From Activation to Initialization: Scaling Insights for Optimizing Neural Fields

Hemanth Saratchandran*
Australian Institute of Machine Learning,
University of Adelaide, Australia

Sameera Ramasinghe
Amazon, Australia

Simon Lucey
Australian Institute of Machine Learning,
University of Adelaide, Australia

Abstract

In the realm of computer vision, Neural Fields have gained prominence as a contemporary tool harnessing neural networks for signal representation. Despite the remarkable progress in adapting these networks to solve a variety of problems, the field still lacks a comprehensive theoretical framework. This article aims to address this gap by delving into the intricate interplay between initialization and activation, providing a foundational basis for the robust optimization of Neural Fields. Our theoretical insights reveal a deep-seated connection among network initialization, architectural choices, and the optimization process, emphasizing the need for a holistic approach when designing cutting-edge Neural Fields.

1. Introduction

Neural Fields have emerged as a compelling paradigm leveraging coordinate-based neural networks to achieve a concise and expressive encoding of intricate geometric structures and visual phenomena [5, 6, 17, 19, 28, 33]. Despite the notable advancements in the application of neural fields across various domains [7, 24, 31, 32], the prevalent approach to understanding and designing these networks remains primarily empirical. Additionally, in the era of big data, practitioners often follow the trend of scaling networks ad hoc with larger datasets, lacking a clear understanding of how such architectures should proportionately adapt to data size.

In this paper, we explore the scaling dynamics of neural fields in relation to data size. Specifically, when given a neural field and a dataset, we inquire about the number of parameters necessary for the neural architecture to facilitate gradient descent convergence to a global minimum. Our theoretical findings imply that the answer to this question depends on the chosen activation and initialization scheme. For shallow networks employing a sine [30], sinc [24], Gaussian [23], or wavelet activation [26] and initialized with standard schemes such as LeCun [16], Kaiming [13], or Xavier [11], the network parameters must scale superlinearly with the number of training samples for gradient descent to converge effectively. In the case of deep networks with the same activations and initializations, we prove that the network parameters need to scale super-quadratically. This contrasts with the work [20], demonstrating that networks employing a ReLU activation, with or without a positional embedding layer [33], scale quadratically in the shallow setting and cubically in the deep setting. Other studies have explored analogous scaling laws [1–4, 21], yet in each instance, the scaling demands were cubic or more and pertained to initializations not commonly employed by practitioners in the field. While Nguyen's work [20] was previously considered state-of-the-art, our theoretical insights challenge this notion, demonstrating that significantly fewer parameters are required when employing a different activation function. For further comparison of our results with the literature we refer the reader to the related work sec. 2.

Theoretical results often find themselves in regimes that may not align with practical applications, rendering the theory insightful but lacking in predictiveness. To underscore the predictive efficacy of our theoretical framework, we design a novel initialization scheme and demonstrate its superior optimality compared to standard practices in the literature. Our findings reveal that neural fields, equipped with the activations sine, sinc, Gaussian, or wavelet and initialized using our proposed scheme, require a linear scaling with data in the shallow case and a quadratic scaling in the deep case, for gradient descent to converge to a global optimum. When compared with standard practical initializa-

*hemanth.saratchandran@adelaide.edu.au

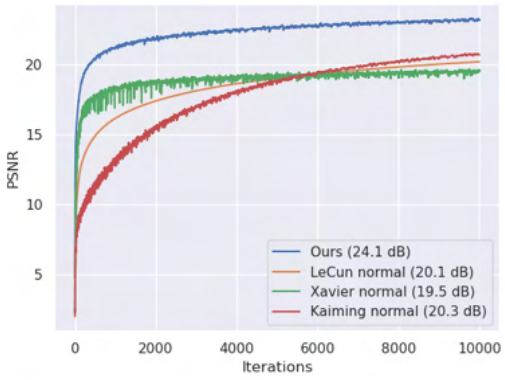

Figure 1. We evaluate Gaussian-activated networks comprising four hidden layers, initialized using four different methods, and trained with full-batch gradient descent. The comparison is performed on an image reconstruction task, with the final train PSNRs displayed in parentheses in the legend.

tions such as LeCun [16], Xavier [11], and Kaiming [13], our initialization proves to be significantly more parameter-efficient. We turn the readers attention to fig. 1 where we compared our initialization with Kaiming normal [13], Xavier normal [11] and Lecun Normal [16] on an image reconstruction task. As predicted by our theory, our initialization shows superior performance.

We extensively test our initialization scheme across various neural field applications including image regression, super resolution, shape reconstruction, tomographic reconstruction, novel view synthesis and physical modeling. Our contributions include:

1. The first theoretical proof of scaling laws for neural fields with the activations sine, sinc, Gaussian, and wavelets, ensuring effective convergence with gradient descent. The proof demonstrates that networks employing these activations surpass state-of-the-art outcomes in terms of parameter efficiency.
2. The development of a superior initialization scheme compared to standard approaches in the literature.
3. Empirical validation of our theoretical predictions on various neural field applications

