Beyond Periodicity: Towards a Unifying Framework for Activations in Coordinate-MLPs

Sameera Ramasinghe and Simon Lucey

Australian Institute for Machine Learning
University of Adelaide
{sameera.ramasinghe,simon.lucey}@adelaide.edu.au

Abstract. Coordinate-MLPs are emerging as an effective tool for modeling multidimensional continuous signals, overcoming many drawbacks associated with discrete grid-based approximations. However, coordinate-MLPs with ReLU activations, in their rudimentary form, demonstrate poor performance in representing signals with high fidelity, promoting the need for positional embedding layers. Recently, Sitzmann *et al.* [24] proposed a sinusoidal activation function that has the capacity to omit positional embedding from coordinate-MLPs while still preserving high signal fidelity. Despite its potential, ReLUs are still dominating the space of coordinate-MLPs; we speculate that this is due to the hyper-sensitivity of networks – that employ such sinusoidal activations – to the initialization schemes. In this paper, we attempt to broaden the current understanding of the effect of activations in coordinate-MLPs, and show that there exists a broader class of activations that are suitable for encoding signals. We affirm that sinusoidal activations are only a single example in this class, and propose several **non-periodic** functions that empirically demonstrate more robust performance against random initializations than sinusoids. Finally, we advocate for a shift towards coordinate-MLPs that employ these non-traditional activation functions due to their high performance and simplicity. [1]

Keywords: Coordinate-networks, implicit neural representations

1 Introduction

Despite the ubiquitous and successful usage of conventional discrete representations in machine learning (*e.g.* images, 3D meshes, and 3D point clouds etc.), coordinate MLPs are now emerging as a unique instrument that can represent multi-dimensional signals as continuously differentiable entities. Coordinate-MLPs – also known as *implicit neural representations* [24] – are fully connected networks that encode continuous signals as weights, consuming low-dimensional coordinates as inputs. Such continuous representations are powerful compared to their discrete grid-based counterparts, as they can be queried up to extremely high resolutions. Furthermore, whereas the memory consumption of grid-based

[1] Code available at https://github.com/samgregoost/Beyond_periodicity

representations entails exponential growth rates against the dimension and the resolution of data, neural representations have displayed a much more compact relationship between the above factors. Consequently, this recent trend has influenced a proliferation of studies in vision-related research including texture generation [7, 14, 7, 30], shape representation [4, 5, 28, 6, 1, 12, 15], and novel view synthesis [11, 13, 21, 25, 31, 17, 17, 19, 10, 29, 16].

Notwithstanding the virtues mentioned above, coordinate MLPs, in their fundamental form, exhibit poor performance in encoding signals with high-frequency components when equipped with common activation functions such as ReLUs. An elemental reason behind this has shown to be the *spectral bias* of MLPs [2, 18]. That is, the corresponding neural tangent kernel (NTK) of MLPs are prone to high-frequency fall-offs, hampering their ability to learn high-frequency functions. The prevalent work-around to this problem involves applying a *positional embedding layer* prior to the MLP, where the low-dimensional inputs are projected to a higher-dimensional space using Fourier features [27].

In contrast, Sitzmann *et al.* [24] recently portrayed that MLPs with sinusoidal activation functions are naturally suited for encoding high-frequency signals, eliminating the need for positional embedding layers. Despite its potential, much of the research that involve coordinate-MLPs still prefer positional embeddings over sinusoidal activations. We postulate that this could be for two reasons. First, Sitzmann *et al.* mainly attributed the success of sinusoidal activations to their periodicity, although the evidence for this relationship still remains scant. Consequently, this lack of understanding obfuscates some of the fundamental principles behind its effectiveness and hampers faithful usage in a wider range of applications. Second, sinusoidal activations are highly sensitive to the initialization of the MLP, showcasing significant performance drops in cases where the MLP is initialized without strictly adhering to the guidelines of Sitzmann *et al.* The above drawbacks have heightened the need for a more rigorous analysis that facilitates more effective usage of activation functions in coordinate-MLPs.

Contributions: We offer a broader theoretical understanding of the role of activation functions within coordinate-MLPs. In particular, we show that the efficacy of a coordinate-MLP is critically bound to its Lipschitz smoothness and the singular value distribution of the hidden-layer representations, and the optimal values of these metrics depend on the characteristics of the signal that needs to be encoded. We further show that the above properties are inherently linked to each other, and by controlling one property, the other can be implicitly manipulated. We further derive formulae to connect the Lipschitz smoothness and the singular value distribution to the properties of the activation functions. The significance of this finding is two-fold: (i) providing guidelines for tuning the hyper-parameters of an activation function based on the given signal and, (ii) enabling a practitioner to theoretically predict the effect of a given activation function when used in a coordinate-MLP, prior to practical implementation. We further show that sinusoidal activations are simply a single example that fulfills such constraints, and the periodicity is not a crucial factor that determine the efficacy of an activation function. Consequently, we propose a much

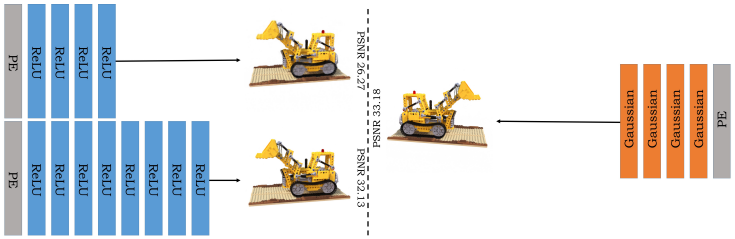

Fig. 1. ReLU vs Gaussian activations (ours). Gaussian activations achieve better results with $\sim$ 50% less parameters. These non-periodic activations also allow embedding-free architectures (see Fig.3), and are robust to different random initializations of coordinate-MLPs than the sinusoid activations advocated in SIREN [24].

