.882 / 0.149 |

[4] Due to the unavailability code of $\text{Zoo}_{\text{[ViT]}}$ [29], we cannot measure their exact computational metrics; however, their transformer-based INR classification network inherently requires more computational resources than our MLP-based approach.

# 5 Related Work

## 5.1 Implicit Neural Representations

Implicit Neural Representation (INR) [48, 45, 30] introduces an advanced approach that employs coordinate-based multilayer perceptrons (MLPs) for multimedia data representation and storage. The spectral bias phenomenon [37], wherein high-frequency signal components are more challenging to fit during the INR encoding process, has prompted numerous enhancement strategies. These improvements span various aspects, including novel activation function designs [45, 41, 39, 27], accelerated sampling techniques [59, 22, 57, 61], meta-learning frameworks [47, 7, 12], data transformation methods [43, 60], and alternative approaches [42, 44, 55, 32]. These advancements have enhanced the capacity of INRs to exploit their inherent memory efficiency and natural suitability for inverse problems, enabling various downstream applications including image enhancement [26, 9, 6], view synthesis [30, 4, 32, 54, 53, 56], PDE solving [38, 8, 20], data compression [13, 46, 17]. Despite extensive research in this field, the role of bias terms in INRs remains unexplored. Our work bridges this gap through comprehensive analysis and leverages these findings to enhance INRs' expressive capability and downstream performance.

## 5.2 Neural Networks for Neural Representations

Unlike discrete representations such as pixels, neural representations contain less interpretable parameters, making them challenging for designing network for processing them. Existing methods in this domain fall into two distinct categories: **(1) Symmetry-equivariant architectures** [33, 62, 23, 19]. This line of research considers that neural parameters exhibit inherent symmetries, such as permutation symmetries, where neurons can be arbitrarily permuted without affecting their behavior. The goal is to design architectures that are equivariant to these symmetries. NFN [62] proposes Equivariant NF-Layers, which separately address the assumed symmetries of hidden neurons (HNP) and all neurons (NP), demonstrating effectiveness in downstream tasks. NG-GNN [23] represents the input of INRs as a neural graph structure and leverages well-established Graph Neural Networks (GNNs) for processing. ScaleGMN [19] extends the scope of such meta-architectures from permutation symmetries to scaling symmetries and introduces a corresponding equivariant message-passing framework, achieving superior performance against other meta-network. **(2) End-to-end methods** [15]. Recently, MWT [15] was introduced to address this problem without explicitly considering the equivariant symmetries of INRs. Instead, it integrates the encoding process of INRs directly into the task-specific network's loop in an end-to-end manner, achieving state-of-the-art performance on the INR classification task. **Differences:** Unlike these approaches, Feat-Bias infuses established features into input-layer bias space. While offering limited insights into permutation-symmetric processing, it achieves significant performance improvements through a lightweight approach, leveraging novel findings about bias terms in INRs.

# 6 Conclusion and Limitations

This paper reveals the previously overlooked role of bias terms in INRs through analytical and empirical studies. Our findings show that, unlike in conventional MLPs, bias terms in INRs contribute minimally to nonlinear representation, with their gradient propagation having negligible impact on performance. Through comprehensive analysis, we demonstrate that bias terms primarily mitigate spatial aliasing caused by coordinate and activation symmetries, with input layer bias terms providing the most significant benefits. Based on these insights, we propose applying bias terms exclusively to input layers, challenging the conventional full-bias architecture and demonstrating consistent performance improvements. Moerover, we introduce Featuring INRs with Bias Terms (Feat-Bias), which initializes input-layer bias with pre-trained encoder features, bypassing the challenging process of parsing intractable neural parameters. This approach achieves substantial performance gains while reducing computational costs compared to existing INR post-processing methods.

The main limitation of our work lies in the lack of rigorous mathematical proof to theoretically explain the surprising behavior of bias terms in INRs, despite our intuitive analysis and empirical evidence. Future work will focus on developing a formal theoretical framework for these findings.