2. Related Work

Several works have considered the effect of overparameterization on gradient descent convergence. The work [10] considered convergence of gradient decent for smooth activations using the Neural Tangent Kernel (NTK) parameterization [15] and showed that if all hidden layers satisfied the growth $\Omega(N^4)$, then gradient decent converges to a global minimum. In [14], using the Neural Tangent Hierarchy it was shown that for a smooth activation, $\Omega(N^3)$ suffices for all the hidden layers to guarantee convergence of gradient decent to a global minimum. Both these papers used the standard normal distribution as initialization $\mathcal{N}(0,1)$, which is rarely used in practise especially in neural fields applications. There have been several works that have studied the convergence of gradient decent for ReLU activated neural networks, [3, 37] prove convergence for gradient decent in the setting where the input and output layers are fixed, while the inner layers are only trained. Their result requires polynomial overparameterization for a large degree polynomial. For two layer ReLU networks, the works [4, 10, 21] study the convergence of gradient decent essentially showing that the width scaling must be at the order of $\Omega(N^4)$. For deep ReLU networks, the state of the art was proved in [20] showing that one could take one hidden layer of order $\Omega(N^3)$. In each case our results show that much less overparamterization is needed and our analysis is carried out with initializations used by practitioners.

3. Notation

We consider a depth L neural network with layer widths $\{n_1, \dots, n_L\}$. We let $X \in \mathbb{R}^{N \times n_0}$ denote the training data, with n_0 being the dimension of the input and N being the number of training samples. We let $Y \in \mathbb{R}^{N \times n_L}$ denote the training labels. The output at layer k will be denoted by F_k and is defined by

$$F_k = \begin{cases} F_{L-1}W_L + b_L, & k = L \\ \phi(F_{k-1}W_k + b_k), & k \in [L-1] \\ X, & k = 0 \end{cases} \tag{3.1}$$

where the weights $W_k \in \mathbb{R}^{n_{k-1} \times n_k}$ and the biases $b_k \in \mathbb{R}^{n_k}$ and ϕ is an activation applied component wise. The notation $[m]$ is defined by $[m] = \{1, \dots, m\}$. For a weight matrix W_k at layer k, the notation W_k^0 will denote the initialization of that weight matrix. These are the initial weights of the network before training begins under a gradient descent algorithm. The networks we use in the paper will all be trained using the MSE loss function, which we denote by $\mathcal{L}$. Our theoretical results will be primarily for the case where ϕ is given by one of the activation sine, sinc, Gaussian or wavelet. We remind the reader that the sinc function is defined by: $sinc(x) = \frac{sin(x)}{x}$ for $x \neq 0$ and $sinc(0) = 1$. For more details on how these activations are used within the context of neural fields we refer the reader to [23, 24, 26, 29].

We will always assume the data set X consists of i.i.d sub-gaussian vectors and for the theoretical proofs will assume they are normalized to have norm $||X_i||_2 = 1$. More details on data assumptions can be found in sec. 1 of the supp. material.

Our networks will be free to have a positional embedding layer (PE), which is simply an embedding of the data into a higher dimensional space. The reader who is unfamiliar with PE should consult the standard references [19, 33].

A neural field is any such network that parameterizes a continuous field. Examples of neural fields can be found in sec. 5. All networks will be trained with the standard Mean Square Error (MSE) loss.

We will use standard complexity notations, $\Omega(\cdot)$, $\mathcal{O}(\cdot)$, $\Theta(\cdot)$ throughout the paper. The reader who is unfamiliar with this notation can consult sec. 1 of the supp. material. Finally, we will use the notation "w.h.p." to denote *with high probability*.

4. Theoretical Scaling Laws

In this section, we provide a theoretical understanding of how much overparameterization a neural field needs in order for gradient descent to converge to a global minimum for a given data set. Furthermore, we prove a scaling law that details how the network must scale as the dataset size grows, in order to facilitate optimum convergence of gradient descent. We will work with the activations sine, sinc and Gaussian. For wavelets see sec. 1 of the supp material. To accommodate space constraints and offer readers a more expansive perspective, we have deferred the exhaustive details of the proofs to sec. 1 of the supp. material.

We begin with a definition of overparameterization that we will use throughout the paper. Our definition is consistent with several works in the literature [2, 8, 20, 37].

Definition 4.1. Given a data set X with N samples and a neural network $F(\theta)$. We say the network $F(\theta)$ is overparameterized with respect to the size of X if the number of parameters θ is greater than N.

In general, it is known via several works in the literature that under various assumptions, overparameterization is necessary for gradient descent to converge to a global minimum [1, 3, 4, 10, 20, 21]

4.1. A scaling law for shallow networks

In this section, we present our theorem, delineating a complexity bound that dictates the extent of overparameterization necessary for a neural field to enable gradient descent convergence to a global minimum, particularly in the context of shallow networks—those with only two layers.

Theorem 4.2. *Let X be a fixed data set with N samples. Let F be a shallow neural network of depth 2 admitting one of the following activation functions:*

1. $sin(\omega x)$,
2. $sinc(\omega x) = \frac{sin(\omega x)}{x}$,
3. $e^{-x^2/2\omega^2}$

where ω ($1/\omega^2$) is a fixed frequency hyperparameter. Let the widths of the network satisfy

$$n_1 = \Omega(N^{3/2}) \text{ and } n_2 = \Theta(m) \tag{4.1}$$

where m is a fixed positive constant. Suppose the network has been initialized according to LeCun's initialization scheme

$$(W_1^0)_{ij} \sim \mathcal{N}(0, \frac{1}{n_0}) \text{ and } (W_2^0)_{ij} \sim \mathcal{N}(0, \frac{1}{n_1}). \tag{4.2}$$

Then for a small enough learning rate gradient descent converges to a global minimum w.h.p. when trained on X.