broader class of *non-periodic* activation functions that can be used in encoding functions/signals with high fidelity, and show that their empirical properties match with theoretical predictions. We further illustrate that the newly proposed activation functions are robust to different initialization schemes, unlike sinusoidal activations. Further, picking one such proposed activation – Gaussian – as an example, we demonstrate that coordinate-MLPs with such activation functions enjoy better results, faster convergence rates, and shallower architectures in comparison to ReLU-MLPs. Finally, we show that these activations allow positional-embedding-free architectures to be used in complex tasks such as 3D view synthesis. To our knowledge, this is the first instance coordinate-MLPs have successfully been employed in such experiments in the absence of positional embeddings.

2 Related works

In recent years, there has been an increasing interest in parameterizing signals using neural networks – commonly referred to as coordinate-MLPs [26] or implicit neural functions [24] – largely owing to the seminal work by Mildenhall *et al.* [11]. The usage of coordinate-MLPs are somewhat different from conventional MLPs: i) conventional MLPs typically operate on high dimensional inputs such as images, sounds, or 3D shapes, and ii) are primarily being used for classification purposes where the decision boundaries do not have to preserve smoothness. In contrast, coordinate-MLPs are used to encode the signals as weights where the inputs are low-dimensional coordinates and the outputs have to preserve smoothness [32]. One of the most remarkable aspects of Mildenhall *et al.*'s work includes demonstrating the generalization properties of such neural signal representations, *i.e.* once trained with a handful of view points, the coordinate-MLP can reconstruct the photometric view projection from an arbitrary angle with fine-details. This ground-breaking demonstration caused a ripple of studies that include neural signal representations as the core entities across many applications including shape representation [4, 5, 28, 6, 1, 12, 15], and novel view synthesis [11, 13, 21, 25, 31, 17, 17, 19, 10, 29, 16]. However, for optimal

performance, these coordinate-MLPs have to use positional embeddings, which allow them to encode high-frequency signal content. In contrast, Sitzmann *et al.* [24] proposed sinusoid activations that enabled coordinate-MLPs to encode signals with higher quality without positional embeddings. But, sinusoid activations have been shown to be extremely sensitive to the initialization scheme of the MLPs. A further limitation to the framework developed by Sitzmann *et al.* is its confinement to periodic activations. In contrast, our work generalizes the current understanding on the effect of activations in coordinate-MLPs and thereby propose a class of non-periodic activations that is robust under random initializations. Recently, Liang *et al.* [9] proposed a novel class of activation functions that can approximate target functions with a smaller number of parameters. However, our framework differs from theirs in two important aspects: *1)* They mix multiple activation types to expand the class of functions that can be approximated. However, NNs with *any* non-polynomial activation – thus all activations we propose – are universal approximators [8]. Thus, our setup is simpler while being more expressive. *2)* Mixing activations leads to poor controllability of memorization/generalization tradeoff, which depends on the problem domain. For instance, it is unclear how to control the coefficients of polynomials (and also other functions, when mixed together) to this end. In contrast, our framework provides a much more clear interpretation of this tradeoff, and shows compelling generalization properties in complex settings as NeRF.

3 Methodology

Notation. The set of real $n-$dimensional vectors are denoted by $\mathbb{R}^n$. The vectors are denoted by bold lower-case letters (*e.g.,* $\mathbf{x}$). The set of $m \times n$ dimensional matrices are denoted by $\mathbb{R}^{m\times n}$, and the matrices are denoted by bold upper-case letters (*e.g.,* $\mathbf{A}$). $\|\cdot\|$ denotes vector norm, $\|\cdot\|_F$ denotes the Frobenius norm, and $\|\cdot\|_o$ is the operator norm. $\mathbb{B}_r^n$ represents the $n-$ dimensional ball with radius r. Further, $g(f(\mathbf{x})) = g \circ f(x)$ where $\circ$ is the compositional operator.

3.1 Rank and memorization

The efficacy of a coordinate-MLP largely depends on its ability to memorize training data. The objective of this section is to identify the key factors that affect memorization. To establish the foundation for our analysis, we first revisit the formulation of an MLP.

An MLP f with $k-1$ non linear hidden-layers can be described by,

$$f : \mathbf{x} \to g^k \circ \psi^{k-1} \circ g^{k-1} \circ \cdots \circ \psi^1 \circ g^1(\mathbf{x}), \tag{1}$$

where $g^i : \mathbf{x} \to \mathbf{A}^i \cdot \mathbf{x} + \mathbf{b}^i$ is an affine projection with trainable weights $\mathbf{A}^i \in \mathbb{R}^{\dim(\mathbf{x}^i)\times\dim(\mathbf{x}^{i-1})}$, $\mathbf{b}^i \in \mathbb{R}^{\dim(\mathbf{x}_i)}$ is the bias, and ψ^i is a non-linear activation function. The final layer is a linear transform such that $f : \mathbf{x} \to g^k \circ \phi(\mathbf{x})$, and ϕ is a composition of the preceding $k-1$ layers within the MLP without the

final linear layer. If the number of training examples is N, we define the total (training) embedding matrix as

$$\mathbf{X} \in \mathbb{R}^{D \times N} \coloneqq \left[\phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)^T \dots \phi(\mathbf{x}_N)^T\right] \tag{2}$$

where $\{\mathbf{x}_n\}_{n=1}^N$ are the raw training inputs.

