Spectral-Prior Guided Multistage Physics-Informed Neural Networks for Highly Accurate PDE Solutions

Yuzhen Li[a,b], Liang Li[a,b,*], Stéphane Lanteri[c], Bin Li[b]

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, 611731, Chengdu, P.R. China
[b] Key Laboratory of Large-Scale Electromagnetic Industrial Software, Ministry of Education, , 611731, Chengdu, P.R. China
[c] Université Côte d'Azur, Inria, CNRS, LJAD, Sophia Antipolis, France

arXiv:2508.17902v1 [math.NA] 25 Aug 2025

Abstract

Physics-Informed Neural Networks (PINNs) are becoming a popular method for solving PDEs, due to their mesh-free nature and their ability to handle high-dimensional problems where traditional numerical solvers often struggle. Despite their promise, the practical application of PINNs is still constrained by several factors, a primary one being their often-limited accuracy. This paper is dedicated to enhancing the accuracy of PINNs by introducing spectral-prior guided multistage strategy. We propose two methods: Spectrum-Informed Multistage Physics-Informed Neural Networks (SI-MSPINNs) and Multistage Physics-Informed Neural Networks with Spectrum Weighted Random Fourier Features (RFF-MSPINNs). The SI-MSPINNs integrate the core mechanism of Spectrum-Informed Multistage Neural Network (SI-MSNNs) and PINNs, in which we extract the Dominant Spectral Pattern (DSP) of residuals by the discrete Fourier transform. This DSP guides the network initialization to alleviate spectral bias, and gradually optimizes the resolution accuracy using a multistage strategy. The RFF-MSPINNs combines random Fourier features with spectral weighting methods, dynamically adjusting the frequency sampling distribution based on the residual power spectral density, allowing the network to prioritize learning high-energy physical modes. Through experimental verification of the Burgers equation and the Helmholtz equation, we show that both models significantly improve the accuracy of the original PINNs.

Keywords: Physics-Informed Neural Networks, Multistage Neural Networks, Spectrum information, Partial Differential Equations, Random Fourier Features

1. Introduction

In recent years, the rapid development of deep learning has brought new opportunities to scientific computing. In 2019, Raissi et al. proposed Physics-Informed Neural Networks (PINNs) [1], which embed physical laws into the neural network training process, improving the interpretability of models and solving PDEs and inverse problems without relying on large amounts of data. This mesh-free approach allows them to handle complex geometries and high-dimensional spaces with greater flexibility than traditional methods like the finite element method (FEM) [1, 2]. By combining data-driven methods with physical constraints [3, 4], PINNs provide powerful modeling capabilities, particularly suitable for solving high-dimensional nonlinear systems. Chen et al. [3] proposed a physics-informed machine learning framework for the reduced-order modeling of parametric PDEs, demonstrating its effectiveness through steady and unstable problems.

However, the widespread adoption of PINNs is still hindered by significant limitations, with sub-optimal accuracy being a primary concern [5, 6, 7]. This issue arises mainly from the complex and often problematic optimization dynamics of the training process [8]. The composite loss function, which combines terms for

*Corresponding author
Email addresses: liyuzhen0201@163.com (Yuzhen Li), plum_liliang@uestc.edu.cn, plum.liliang@gmail.com (Liang Li), stephane.lanteri@inria.fr (Stéphane Lanteri), libin@uestc.edu.cn (Bin Li)

PDE residuals, boundary, and initial conditions, frequently suffers from imbalanced gradients. Furthermore, the inherent spectral bias of neural networks makes them slow to learn high-frequency or multi-scale features of a solution, a critical weakness for many complex physical systems [9, 10, 11]. These optimization challenges make PINNs difficult to have high accuracy, and thus less robust and reliable than mature numerical methods like FEM or FVM for many applications [12, 13, 14, 15].

In order to improve the accuracy of the Neural Network (NN) based methods, Ng et al. [16] proposed the Multistage Neural Networks (MSNNs) method, which divides the neural network training process into multiple stages. Each stage learns the residual of the previous stage (i.e. the difference between the objective function and the output of the trained network) through a new neural network. And by introducing scale factors to adjust weights and other methods, the network settings are optimized for the high-frequency characteristics of residuals to alleviate the spectral bias [17] of the neural network and achieve a high-precision approximation of the objective function. However, the convergence efficiency decreases significantly in high-dimensional problems. The author applied it to regression problems and ordinary/partial differential equations to verify its effectiveness. Ng et al. [18] proposed an improved Spectrum-Informed Multistage Neural Network (SI-MSNN) method based on MSNN. This method further alleviates the spectral bias of the neural network through a spectral information guided initialization strategy, enabling the network to quickly converge to the spectral model of the objective function. By combining the strategy of learning the residuals of the previous stage in multi-stage training, a high-precision approximation of the two-dimensional regression problem was ultimately achieved, but it was not applied to the PINNs framework to solve the PDEs problem.

Based on the above considerations, this paper integrates the SI-MSNN method into the PINNs framework and establishes the Spectrum-Informed Multistage Physics-Informed Neural Networks (SI-MSPINNs). SI-MSPINNs utilizes a spectrum guided initialization strategy to enable the network to more accurately capture high-frequency features when learning physics equation residuals, thereby improving the accuracy and efficiency of solving complex physics problems. In addition, a spectral weighting method incorporating random Fourier features (RFF) was introduced to initialize MSNN, and an Multistage Physics-Informed Neural Networks with Spectrum Weighted Random Fourier Features (RFF-MSPINNs) model was established. By assigning spectral weights to random Fourier features, the accuracy of MSNN in handling 2D problems in PINNs has been improved, and the spectral bias problem in neural networks has also been solved.

The remainder of this paper is organized as follows. Section 2 briefly introduces the background and theoretical knowledge of related works such as PINNs and MSNNs. Section 3 introduces the framework and algorithm of SI-MSPINNs. Section 4 introduces the framework and algorithm of RFF-MSPINNs. Section 5 applies the two newly proposed models to the Burgers equation and the Helmholtz equation, respectively. Section 6 provides a summary of the entire text and offers prospects for future work.

2. Related Work

2.1. PINNs: Physics-Informed Neural Networks

PINNs is a PDEs solving paradigm that deeply integrates physical laws into deep learning frameworks. Its core idea is to achieve unified modeling of physical laws and data constraints through a dual loss collaborative mechanism. This method constructs an approximate solution function for neural networks and synchronously embeds control equation residuals and boundary/initial condition constraints in the loss function. And by using automatic differentiation techniques to calculate higher-order derivative terms, without relying on grid discretization in traditional numerical methods, the overall loss is minimized by optimizing network parameters, ultimately obtaining a solution that conforms to both data and physical laws.