Remark 4.3. Thm. 4.2 has been expressed in the context of LeCun's initialization. However, a similar theorem can be proved for the two other standard initializations used in the literature, namely Xavier normal initialization [11] and Kaiming normal initialization [13].

4.2. A scaling law for deep networks

In this section we present a generalization of thm. 4.2 to the setting of deep networks.

Theorem 4.4. *Let X be a fixed data set of size N. Let F be a deep neural network of depth L, $L > 2$, admitting one of the following activation functions:*

1. $sin(\omega x)$,
2. $sinc(\omega x) = \frac{sin(\omega x)}{x}$,
3. $e^{-x^2/2\omega^2}$

where $\omega > 0$ ($1/\omega^2$) is a fixed frequency hyperparameter. Let the widths of the network satisfy

$$n_l = \Theta(m), \forall l \in [L] \text{ with } l \neq L-1, \tag{4.3}$$

$$n_{L-1} = \Omega(N^{5/2}), \tag{4.4}$$

where m is a fixed constant that is allowed to depend linearly on N. Suppose the network is initialized according to LeCun's initialization scheme

$$(W_l^0)_{ij} \sim \mathcal{N}(0, 1/n_{l-1}), \text{ for } l \in [L]. \tag{4.5}$$

Then for a small enough learning rate gradient descent converges to a global minimum w.h.p. when trained on X.

Remark 4.5. A similar theorem can be proved for Xavier and Kaiming normal initializations. See sec. 1 of the supp. material for learning rate bounds.

Remark 4.6. Thms. 4.2 and 4.4 reveal a significant connection between activation functions and initialization methods in neural network training. These theorems underscore the importance of carefully choosing the architecture (via activation functions) and initialization for crafting parameter-efficient networks that converge to optimal solutions.

4.3. Analyzing the proof methodology

In sec. 4.1 and 4.2, we explored the impact of initialization and activation choices on overparameterization in order for gradient descent to converge to a global minimum.

This naturally leads to the question of whether there are initialization schemes for neural fields utilizing the activations $sin(\omega x)$, $sinc(\omega x)$, and $e^{-x^2/2\omega^2}$ that necessitate less overparameterization compared to the results established in thms. 4.2 and 4.4.

In order to answer this question we start by giving a concise sketch of the main idea of the proof methodology of thm. 4.2. We let F be a fixed 2-layer network. Let $\sigma_0 = \sigma_{\min}(F_1^0)$, denote the smallest singular value of the hidden layer of F and let $W_2^0 \in \mathbb{R}^{n_1 \times n_2}$ denote the initial weight matrix of the second layer of F. The key step in proving thm. 4.2 is to establish the lower bound

$$\sigma_0^2 \geq 16\sqrt{N}\sqrt{n_1}\sqrt{2\mathcal{L}(\theta_0)}||W_2^0||_2 \tag{4.6}$$

when $n_1 = \Omega(\lceil N^{3/2} \rceil)$. Once this is achieved a routine use of convergence theory leads to the proof of thm. 4.2. The interested reader can consult sec. 1 of supp. material for full details.

To obtain (4.6), the proof proceeds by first establishing the following two complexity bounds:

$$\sigma_0 \geq \Omega(\sqrt{n_1}) \text{ and } \sqrt{2\mathcal{L}(\theta_0)} = \mathcal{O}(\sqrt{N}). \tag{4.7}$$

We pause to mention that the two inequalities in (4.7) are activation dependent and is precisely the place where we need to use that the activation is one of sine, Gaussian, sinc, or wavelet.

Substituting (4.7) into (4.6) shows us that proving inequality (4.6) boils down to showing

$$n_1 \geq CNn_1^{3/4}||W_2^0||_2, \tag{4.8}$$

where $C > 0$ is a constant (coming from the complexity bounds in (4.7)) which we won't worry about for this discussion. In order to establish inequality (4.8) we need to understand the 2-norm of the random matrix W_2^0. This is precisely where LeCun's initialization is used. By appealing to thm. 2.13 of [9], it can be shown that if a random $n_1 \times n_2$-matrix W has entries sampled from a Normal distribution of the form $\mathcal{N}(0, 1/n_1)$ then

$$||W||_2 = \mathcal{O}(\sqrt{n_2}/\sqrt{n_1}). \tag{4.9}$$

Applying (4.9) to the random weight matrix W_2^0, initialized using LeCun's initialization (4.2), we see that (4.8) holds if $n_1 = \Omega(N^{3/2})$.

Inequality (4.8) illuminates the intrinsic connection between initialization and overparameterization. The reduction of the product $n_1^{3/4}|W_2^0|_2$ serves as a theoretical indicator for the diminished necessity of overparameterization. Achieving a small value for $n_1^{3/4}|W_2^0|_2$ entails minimizing the norm $|W_2^0|_2$, and leveraging (4.9) (refer to thm. 2.13 in [9]), we ascertain that the norm $|W_2^0|_2$ possesses a complexity bound of $\mathcal{O}(\sqrt{n_2}/\sqrt{n_1})$. Consequently, for a smaller norm $|W_2^0|_2$, theoretical sampling from a Gaussian distribution with an exceedingly small variance is warranted. Suppose the entries of W_2^0 are sampled from $\mathcal{N}(0, 1/n_1^p)$. Employing (4.9) once more, we deduce that $|W_2^0|_2 = \mathcal{O}(\sqrt{n_2}/n_1^{p/2})$. Substituting this result back into (4.8), we observe that a larger value of p corresponds to a reduced complexity requirement for n_1. In essence, ***sampling the final layers weight matrix from a Normal distribution with smaller variance necessitates less overparameterization for gradient descent to converge to a global minimum***. However, it's crucial to note that excessively small variances in the Gaussian distribution pose the challenge of vanishing gradients in the network.