Recall that the final layer of an MLP is (typically) an affine projection without any non-linearity. Dropping the bias for simplified notation, we get,

$$\tilde{\mathbf{Y}} = \mathbf{A}^k \mathbf{X}, \tag{3}$$

where $\tilde{\mathbf{Y}} \in \mathbb{R}^{q \times N}$ are the outputs of the MLP. Suppose $\mathbf{Y} \in \mathbb{R}^{q \times N}$ are the ground truth training outputs the MLP is attempting to learn. Observe that if the MLP is perfectly memorizing the training set — if $\tilde{\mathbf{Y}} = \mathbf{Y}$ — then each row of $\mathbf{Y}$ is a linear combination of the rows of $\mathbf{X}$. Assume we have no prior knowledge of $\mathbf{Y}$, that is, the rows of $\mathbf{Y}$ can be *any* arbitrary vector in $\mathbb{R}^N$. If the rows of $\mathbf{X}$ are linearly independent, they form a basis for $\mathbb{R}^N$ (assuming $D \geq N$). Therefore, if $\text{rank}(\mathbf{X}) = N$, it is guaranteed that (assuming perfect convergence) the MLP can find a weight matrix $\mathbf{A}^k$ that ensures perfect reconstruction of $\mathbf{Y}$.

One can raise the valid question: could this conclusion hold in the practical case where $D \ll N$? The answer to this question depends on the nature of the ground truth signal. Note that although the condition $\text{rank}(\mathbf{X}) = N$ is sufficient to ensure perfect memorization for *any* signal, it might not always be necessary since natural signals are typically redundant – that is of limited bandwidth. The bandwidth of a category of signals can be defined [23] as the number of linearly independent (normalized) bases required to represent them. Thus, $\text{rank}(\mathbf{X})$ can be less than N for many categories of signals whilst still enjoying perfect signal recovery by the MLP. Fig. 2 is a perfect example that illustrates the above point. Note that the stable rank is a lower bound for rank [20]. Better reconstructions are shown when the stable rank is high, but the measure is bounded by the network width (D), which is lower than the number of points (N). In contrast, encoding noise signals which have limited to no redundancy – would require a larger network width – and yields poorer results with $D \ll N$ (see Appendix) as predicted. Rigorously speaking, the analysis so far only considers the penultimate layer. However, based on the gathered insights, we make the following general claim: *the potential of the hidden-layers to induce high-rank representations – that is those with very few zero singular values within $\boldsymbol{X}$ – correlates with the memorization capacity of a coordinate-MLP.*

One could also view the above result as a refashioning of the well known Nyquist-Shannon sampling theory [23] applied to-coordinate MLPs. The result is important, however, when it comes to the exposition of the rest of this paper. But, a critical component is overlooked in the above analysis. In many applications that utilize coordinate-MLPs, the ability predict values at unseen coordinates, *i.e.*, generalization, is important. For instance, in novel view synthesis of a 3D scene, the network only observes a handful of views, in which the network then has to predict the views from new angles. Therefore, the immediate question arises: *is having the ability to induce high-rank representations*

(i.e. very few zero singular values within $\boldsymbol{X}$) sufficient for both memorization and generalization? In Section 3.2, we shall see that this is indeed not the case.

3.2 Smoothness and generalization

To show that the rank alone is not sufficient to guarantee good generalization, we perform a simple thought experiment on 1-D input coordinates $x \in \mathbb{R}$ and single channel outputs. Let us construct a very wide layer $\phi : \mathbb{R} \rightarrow \mathbb{R}^D$ such that $D = N$, and define the layer output $\phi(x) = [e^{\frac{-(x-x_1)^2}{\sigma^2}}, \dots, e^{\frac{-(x-x_N)^2}{\sigma^2}}]$, where $x_1, \dots, x_N$ are the training points. With extremely small σ, $\phi(\cdot)$ is equivalent to one-hot encoding, ensuring $\text{rank}(\mathbf{X}) = N$. Then, it is guaranteed that an $\mathbf{A}$ can be found to memorize all the ground truth outputs $y_1, \dots, y_N$. However, all the unseen points will map to $\vec{\mathbf{0}}$, and thus, the network will be obtain extremely poor generalization. In summary, having a higher rank for $\mathbf{X}$ will help in memorization, but, it will not necessarily ensure good generalization.

Moreover, strictly speaking, generalization cannot be measured independently without context. For instance, given sparse training points a neural network can, in theory, learn infinitely many functions while fitting the training points. Therefore, for good generalization, the network has to learn a function within a space restricted by certain priors and intuitions regarding the ground truth signal. The generalization then depends on the extent to which the learned function is close to these prior assumptions about the task. When no such priors are available, one intuitive solution that is widely accepted for regression (at least from an engineering perspective) is to have "smooth" interpolation between the training points [3].

In order to ensure such smooth interpolations (where second order derivatives are bounded) it is critical to preserve the smoothness across non-linear layers $\phi(\cdot)$ *locally* as $\frac{||\phi(\mathbf{x}_1)-\phi(\mathbf{x}_2)||}{||\mathbf{x}_1-\mathbf{x}_2||} = C$, where C is some constant (since the final layer is linear). Although the above condition seems overly restrictive, recall that the embeddings $\phi(\cdot)$ are learned via hidden-layers, as opposed to being analytically designed. Therefore, it is enough to reduce the search space of the parameters accordingly, as opposed to explicitly enforcing the above constraint. Thus, we can slightly relax the above equality to an inequality interms of the local Lipschitz smoothness. More precisely, in practice, it is enough to ensure

$$\|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\| \leq C\|\mathbf{x}_1 - \mathbf{x}_2\|, \tag{4}$$

locally, where C is a non-negative, locally varying constant that depends on the magnitude of the local first-order derivatives (*i.e.*, frequencies) of the encoded signal. That is, at intervals where encoding points exhibit large fluctuations, C needs to be higher, and vise-versa.