Consider the following partial differential equation defined on a spatial domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$:

$$\begin{aligned} &\mathcal{L}\big(u(\mathbf{x})\big) = f(\mathbf{x}) && \mathbf{x} \in \Omega, \quad t \in [0,T], \\ &\mathcal{B}(u(\mathbf{x},t);\mu) = 0 && \mathbf{x} \in \partial\Omega, \quad t \in [0,T], \\ &\mathcal{I}(u(\mathbf{x},0)) = u_0(\mathbf{x};\mu) && \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\mathcal{L}$ is a (nonlinear) differential operator, $\mathcal{B}$ and $\mathcal{I}$ are operators for boundary condition and initial condition, respectively. The goal of PINNs is to approximate the solution of the physical equation using a neural network $P^\theta(\mathbf{x}, t)$ where θ represents the parameters of the neural network, including weights and bias vectors. A loss function is defined to learn the parameters θ by minimizing the value of the loss function. The loss function in PINNs is defined as a weighted sum of the residuals of the physical equation, the boundary conditions and the initial conditions [1]:

$$\begin{aligned}\mathcal{L}_{\text{PINN}}(P^\theta) = &\frac{\lambda_f}{|C_f|} \sum_{(\mathbf{x},t)\in C_f} \left\| \mathcal{L}\big(u(\mathbf{x})\big) + f(\mathbf{x}) \right\|_2^2 + \\ &\frac{\lambda_b}{|C_\partial|} \sum_{(\mathbf{x},t)\in C_\partial} \left\| \mathcal{B}(P^\theta)(\mathbf{x},t) \right\|_2^2 + \frac{\lambda_i}{|C_i|} \sum_{\mathbf{x}\in C_i} \left\| P^\theta(\mathbf{x},0) - u_0(\mathbf{x}) \right\|_2^2,\end{aligned} \tag{2}$$

where $C_f \subset \Omega \times [0,T]$, $C_\partial \subset \partial\Omega \times [0,T]$ and $C_i \subset \Omega$ represent the sampling spaces, $|C_f|$, $|C_\partial|$ and $|C_i|$ denote the number of points of insertion for the equation, boundary, and initial conditions, respectively, and λ_f, λ_b and λ_i represent the weights of the three loss terms. In this way, PINNs combine the data-driven approach with the physics-based approach to improve the accuracy and reliability of the neural network model.

2.2. MSNNs: Multistage Neural Networks

MSNNs effectively alleviate the spectral bias of traditional neural networks and significantly improve the accuracy of the approximation of functions by dividing the training process into multiple stages [19, 20], each stage using a new network to fit the residual of the previous stage. The key idea is to normalize the residual magnitude and adjust the scale factor based on the residual characteristics of different stages, to better initialize the network. Below, the processes of these two core mechanisms are briefly introduced, respectively.

2.2.1. Magnitude Normalization

When traditional neural networks fit residuals, if the residual magnitude is much smaller than the original data, conventional weight initialization methods such as Xavier are difficult to effectively capture small signals, resulting in limited accuracy in subsequent network fitting [16]. To address this issue, MSNN adopts the following strategy.

- Residual normalization: Calculate the root mean square value (RMS) of the residual $e_1(x)$ from the previous stage, defined as

$$\epsilon_1 = \text{RMS}(e_1(x)) = \sqrt{\frac{1}{N_d}\sum_{i=1}^{N_d}\left[e_1(x^{(i)})\right]^2} = \sqrt{\frac{1}{N_d}\sum_{i=1}^{N_d}\left[u_g^{(i)} - u_0(x^{(i)})\right]^2}, \tag{3}$$

  where u_0 represents the neural network trained in the first stage on data points N_d and u_g represents the objective function. Then normalize the residual to $e_1(x)/\epsilon_1$ as the training target for the second stage network.

- Scale integration: The output of the second stage network $u_1(x)$ must be multiplied by a normalization coefficient of ϵ_1, and then combined with the previous stage network $u_0(x)$, which is $u_c^{(1)}(x) = u_0(x) + \epsilon_1 u_1(x)$. This process can be recursively extended to more stages, and the final model is $u_c^n(x) = \sum_{j=0}^{n} \epsilon_j u_j(x)$, where ϵ_j is the magnitude coefficient of the residuals in each stage.

2.2.2. Frequency Adaptation

Traditional neural networks suffer from spectral bias, making it difficult to fit high-frequency features. In multistage training, the high-frequency components of residuals are often more significant. Therefore, for high-frequency residuals, it is necessary to increase the weight scale from the input layer to the first hidden layer. MSNN accelerates the convergence of weights to the optimal value required for high frequencies

(approximately $O(2\pi f_d)$, where f_d is the residual dominant frequency) by introducing scale factor κ and multiplying the weights of the layer by κ. The schematic diagram is shown in Figure 1, considering a shallow neural network with a single input, a single output, and a hidden layer, where $w_i^{(0)}$ represents the weight between the input layer and the hidden layer, and $w_i^{(1)}$ corresponds to the weight between the hidden layer and the output layer, and σ represents the activation function. Multiply the weights $w_i^{(0)}$ between the input layer and the first hidden layer by the scale factor κ.

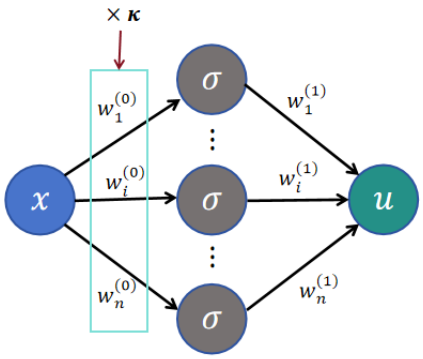


Figure 1: Schematic diagram of scale factor κ applied to a single hidden layer neural network.

2.3. SI-MSNNs: Spectrum-Informed Multistage Neural Networks [18]

SI-MSNNs is an improved model for high-precision function approximation problems. This method aims to solve the problem of difficulty in capturing high-frequency features and limited accuracy of MSNNs in high-dimensional scenes. By integrating spectral information of the objective function to optimize network initialization, the accuracy limit of high-dimensional problems is significantly improved. The core innovation of SI-MSNNs lies in the spectrum-informed initialization strategy, which extracts the dominant frequency features of the objective function based on discrete Fourier transform (DFT) and embeds them into the first layer parameters of the network. The specific steps are as follows:

1. Calculate the discrete Fourier transform $\tilde{u}(k)$ of the objective function u or residual, and obtain its polar coordinate form $\tilde{u}(k) = a(k)e^{i\theta(k)}$, where $a(k)$ is the amplitude and $\theta(k)$ is the phase.
2. Select the top n_f Fourier modes with the largest amplitude ($k^{(j)}$), and calculate the normalized amplitude $a^{(j)}$ and the corresponding phase $\theta^{(j)}$. For two-dimensional problems, use the conjugate symmetry of Fourier transform to screen patterns with $k_y > 0$ to avoid redundancy.
3. Map the selected patterns to the first layer of the neural network, the Fourier feature embedding layer, which has an activation function of $\gamma(x) = A\cos(Bx + b)$ and directly encodes the main spectral information of the objective function, where the weight matrix $B = [k^{(1)}, \cdots, k^{(n_f)}]^{\mathrm{T}}$ corresponds to the dominant frequency, the bias vector $b = [\theta^{(1)}, \cdots, \theta^{(n_f)}]^{\mathrm{T}}$ corresponds to the phase, and the amplitude parameter $A = [\alpha^{(1)}, \cdots, \alpha^{(n_f)}]^{\mathrm{T}}$ corresponds to the feature strength.