## Acknowledgments and Disclosure of Funding

This work is supported in part by National Key Research and Development Project of China (Grant No. 2023YFF0905502), National Natural Science Foundation of China(Grant No. 92467204 and 62472249), and Shenzhen Science and Technology Program (Grant No. JCYJ20220818101014030 and KJZD20240903102300001).

## References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.

[2] Hmrishav Bandyopadhyay, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Tao Xiang, Timothy M. Hospedales, and Yi-Zhe Song. Sketchinr: A first look into sketches as implicit neural representations. *CoRR*, abs/2403.09344, 2024.

[3] Nuri Benbarka, Timon Höfer, Hamd ul Moqeet Riaz, and Andreas Zell. Seeing implicit neural representations as fourier series. In *WACV*, pages 2283–2292. IEEE, 2022.

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022.

[5] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. In *NeurIPS*, pages 21557–21568, 2021.

[6] Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *CVPR*, pages 25627–25636. IEEE, 2024.

[7] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *ECCV (17)*, volume 13677 of *Lecture Notes in Computer Science*, pages 170–187. Springer, 2022.

[8] Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. In *NeurIPS*, 2023.

[9] Tomás Chobola, Yu Liu, Hanyi Zhang, Julia A. Schnabel, and Tingying Peng. Fast context-based low-light image enhancement via neural implicit representations. In *ECCV (86)*, volume 15144 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 2024.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

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

[12] Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 5694–5725. PMLR, 2022.

[13] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Golinski, Yee Whye Teh, and Arnaud Doucet. COIN++: neural compression across modalities. *Trans. Mach. Learn. Res.*, 2022, 2022.

[14] E.Kodak. Kodak dataset, 1999.

[15] Alexander Gielisse and Jan van Gemert. End-to-end implicit neural representations for classification. *CVPR*, 2025.

[16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[17] Jiajun He, Gergely Flamich, Zongyu Guo, and José Miguel Hernández-Lobato. RECOMBINER: robust and enhanced compression with bayesian implicit neural representations. In *ICLR*. OpenReview.net, 2024.

[18] Langwen Huang and Torsten Hoefler. Compressing multidimensional weather and climate data into neural networks. In *ICLR*. OpenReview.net, 2023.

[19] Ioannis Kalogeropoulos, Giorgos Bouritsas, and Yannis Panagakis. Scale equivariant graph metanetworks. In *NeurIPS*, 2024.

[20] Namgyu Kang, Jaemin Oh, Youngjoon Hong, and Eunbyung Park. PIG: physics-informed gaussians as adaptive parametric mesh representations. *CoRR*, abs/2412.05994, 2024.

[21] Amirhossein Kazerouni, Reza Azad, Alireza Hosseini, Dorit Merhof, and Ulas Bagci. INCODE: implicit neural conditioning with prior knowledge embeddings. In *WACV*, pages 1287–1296. IEEE, 2024.

[22] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Accelerating neural field training via soft mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20071–20080, 2024.

[23] Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J. Burghouts, Efstratios Gavves, Cees G. M. Snoek, and David W. Zhang. Graph neural networks for learning equivariant representations of neural networks. In *ICLR*. OpenReview.net, 2024.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[25] Zhemin Li, Hongxia Wang, and Deyu Meng. Regularize implicit neural representation by itself. In *CVPR*, pages 10280–10288. IEEE, 2023.

[26] Zhiheng Li, Muheng Li, Jixuan Fan, Lei Chen, Yansong Tang, Jiwen Lu, and Jie Zhou. Learning dual-level deformable implicit representation for real-world scale arbitrary super-resolution. In *ECCV (69)*, volume 15127 of *Lecture Notes in Computer Science*, pages 352–368. Springer, 2024.

[27] 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. *CoRR*, abs/2312.02434, 2023.

[28] Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Deep learning on implicit neural representations of shapes. In *ICLR*. OpenReview.net, 2023.

[29] Qi Ma, Danda Pani Paudel, Ender Konukoglu, and Luc Van Gool. Implicit zoo: A large-scale dataset of neural implicit functions for 2d images and 3d scenes. In *NeurIPS*, 2024.

[30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020.

[31] Amirali Molaei, Amirhossein Aminimehr, Armin Tavakoli, Amirhossein Kazerouni, Bobby Azad, Reza Azad, and Dorit Merhof. Implicit neural representation in medical imaging: A comparative survey. In *ICCV (Workshops)*, pages 2373–2383. IEEE, 2023.