Thus far, we have established that the singular values of the $\mathbf{X}$ correlates with the memorization of seen coordinates and the (Lipschitz) smoothness of $\phi(\cdot)$ correlates with the generalization performance of an MLP. Thus, it is intriguing to investigate if there exists a connection between these two forces at a fundamental level, as such an analysis has the potential to provide valuable insights that enable efficient manipulation of these factors.

3.3 Singular value distribution as a proxy for smoothness

This section is devoted to exploring the interrelation between the smoothness and the singular value distribution of the hidden representations. Suppose that for coordinates $\mathbf{x}_i$ in a given small neighborhood, $\phi(\cdot)$ is Lipschitz bounded with a constant C. Then,

$$\frac{\sqrt{(\phi(\mathbf{x}_1)\phi(\mathbf{x}_1)^T - 2\phi(\mathbf{x}_1)\phi(\mathbf{x}_2)^T + \phi(\mathbf{x}_2)\phi(\mathbf{x}_2)^T)}}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \leq C \tag{5}$$

With Eq. 5 in hand, let us consider two cases for $\mathbf{X}$.

Case 1. The columns of $\mathbf{X}$ are orthogonal and the singular values of $\mathbf{X}$ are identically distributed.

One can see that,

$$\frac{\sqrt{\|\phi(\mathbf{x}_i)\|^2 + \|\phi(\mathbf{x}_j)\|^2}}{\|\mathbf{x}_i - \mathbf{x}_j\|} \leq C \Rightarrow \lim_{\|\mathbf{x}_1 - \mathbf{x}_2\| \to 0} C = \infty \tag{6}$$

In other words, having (approximately) equally distributed singular values violates the Lipschitz smoothness of the network.

Case 2. The singular values of $\mathbf{X}$ are non-zero and the angle between the columns of $\mathbf{X}$ are upper-bounded by $0 < \alpha < \frac{\pi}{2}$.

Consider

$$\mathbf{x}_i^*, \mathbf{x}_j^* = \underset{\mathbf{x}_i, \mathbf{x}_i}{\arg} \Big(\frac{\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|}{\|\mathbf{x}_i - \mathbf{x}_j\|} = C \Big).$$

Then, we can define an upper bound on C as

$$C \leq \frac{\sqrt{\|\phi(\mathbf{x}_i^*)\|^2 + \|\phi(\mathbf{x}_j^*)\|^2 - 2\|\phi(\mathbf{x}_i^*)\|\|\phi(\mathbf{x}_j^*)\| \cos\alpha}}{\|\mathbf{x}_i^* - \mathbf{x}_j^*\|},$$

which can be minimized by decreasing α. Strictly speaking, C can still be considerable with a small α, if $|(\|\phi(\mathbf{x}_i^*)\| - \|\phi(\mathbf{x}_j^*)\|)|$ is large enough. However, in practice, we observe that the $\|\phi(\mathbf{x})\|$'s do not deviate from their maximum norm within the set significantly. That is, within a small sub set of $\mathbf{x}$, the vectors $\phi(\mathbf{x})$ approximately lie on a sphere (see Appendix). Therefore, we make the following claim: *the local Lipschitz constant of a network layer can be minimized by reducing the angles between the output vectors.* Below, we justify this claim from another perspective.

Consider a set of coordinates $\{\mathbf{x}_i\}_{i=1}^N$ and the function $\phi(\cdot)$ induced by a hidden-layer of an MLP. Let $\{\lambda\}_{i=1}^N$ be the singular values of $\mathbf{X}$ where $\mathbf{X} = \big[\phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)^T \dots \phi(\mathbf{x}_N)^T\big]$. Intuitively, if the angles α between the columns of $\mathbf{X}$ are small, most of the energy of the singular values should be concentrated on the first few components. On the other hand, if α is high, the energies should be distributed. Therefore, we advocate in this paper that the stable rank, defined as $\mathcal{S}(\mathbf{X}) = \sum_{i=1}^N \frac{\sqrt{\lambda_i}}{\max_i(\sqrt{\lambda_i})}$ [20], can be used as a useful proxy measure for the

spread (angles) of the column vectors of $\mathbf{X}$, *i.e.*, $\mathcal{S}(\mathbf{X})$ is large if the spread is large, and vise-versa. We empirically demonstrate that MLPs are *not* able to obtain a high Lipschitz constant with a small $\mathcal{S}(\mathbf{X})$ (see Fig. 2 and Fig. 7). If our intuition was incorrect (*i.e.*, if the network could obtain a high Lipschitz constant with a small α by varying the norm of the layer outputs significantly), we should be able to observe high local Lipschitz constants with small $\mathcal{S}(\mathbf{X})$. Our experimental results in strongly counters this. That is, networks can *not* obtain a high Lipschitz constant if $\mathcal{S}(\mathbf{X})$ are low. In other words, coordinate-MLPs primarily try to increase the Lipschitz constant by increasing the angles between the network outputs.

Based on the gathered insights within this section, we argue that $\mathcal{S}(\mathbf{X})$ is a potentially useful proxy measure for the local Lipschitz smoothness of network layers. More precisely, if $\mathcal{S}(\mathbf{X})$ is larger, then the Lipschitz constant C tends to become larger, and vice-versa. This is a useful result, as computing the exact Lipschitz constant of an MLP is an NP-hard problem [22]. Although one can efficiently obtain upper-bounds for the Lipschitz constant, that requires calculating the gradients of the function. Instead, we can gain a rough understanding on the behavior of the Lipschitz smoothness of a particular layer by observing $\mathcal{S}$ at run-time. We should emphasize that these insights are based on intuition and empirical evaluation. A more rigorous proof on this relationship is outside the scope of this paper, but the established relationship is sufficient to allow us to make some useful architectural predictions. In Section 3.4, we will connect these gained insights to the *local* Lipshchitz smoothness of the signal and the properties of activation functions.