In summary, SI-MSNNs provide an efficient framework for high-precision function approximation and validate their effectiveness in multiscale scientific problems.

3. Spectrum-Informed Multistage Physics-Informed Neural Network

This section integrates the core ideas of SI-MSNNs with the PINNs framework to propose the SI-MSPINNs model, which solves the accuracy bottleneck of traditional PINNs in solving high-frequency PDEs. This model inherits the residual learning mechanism of MSNN and uses the SI-MSNNs method to extract the spectral features of residuals through discrete Fourier transform, guiding the initialization of network weights

and effectively alleviating spectral bias. At the same time, the physical constraint core of PINNs is retained, and the residual of partial differential equations is incorporated into the loss function to achieve collaborative optimization of data and physical laws.

Traditional PINN simultaneously fits data and physical equation residuals through a single-stage network, which is susceptible to spectral bias limitations and difficult to capture high-frequency physical features. SI-MSPINNs borrows the multistage training logic of SI-MSNN and decomposes PDE solving into a progressive residual learning process. Given PDE:

$$\mathcal{L}\big(u(\mathbf{x})\big) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{4}$$

and boundary condition:

$$\mathcal{B}\big(u(\mathbf{x})\big) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \tag{5}$$

First, train the basic PINN network u_0, where the loss function includes data fitting loss (initial/boundary conditions) and PDE residual loss, consistent with traditional PINN,

$$\mathcal{L}_{\text{PINN}} = \|\mathcal{L}(u_0) - f\|^2 + \|\mathcal{B}(u_0) - g\|^2. \tag{6}$$

Then, calculate initial residual $r_0 = \mathcal{L}(u_0) - f$ Stage $n \geq 1$. Specifically, perform DFT on residual r_{n-1} to extract the first n_f main modes (amplitude a_j and corresponding phase θ_j). Then build a spectral embedding layer:

$$\gamma(\mathbf{x}) = [\alpha_j \cos(\mathbf{k}_j \cdot \mathbf{x} + \theta_j)]^{\mathrm{T}}, \quad j = 1, \ldots, n_f. \tag{7}$$

Finally, calculate the RMS value ϵ_{n-1} of the residual and train u_n to fit the normalized residual r_{n-1}/ϵ_{n-1} to $\min \left\| \mathcal{L}(u_n) - \frac{r_{n-1}}{\|\epsilon_{n-1}\|} \right\|^2$. Finally update total solution $u_{\text{total}} = u_{\text{total}} + \epsilon_{n-1} u_{\theta_n}$ and residual $\epsilon_k = \mathcal{L}(u_{\text{total}}) - f$. The algorithm and flowchart of SI-MSPINNs are given by Algorithm 1 and Figure **??** respectively.

Algorithm 1 Training Scheme for SI-MSPINNs

Input: PDE operator $\mathcal{L}[\cdot]$, training data $\{(x_i, f(x_i))\}_{i=1}^{N_d}$

1: Train the 0-th stage PINN network u_0 with loss $\mathcal{L}_{\text{PINN}}(u_0)$, yielding the initial solution u_0.
2: Compute the initial residual $r_0 = u - u_0$ and the normalized coefficient $\epsilon_0 = \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} r_{0,i}^2}$.
3: **while** $n \leq s$ **do**
4: **Spectrum Extraction:** Perform DFT on r_{n-1} to obtain the spectrum $\tilde{r}_{n-1}(k) = a(k) e^{i\theta(k)}$, and select the top n_f dominant spectral modes $\{k^{(j)}, \alpha^{(j)}, \theta^{(j)}\}_{j=1}^{n_f}$.
5: **Network Initialization:** Initialize the first-layer weights B, biases b, and amplitudes A of the n-th stage network u_n using $\{k^{(j)}, \alpha^{(j)}, \theta^{(j)}\}$; other layers are initialized via Xavier initialization.
6: **Train** u_n**:** Minimize the loss $\mathcal{L}_{\text{PINN}}(u_0)$ to obtain u_n.
7: **Update Residual:** $r_n(x) = \mathcal{L}[u_n](x) - f(x)$, and compute $\epsilon_n = \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} r_{n,i}^2}$.
8: Set $n = n + 1$.
9: **end while**

Output: The final solution: $u_s = \sum_{i=0}^{s} \epsilon_i u_i$.

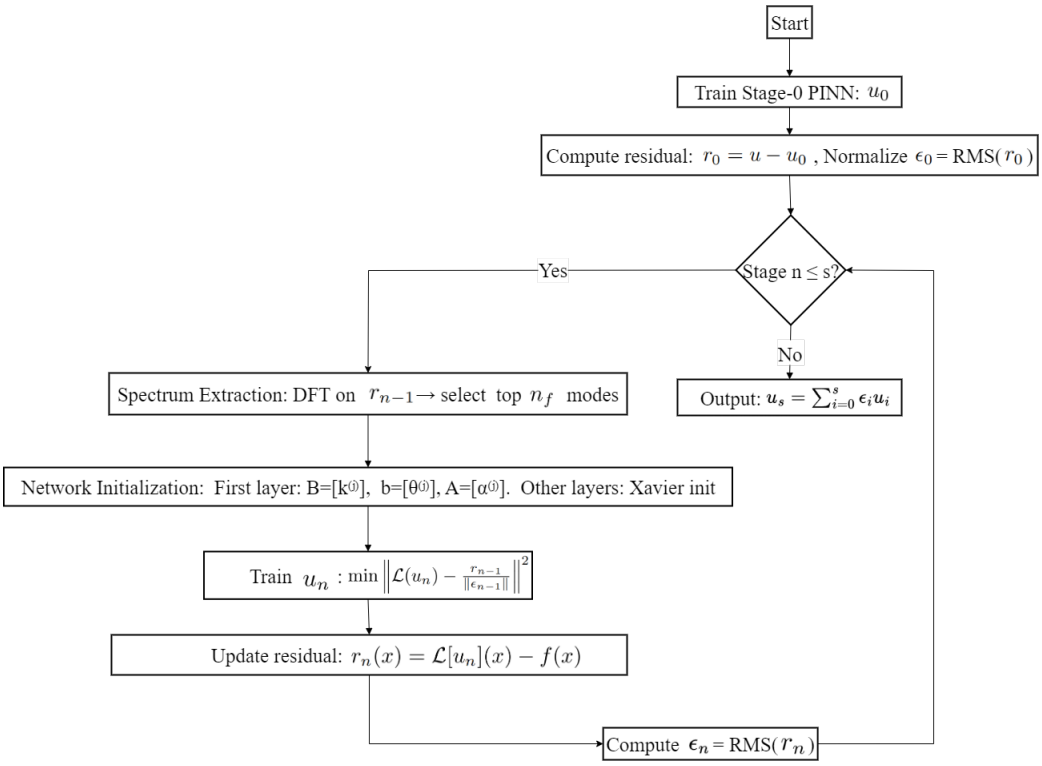


Figure 2: The flowchart of the SI-MSPINNs.