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022.

[33] Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 25790–25816. PMLR, 2023.

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.

[35] Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174. Computer Vision Foundation / IEEE, 2019.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[37] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron C. Courville. On the spectral bias of neural networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 2019.

[38] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[39] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *ECCV (33)*, volume 13693 of *Lecture Notes in Computer Science*, pages 142–158. Springer, 2022.

[40] Simone Saitta, Marcello Carioni, Subhadip Mukherjee, Carola-Bibiane Schönlieb, and Alberto Redaelli. Implicit neural representations for unsupervised super-resolution and denoising of 4d flow MRI. *Comput. Methods Programs Biomed.*, 246:108057, 2024.

[41] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. WIRE: wavelet implicit neural representations. In *CVPR*, pages 18507–18516. IEEE, 2023.

[42] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. MINER: multiscale implicit neural representation. In *ECCV (23)*, volume 13683 of *Lecture Notes in Computer Science*, pages 318–333. Springer, 2022.

[43] Junwon Seo, Sangyoon Lee, Kwang In Kim, and Jaeho Lee. In search of a data transformation that accelerates neural field training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[44] Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicity neural representation with fourier bases reparameterized training. *CoRR*, abs/2401.07402, 2024.

[45] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

[46] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *ECCV (26)*, volume 13686 of *Lecture Notes in Computer Science*, pages 74–91. Springer, 2022.

[47] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *CVPR*, pages 2846–2855. Computer Vision Foundation / IEEE, 2021.

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

[49] Shengjie Wang, Tianyi Zhou, and Jeff A. Bilmes. Bias also matters: Bias attribution for deep neural network explanation. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 6659–6667. PMLR, 2019.

[50] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[51] David Wiesner, Julian Suk, Sven Dummer, David Svoboda, and Jelmer M. Wolterink. Implicit neural representations for generative modeling of living cell shapes. In *MICCAI (4)*, volume 13434 of *Lecture Notes in Computer Science*, pages 58–67. Springer, 2022.

[52] Jelmer M. Wolterink, Jesse C. Zwienenberg, and Christoph Brune. Implicit neural representations for deformable image registration. In *MIDL*, volume 172 of *Proceedings of Machine Learning Research*, pages 1349–1359. PMLR, 2022.

[53] Shuzhao Xie, Jiahang Liu, Weixiang Zhang, Shijia Ge, Sicheng Pan, Chen Tang, Yunpeng Bai, Cong Zhang, Xiaoyi Fan, and Zhi Wang. Sizegs: Size-aware compression of 3d gaussian splatting via mixed integer programming. In *ACM MM*, 2025.

[54] Shuzhao Xie, Weixiang Zhang, Chen Tang, Yunpeng Bai, Rongwei Lu, Shijia Ge, and Zhi Wang. Mesongs: Post-training compression of 3d gaussians via efficient attribute transformation. In *European Conference on Computer Vision*. Springer, 2024.

[55] Runzhao Yang. TINC: tree-structured implicit neural compression. In *CVPR*, pages 18517–18526. IEEE, 2023.

[56] Wei Yao, Shuzhao Xie, Letian Li, Weixiang Zhang, Zhixin Lai, Shiqi Dai, Ke Zhang, and Zhi Wang. Sd-gs: Structured deformable 3d gaussians for efficient dynamic scene reconstruction, 2025.

[57] Chen Zhang, Steven Tin Sui Luo, Jason Chun Lok Li, Yik-Chung Wu, and Ngai Wong. Non-parametric teaching of implicit neural representations. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[58] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018.

[59] Weixiang Zhang, Shuzhao Xie, Shijia Ge, Wei Yao, Chen Tang, and Zhi Wang. Expansive supervision for neural radiance field. *arXiv preprint arXiv:2409.08056*, 2024.

[60] Weixiang Zhang, Shuzhao Xie, Chengwei Ren, Shijia Ge, Mingzi Wang, and Zhi Wang. Enhancing implicit neural representations via symmetric power transformation. In *AAAI*, pages 10157–10165. AAAI Press, 2025.

