Spectrum-Informed Multistage Neural Network: Multiscale Function Approximator of Machine Precision

Jakin Ng[1 2] **Yongji Wang**[1 3] **Ching-Yao Lai**[1]

Abstract

Deep learning frameworks have become powerful tools for approaching scientific problems such as turbulent flow, which has wide-ranging applications. In practice, however, existing scientific machine learning approaches have difficulty fitting complex, multi-scale dynamical systems to very high precision, as required in scientific contexts. We propose using the novel multistage neural network approach with a spectrum-informed initialization to learn the residue from the previous stage, utilizing the spectral biases associated with neural networks to capture high frequency features in the residue, and successfully tackle the spectral bias of neural networks. This approach allows the neural network to fit target functions to double floating-point machine precision $O(10^{-16})$.

1. Introduction

1.1. Precision machine learning

Typical machine learning applications, such as computer vision or natural language processing, do not necessarily require neural networks to fit data to extremely high precision. For instance, the training loss function may simply be a proxy for the true metric, and moreover noise present in the training data may cause models with very low training loss to overfit (Michaud et al., 2023). Recently, deep learning techniques are being increasingly developed for scientific purposes where high precision is desirable or required (Wang et al., 2023; Michaud et al., 2023; Wang & Lai, 2024; Müller & Zeinhofer, 2023; Jnini et al., 2024). For instance, neural networks used as interpolators for equation discovery, which infers an exact equation or formula based on data, the correctness of the learned equation requires very high accuracy across multiple scales (Udrescu & Tegmark, 2020; Mojgani et al., 2024). Another application requiring high precision is physics-informed neural networks (PINNs), which behave as numerical solvers for partial differential equations (PDEs), where high accuracy is an intrinsic requirement. When solving a well-posed PDE problem, there exists a theoretical global minimum for the PINN training where the training loss should converge to zero (Raissi et al., 2019).

In this study, we focus on studying the regression problem as a preliminary example to demonstrate the challenge of training neural networks to approach a target function with high precision. The regression problem is given as: given a dataset $\{(\mathbf{x_i}, u_i = u(\mathbf{x_i}))\}$ sampled from a continuous target function u, the neural network $\mathcal{N}_\theta$, with parameters θ, is trained to fit the data points to approximate the function u. We assume that the number of data points is sufficient, and that each of them has zero noise, which guarantees that the data set contains sufficient authentic information of the target functions. Although the universal approximation theorem is a theoretical guarantee that neural networks of large enough size can approximate any function arbitrarily well (Hornik et al., 1989), in practice, the training loss is easily trapped in local minima and eventually plateaus after a certain number of iterations. Many advanced techniques exist to expedite training (Sitzmann et al., 2020; Michaud et al., 2023; Liu et al., 2020), but are unable to consistently reduce the training error down to the required precision, which is one of the significant challenges of using deep learning methods for scientific objectives. In contrast, classical numerical methods can consistently reduce error, for instance by increasing the mesh resolution.

1.2. Spectrum-informed initialization for regression

To resolve the precision limit of PINNs, Wang & Lai (2024) proposed the multistage training scheme which divides neural network training into different stages. For each stage of training, a new neural network was introduced and optimized to learn the residues from previous stages, which largely increases the convergence rate from linear decay

[1]Stanford University, Palo Alto, CA. [2]Massachusetts Institute of Technology, Cambridge, MA. [3]New York University, New York, NY.. Correspondence to: Ching-Yao Lai <cyaolai@stanford.edu>, Yongji Wang <yw8211@nyu.edu>, Jakin Ng <jakinng@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

to approximately exponential decay. The combined neural networks after several stages of training can approximate the target function up to double floating-point machine precision $O(10^{-16})$ for one-dimensional problems. However, this fails to hold for two- or higher-dimensional problems (Figure 1a). This indicates that more advanced techniques are required to enhance the multistage neural networks (MSNNs) to ensure high-precision approximation for higher-dimensional problems.