The key idea of the proposed SI-MSPINNs method in this section is to integrate spectrum information initialization and multistage physical constraints.

4. Multistage Physics-Informed Neural Network with Spectrum Weighted Random Fourier Features

RFF-MSPINNs integrates the spectral weighting method of random Fourier features (RFF) with MSNNs into the PINNs framework, significantly improving the accuracy and efficiency of PINNs. RFF-MSPINNs utilizes the power spectral density (PSD) of objective function residuals to guide RFF frequency sampling, enabling the neural network to prioritize learning the energy dominated physical modes and break through the bottleneck of high-frequency learning.

RFF is a technique that constructs high-dimensional feature maps using sine and cosine functions to approximate the kernel functions by randomly sampling frequency vectors and phases [21, 22]. Its core idea is to map the low dimensional inputs to high-dimensional space, enhance the model's ability to learn high-frequency features, and alleviate the spectral bias of neural networks. RFF uses the Equation 8 for random frequency sampling to approximate shift invariant kernels,

$$\phi(\mathbf{x}) = \sqrt{\frac{2}{m}\left[\cos\left(\boldsymbol{\omega}_1^T\mathbf{x} + b_1\right), \ldots, \cos\left(\boldsymbol{\omega}_m^T\mathbf{x} + b_m\right)\right]^T} \tag{8}$$

where $\boldsymbol{\omega}_j \sim \mathcal{N}(0, \sigma^2)$, $b_j \sim \mathcal{U}[0, 2\pi]$. Insufficient high-frequency sampling in traditional RFF may result in spectral bias.

In order to address this, PSD [23, 24, 25] is introduced, which is a physical quantity that describes the energy distribution of a signal or function in the frequency domain. It quantitatively characterizes the contribution of different frequency components to the total energy of the signal or function. The larger the

PSD value at a certain frequency, the higher the proportion of that frequency component in the signal or function [26]. It is a key indicator for analyzing the multiscale and spectral characteristics of signals.

In the proposed RFF-MSPINNs, we adopt a spectrum weighted RFF mechanism, whose core idea is to adjust the sampling distribution based on the residual PSD and prioritize the energy dominant frequency. The key process consists of the following four steps:

1. Firstly, perform DFT on the residual function $r_n(x)$, that is, $\tilde{r}_n(k) = \mathcal{F}\{r_n(x)\}$. And calculate the residual PSD $P(k) = |\tilde{r}_n(k)|^2$.
2. Then use Equation 9 to normalize PSD to a probability distribution, in order to construct a frequency sampling distribution
$$p(k) = \frac{P(k)}{\sum_k P(k)}. \tag{9}$$
3. Next, in order to utilize the spectral information of the residuals, we change the frequency sampling distribution to be proportional to the PSD of the residuals. As shown in Equation 10, extract m frequencies from $p(k)$.
$$k_j \sim p(k),\ j = 1, 2, \ldots, m \tag{10}$$
4. Finally, initialize the RFF layer by constructing frequency matrix $\mathbf{B} \in \mathbb{R}^{m \times d}$:
$$\mathbf{B} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix}. \tag{11}$$
Randomly generate phase shifts $b_j \sim \mathcal{U}[0, 2\pi]$, and finally defining the feature mapping:
$$\gamma(x) = \cos(2\pi \mathbf{B}x + \mathbf{b}). \tag{12}$$

This RFF module enhances the network's ability to learn high-frequency components by using a frequency sampling distribution proportional to the residual PSD. By sampling frequency based on power spectral density, we ensure that more features are allocated on important frequencies (high-energy frequencies) and fewer features are allocated on low-energy frequencies. In this way, the neural network has features that match the frequency spectrum of the objective function during initialization, allowing it to learn the main frequency components of the objective function faster, especially those with higher energy. The algorithm and flowchart of SI-MSPINNs are given by Algorithm 2 and Figure 3 respectively.

Algorithm 2 Training Scheme for RFF-MSPINNs

Input: PDE operator $\mathcal{L}[\cdot]$, training data $\{(x_i, f(x_i))\}_{i=1}^{N_d}$

Initialization
  Train the 0-th stage PINN network u_0 with loss $\mathcal{L}_{\text{PINN}}(u_0)$, yielding the initial solution u_0.
  Compute the initial residual $r_0 = u - u_0$ and the normalized coefficient $\epsilon_0 = \sqrt{\frac{1}{N_d}\sum_{i=1}^{N_d} r_{0,i}^2}$.
while $n \leq s$ **do**
  Compute the PSD $P_{n-1}(k)$ of $r_{n-1}(x)$ via DFT.
  Sample RFF frequencies $\{\mathbf{b}_i\}_{i=1}^{M}$ according to $p(k) \propto P_{n-1}(k)$, and phases $b_i \sim \mathcal{U}(0, 2\pi)$.
  Construct network $\mathcal{N}_n$ containing the RFF layer $\gamma(x)$ and L hidden layers.
  Normalize the residual: $\hat{r}_{n-1}(x) = \frac{r_{n-1}(x)}{\epsilon_{n-1}}$, where $\epsilon_{n-1} = \text{RMS}(r_{n-1})$.
  Train $\mathcal{L}_n$ by minimizing the PINN loss $\mathcal{L}_{\text{PINN}}(u_n)$.
  Update the solution: $u_n(x) = u_{n-1}(x) + \epsilon_{n-1} \cdot \mathcal{L}_n(x)$.
  Compute the new residual: $r_n(x) = \mathcal{L}[u_n](x) - f(x)$.
  Set $n = n + 1$.
end while