[61] Weixiang Zhang, Shuzhao Xie, Chengwei Ren, Siyi Xie, Chen Tang, Shijia Ge, Mingzi Wang, and Zhi Wang. Evos: Efficient implicit neural training via evolutionary selector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[62] Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J. Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. In *NeurIPS*, 2023.

# A Additional Verification for Empirical Studies

In this section, we provide additional verification of our empirical studies (Sec.2.2) and INR representation experiments (Sec.4.1). We conduct experiments on the Kodak dataset [14], maintaining all configurations identical to those described in Sec. 4.1. The quantitative results are presented in Tab. 4. The configurations Standard, Bias Free, $H_1$Bias, Output Bias, and Input Bias align with those in Tab. 1. Additionally, we include $H_2$ Bias and $H_3$ Bias, representing networks with bias terms exclusively in the second and third layers, respectively. Settings without underlines represent standard SIREN implementations, while underlined settings correspond to FINER. The symbol ⊘ indicates reconstructions exhibiting *spatial aliasing*. For each configuration block, we report averaged metrics both with and without bias term backpropagation. Complementary results on the DIV2K dataset are presented in Tab. 5, supplementing the findings in Tab. 1. The results in Tab. 5 and Tab. 4 further validate our observations from Sec. 2.2 and conclusions from Sec. 4.1. These findings substantiate both the magnitude of bias terms' impact in INRs and the effects of their gradient optimization, lending additional support to Proposition 1. This reinforces our claim that the primary role of bias terms in neural representations is to eliminate *spatial aliasing* arising from the symmetry of coordinate space and activation functions.

Table 4: Results of 2D image fitting tasks with varying bias configurations (Kodak dataset)

| Settings | 1k Iterations PSNR ↑ | 1k Iterations SSIM↑ | 3k Iterations PSNR↑ | 3k Iterations SSIM↑ | 5k Iterations PSNR↑ | 5k Iterations SSIM↑ |
|---|---|---|---|---|---|---|
| Standard | 30.05 / 30.05 | 0.832 / 0.832 | 33.93 / 33.92 | 0.912 / 0.912 | 35.24 / 35.23 | 0.929 / 0.929 |
| Bias Free⊘ | 15.06 / 15.06 | 0.599 / 0.599 | 15.09 / 15.09 | 0.620 / 0.620 | 15.10 / 15.10 | 0.626 / 0.626 |
| $H_1$ Bias | 29.67 / 29.68 | 0.822 / 0.822 | 33.25 / 33.25 | 0.902 / 0.902 | 34.44 / 34.45 | 0.919 / 0.919 |
| $H_2$ Bias | 29.96 / 29.96 | 0.831 / 0.831 | 33.33 / 33.32 | 0.903 / 0.903 | 34.41 / 34.42 | 0.918 / 0.918 |
| $H_3$ Bias | 30.07 / 30.05 | 0.829 / 0.829 | 33.17 / 33.16 | 0.895 / 0.894 | 34.18 / 33.17 | 0.910 / 0.910 |
| Output Bias⊘ | 16.34 / 16.04 | 0.613 / 0.610 | 17.03 / 16.78 | 0.641 / 0.639 | 17.31 / 17.02 | 0.648 / 0.646 |
| Input Bias | 30.78 / 30.78 | 0.854 / 0.854 | 34.31 / 34.30 | 0.917 / 0.917 | 35.42 / 35.42 | 0.930 / 0.930 |
| Standard | 32.05 / 32.06 | 0.880 / 0.881 | 36.18 / 36.18 | 0.940 / 0.940 | 37.52 / 37.53 | 0.952 / 0.952 |
| Bias Free⊘ | 15.09 / 15.09 | 0.617 / 0.617 | 15.11 / 15.11 | 0.632 / 0.632 | 15.11 / 15.11 | 0.636 / 0.636 |
| $H_1$ Bias | 30.83 / 30.82 | 0.853 / 0.853 | 34.46 / 34.45 | 0.921 / 0.921 | 35.82 / 35.81 | 0.935 / 0.935 |
| $H_2$ Bias | 31.28 / 31.28 | 0.866 / 0.866 | 34.74 / 34.75 | 0.924 / 0.924 | 35.66/ 35.67 | 0.934 / 0.935 |
| $H_3$ Bias | 31.37 / 31.36 | 0.865 / 0.864 | 34.60 / 34.60 | 0.919 / 0.919 | 35.60 / 35.59 | 0.931 / 0.931 |
| Output Bias⊘ | 16.19 / 16.07 | 0.628 / 0.627 | 16.82 / 16.79 | 0.652 / 0.651 | 17.10 / 17.04 | 0.655 / 0.657 |
| Input Bias | 33.06 / 33.05 | 0.901 / 0.900 | 36.81 / 36.80 | 0.948 / 0.950 | 37.92 / 37.91 | 0.955 / 0.955 |