The training of neural networks is known to suffer from spectral bias, which is a phenomenon in which neural networks tend to fit low-frequency features of the target function, and may fail to capture high-frequency information (Rahaman et al., 2019; Xu et al., 2022). Spectral bias is particularly problematic when using neural networks as function approximators for multiscale problems, such as turbulence (Mojgani et al., 2024; Rybchuk et al., 2023; Chattopadhyay & Hassanzadeh, 2023; Lai et al., 2024). Various frequency domain approaches have been proposed to mitigate this problem, including the scale factor approach (Cai et al., 2019; Liu et al., 2020; Jin et al., 2024; Li et al., 2023), which were applied in the multistage neural networks (Wang & Lai, 2024) by multiplying a large scale factor to the weight between the input and first hidden layer. The optimal value of the scale factor was found to be $\kappa = \pi f_d/\sqrt{V_{ar}}$, where f_d is the domain frequency of the target function and V_{ar} is the variance of first layer weights. Although this setting works effectively for fitting one-dimensional functions, it remains limited when approximating high-frequency function in higher dimensions. Here, to resolve the issue, we provide an advanced method that can further alleviate spectral bias and enhance the multistage neural network training, by initializing the neural network weights based on more straightforward spectral information from the target function. In more details, this spectrum-informed initialization uses the discrete Fourier transform of the target function or the residues to inform the initialization of the neural network, ensuring fast convergence across the spectral modes present. In doing so, the spectrum-informed multistage neural network (SI-MSNN) is able to approximate the target function down to $O(10^{-16})$. Moreover, the regression provided by the neural network is accurate across the Fourier spectrum, even for challenging multi-scale target functions.

In Section 2, we provide an overview of Fourier feature embeddings, the discrete Fourier transform, and neural tangent kernel theory. In Section 3, we introduce the spectrum-informed initialization of network weights, and its usage for multi-stage neural networks (MSNNs). In Section 4, we provide experimental results demonstrating the advantage of SI-MSNNs over the original MSNNs on error reduction and its ability to reach machine precision for complex regression problems. Lastly, in Section 5, we provide further discussions and future work on the spectrum-informed initialization for multistage neural networks.

2. Preliminaries

2.1. Discrete Fourier transform

The Fourier transform decomposes a function into the various frequencies and corresponding amplitudes that are present, writing the function in terms of a complex exponential or sinusoidal/cosinusoidal basis (Sneddon, 1995). The discrete Fourier transform (DFT) of a function with values provided at N equally spaced points can be computed using the fast Fourier transform (FFT) algorithm in $O(N \log N)$. The one-dimensional DFT of a function u, specified at N points x_n, is given by

$$\tilde{u}(k) = \frac{1}{N}\sum_{n=0}^{N-1} u(x_n)\exp(-ikx_n), \tag{1}$$

for which the inverse Fourier transform is

$$\tilde{u}(k) = \sum_{n=0}^{N-1} u(x_n)\exp(ikx_n), \tag{2}$$

where k is the wavenumber. Analogously, the two-dimensional DFT of a function specified on a grid with N_x values x_n and N_y values y_m is given by

$$\tilde{u}(\mathbf{k}) = \sum_{n=0}^{N_x}\sum_{m=0}^{N_y} \exp(-i(k_x x_n + k_y y_m)), \tag{3}$$

where $\mathbf{k} = (k_x, k_y)$ is the wavenumber vector.

2.2. Fourier feature networks

Spectral bias, which is also referred to as the frequency principle, is a well-known phenomenon of standard multi-layer perceptrons (MLPs), which tend to exhibit a bias towards fitting low-frequency information in target functions, and learn high-frequency information more slowly, or fail to converge on high frequency modes (Rahaman et al., 2019; Xu et al., 2022). To allow MLPs to learn high-frequencies, Fourier feature networks first pass input points x through a Fourier feature mapping $\gamma(x)$, before passing the result through the MLP (Rahimi & Recht, 2007; Tancik et al., 2020). The Fourier feature mapping $\gamma(x)$ can be alternatively considered as the first layer of the MLP, as a composition of a nonlinear, periodic activation function and a linear function with added bias weights.