3.4 Local Lipschitz smoothness and the activation function

Activation (ψ)	Equation	parameterized	ψ'	ψ''	**R1**	**R2**
ReLU	$\max(0, x)$	✗	$\begin{cases}1, & \text{if } x>0\\ 0, & \text{otherwise}\end{cases}$	0	✗	✗
PReLU	$\begin{cases}x, & \text{if } x>0\\ ax, & \text{otherwise}\end{cases}$	✓	$\begin{cases}1, & \text{if } x>0\\ a, & \text{otherwise}\end{cases}$	0	✓	✗
Sin	$\sin(ax)$	✓	$a\cos(ax)$	$-a^2\sin(ax)$	✓	✓
Tanh	$\frac{e^x-e^{-x}}{e^x+e^{-x}}$	✗	$\frac{4e^{2x}}{\left(e^{2x}+1\right)^2}$	$-\frac{8\left(e^{2x}-1\right)e^{2x}}{\left(e^{2x}+1\right)^3}$	✗	✓
Sigmoid	$\frac{1}{1+e^{-x}}$	✗	$\frac{e^x}{(e^x+1)^2}$	$-\frac{(e^x-1)e^x}{(e^x+1)^3}$	✗	✓
SiLU	$\frac{x}{1+e^{-x}}$	✗	$\frac{e^x(e^x+x+1)}{(e^x+1)^2}$	$-\frac{e^x((x-2)e^x-x-2)}{(e^x+1)^3}$	✗	✓
SoftPlus	$\frac{1}{a}\log(1+e^{ax})$	✓	$\frac{e^{cx}}{1+e^{cx}}$	$\frac{ce^{cx}}{(e^{cx}+1)^2}$	✓	✗
Gaussian	$e^{\frac{-0.5x^2}{a^2}}$	✓	$-\frac{xe^{-\frac{x^2}{2a^2}}}{a^2}$	$\frac{\left(x^2-a^2\right)e^{-\frac{x^2}{2a^2}}}{a^4}$	✓	✓
Quadratic	$\frac{1}{1+(ax)^2}$	✓	$-\frac{2a^2x}{\left(a^2x^2+1\right)^2}$	$\frac{2a^2\left(3a^2x^2-1\right)}{\left(a^2x^2+1\right)^3}$	✓	✓
Multi Quadratic	$\frac{1}{\sqrt{1+(ax)^2}}$	✓	$-\frac{a^2x}{\left(a^2x^2+1\right)^{\frac{3}{2}}}$	$\frac{2a^4x^2-a^2}{\left(a^2x^2+1\right)^{\frac{5}{2}}}$	✓	✓
Laplacian	$e^{\left(\frac{-\lvert x\rvert}{a}\right)}$	✓	$\frac{xe^{\frac{\lvert x\rvert}{a}}}{a\lvert x\rvert}$	$\frac{e^{\frac{\lvert x\rvert}{a}}}{a^2}$	✓	✓
Super-Gaussian	$[e^{\frac{-0.5x^2}{a^2}}]^b$	✓	$-\frac{bxe^{-\frac{bx^2}{2a^2}}}{a^2}$	$\frac{b\left(bx^2-a^2\right)e^{-\frac{bx^2}{2a^2}}}{a^4}$	✓	✓
ExpSin	$e^{-\sin(ax)}$	✓	$ae^{\sin(ax)}\cos(ax)$	$-a^2e^{\sin(ax)}\left(\sin(ax)-\cos^2(ax)\right)$	✓	✓

Table 1. Comparison of existing activation functions (top block) against the proposed activation functions (bottom block). The proposed activations and the sine activations fulfill **R1** and **R2**, implying better suitability to encode high-frequency signals.

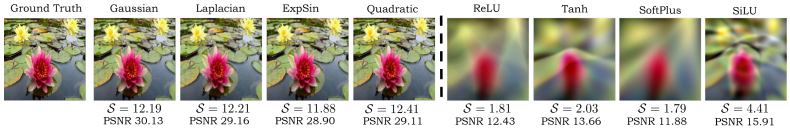

Fig. 2. Proposed activations (left block) vs. existing activations (right block) and their respective stable ranks ($\mathcal{S}$) in image encoding without positional embeddings. As predicted by Table 1, the proposed activations are better suited for encoding signals with high fidelity. As Sec. 3.2 stated, the stable ranks of the proposed activations are higher, indicating larger local Lipschitz constants which allow sharper edges.

Natural signals have varying local Lipschitz smoothness. For instance, an image may contain high variations within a particular subset of the pixels and may consist of constant pixel values within another subset. Since the final layer of an MLP is linear, the hidden non-linear layers should then have the ability to construct representations with varying local Lipschitz smoothness for better signal encoding. In this section, we show that this ability is primarily linked to the first and second-order gradients of the activation function.