Output: Final solution $u_s(x)$

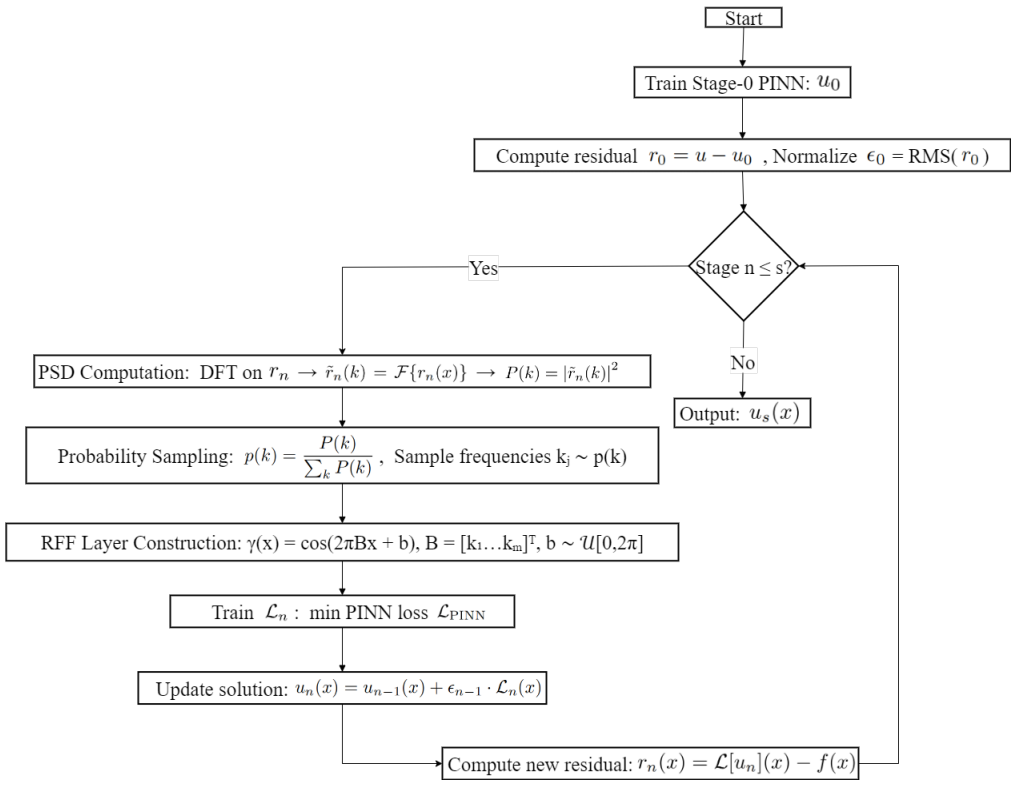


Figure 3: The flowchart of RFF-MSPINNs.

The SI-MSNN directly select the first n_f maximum amplitude frequencies and explicitly sets their amplitude and phase, which is equivalent to constructing a deterministic initialization using these frequency components. While, the RFF-MSPINNs select frequencies through probability sampling, and the probability of each frequency being selected is proportional to its energy. The initialization constructed in this way contains randomness, but statistically tends to favor important frequencies. It should be noted that RFF-MSPINNs guides the network to focus on important frequencies while maintaining randomness, which is a method that falls between completely deterministic (SI-MSNN) and completely random (traditional RFF). It may be more scalable in high-dimensional problems because it does not require sorting of all frequencies (SI-MSNN requires sorting) but only sampling based on probability distribution.

5. Experiments

This section applies the SI-MSPINNs and RFF-MSPINNs proposed above to the Burgers equation and the Helmholtz equation, respectively, to verify the performance of SI-MSPINNs and RFF-MSPINNs. All experiments are implemented using Torch 2.5.1 on a workstation with an Intel i9-10900 CPU and an NVIDIA Quadro RTX 5000 GPU. Codes and the corresponding data are available on GitHub [1].

5.1. Burgers Equation

Consider the Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2}, \quad (x,t) \in [-1,1] \times [0,1], \tag{13}$$

[1] available at https://github.com/liyuzhen0201/MPINN

with Dirichlet boundary conditions and initial conditions $u(-1,t) = u(1,t) = 0$ and $u(x,0) = -\sin(\pi x)$.

Set the number of training residual point samples within the domain to 2540, the number of boundary sampling points to 80, and the number of initial condition residual points. We use a three-stage architecture. The three stages of this case all use fully connected neural networks with a depth of 4 (equivalent to 3 hidden layers) and a width of 20. Adam+L-BFGS is used as the optimizer for each of the three stages.

5.1.1. Results of the SI-MSPINNs

The results of SI-MSPINNs solving the Burgers equation are shown in Figures 4.