While the preceding discussion focused on shallow networks, a parallel argument can be extended to deep networks, illustrating that the level of overparameterization required for the last hidden layer to achieve convergence to a global minimum is linked to the variance of the Normal distribution from which we initialize the final layer's weights. Moreover, a smaller variance corresponds to a reduced need for overparameterization.

4.4. Designing new initializations

The discussion in the previous section suggested a new approach to initializing weights for a neural network. The main point was that by controlling the variance of the Normal distribution the final layer weights are sampled from, we can use less parameters and still converge to a global minimum under gradient descent.

Fix a deep neural network F with L layers. We define the following initialization.

Initialization 1:

$$(W_l^0)_{ij} \sim \mathcal{N}(0, 1/n_{l-1}) \text{ for } l \in [L-1]. \tag{4.10}$$

$$(W_L^0)_{ij} \sim \mathcal{N}(0, 2/(n_{l-1}^{3/2})). \tag{4.11}$$

As suggested by the discussion in sec. 4.3 the variance of the Gaussian we sample from for the last layer is smaller by a factor of $1/\sqrt{n_{L-1}}$. The biases of the network will be initialized to 0 or a very small number such as 0.01 as is the standard practice for many common normal initializations. Fig. 2, gives a diagrammatic rendition of initialization 1.

Theorem 4.7. *Let X be a fixed dataset with N training samples. Let F be a shallow neural network of depth 2 admitting one of the following activation functions:*

1. $sin(\omega x)$,
2. $sinc(\omega x) = \frac{sin(\omega x)}{x}$,
3. $e^{-x^2/2\omega^2}$.

Let the widths of the network satisfy

$$n_1 = \Omega(N) \text{ and } n_2 = \Theta(m) \tag{4.12}$$

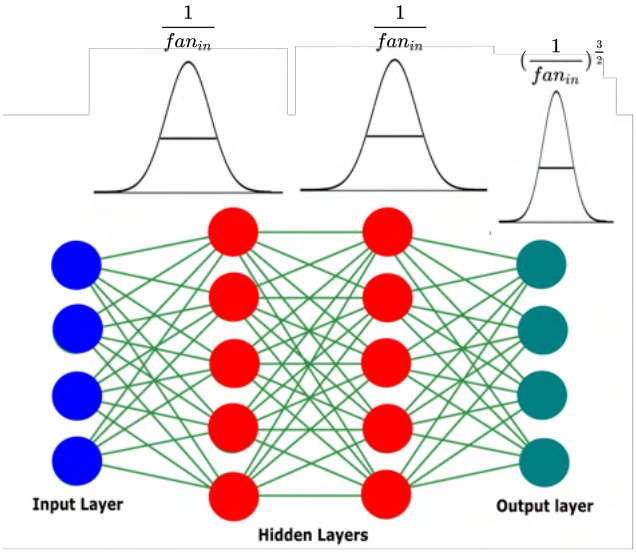

Figure 2. Diagram showing how to initialize weight matrices according to **Initialization 1**. The final output layer is initialized with a Normal distribution of smaller variance than the previous layers by a factor of $1/\sqrt{fan_{in}}$, where fan_{in} denotes the input dimension to the layer.

where m is a fixed positive constant. Suppose the network has been initialized according to initialization 1, see (4.10), (4.11). Then for a small enough learning rate gradient descent converges to a global minimum w.h.p.

Remark 4.8. Comparing thm. 4.7 with thm. 4.2, we observe that utilizing Initialization 1 requires only linear overparameterization in the final hidden layer, as opposed to superlinear overparameterization.

Theorem 4.9. *Let X be a fixed dataset with N training samples. Let F be a deep neural network of depth L, $L > 2$, admitting one of the following activation functions:*

1. $sin(\omega x)$,
2. $sinc(\omega x) = \frac{sin(\omega x)}{x}$,
3. $e^{-x^2/2\omega^2}$.

Let the widths of the network satisfy

$$n_l = \Theta(m), \forall l \in [L] \text{ with } l \neq L-1, \tag{4.13}$$

$$n_{L-1} = \Omega(N^2), \tag{4.14}$$

where m is a fixed constant that is allowed to depend linearly on N. Suppose the network is initialized by initialization 1, see (4.10), (4.11). Then for a small enough learning rate gradient descent converges to a global minimum w.h.p.

Remark 4.10. Comparing thm. 4.7 with thm. 4.4, we observe that utilizing Initialization 1 requires quadratic overparameterization in the final hidden layer, as opposed to super quadratic overparameterization, for gradient descent to converge to a global minimum.

Many practitioners in machine learning often use the Uniform distribution to sample the weight matrices of a neural network at initialization. Motivated by this we define a second uniform initialization scheme as follows.