In Sec. 3.3, we established that in practice, the angle between the network outputs determines the Lipschitz smoothness. It is easy to see that both the affine transformation and the activation function contribute to the composite Lipschitz constant of a hidden layer. However, the Lipschitz constant of the affine transformation is the operator norm of its weight matrix: Let $\mathbf{x} \in \mathbb{B}_{\delta}^{m}$ with center $\mathbf{x}_0$. Then as $\lim_{\delta \to 0}$,

$$\|(\mathbf{A}\mathbf{x} + \mathbf{b}) - (\mathbf{A}\mathbf{x}_0 + \mathbf{b})\| \leq C_{\mathbf{x}_0,\delta} \|\mathbf{x} - \mathbf{x}_0\| \tag{7}$$

$$C_{\mathbf{x}_0,\delta} = \sup_{\|\mathbf{x}-\mathbf{x}_0\| \neq 0} \frac{\|\mathbf{A}(\mathbf{x} - \mathbf{x}_0)\|}{\|\mathbf{x} - \mathbf{x}_0\|}, \tag{8}$$

which is not a local property. In other words, the network can only control the Lipschitz smoothness of the network via the affine layer globally, which is not useful in encoding natural signals. Hence, we direct our attention towards the activation function. However, it is not trivial to establish the connection between the point-wise activation function $\psi : \mathbb{R} \to \mathbb{R}$ and the composite Lipschitz smoothness, given that the vector norms stays approximately the same (which is our empirical observation). Hence, we strive to obtain mathematical intuition as described next.

Consider an input vector $\mathbf{x}_0 = [x_1, \ldots, x_N]$. Further, let $\mathbf{x}_{\epsilon_1} = [x_1 + \epsilon_1, \ldots, x_N + \epsilon_1]$ and $\mathbf{x}_{\epsilon_2} = [x_1 + \epsilon_2, \ldots, x_N + \epsilon_2]$. Our intention is to obtain a measure for $\measuredangle(\mathbf{x}_0, \mathbf{x}_{\epsilon_1}) - \measuredangle(\mathbf{x}_0, \mathbf{x}_{\epsilon_2})$. Further,

$$\measuredangle(\mathbf{x}_0, \mathbf{x}_{\epsilon_1}) = \cos^{-1}\Big(\frac{\psi(\mathbf{x}_0) \cdot \psi(\mathbf{x}_{\epsilon_1})}{\|\psi(\mathbf{x}_0)\| \|\psi(\mathbf{x}_{\epsilon_1})\|}\Big).$$

Since the norms are approximately constant, we can use a proxy for $\measuredangle(\mathbf{x}_0, \mathbf{x}_{\epsilon_1}) - \measuredangle(\mathbf{x}_0, \mathbf{x}_{\epsilon_2})$ as,

$$\begin{aligned}
|\tilde{\measuredangle}(\mathbf{x}_0, \mathbf{x}_{\epsilon_1}) - \tilde{\measuredangle}(\mathbf{x}_0, \mathbf{x}_{\epsilon_2})| &= |\psi(\mathbf{x}_0) \cdot \psi(\mathbf{x}_{\epsilon_1}) - \psi(\mathbf{x}_0) \cdot \psi(\mathbf{x}_{\epsilon_2})| \\
&= |\sum_{i=i}^{N} \Big(\psi(x_i + \epsilon_1) - \psi(x_i + \epsilon_2)\Big)\psi(x_i)| \\
&\leq \sum_{i=i}^{N} |\Big(\psi(x_i + \epsilon_1) - \psi(x_i + \epsilon_2)\Big)||\psi(x_i)| \\
&\leq C_\psi |\epsilon_1 - \epsilon_2| \sum_{i=i}^{N} |\psi(x_i)|,
\end{aligned}$$

Fig. 3. Novel view synthesis without positional embedding (zoom in for a better view). Gaussian activations can completely omit positional embeddings while producing results with significantly better fidelity. In contrast, the performance of ReLU-MLPs severely degrade when positional embeddings are not used. We use 8-Layer MLPs for this experiment.

where C_ψ is the local Lipschits constant of the activation function in the corresponding interval I. We then obtain,

$$\frac{|\tilde{\measuredangle}(\mathbf{x}_0, \mathbf{x}_{\epsilon_1}) - \tilde{\measuredangle}(\mathbf{x}_0, \mathbf{x}_{\epsilon_2})|}{|\epsilon_1 - \epsilon_2|} \leq C_\psi |\sum_{i=i}^{N} |\psi(x_i)| \tag{9}$$

Therefore, the upper-bound on the Lipschitz constant of the angle variation in a local interval can be increased by increasing the local Lipschitz constant of the activation function. Further, by definition, the local Lipschits constant $C_\psi = \sup_{x \in I}(|\frac{d\psi}{dx}|)$. Therefore, we come to the conclusion that in order to encode signals with high frequencies (large fluctuations), one needs to use activation functions that contain first-order derivatives with large magnitudes (the converse is also true). Also, it is important to note that the magnitudes of the local variations depend on the signal. For instance, one can have an extremely smooth signal which can be encoded using activation functions with smaller magnitudes of first-order derivatives. However, the same activations would not be suitable for encoding signals with large fluctuations. Therefore, for better usability across signals with different smoothness properties, activation functions

need to be parameterized where the first-order derivates can be controlled via the hyperparameters. We denote this as the requirement 1 (**R1**)

However, **R1** is not necessarily sufficient to ensure good signal fidelity when considering a particular signal with significantly varying fluctuations across different intervals. Thus, for better performance, activations should consist of varying first-order derivatives across a considerable interval, and equivalently, non-negligible second-order derivatives (to obtain varying Lipschitz smoothness). We denote this as requirement 2 (**R2**). This gives the affine transformations the ability to project the points to different regions of the activation function and achieve varying local Lipschitz smoothness.

Fig. 4. Novel view synthesis with positional embedding (zoom in for a better view). With Gaussian activations, shallow MLPs can obtain high-fidelity reconstructions. In contrast, the performance of ReLU-MLPs degrade when the depth of the MLP reduces. We use 4-layer MLPs for this comparison.