Table 5: Results of 2D image fitting tasks with varying bias configurations (DIV2K dataset)

| Settings | 1k Iterations PSNR ↑ | 1k Iterations SSIM↑ | 3k Iterations PSNR↑ | 3k Iterations SSIM↑ | 5k Iterations PSNR↑ | 5k Iterations SSIM↑ |
|---|---|---|---|---|---|---|
| Standard | 30.30 / 30.30 | 0.882 / 0.882 | 34.68 / 34.69 | 0.946 / 0.946 | 36.17 / 36.18 | 0.956 / 0.959 |
| Bias Free⊘ | 13.98 / 13.98 | 0.556 / 0.556 | 14.00 / 14.00 | 0.574 / 0.574 | 14.03 / 14.03 | 0.577 / 0.577 |
| $H_1$ Bias | 29.88 / 29.87 | 0.872 / 0.872 | 34.02 / 34.00 | 0.938 / 0.938 | 35.33 / 35.34 | 0.950 / 0.950 |
| $H_2$ Bias | 30.14 / 30.13 | 0.875 / 0.875 | 33.98 / 33.96 | 0.935 / 0.935 | 35.20 / 35.19 | 0.946 / 0.946 |
| $H_3$ Bias | 30.18 / 30.14 | 0.871 / 0.869 | 33.65 / 33.63 | 0.927 / 0.926 | 34.77 / 34.76 | 0.934 / 0.933 |
| Output Bias⊘ | 14.67 / 14.49 | 0.567 / 0.562 | 15.11 / 14.96 | 0.593 / 0.583 | 15.11 / 15.13 | 0.586 / 0.588 |
| Input Bias | 31.20 / 31.18 | 0.903 / 0.902 | 35.33 / 35.32 | 0.953 / 0.953 | 36.61 / 36.63 | 0.962 / 0.962 |
| Standard | 32.26 / 32.25 | 0.916 / 0.916 | 36.84 / 36.83 | 0.962 / 0.962 | 38.47 / 38.47 | 0.971 / 0.971 |
| Bias Free⊘ | 13.99 / 13.99 | 0.568 / 0.568 | 14.01 / 14.01 | 0.581 / 0.581 | 14.03 / 14.03 | 0.583 / 0.583 |
| $H_1$ Bias | 31.06 / 31.08 | 0.894 / 0.894 | 35.21 / 35.23 | 0.950 / 0.949 | 36.63 / 36.62 | 0.958 / 0.958 |
| $H_2$ Bias | 31.49 / 31.49 | 0.901 / 0.901 | 35.28 / 35.24 | 0.948 / 0.947 | 36.62 / 36.60 | 0.957 / 0.957 |
| $H_3$ Bias | 31.66 / 31.60 | 0.901 / 0.899 | 35.21 / 35.18 | 0.946 / 0.944 | 36.32 / 36.30 | 0.952 / 0.951 |
| Output Bias⊘ | 14.55 / 14.51 | 0.577 / 0.573 | 15.13 / 14.98 | 0.599 / 0.590 | 15.22 / 15.14 | 0.595 / 0.594 |
| Input Bias | 33.39 / 33.38 | 0.933 / 0.932 | 37.95 / 37.90 | 0.968 / 0.968 | 39.30 / 39.29 | 0.974 / 0.974 |