Fix a deep neural field F with L layers We define the following uniform initialization.

Initialization 2

$$(W_l^0)_{ij} \sim \mathcal{U}(-1/\sqrt{n_{l-1}}, 1/\sqrt{n_{l-1}}) \text{ for } l \neq L. \tag{4.15}$$

$$(W_L^0)_{ij} \sim \mathcal{U}(-1/(n_{l-1}^{3/4}), 1/(n_{l-1}^{3/4})) \tag{4.16}$$

where $\mathcal{U}(a, b)$ denotes the Uniform distribution on the interval $[a, b]$. The biases can be initialized to be 0 or a very small number. In sec. 1 of the supp. material we give another way to initialize the biases.

Remark 4.11. In [29], Sitzmann et al. presented a uniform initialization for networks using a sine activation. In sec. 1 of the supp. material, we demonstrate the combination of our initialization (4.15), (4.16) with theirs and provide comparisons in sec. 2 of the supp. material.

5. Experiments: Applications to Neural Fields

In this section, we empirically test the theory developed in sec. 4, focusing exclusively on the widely used initializations LeCun normal [16], Xavier normal [11], Kaiming normal [13], and their uniform counterparts. If unfamiliar with these initializations, please refer to sec. 2 of the supp. material. Additionally, we employ four activations: sinc [24], Gaussian [23], Gabor wavelet [26], each with a frequency parameter ω (or $1/\omega^2$ for the Gaussian) and ReLU-PE [33] with a positional embedding layer. For details on tuning this hyperparameter and the frequencies used in each experiment, consult sec. 2 of the supp. material. Further experiments can also be found in sec. 2 of the supp. material.

5.1. Practical Validation of the Theoretical Analysis

We conduct empirical testing to validate the theories presented in sections 4.1 and 4.2. In section 4.1, we derived theorem 4.2, demonstrating that a shallow feedforward neural network utilizing activation functions such as sine, sinc, Gaussian, or wavelet, and initialized according to LeCun, Xavier, or Kaiming initialization, necessitates superlinear growth in the width of the hidden layer as the dataset size increases for gradient descent to converge to a global minimum.

It is noteworthy that this growth requirement is less stringent than what is established for ReLU activation (with or without PE) in Ngyuen's work [20], which asserts a quadratic growth in the number of data samples. Consequently, we anticipate observing that a ReLU (with or without PE) network demands more parameters than its sinc-activated counterpart when trained on the same dataset until convergence.

Shallow Experiment: In our investigation, we performed a 1-dimensional curve fitting experiment on the function $f(x) = \sin(2\pi x) + \sin(6\pi x) + \sin(10\pi x)$. We systematically sampled the curve at intervals of 10, 20, 50, 75, 100, 125, 150, 175, and 200 points, creating nine datasets of varying sizes for our training data.

Subsequently, we employed three networks with sinc activation and one hidden layer, along with three networks featuring ReLU-PE activation and one hidden layer. The positional embedding layer had dimension 8 and employed a Random Fourier Feature (RFF) type embedding [33]. Each dataset underwent training using full batch gradient descent until reaching a PSNR value of 35dB. We increased the number of parameters with the growth in dataset size, adjusting them until the respective networks achieved convergence at the target PSNR.

Initialization for both sinc and ReLU-PE networks was performed using LeCun, Xavier, and Kaiming initializations. The results of this experiment are illustrated in fig. 3 (left). As anticipated by thm. 4.2, the sinc-activated networks exhibited a significantly lower parameter requirement for convergence as the dataset size increased. Furthermore, as fig. 3 (left) shows the sinc networks had parameter growth similar to $\mathcal{O}(N^{3/2})$, as predicited by thm. 4.2. The ReLU-PE networks had parameter growth as $\mathcal{O}(N^2)$ as predicted in [20].

Deep Experiment: For the case of deep networks we ran a similar experiment to the above shallow networks except that this time we used an image regression task.

The task was to reconstruct a 512×512 Lion image. Given pixel coordinates $\mathbf{x} \in \mathbb{R}^2$, the aim of the task is to a network $\mathcal{N}$ to regress the associated RGB values $\mathbf{c} \in \mathbb{R}^3$ [23, 29]. We sampled 1000, 5000, 10000, 25000, 50000, 100000, 150000 and 200000 pixel coordinates, creating a total of 8 datasets of varying size as the training data.

We employed three networks with sinc activation and four hidden layers, along with three networks featuring ReLU-PE activation and four hidden layers. The PE-layer was 16 dimensional and used Random Fourier Features (RFF) as the positional embedding technique [33]. The first three hidden layers of all networks had 64 neurons each. We increased the number of parameters by only increasing the width of the last hidden layer as this was shown to suffice from thm. 4.4.

Each dataset underwent training using full batch gradient descent until reaching a PSNR value of 25dB. Initialization for both sinc and ReLU-PE networks was performed using LeCun, Xavier, and Kaiming initializations. The results of this experiment are illustrated in fig. 3 (right). As predicted by thm. 4.2, the sinc-activated networks exhibited a significantly lower parameter requirement for convergence as the dataset size increased. Nevertheless, it is essential to highlight that beyond a dataset size of $30,000$, the sinc networks achieved convergence using fewer parameters than the actual number of samples. This discrepancy arises from our decision to set the cutoff point at 25dB. The rationale behind this choice was rooted in practical considerations related to memory constraints. Given that we employed full-batch training, higher dB values necessitated a substantially greater number of parameters, leading to memory-related issues.