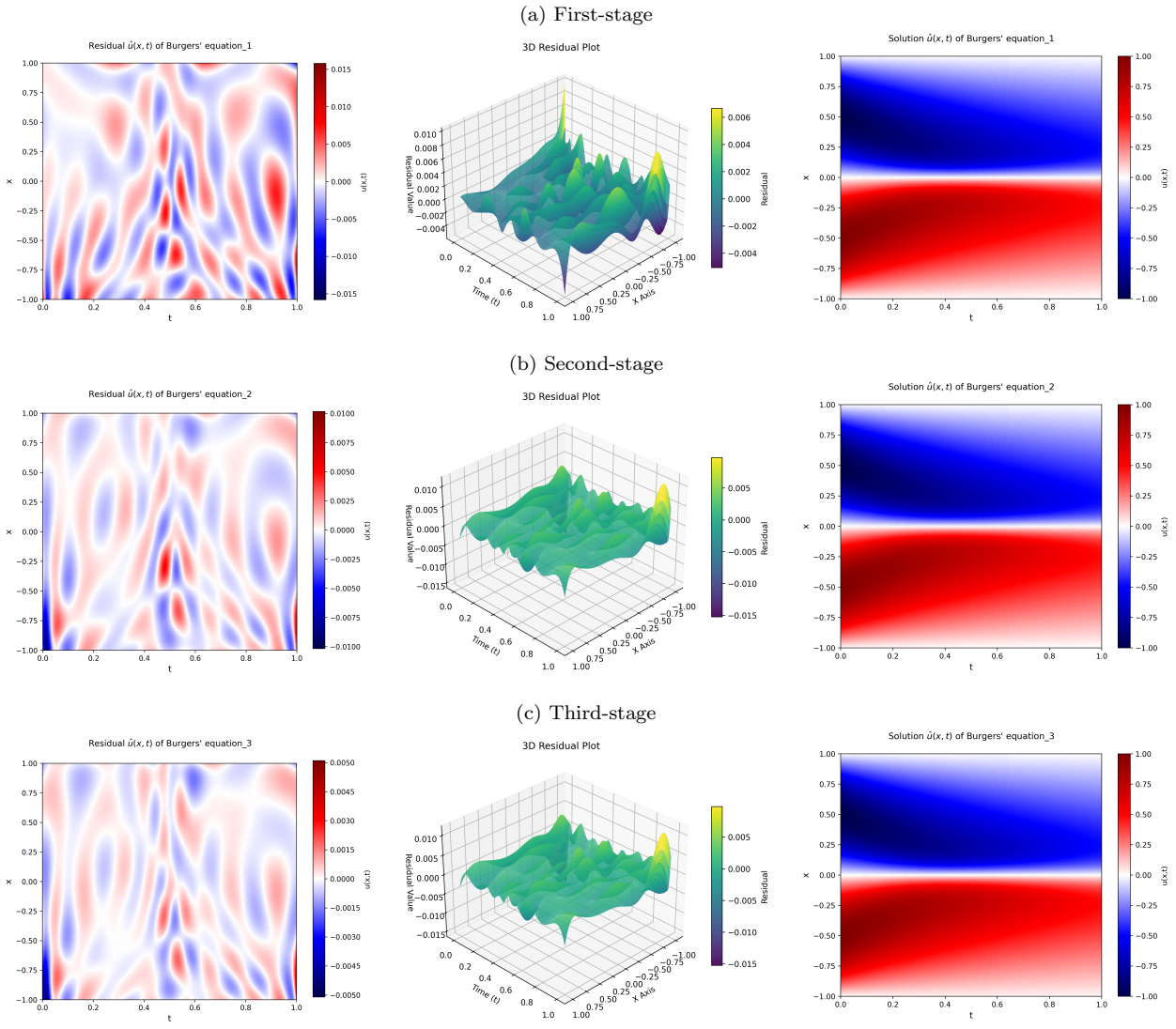


Figure 4: SI-MSPINNs for Burgers equation. (The left and middle figures show the two-dimensional and three-dimensional residual results of each stage, while the right figure shows the solution of the Burgers equation.)

5.1.2. Results of the RFF-MSPINNs

The results of RFF-MSPINNs solving the Burgers equation are shown in Figures 5.

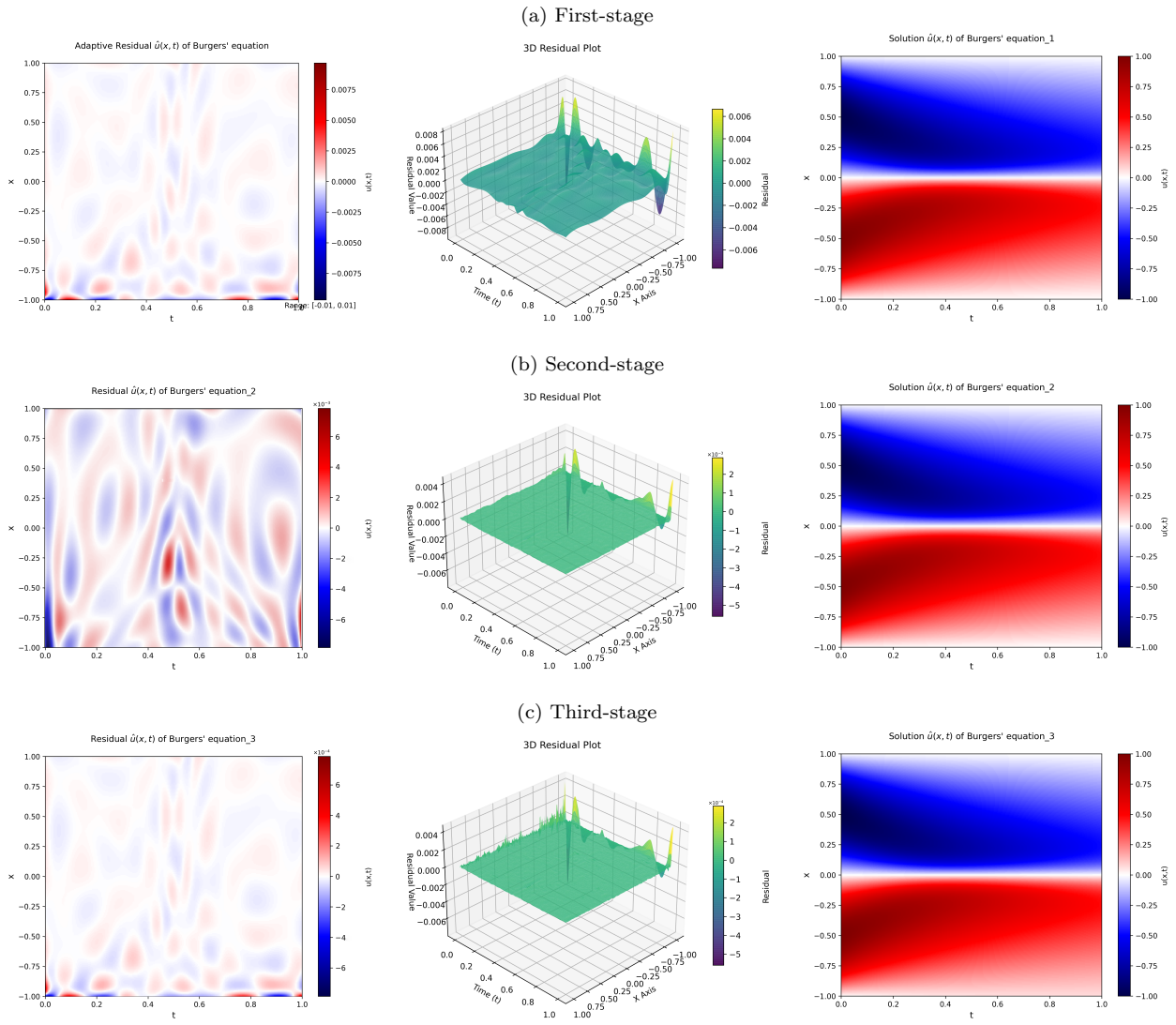
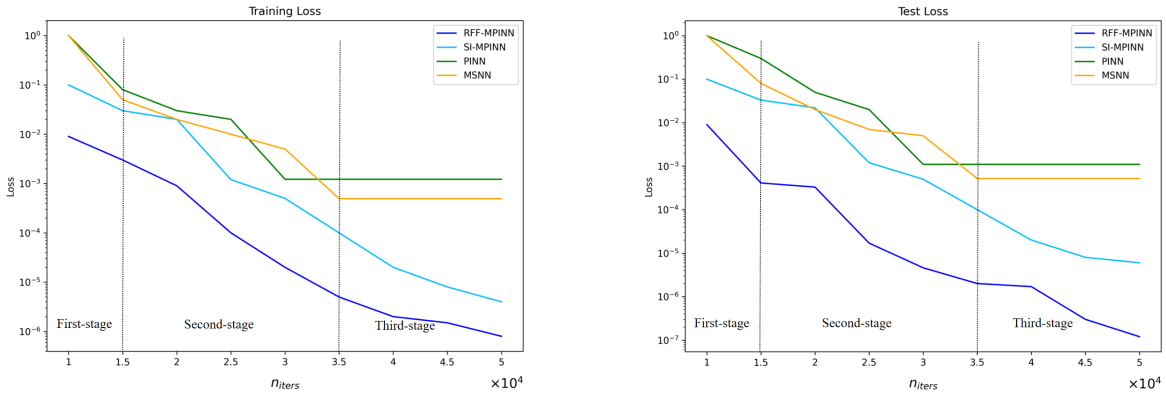


Figure 5: RFF-MSPINNs for Burgers equation. (The left and middle figures show the two-dimensional and three-dimensional residual results of each of the three stages, while the right figure shows the solution of the Burgers equation.)