## B Additional Experiments for Feat-Bias

In this section, we present additional experiments to validate the effectiveness of Feat-Bias (Sec.3). Following prior work[62, 19, 15], we evaluate our method on INR classification tasks using MNIST and F-MNIST datasets. In addition to the baselines discussed in Sec.D, we include comparisons with recent methods including Inr2Vec[28], DWS [33], and NG-GNN [23]. Feat-Bias$_{[\text{ViT}]}$ and Feat-Bias$_{[\text{DINOV2}]}$ represent our method using features extracted from ViT [10] and DINOv2 [34], respectively. We maintain identical configurations as described in Sec. D, employing a $3 \times 256$ MLP downstream network, training for 1000 iterations with a cosine scheduler and learning rate of $1\text{e}{-3}$. The quantitative results are presented in Tab. 6. Our Feat-Bias$_{[\text{DINOV2}]}$ achieves the highest classification accuracy on both datasets, surpassing the recent state-of-the-art MWT-L. Notably, on F-MNIST, our method improves accuracy from 87.32% to 92.38%, demonstrating significant advancement in INR downstream tasks.

Table 6: INR classification on MNIST and Fashion-MNIST Datasets

| Method | MNIST | F-MNIST | Method | MNIST | F-MNIST |
|---|---|---|---|---|---|
| MLP | 17.55±0.01 | 19.91±0.47 | Inr2Vec [28] | 23.69±0.10 | 22.33±0.41 |
| $\text{NFN}_{\text{NP}}$ [62] | 78.50±0.23 | 68.19±0.28 | $\text{NFN}_{\text{HNP}}$ [62] | 79.11±0.84 | 68.94±0.64 |
| DWS [33] | 85.71±0.57 | 67.06±0.29 | NG-GNN [23] | 91.40±0.60 | 68.00±0.20 |
| ScaleGMN [19] | 96.57±0.10 | 80.46±0.32 | ScaleGMN-B [19] | 96.59±0.24 | 80.78±0.16 |
| WT [15] | 93.08±2.26 | 73.81±1.43 | $\text{MWT}_{\text{Mid-Task}}$ [15] | 95.57±0.30 | 77.23±0.56 |
| MWT [15] | 96.58±0.32 | 83.86±0.91 | MWT-L [15] | 98.33±0.13 | 87.32±0.16 |
| Feat-Bias$_{[\text{ViT}]}$ | 95.79±0.04 | 90.13±0.49 | Feat-Bias$_{[\text{DINOV2}]}$ | 98.48±0.05 | 92.38±0.06 |

## C Additional Experiments for Ablation Studies

We present additional ablation studies to further verify the source of *spatial aliasing*. All experimental settings strictly align with those in Sec. 4.3. Tab 7 present ablation result for spatial aliasing on Kodak dataset, demonstrating consistency with the findings presented in Tab. 3.

Table 7: Ablation study for *spatial aliasing* (Kodak Dataset)

| Settings | standard | w/o coord. sym. | w/o act. sym. $(+1)$ | w/o act. sym. $(-1)$ |
|---|---|---|---|---|
| Full-Bias | 35.23 / 0.929 / 0.089 | 32.23 / 0.880 / 0.191 | 30.38 / 0.838 / 0.257 | 30.44 / 0.840 / 0.254 |
| Bias-Free | **15.10 / 0.626 / 0.461** | 32.17 / 0.877 / 0.200 | 30.11 / 0.832 / 0.258 | 30.11 / 0.832 / 0.259 |

# D Additional Experiments for classification task

**Experimental Settings.** In this section, we further evaluate the effectiveness of our proposed Feat-Bias (Sec. 3) on INR downstream tasks. The downstream network of Feat-Bias consists of a lightweight $3 \times 256$ MLP classifier. Since the CIFAR-100 dataset is more complex than CIFAR-10, the number of iterations is changed to 5000, using a cosine scheduler with a learning rate of $1\mathrm{e}{-3}$. In addition, the iteration count of MWT has also been modified to 20 epochs. In tab 8, we list the performance metrics at 10 epochs (i.e., the original default setting) and 20 epochs. All experiments are repeated three times, reporting both mean and standard deviation.