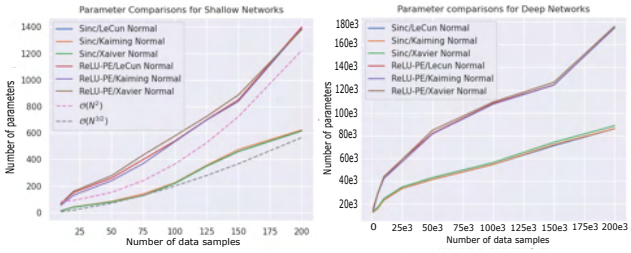

Figure 3. Comparing how many parameters are needed for a ReLU-PE and sinc network to converge with different intializations and data set sizes. Left figure shows results for shallow networks on a 1-dim. curve fitting task. Right figure shows results for deep networks on a image regression task. For all initializations, the sinc activated networks require much less parameters to converge than the ReLU-PE ones.

The second experiment was to test thm. 4.9. In order to do this we carried out a similar experiment to the above deep network experiment, using the same datasets and data instance. In this setting we considered four sinc networks, each initialized with LeCun normal, Xavier normal, Kaiming normal and **initialization 1** (see (4.10)). We trained each each network until it reached a PSNR of 25dB, increasing the width of the final hidden layer as the dataset size increased. Fig. 4 (left) shows the results of the experiment. As predicted by thm. 4.9, the network employing our initialization 1 needs far less parameters to converge. We repeated the experiments this time initializing the networks with the uniform distributions, **initialization 2** (see (4.11)), LeCun uniform, Xavier uniform, Kaiming uniform. Fig. 4 (right) shows that the network initialized with initialization 2 required less parameters to converge than all other networks.

5.2. Single Image Super Resolution

We test our initializations on an image super resolution task. We take the approach considered in [26], where a $4\times$ super resolution is conducted on an image from the DIV2K dataset [1, 34]. The problem is cast as solving $y = Ax$, where the operator A implements a $4\times$ downsampling (with no aliasing). We then solve for x as the output of a neural field.

We explored the impact of four normal initializations and four uniform initializations on a Gaussian-activated net-

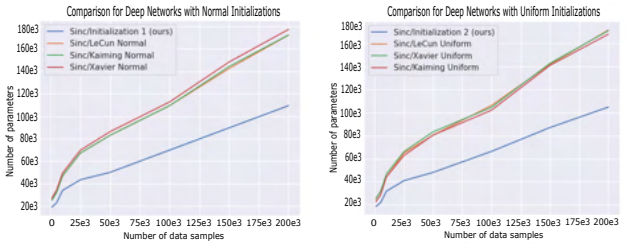

Figure 4. Comparing the performance of deep networks with sinc activation across image regression tasks, utilizing four distinct initialization schemes. Networks were trained until reaching a 25dB PSNR. On the left, we observe the outcomes with four normal initializations, showcasing that our initialization demands the fewest parameters for convergence. On the right, the comparison extends to four different uniform initializations, where our approach emerges as the most effective.

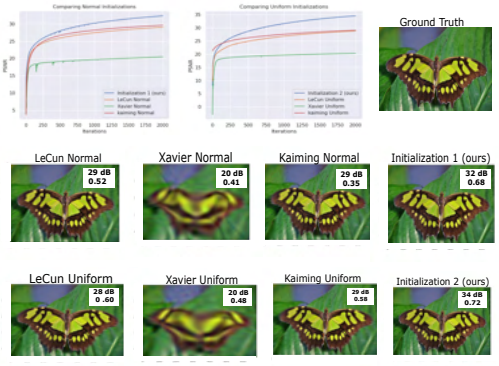

Figure 5. The figure shows the results for a $4\times$ single image super resolution with four normal initializations and four uniform initializations. Networks initialized with initialization 1 (our) and initialization 2 (our) produced the highest train dB and SSIM at convergence. Zoom in for better viewing.

work with two hidden layers. The Gaussian activation was characterized by a variance of 0.1^2, a choice validated as optimal across all initializations and commonly adopted by practitioners [7, 23, 24, 27]. Subsequently, all networks underwent training using the Adam optimizer. Figure 5 presents the outcomes of this investigation. Notably, among the normal initializations, our first initialization demonstrated superior performance, while among the uniform initializations, our second initialization excelled. In each case, Structural Similarity (SSIM [36].) was computed, with both our first and second initializations consistently producing the highest SSIM values.

5.3. Occupancy Fields

We optimize a binary occupancy field, which represents a 3D shape as the decision boundary of a neural network [12, 35]. We use the *thai statue* instance obtained from XYZ RGB Inc. We trained two groups of four networks, each

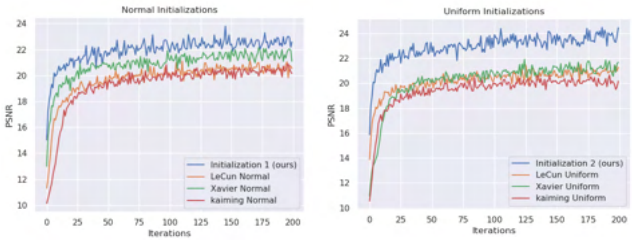

Figure 6. Top left; comparison of normal initializations. Top right; comparison of uniform initializations. In both cases our initialization performs better reaching a higher PSNR. Bottom summary of final train PSNR and IOU accuracy.