It is interesting to note that most of the commonly used activations in deep learning do not satisfy above properties. For instance, consider the ReLU activation $\psi(x) = \max(0, x)$. The derivative of the ReLU then cannot be more than 1, which hampers its ability to encode function with large local variations. Activations such as Sigmoid, Tanh, GELU also have bounded first-order gradient magnitudes within a smaller range and are not parameterized, which violates **R1**. On the other hand, PReLUs, $\psi(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{otherwise} \end{cases}$, is a parameterized activation that can have extremely large derivatives by controlling the hyperparameter a. However, this derivative is either 1 or a, which violates **R2** and hampers the network's ability to obtain varying local smoothness. In contrast, recently proposed sine activations [24] $\psi = \sin(ax)$ satisfy both **R1** and **R2**, and thus, are suitable for encoding signals. However, we show that the periodicity, as advocated in [24], is not a crucial requirement, as long as **R1** and **R2** are satisfied. Instead, we affirm that there is a much broader class of activations that can be used in coordinate-MLPs, and propose several parameterized activation functions that originate from the family of infinitely differentiable functions as examples. Table 1 compares existing and several novel activation functions that we propose, against **R1** and **R2**. Finally, it is important to note that even with-

out the restriction that the norms of the vectors are approximately constant, the above conclusions hold (see Appendix).

Activation	Depth	PE	PSNR	SSIM
ReLU	4L	✓	27.44	0.922
Gaussian	4L	✓	**31.13**	**0.947**
ReLU	8L	✗	26.55	0.918
Gaussian	8L	✗	**31.17**	**0.949**
ReLU	8L	✓	30.91	0.941
Gaussian	8L	✓	**31.58**	**0.951**

Fig. 5. Quantitative comparison in novel view synthesis on the real synthetic dataset [11]. Gaussian activations can achieve high-fidelity reconstructions without positional embeddings. When equipped with positional embeddings, they demonstrate similar performance with $\sim 50\%$ less parameters.

Fig. 6. Qualitative comparison of convergence when MLPs are initialized without following [24] (after 3000 epochs). Unlike Sine activations, Gaussian activations are robust to various initialization schemes (example shown used Xavier normal initialization).

4 Experiments

4.1 Comparison of activation functions

We compare the capacity of a coordinate-MLP in encoding signals when equipped with different activation functions. Fig. 2 illustrates an example where an image is encoded as the weights of an MLP. As shown, newly proposed Gaussian, Laplacian, ExpSin, and Quadratic activation functions are able to encode the image with significantly better fidelity with sharper gradients (high Lipschitz constants), compared to the existing activations such as ReLU, Tanh, SoftPlus, and SiLU. Also, note that the stable ranks (the energy distribution between the singular values) of the hidden representations are higher for the proposed activation functions than the rest. This matches with our theoretical predictions in Sec.3.2

4.2 Novel view synthesis

Without positional embeddings: We leverage the real synthetic dataset released by [11] to test the capacity of the Gaussian activations in encoding high-dimensional signals. Fig. 3 qualitatively contrasts the performance of ReLU vs. Gaussian activations without the positional embeddings. When the positional embeddings are not used, the ReLU MLPs demonstrate poor performance in capturing high-frequency details. On the contrary, Gaussian activations can capture information with higher fidelity in the absence of positional embedding. We believe this is an interesting result that opens up the possibility of positional-embedding-free architectures.

With positional embeddings: Although suitably chosen activation functions can omit positional embeddings, the combination of the two can still enable shallower networks to learn high-frequency functions. Fig. 4 depicts an example with 4-layer MLPs. As evident, when the network is shallower, ReLU MLPs showcase reduced quality, while the performance of Gaussian activated MLPs is on-par with deeper ReLU MLPs. This advocates that practitioners can enjoy significantly cheaper architectures when properly designed activation functions are used. Table 5 depicts the quantitative results that include above comparisons.

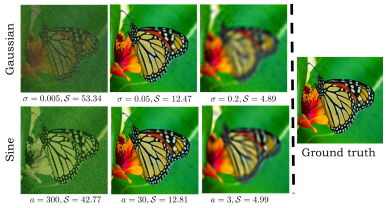

Fig. 7. Stable rank ($\mathcal{S}$) vs the fidelity of reconstructions. Having an extremely high or low stable rank (or equivalently a Lipschitz constant) hampers the ability of an MLP in encoding functions with fine details (Sec. 3.2). Thus, it is important to adjust the hyper-parameters of an activation function to tune the above metrics to a suitable range.

Fig. 8. Distribution of the upper-bound of the point-wise Lipschitz constant $\|\mathbf{J}(f)_{\mathbf{x}}\|_F$ with Gaussian reconstructions. Having an activation function with a suitable bound for the local Lipschitz constant helps the network to learn functions with properly distributed derivatives.

4.3 Convergence

Sitzmann *et al.* comprehensively demonstrated that sine activations enable MLPs to encode signals with fine details. However, a drawback entailed with the sine activations is that they are extremely sensitive to the initialization of the MLP. In comparison, the proposed non-periodic activation functions do not suffer from such a problem. Fig. 6 illustrates a qualitative example. When the initialization method of the MLP does not strictly follow the method proposed in Sitzmann *et al.*, the sine activated MLPs do not converge even after 3000 epochs. In contrast, Gaussian activations demonstrate much faster convergence. Fig. 9 illustrates a quantitative comparison of convergence. We trained the networks on the natural image dataset released by [27] and the average PSNR value after each iteration is shown in Fig. 9. As clearly evident, the Gaussian activations enjoy higher robustness against the various initialization schemes of an MLP.