**Quantitative Results.** Tab. 8 shows the classification results for CIFAR-100 dataset [24], where we also adopt Implicit-Zoo's default configuration with a $1 \times 64$ SIREN network like the settings in the main text to ensure consistent experimental conditions across all comparisons. Since Implicitly Zoo does not provide the INR dataset of CIFAR-100, only MWT is compared with our method here. As evidenced by the results, our method achieves superior performance across accuracy, precision, and F1 score metrics meanwhile greatly reduces the training time,further demonstrating Feat-Bias's effectiveness in INR post-processing tasks.

Table 8: INR classification on CIFAR-100 Datasets

| Method | ***Classification Task*** | | | | | ***INRs*** | |
|---|---|---|---|---|---|---|---|
| | Accuracy (%) ↑ | Precision (%) ↑ | F1 ↑ | Time ↓ | Params (kilo #) ↓ | PSNR (dB) ↑ | SSIM ↑ |
| $\text{WT}_{\text{10epochs}}$ [15] | $12.21_{\pm 0.89}$ | $10.90_{\pm 0.62}$ | $10.09_{\pm 0.87}$ | $73.52_{\pm 1.78}$ (min) | 261 | 32.33 | 0.942 |
| $\text{WT}_{\text{20epochs}}$ [15] | $15.83_{\pm 0.71}$ | $13.82_{\pm 0.85}$ | $13.84_{\pm 0.79}$ | $151.78_{\pm 2.93}$ (min) | 261 | 34.89 | 0.964 |
| $\text{MWT}_{\text{Mid-Task-10epochs}}{}^{\ddagger}$ [15] | $15.81_{\pm 0.44}$ | $14.94_{\pm 0.86}$ | $13.86_{\pm 0.74}$ | $83.41_{\pm 1.55}$ (min) | 261 | 30.17 | 0.910 |
| $\text{MWT}_{\text{Mid-Task-20epochs}}{}^{\ddagger}$ [15] | $19.87_{\pm 0.21}$ | $17.88_{\pm 0.11}$ | $18.06_{\pm 0.15}$ | $169.42_{\pm 2.80}$ (min) | 261 | 33.06 | 0.948 |
| $\text{MWT}_{\text{10epochs}}$ [15] | $19.46_{\pm 0.88}$ | $18.09_{\pm 0.58}$ | $17.67_{\pm 0.69}$ | $83.74_{\pm 0.97}$ (min) | 261 | 24.45 | 0.752 |
| $\text{MWT}_{\text{20epochs}}$ [15] | $23.29_{\pm 0.57}$ | $21.56_{\pm 0.67}$ | $21.78_{\pm 0.65}$ | $168.0_{\pm 1.27}$ (min) | 261 | 25.80 | 0.806 |
| $\text{Feat-Bias}_{\text{[ViT]}}$ | $74.83_{\pm 0.07}$ | $75.03_{\pm 0.09}$ | $74.79_{\pm 0.07}$ | $144.70_{\pm 1.83}$ (sec) | 151 | / | / |
| $\text{Feat-Bias}_{\text{[DINOV2]}}$ | $78.51_{\pm 0.16}$ | $78.71_{\pm 0.13}$ | $78.46_{\pm 0.15}$ | $145.38_{\pm 1.65}$ (sec) | 151 | / | / |

‡ denotes $\omega_{task}$ reported in MWT [15].

## E Additional visualizations for *spatial aliasing*

Figure 5: Visualization for *spatial aliasing*: reconstructed signals demonstrate central symmetry, manifesting as aliasing artifacts between distinct regions of the input. Green and Pink denote the ground truth and reconstructed signals with *spatial aliasing*, respectively.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction have effectively reflected our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed our limitations in the final part of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have established sufficient assumptions for all theoretical result in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provide all details to reproduce the main experimental results in the section of Experimental Settings. We will also release the codes and checkpoints.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide open access to the data and code if the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have specified all the training and test details in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have reported error bar in the experiment sections.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer:[Yes]

   Justification: We have provided sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conformed with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work performed demonstrates no significant societal impact. Currently, there are no evident examples suggesting that INR might lead to potential negative societal consequences.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.