From Figures 4 and 5, we can observe that both the RFF-MSPINNs and SI-MSPINNs can improve accuracy through multistage residual learning, and enhance high-frequency capture capability by guiding multistage PINNs through spectral priors. In addition, RFF-MSPINNs does not directly use frequency components to construct a deterministic initialization in the first layer of the neural network, but uses frequency to guide the training of the neural network. Therefore, the final result of the third stage of RFF-MSPINNs has higher accuracy than SI-MSPINNs by an order of magnitude.

As shown in Figures 6a and 6b, the loss values of traditional PINN and MSNN are 1.22×10^{-3} and 4.94×10^{-4}, respectively. Meanwhile, the training loss values of SI-MSPINNs and RFF-MSPINNs in the third stage can reach 6.91×10^{-6} and 7.34×10^{-7}, respectively. The three stages of training for SI-MSPINNs and RFF-MSPINNs took a total of 1326 seconds and 711 seconds, respectively.



(a) Training loss figure for different methods.

(b) Test loss figure for different methods.

Figure 6: Comparison figure of training loss and test loss for different methods.

The RFF-MSPINNs guides the network to focus on important frequencies while maintaining randomness, and is a method that falls between complete certainty and complete randomness. It does not require sorting all frequencies, but only sampling based on probability distribution. So the convergence speed of RFF-MSPINNs is faster.

5.2. Helmholtz equation

Consider the electromagnetic scattering problem of a dielectric disk described by the two-dimensional Helmholtz equation, defined as follows:

$$\begin{cases} \frac{\partial^2 E_{rz}}{\partial x^2} + \frac{\partial^2 E_{rz}}{\partial y^2} + \omega^2 \mu \epsilon E_{rz} = 0 \\ \frac{\partial^2 E_{iz}}{\partial x^2} + \frac{\partial^2 E_{iz}}{\partial y^2} + \omega^2 \mu \epsilon E_{iz} = 0 \end{cases}, \tag{14}$$

where E_{rz} and E_{iz} represent the real and imaginary parts of the electric field intensity E in the z-direction, respectively. ϵ and μ are the permittivity and permeability of the material, satisfying $\epsilon = \epsilon_0 \epsilon_r$ and $\mu = \mu_0 \mu_r$, where ϵ_0 and μ_0 are the permittivity and permeability of vacuum, ϵ_r and μ_r are the relative permittivity and relative permeability of the material.

Since this scattering problem is defined in open domain, we introduce an artificial boundary to truncate it. On this boundary, a first-order absorbing boundary condition is imposed:

$$\begin{cases} \mathbf{n} \times \mathbf{E}_r - \sqrt{\frac{\mu}{\epsilon}} \mathbf{n} \times (\mathbf{H}_r \times \mathbf{n}) = \mathbf{n} \times \mathbf{E}_{r,inc} - \sqrt{\frac{\mu}{\epsilon}} \mathbf{n} \times (\mathbf{H}_{r,inc} \times \mathbf{n}) \\ \mathbf{n} \times \mathbf{E}_i - \sqrt{\frac{\mu}{\epsilon}} \mathbf{n} \times (\mathbf{H}_i \times \mathbf{n}) = \mathbf{n} \times \mathbf{E}_{i,\text{ inc}} - \sqrt{\frac{\mu}{\epsilon}} \mathbf{n} \times (\mathbf{H}_{i,inc} \times \mathbf{n}) \end{cases}, \tag{15}$$

where $\mathbf{n}$ is the unit outward normal vector on the truncated boundary, $\mathbf{H}_r$ and $\mathbf{H}_i$ represent the real and imaginary parts of the magnetic field $\mathbf{H}$, $\mathbf{E}^{\mathbf{inc}}$ and $\mathbf{H}^{\mathbf{inc}}$ represent the incident electric and magnetic fields, respectively.

One can obtain the analytical solution of this problem [27]:

$$\begin{cases} u(r,\theta) = \sum_{n=-\infty}^{+\infty} i^n \left[J_n(Kr) + \alpha_n H_n^{(1)}(Kr) \right] e^{in\theta} & \text{for } r \le R \\ u(r,\theta) = \sum_{n=-\infty}^{+\infty} i^n \, \beta_n \, J_n\big(Kr\big) \, e^{in\theta} & \text{for } r \ge R \end{cases}, \tag{16}$$

where α_n and β_n have the following forms:

$$\alpha_n = \frac{\mu \, K_i \, J_n'(K_i R) \, J_n(KR) \; - \; K \, J_n(K_i R) \, J_n'(KR)}{K \, H_n^{(1)'}(KR) \, J_n(K_i R) \; - \; \mu \, K_i \, J_n'(K_i R) \, H_n^{(1)}(KR)}, \tag{17}$$

$$\beta_n = \frac{K\,H_n^{(1)'}(KR)\,J_n(KR)\;-\;K\,H_n^{(1)}(KR)\,J_n'(KR)}{K\,H_n^{(1)'}(KR)\,J_n(K_iR)\;-\;\mu\,K_i\,J_n'(K_iR)\,H_n^{(1)}(KR)}, \tag{18}$$

where R is the radius of the disk, $H_n^{(1)}$ and $H_n^{(1)'}$ are the first-kind Hankel function and its derivative, J_n and J_n' are the first-kind Bessel function and its derivative. This analytical solution is used as the ground truth for evaluating the accuracy of the methods.

Using a plane wave as the excitation source $H_x^{\text{inc}}(x,y) = 0, H_y^{\text{inc}}(x,y) = \cos\big(kx\big), E_z^{\text{inc}}(x,y) = -\cos\big(kx\big)$ with a frequency $f = 300\text{MHz}$, where $k = \frac{\omega}{c}$ is the wavenumber, c is the speed of light in vacuum, $\omega = 2\pi f$ is the angular frequency, and f is the wave frequency. The computational domain is $\Omega = [-1,1]^2$, and the disk radius is 0.25, as shown in Figure 7. This equation is difficult to solve for PINNs method, because we could have discontinuity of the boundary of the disk.