2.2.1. Random Fourier features

Random Fourier features are a non-trainable input mapping

$$\gamma(x) = \cos(2\pi Bx + b) \tag{4}$$

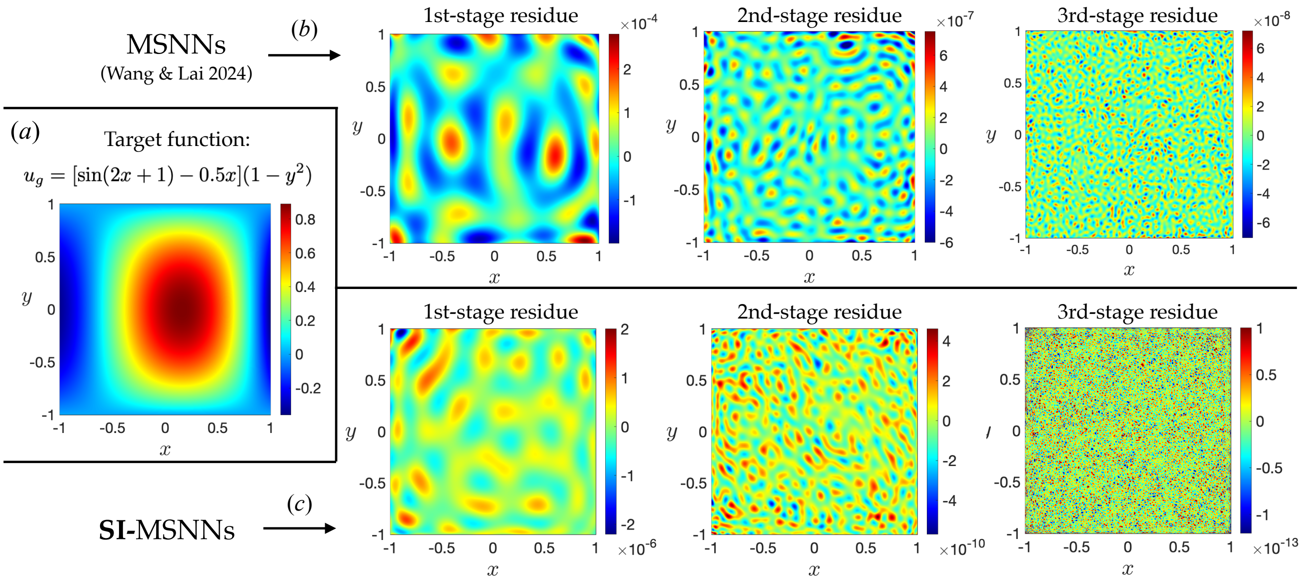

Figure 1. (a) Target function $u_g(x, y)$ for a 2-D regression problem. (b) Errors of neural networks with the target function u_g after different stages of training using the original multi-stage neural networks (MSNNs) where the scaling factor was used to mitigate the spectral biases of the network training. The 3rd-stage residue reaches $O(10^{-8})$. (c) Errors after different stages of training using spectrum-informed initialization for both weights and biases of the network, which reaches $O(10^{-13})$ by the 3rd stage of training.

where B and b are fixed random variables, and the cosinusoidal function is applied component-wise (Rahimi & Recht, 2007).

2.2.2. Fourier feature mapping

Fourier feature mapping extends this technique, using the positional encoding

$$\gamma(x) = [A\cos(2\pi Bx), A\sin(2\pi Bx)]^T, \quad (5)$$

where $B \in \mathbb{R}^{m\times d}$ are the weights for the feature mapping, which can be either fixed or trainable. Typically, these weights are initialized as $B \sim \mathcal{N}(0, \sigma^2)$, where σ is a hyperparameter that determines the span of the initialized weights and can be set to 1 as a default, or determined based on the problem (Tancik et al., 2020).

2.2.3. Cosinusoidal activation

Since cosine and sine differ only by a phase, which can be recovered using the trainable phase parameter b, it is comparable to use the cosinusoidal mapping

$$\gamma(x) = [A\cos(2\pi Bx + b)]^T, \quad (6)$$

where the weights B are initialized as in the previous section, and the biases b can be initialized to zero or following a normal distribution. Equivalently, the cosinusoidal feature mapping can be seen as a modified first layer of an MLP, where the activation function is $\sigma(x) = A\cos(2\pi x)$ rather than the typical choices $\sigma(x) = \tanh(x)$ or $\sigma(x) = \mathrm{ReLU}(x)$.

2.3. Neural tangent kernel

The neural tangent kernel (NTK) theory describes the evolution of neural networks by gradient descent during training. In the limit of an infinite-width neural network, the NTK can be used to approximate the result of training a neural network as the learning rate approaches zero. In fact, the neural network converges in the eigenvectors of the NTK at an exponential rate with respect to the corresponding eigenvalue (Jacot et al., 2018; Tancik et al., 2020). For a conventional MLP, the spectral bias can be viewed as an eigenvector bias, as the eigenvalues decay rapidly (Wang et al., 2021). When using a Fourier feature mapping, the principal eigenvectors of the NTK correspond to the embedded frequencies from B, and thus embedding the frequencies of the target function u into the neural network using Fourier feature mapping allows rapid convergence in those directions (Tancik et al., 2020).

3. Methods

3.1. Problem setup

We consider the supervised learning regression problem, where N_d data points $\{(\mathbf{x_i}, u(\mathbf{x_i}))\}$ are provided, and the task is to fit a neural network to the target function u. In the precision machine learning setting, we assume that the data is exact, in order to test the capacity of neural networks as universal function approximators.