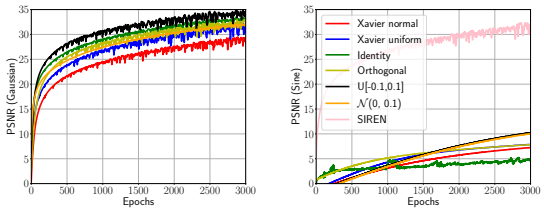

Fig. 9. Convergence rates of Gaussian and sine MLPs on natural images by [27] under different initialization schemes. Gaussian activations are significantly robust to various initialization methods compared to sine activations. Other proposed non-periodic activation functions (not shown in the figure) also demonstrate similar robustness.

4.4 Local Lipschitz smoothness

The local Lipschitz smoothness of a function converges to the Jacobian norm at the corresponding point (see Appendix). In Section 3.4, we showed that a good proxy measure for the Lipschitz constant is the range of the first-order derivative of the activation function. We further affirmed that the Lipschitz constant should be suitably chosen for better performance *i.e.* a too high or too low Lipschitz constant can prevent the network from properly learning a signal. Fig. 7 illustrates an example that confirms this statement. When σ increases, Range$|\psi'|$ of the Gaussian activation decreases, decreasing the Lipschitz constant (see Sec. 3.4). In contrast, when a increases, the Range$|\psi'|$ of the sine activation increases, increasing the Lipschitz constant. A lower Lipschitz constant results in blurry edges as it does not allow sharp changes locally. On the other hand, an extremely large Lipschitz constant allows unwanted fluctuations. Hence, choosing the parameters to be in a suitable range is vital for better performance. Fig. 8 shows the distribution of local Lipschitz constants after encoding a signal with Gaussian activations with properly chosen parameters.

5 Conclusion

We seek to extend the current understanding of activation functions that allow coordinate-MLPs to encode functions with high fidelity. We show that the previously proposed sinusoid activation [25] is a single example of a much broader class of activation functions that enable coordinate-MLPs to encode high-frequency signals. Further, we develop generic guidelines to devise and tune an activation function for coordinate-MLPs and propose several non-periodic activation functions as examples. The proposed activation functions allow positional-embedding-free coordinate-MLPs, and show much better convergence properties against various initialization schemes compared to sinusoid activations. Finally, choosing Gaussian activations from the proposed list, we demonstrate compelling results across various signal encoding tasks.

References

1. Basher, A., Sarmad, M., Boutellier, J.: Lightsal: Lightweight sign agnostic learning for implicit surface representation. arXiv preprint arXiv:2103.14273 (2021)
2. Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., Kritchman, S.: Frequency bias in neural networks for input of non-uniform density. In: International Conference on Machine Learning. pp. 685–694. PMLR (2020)
3. Biship, C.M.: Pattern recognition and machine learning (information science and statistics) (2007)
4. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
5. Deng, B., Lewis, J.P., Jeruzalski, T., Pons-Moll, G., Hinton, G., Norouzi, M., Tagliasacchi, A.: Nasa neural articulated shape approximation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. pp. 612–628. Springer (2020)
6. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4857–4866 (2020)
7. Henzler, P., Mitra, N.J., Ritschel, T.: Learning a neural 3d texture space from 2d exemplars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8356–8364 (2020)
8. Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural networks **6**(6), 861–867 (1993)
9. Liang, S., Lyu, L., Wang, C., Yang, H.: Reproducing activation function for deep learning. arXiv preprint arXiv:2101.04844 (2021)
10. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7210–7219 (2021)
11. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision. pp. 405–421. Springer (2020)
12. Mu, J., Qiu, W., Kortylewski, A., Yuille, A., Vasconcelos, N., Wang, X.: A-sdf: Learning disentangled signed distance functions for articulated shape representation. arXiv preprint arXiv:2104.07645 (2021)
13. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
14. Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., Geiger, A.: Texture fields: Learning texture representations in function space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4531–4540 (2019)
15. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
16. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5865–5874 (2021)

17. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10318–10327 (2021)
18. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019)
19. Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K.M., Tagliasacchi, A.: Derf: Decomposed radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14153–14161 (2021)
20. Rudelson, M., Vershynin, R.: Sampling from large matrices: An approach through geometric functional analysis. Journal of the ACM (JACM) **54**(4), 21–es (2007)
21. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2304–2314 (2019)
22. Scaman, K., Virmaux, A.: Lipschitz regularity of deep neural networks: analysis and efficient estimation. arXiv preprint arXiv:1805.10965 (2018)
23. Shannon, C.E.: Communication in the presence of noise. Proceedings of the IRE **37**(1), 10–21 (1949)
24. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems **33** (2020)
25. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. arXiv preprint arXiv:1906.01618 (2019)
26. Sun, Y., Liu, J., Xie, M., Wohlberg, B., Kamilov, U.S.: Coil: Coordinate-based internal learning for imaging inverse problems. arXiv preprint arXiv:2102.05181 (2021)
27. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. arXiv preprint arXiv:2006.10739 (2020)
28. Tiwari, G., Sarafianos, N., Tung, T., Pons-Moll, G.: Neural-gif: Neural generalized implicit functions for animating people in clothing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11708–11718 (2021)
29. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021)
30. Xiang, F., Xu, Z., Hasan, M., Hold-Geoffroy, Y., Sunkavalli, K., Su, H.: Neutex: Neural texture mapping for volumetric neural rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7119–7128 (2021)
31. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021)
32. Zheng, J., Ramasinghe, S., Lucey, S.: Rethinking positional encoding. arXiv preprint arXiv:2107.02561 (2021)