	Train PSNR (dB)	**IOU**
Initialization 1 (ours)	**22.7**	**0.89**
LeCun Normal	20.3	0.81
Xavier Normal	21.2	0.84
Kaiming Normal	19.9	0.80
Initialization 2 (ours)	**24.5**	**0.91**
LeCun Uniform	21.2	0.86
Xaiver Uniform	21.6	0.87
Kaiming Uniform	20.2	0.82

Table 1. Table showing results of each initialization on an Occupancy task.

activated with the Gabor wavelet [26]. The first group of four networks were trained with four normal initializations, Lecun normal, Xavier normal, Kaiming normal and initialization 1 (see (4.10), (4.11)). The second group of four networks were trained with LeCun uniform, Xavier uniform, Kaiming Uniform and initialization 2 (see (4.15), (4.16)). All networks were trained with the Adam optimizer. For accuracy testing we used the performance metric given by Intersection Over Union (IOU) . Fig. 6 and table 1 details the results of the experiments. As can be seen from the figure, both our initializations converge to a higher PSNR, $2-4$dB higher than the others, and have the highest IOU. Figures of reconstructed meshes can be found in the sec. 2 of supp. material.

5.4. Neural Radiance Fields (NeRF)

NeRF has recently emerged as a compelling method, leveraging a Multi-Layer Perceptron (MLP) to model 3D objects and scenes based on multi-view 2D images. This innovative approach exhibits promise in achieving high-fidelity reconstructions for novel view synthesis tasks [7, 18, 19, 25]. Given 3D points $\mathbf{x} \in \mathbb{R}^3$ and viewing directions, NeRF is designed to estimate the radiance field of a 3D scene. This field maps each input 3D coordinate to its corresponding volume density $\sigma \in \mathbb{R}$ and directional emitted color $\mathbf{c} \in \mathbb{R}^3$ [7, 18, 19].

NeRF is commonly trained using Kaiming uniform initialization for optimal outcomes. Our experiment involved

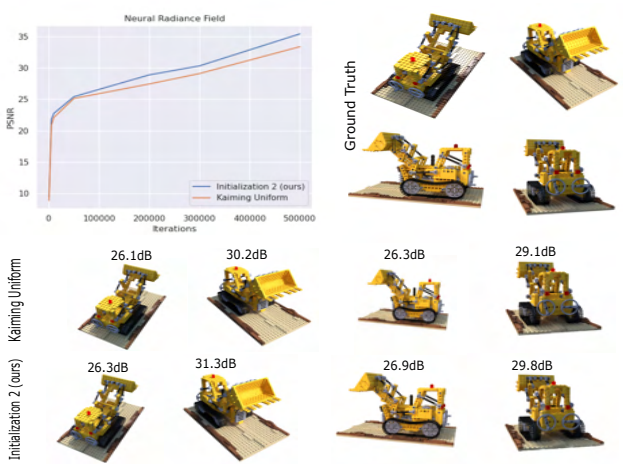

Figure 7. Training results comparison for two Gaussian-activated NeRFs: one with Kaiming uniform initialization and the other with Initialization 2 (see (4.15) and (4.16)). Top-left: Training PSNR. Top-right: Four ground truth instances for testing. Bottom: Test PSNRs. Initialization 2 consistently outperforms the Kaiming uniform-initialized NeRF in both training and test accuracy.

two Gaussian-activated NeRFs: one initialized with Kaiming uniform and the other with Initialization 2 (see (4.15), (4.16)). We used the Lego instance from the NeRF real synthetic data set. All networks were trained with the Adam optimizer. Fig. 7 displays the results, indicating that Initialization 2 achieved a higher training PSNR by 1.1dB. Testing across 24 unseen views revealed that Initialization 2 consistently outperformed Kaiming initialization, with a test difference ranging from 0.1 to 1.1dB across different scenes. For more details please see sec. 2 of the supp. material.

5.5. Physics Informed Neural Networks (PINNs)

Physics informed neural networks are an innovative neural architecture that parameterize a physics field arising as the solution of a partial differential equation (PDE). For an introduction to PINNs we refer the reader to [22].

We consider the 2D incompressible Navier-Stokes equations as considered in [22].

$$u_t + uu_x + 0.01u_y = -p_x + 0.01(u_{xx} + u_{yy}) \quad (5.1)$$

$$v_t + uv_x + 0.01v_y = -p_y + 0.01(v_{xx} + v_{yy}) \quad (5.2)$$

where $u(x, y, t)$ denotes the x-component of the velocity field of the fluid, and $v(x, y, t)$ denotes the y-component of the velocity field. The term $p(t, x, y)$ is the pressure. The domain of the problem is $[-15, 25] \times [-8, 8] \times [0, 20]$.

The fluid field solution to equations (5.1) and (5.2) can be parameterized by a neural field that is a PINN. The two PDE equations (5.1) and (5.2) are embedded into the loss function enforcing a physical constraint on the network as it trains.