A neural network with L hidden layers, given a feature embedding $\gamma(x)$, is a mapping $\mathcal{N} : \mathbb{R}^d \to \mathbb{R}$ which acts as $\mathcal{N}(x;\theta) = f_L \circ \sigma_L \circ \cdots\cdots \circ \sigma_1 \circ f_1 \circ \gamma(x)$, where $f_k(x) = W_k x + b_k$ is a linear function and $\sigma_k(x)$ is a nonlinear activation function, such as $\tanh$ or reLU.

3.2. Multi-stage neural networks (MSNN)

As mentioned above, multi-stage neural networks (Wang & Lai, 2024) are a novel approach dividing the training into stages, maintaining a high rate of convergence and allowing the regression for one-dimensional problems to reach machine precision (Wang & Lai, 2024).

Given a target function u, a typical neural network $u_0(x)$ with Xavier weight initialization is trained to approximate u/ϵ_0, with the input coordinates $\mathbf{x}$ normalized to within $[-1, 1]$. The normalizing factor ϵ_0 is taken to be the root mean square value $\epsilon_0 = \sqrt{\frac{1}{N_d}\sum_{i=0}^{N_d-1} u_i^2}$, where u/ϵ_0 has order of magnitude $O(1)$ matching the output of the network at initialization. The first stage residue is $e_1(x) = u(x) - u_0(x)$, and is typically a high-frequency function.

Then, the second-stage neural network is trained to approximate the normalized first stage residue e_1/ϵ_1 as a target function, where ϵ_1 is the root mean square value of the error e_1 on the provided data points. To effectively fit the high-frequency first stage residue given the dominant frequency f_d and the variance of weights in the first layer V_{ar}, the activation function of the first layer is taken to be the periodic, high-frequency $\sigma(x) = \cos(\kappa x)$, where its inputs are first multiplied by the optimal scale factor $\kappa = \pi f_d/\sqrt{V_{ar}}$ (Wang & Lai, 2024). The re-scaling of the input to enable learning of high-frequency functions has also been implemented as adaptive activation functions (Jagtap et al., 2020).

In general, the $(n+1)$th stage neural network u_{n+1} is the previous stage residue $e_n(x) = u(x) - \sum_{i=1}^{n} \varepsilon_n u_n(x)$, normalized by its root mean square value ϵ_n. The final result with s stages is $\sum_{n=1}^{s} \varepsilon_n u_n(x) \approx u(x)$. By tuning the neural network in each stage to capture the desired magnitudes and frequencies of the residue, the multistage neural network approach is able to reach machine precision $O(10^{-16})$ for the regression problem in one dimension.

3.3. L^p Norm

The typical loss function for training and evaluation in regression problems is the mean squared error (MSE), which is the squared L^2 norm, defined as $\|x\|_{L_2^2} = \sum_{n=1}^{N_d} x_n^2$. A more general loss function uses the L^p norm, where

$$\|x\|_{L_p} = \left(\sum_{n=1}^{N_d} |x_n|^p\right)^{1/p}, \tag{7}$$

which has a comparable order of magnitude to the root mean squared error (RMSE). As p increases, larger deviations are more heavily penalized. For our purposes, we use the L^p norm with $p = 10$, which prevents large spikes in the residue so the next stage of the multistage neural network can learn the residue more readily.

3.4. Spectrum-informed initialization

In two or more dimensions, the multistage neural network is not able to reach machine precision, and fails to capture the high frequencies present in the later stage residues. We propose a spectrum-informed initialization to replace the cosinusoidal mapping with a scale factor κ used in Wang & Lai (2024). Instead, here we tailor the neural network to the specific frequencies present in the dataset. Consider a dataset with input-output pairs $(\mathbf{x}_i, u_i = u(\mathbf{x}_i))$ for $\mathbf{x}_i \in [-1, 1]^2$ on a grid with N points in both dimensions, where the input domain is assumed to already be transformed to $[-1, 1]$, and $u_i \in \mathbb{R}$. The spectrum-informed initialization provides a targeted initialization of B and b for the input mapping layer $\gamma(x; B, b) = [A\cos(Bx + b)]^T$ of a neural network predicting the quantity $u(x)$ given the input coordinates x, where B and b are tunable parameters and A is a fixed hyperparameter. The remainder of the neural network layers can be initialized with the usual Xavier initialization scheme, which prevents gradient vanishing or explosion (Glorot & Bengio, 2010).

Algorithm 1 Spectrum-informed initialization

```
Compute the discrete Fourier transform $\tilde{u}$ of $u$, where $\tilde{u}(\mathbf{k}) = a(\mathbf{k})e^{i\theta(\mathbf{k})}$ in polar form. Let $\alpha(\mathbf{k}) = \frac{2}{N^2}a(\mathbf{k})$ if $k_y > 0$, and $\alpha(\mathbf{k}) = \frac{1}{N^2}a(\mathbf{k})$ if $k_y = 0$ be a normalized version of the magnitude.
Let $\mathbf{k}^{(j)}$ be the Fourier mode corresponding to the $j$th largest magnitude $|\tilde{u}(\mathbf{k}^{(j)})|$, considering only the modes such that $k_y \geq 0$. Take $\alpha^{(j)} = \alpha(\mathbf{k}^{(j)})$ and $\theta^{(j)} = \theta(\mathbf{k}^{(j)})$.
Given a desired first layer width $n_f$, let $B = [\mathbf{k}^{(1)}, \cdots, \mathbf{k}^{(n_f)}]^T$, $b = [\theta^{(1)}, \cdots, \theta^{(n_f)}]^T$ and $A = \left[\alpha^{(1)}, \cdots, \alpha^{(n_f)}\right]^T$
Initialize the first layer of the neural network, which has width $n_f$, with the Fourier feature mapping $\gamma(x; B, b) = [A\cos(Bx + b)]^T$. Initialize the other parameters based on the Xavier scheme.
```

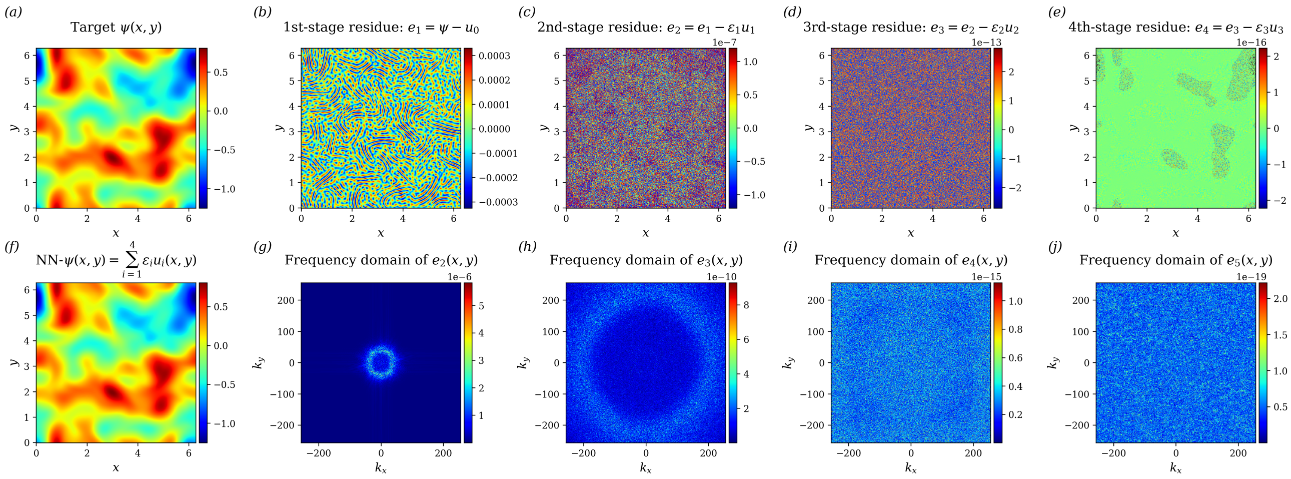

Figure 2. (a) Target function $\psi(x, y)$, a single time snapshot of the numerical solution of the stream function for the 2D incompressible Navier-Stokes equations (11) and (12) with Re = 2000. (b–e) The residues after each of the four stages of training, which are the target functions for the next stages. After four stages, the residue has approached machine precision $O(10^{-16})$. (g) Result of the Spectrum-Informed Multistage Neural Network (SI-MSNN) with four stages of training. (h–j) The spectral domain of each of the residues.

The two-dimensional DFT of u satisfies

$$u(x, y) = \frac{1}{N^2} \sum_{k_x, k_y} \tilde{u}(k_x, k_y) \exp(i(k_x x + k_y y)) \tag{8}$$

$$= \sum_{k_x, k_y} \frac{1}{N^2} a(\mathbf{k}) \exp(i(\mathbf{k} \cdot \mathbf{x} + \theta(\mathbf{k}))) \tag{9}$$

$$= \sum_{k_x, k_y \geq 0} \alpha(\mathbf{k}) \cos(\mathbf{k} \cdot \mathbf{x} + \theta(\mathbf{k})), \tag{10}$$

where $\alpha(\mathbf{k})$ are as defined in Algorithm 1. The last equality comes from the conjugate symmetry $\tilde{u}(\mathbf{k}) = \overline{\tilde{u}(-\mathbf{k})}$ of $\tilde{u}$, since u is a real-valued function. Using the conjugate symmetry avoids embedding redundant information.