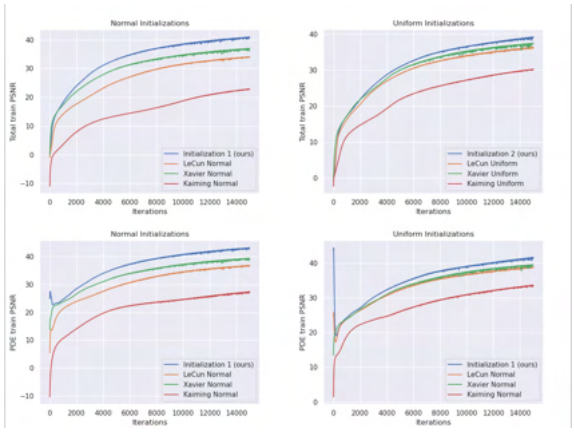

Figure 8. Results from training eight different Gaussian activated PINNs, each with a different initialization. Top row plots the total loss PSNR (MSE loss + PDE loss) and bottom row plots PDE loss PSNR. In both cases the PINNs initialized with our initialization reach a higher dB in both total and PDE loss.

We trained eight Gaussian-activated PINNs with three hidden layers and a width of 128. Four PINNs used LeCun, Xavier, Kaiming, and Initialization 1 ((4.10), (4.11)) for normal initialization, while the other four used corresponding uniform initializations. Training employed an Adam optimizer with full batch. For detailed training setup, please refer to sec. 3 in the supplementary material.

Fig. (8) shows the training PSNR's of each of the networks. As can be seen from the figure, the PINNs initialized with initialization 1 and 2 reach a higher dB in total loss and PDE loss.

6. Conclusion

This paper established a theoretical framework for the optimal scaling of neural fields as dataset sizes expand, ensuring optimal convergence during gradient descent. We uncovered a link between this scalability challenge and the activation and initialization of the network. Our theoretical framework yielded state-of-the-art results for both shallow and deep networks. Moreover, leveraging our theoretical insights, we proposed a novel initialization scheme and validated its efficacy across diverse neural field applications.

7. Limitations

This paper delves into the theory to identify the necessary degree of overparameterization for gradient descent to reach a global minimum. For practical scenarios, training typically uses mini-batches. Unfortunately, our theoretical results do not yet apply to mini-batch training and we suggest that future research applying our findings to mini-batch scenarios could provide useful methodologies on how to scale networks for such training.

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. 1, 3, 6

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019. 3

[3] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019. 2, 3

[4] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019. 1, 2, 3

[5] Boyuan Chen, Robert Kwiatkowski, Carl Vondrick, and Hod Lipson. Fully body visual self-modeling of robot morphologies. *Science Robotics*, 7(68):eabn1944, 2022. 1

[6] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 1

[7] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 264–280. Springer, 2022. 1, 7

[8] Yaim Cooper. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018. 3

[9] Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and banach spaces. *Handbook of the geometry of Banach spaces*, 1(317-366):131, 2001. 4

[10] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019. 2, 3

[11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 1, 2, 3, 5

[12] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *ICML*, 2020. 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1, 2, 3, 5

[14] Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In *International conference on machine learning*, pages 4542–4551. PMLR, 2020. 2

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

[16] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002. 1, 2, 5

[17] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pages 112–123. PMLR, 2022. 1

[18] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 7

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 7

[20] Quynh Nguyen. On the proof of global convergence of gradient descent for deep relu networks with linear widths. In *International Conference on Machine Learning*, pages 8056–8062. PMLR, 2021. 1, 2, 3, 5, 6

[21] Samet Oymak and Mahdi Soltanolkotabi. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1):84–105, 2020. 1, 2, 3

[22] 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. *Journal of Computational physics*, 378:686–707, 2019. 8

[23] S. Ramasinghe and S. Lucey. Beyond Periodicity: Towards a Unifying Framework for Activations in Coordinate-MLPs. In *ECCV*, 2022. 1, 2, 5, 6, 7

[24] Sameera Ramasinghe, Hemanth Saratchandran, Violetta Shevchenko, and Simon Lucey. On the effectiveness of neural priors in modeling dynamical systems. *arXiv preprint arXiv:2303.05728*, 2023. 1, 2, 5, 7

[25] C. Reiser, S. Peng, Y. Liao, and A. Geiger. KiloNeRF: Speeding Up Neural Radiance Fields With Thousands of Tiny MLPs. In *ICCV*, 2021. 7

[26] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 1, 2, 5, 6, 7

[27] Hemanth Saratchandran, Shin-Fang Chng, Sameera Ramasinghe, Lachlan MacDonald, and Simon Lucey. Curvature-aware training for coordinate networks. *arXiv preprint arXiv:2305.08552*, 2023. 7

[28] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-

structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[29] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, G., and Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *NIPS*, 2020. 2, 5, 6

[30] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 1

[31] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021. 1

[32] Yu Sun, Jiaming Liu, Mingyang Xie, Brendt Wohlberg, and Ulugbek S Kamilov. Coil: Coordinate-based internal learning for imaging inverse problems. *arXiv preprint arXiv:2102.05181*, 2021. 1

[33] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 1, 2, 5, 6

[34] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 6

[35] Peng-Shuai Wang, Yang Liu, Yu-Qi Yang, and Xin Tong. Spline positional encoding for learning 3d implicit signed distance fields. *arXiv preprint arXiv:2106.01553*, 2021. 7

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[37] Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. *Advances in neural information processing systems*, 32, 2019. 2, 3