Then, given the n_f largest modes, the Fourier feature mapping, which can be considered the first layer of the neural network, is $\gamma(x) = A \cos(Bx + b) = \sum_{i \leq n_f} \alpha^{(i)} \cos(\mathbf{k}^{(i)} \cdot \mathbf{x} + \theta^{(i)})$, which contains precisely the information from the top n_f modes of the Fourier transform of u. If $n_f = N^2/2$ and the neural network contains only one layer, which is the Fourier feature embedding, the single-layer neural network will exactly match the DFT representation of u, as in Equation 10.

Based on the neural tangent kernel (NTK) theory (Jacot et al., 2018), embedding these primary Fourier modes $\mathbf{k}$ allows the neural network to effectively learn the target function across the frequency spectrum.

4. Experiments

In this section, we experimentally demonstrate the ability of the spectrum-informed initialization to learn the target frequencies, and to train neural networks as approximators down to the limits of numerical precision.

4.1. Comparison with scale factor approach

In Figure 1, we compare the results of a three-stage neural net with the original scale factor approach and the spectrum-informed initialization, fitting a simple target function $u_g(x) = [\sin(2x + 1) - 0.5x](1 - y^2)$. For the scale factor approach, the Fourier feature embedding is of the form $\gamma(x) = \cos(\kappa w^{(0)} x + b)$, where $w^{(0)}$ are the first-layer weights following the Gaussian distribution $\mathcal{N}(0, V_{ar})$ with the variance V_{ar} set by the Xavier initialization method. The scale factor $\kappa = \pi f_d / \sqrt{V_{ar}}$ is determined based on the dominant frequency f_d of the target function. In contrast, the spectrum-informed initialization employs the Fourier feature embedding $\gamma(x) = A \cos(Bx + b)$ with A, B, and b being directly initialized by the spectral information of the target function, as described in Algorithm 1.

Experimentally, the scale factor approach yields a 3rd-stage residue of $O(10^{-8})$ (Figure 1*b*), whereas the spectrum-informed initialization allows the MSNN to fit the data up to $O(10^{-13})$ (Figure 1*c*), which is five orders of magnitudes more accurate. We note that the number of iterations used in each stage of the two experiments are the same.

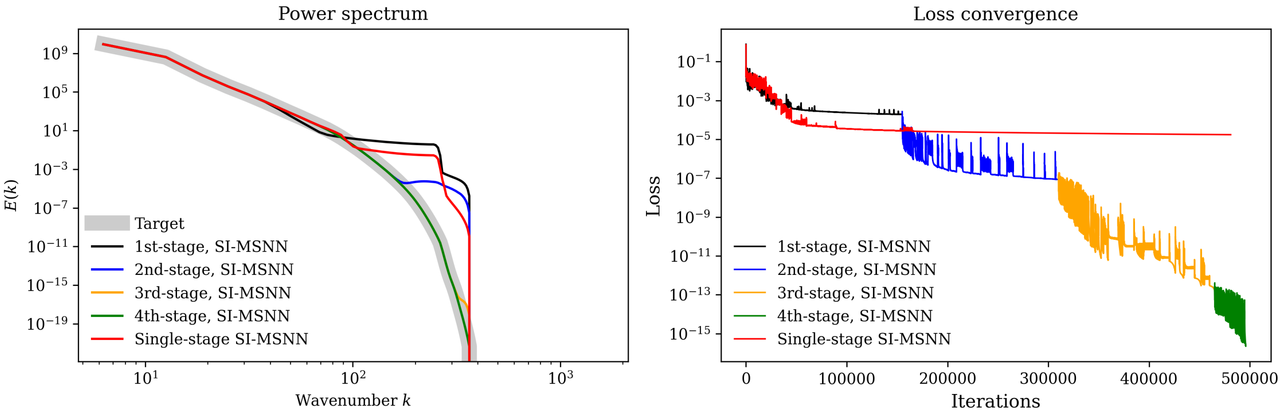

Figure 3. Left: A comparison of the power spectrum of the 2D Navier-Stokes example in Fig. 2 given by a single-stage SI-MSNN and a four-stage SI-MSNN, each with three hidden layers of width 30. The single-stage SI-MSNN has first layer width $n_f = 10000$. The first stage of the four-stage SI-MSNN has first layer width $n_f = 1359$, based on the number of primary Fourier modes present, and the remaining stages all have first layer width $n_f = 10000$. Right: The loss convergence of a single-stage SI-MSNN compared to a four-stage SI-MSNN, where training terminates when machine precision is reached.

4.2. 2D incompressible Navier-Stokes

As an example problem, we consider fitting a snapshot of two-dimensional homogeneous isotropic decaying turbulence (2D-DHIT), which is a broadly multi-scale problem. Random Fourier features have been shown to be effective for similar multi-scale turbulence problems (Mojgani et al., 2024). The Fourier coefficients of the target function decay exponentially with the wavenumber, where the high-frequency, small-scale information plays a large role in the behavior of the system. Taking the curl of the 2D incompressible Navier-Stokes equations yields the vorticity equation, which can be written in terms of the stream function ψ and the vorticity ω:

$$\frac{\partial \omega}{\partial t} + \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} = \frac{1}{\mathrm{Re}}\nabla^2 \omega \qquad (11)$$

and

$$\nabla^2 \psi = -\omega, \qquad (12)$$

which can model a variety of geophysical flows (Subel et al., 2023). Using a direct numerical simulation (DNS), on a $[0, 2\pi)^2$ grid with resolution $N = 512$ and Reynolds number $Re = 2000$, the 2D-DHIT equations are solved using a pseudo-spectral solver with third-order Runge-Kutta time stepping, with the timestep determined by the CFL condition. The initial condition is taken to be a random vorticity field with a broad banded energy spectrum (McWilliams, 1984).

We perform regression with the target function as a snapshot of the stream function $\psi(\mathbf{x}, t = 1.25)$. The coordinate inputs to the neural network are re-scaled to the domain $[-1, 1]$, and the spectrum-informed initialization is based on the re-scaled coordinates. The results are transformed back to the original coordinates from $[0, 2\pi)$. In terms of experimental setup, we employ a fully-connected neural network with the first layer as the spectrum-informed Fourier feature embedding with cosine as the activation function, as well as three hidden layers with 30 units in each layer using hyperbolic tangent activation functions. The first layer contains $n_f = 1359$ units for the first-stage neural network, and $n_f = 10000$ units for the second, third, and fourth-stage neural networks, based on the number of primary Fourier modes in the target function of each stage. The spectrum-informed initialization is used for the first Fourier feature layer, and the Xavier scheme is used to initialize the other parameters, as described in Algorithm 1. The neural networks are trained with full batch gradient descent using Adam optimizer with initial learning rate of 10^{-3} and incorporating learning rate annealing. The models are trained for 150000 iterations, with the fourth stage terminating when machine precision is reached.

In Figure 2, the SI-MSNN method is successfully able to reduce the error down to the machine precision of a 64-bit double float, $O(10^{-16})$, after four stages of training. This result experimentally validates the ability of SI-MSNNs to approximate a target function with arbitrarily high accuracy.

In Figure 3, the multistage setup is shown to be necessary. With a single stage of training, the loss convergence plateaus to a linear decay rate, and the neural network is not able to correct the residues relative to the target function. However, by incorporating multiple stages, with a neural network initialized based on the magnitude and spectrum of the residue, an exponential convergence rate can be maintained, allowing efficient learning.

Moreover, the spectrum-informed initialization allows the multistage neural network to accurately represent the function across the spectral domain. Here, the two-dimensional Fourier transform of the stream function ψ satisfies $\tilde{\psi}(k_x, k_y) = \sum_{n=0}^{N_x} \sum_{m=0}^{N_y} \psi(x_n, y_m) \exp(-i(k_x x_n + k_y y_m))$, where $\mathbf{k} = (k_x, k_y)$ is the wavenumber vector. Given the Fourier transform $\tilde{\psi}$, the angle-averaged power spectrum $E(k)$ is defined as

$$E(k) = \sum_{k-\Delta k \le |\mathbf{k}| \le k+\Delta k} |\tilde{\psi}(\mathbf{k})|^2, \tag{13}$$

for a given spectral band Δk (Kag et al., 2022). The spectral band is defined so that the wavenumber k falls into 200 bins.

In Figure 3, as the number of stages increases, the SI-MSNN is able to match the power spectrum $E(k)$ of the target function $\psi(x, y)$ up to increasing wavenumbers. Successfully fitting both the target function and its power spectrum is important for multi-scale scientific problems, and the capacity to do so is shown with this 2D Navier-Stokes example.

5. Discussions

We introduce a novel spectrum-informed initialization, allowing efficient training of neural networks for solving the regression problem to machine precision. By utilizing the spectral biases of neural networks, a spectrum-informed Fourier embedding of the input allows the neural network to learn in the spectral domain and converge rapidly. The spectrum-informed initialization for multistage neural networks (SI-MSNN) allows the neural network to fit target functions down to machine precision, even for multi-scale target functions, as validated experimentally. The results using the SI-MSNN demonstrate that this approach can achieve residues many orders of magnitude smaller than state-of-the-art approaches, down to machine precision.

In the future, we plan on applying the spectrum-informed initialization for scientific machine learning problems requiring precision as high as possible. In particular, we propose using the spectrum-informed initialization for multistage physics-informed neural networks as partial differential equation solvers (Wang & Lai, 2024; Raissi et al., 2019), where the boundary and initial conditions are the error-free data provided, along with a set of governing equations. Our work provides a promising approach towards precision machine learning.

Impact Statement

This paper presents work which aims to advance scientific machine learning, by providing an approach allowing neural networks to reach machine precision in scientific contexts where very high precision is necessary. Our work has broader impact in many scientific fields, with a goal of developing useful tools for machine learning for science, with potential applications in using neural networks as partial differential equation solvers, equation discovery, and climate science. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Cai, W., Li, X., and Liu, L. Phasednn-a parallel phase shift deep neural network for adaptive wideband learning. *arXiv preprint arXiv:1905.01389*, 2019.

Chattopadhyay, A. and Hassanzadeh, P. Long-term instabilities of deep learning-based digital twins of the climate system: The cause and a solution. *arXiv preprint arXiv:2304.07029*, 2023.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

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

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

Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.

Jin, G., Wong, J. C., Gupta, A., Li, S., and Ong, Y.-S. Fourier warm start for physics-informed neural networks. *Engineering Applications of Artificial Intelligence*, 132: 107887, 2024.

Jnini, A., Vella, F., and Zeinhofer, M. Gauss-newton natural gradient descent for physics-informed computational fluid dynamics. *arXiv preprint arXiv:2402.10680*, 2024.

Kag, V., Seshasayanan, K., and Gopinath, V. Physics-informed data based neural networks for two-dimensional turbulence. *Physics of Fluids*, 34(5), 2022.

Lai, C.-Y., Hassanzadeh, P., Sheshadri, A., Sonnewald, M., Ferrari, R., and Balaji, V. Machine learning for climate physics and simulations. *arXiv preprint arXiv:2404.13227*, 2024.

Li, S., Xia, Y., Liu, Y., and Liao, Q. A deep domain decomposition method based on fourier features. *Journal of Computational and Applied Mathematics*, 423:114963, 2023.

Liu, Z., Cai, W., and Xu, Z.-Q. J. Multi-scale deep neural network (mscalednn) for solving poisson-boltzmann equation in complex domains. *arXiv preprint arXiv:2007.11207*, 2020.

McWilliams, J. C. The emergence of isolated coherent vortices in turbulent flow. *Journal of Fluid Mechanics*, 146:21–43, 1984.

Michaud, E. J., Liu, Z., and Tegmark, M. Precision machine learning. *Entropy*, 25(1):175, 2023.

Mojgani, R., Chattopadhyay, A., and Hassanzadeh, P. Interpretable structural model error discovery from sparse assimilation increments using spectral bias-reduced neural networks: A quasi-geostrophic turbulence test case. *Journal of Advances in Modeling Earth Systems*, 16(3): e2023MS004033, 2024.

Müller, J. and Zeinhofer, M. Achieving high accuracy with pinns via energy natural gradients. *arXiv preprint arXiv:2302.13163*, 2023.

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

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. 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.

Rybchuk, A., Hassanaly, M., Hamilton, N., Doubrawa, P., Fulton, M. J., and Martínez-Tossas, L. A. Ensemble flow reconstruction in the atmospheric boundary layer from spatially limited measurements through latent diffusion models. *Physics of Fluids*, 35(12), 2023.

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

Sneddon, I. N. *Fourier transforms*. Courier Corporation, 1995.

Subel, A., Guan, Y., Chattopadhyay, A., and Hassanzadeh, P. Explaining the physics of transfer learning in data-driven turbulence modeling. *PNAS nexus*, 2(3):pgad015, 2023.

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

Udrescu, S.-M. and Tegmark, M. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Wang, S., Wang, H., and Perdikaris, P. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.

Wang, Y. and Lai, C.-Y. Multi-stage neural networks: Function approximator of machine precision. *Journal of Computational Physics*, pp. 112865, 2024.

Wang, Y., Lai, C.-Y., Gómez-Serrano, J., and Buckmaster, T. Asymptotic self-similar blow-up profile for three-dimensional axisymmetric euler equations using neural networks. *Physical Review Letters*, 130(24):244002, 2023.

Xu, Z.-Q. J., Zhang, Y., and Luo, T. Overview frequency principle/spectral bias in deep learning. *arXiv preprint arXiv:2201.07395*, 2022.