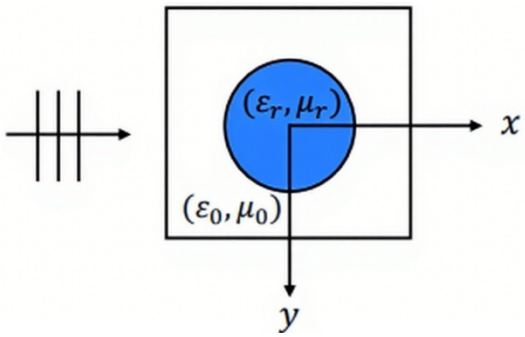


Figure 7: Scattering of plane waves by a dielectric disk

Set the number of training residual point samples within the domain to 2540, and the number of boundary sampling points to 80. We still use a three-stage architecture. All the three networks use fully connected neural networks with a depth of 4 (equivalent to 3 hidden layers) and a width of 20, and the optimizer is Adam+L-BFGS for all the three stages, the same as those in the previous example.

5.2.1. Results of the SI-MSPINNs

When $\epsilon = 1$, the results of SI-MSPINNs are shown in Figure 8.

5.2.2. Results of the RFF-MSPINNs

In Figure 9 we show the results of RFF-MSPINNs when $\epsilon = 1.5$.

In Table 1, we compare the L_2 error of PINN, MSNN, SI-MSPINNs, and RFF-MSPINNs. We can see that both SI-MSPINNs and RFF-MSPINNs effectively improve the accuracy of the original PINNs as well as MSNN. On can also observe that RFF-MSPINNs has higher accuracy, because the frequency sampling of RFF-MSPINNs is proportional to the PSD of the objective function, prioritizing the coverage of critical frequencies. This ensures that more features are allocated to important frequencies and fewer to low-energy frequencies. In this way, the neural network is initialized with features that match the spectrum of the objective function, enabling it to learn the main frequency components of the objective function more quickly, especially the high-frequency components with higher energy. Furthermore, through multistage training, residuals are gradually refined, with each stage initialized based on the spectral information of residuals from the previous stage. This collaborative design allows RFF-MSPINNs to more effectively capture both low-frequency and high-frequency features, address interface discontinuity problems, and thus achieve high accuracy.

6. Conclusion & Future Work

We propose the SI-MSPINNs and RFF-MSPINNs for solving PDEs with high accuracy.

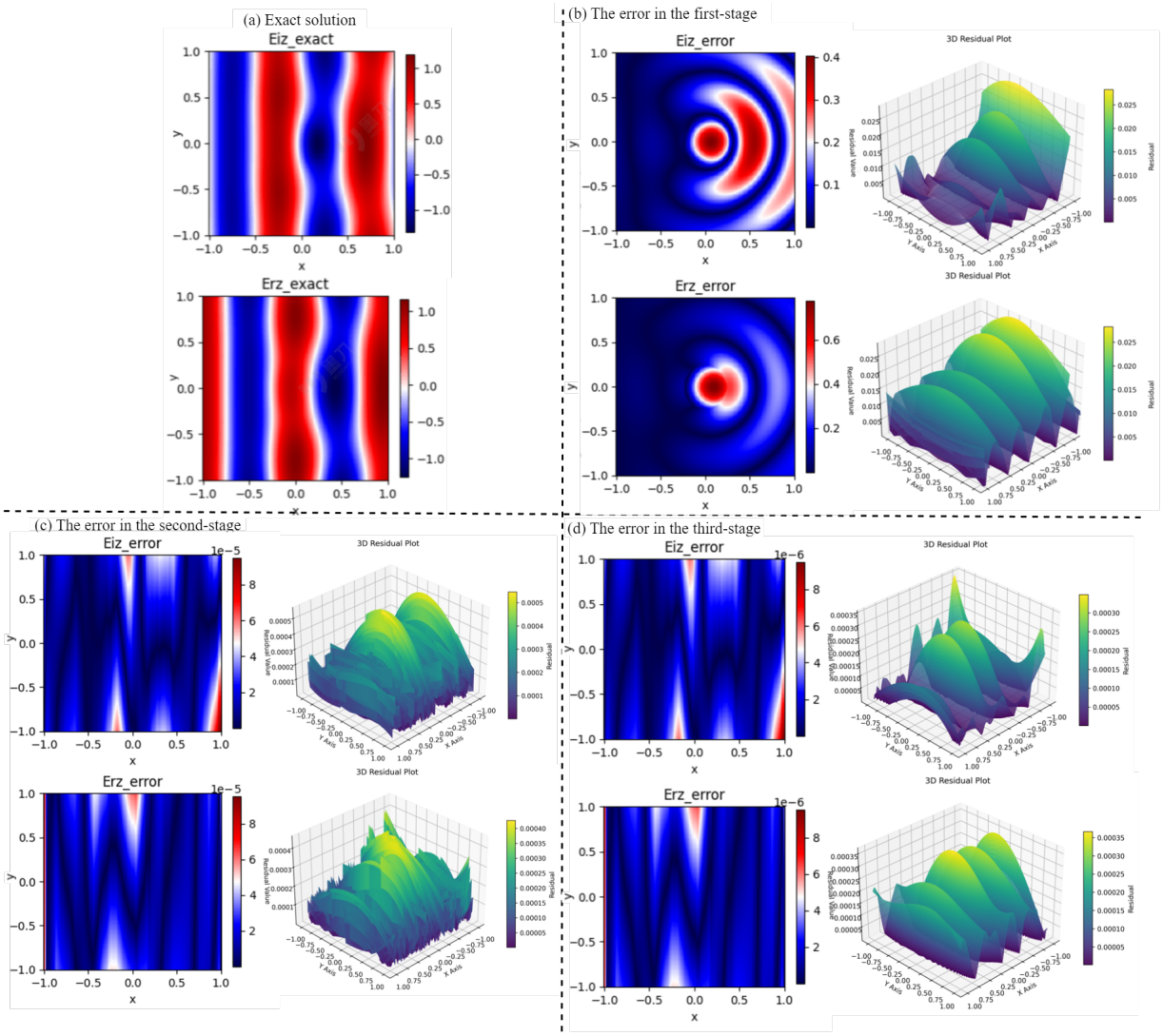


Figure 8: SI-MSPINNs for Helmholtz equation ($\epsilon = 1$).

Table 1: Comparison of results of different methods.

method	$\epsilon = 1$		$\epsilon = 1.5$		$\epsilon = 2$	
	$L_2(E_{rz})$	$L_2(E_{iz})$	$L_2(E_{rz})$	$L_2(E_{iz})$	$L_2(E_{rz})$	$L_2(E_{iz})$
PINN	0.0092	0.0084	0.0206	0.035	0.207	0.158
MSNN	0.0007	0.0005	0.0024	0.0098	0.034	0.061
SI-MSPINNs	3.2×10^{-6}	2.1×10^{-6}	0.0019	0.002	0.0075	0.031
RFF-MSPINNs	2.9×10^{-6}	4.1×10^{-5}	1.7×10^{-4}	2.6×10^{-4}	0.0012	0.0003

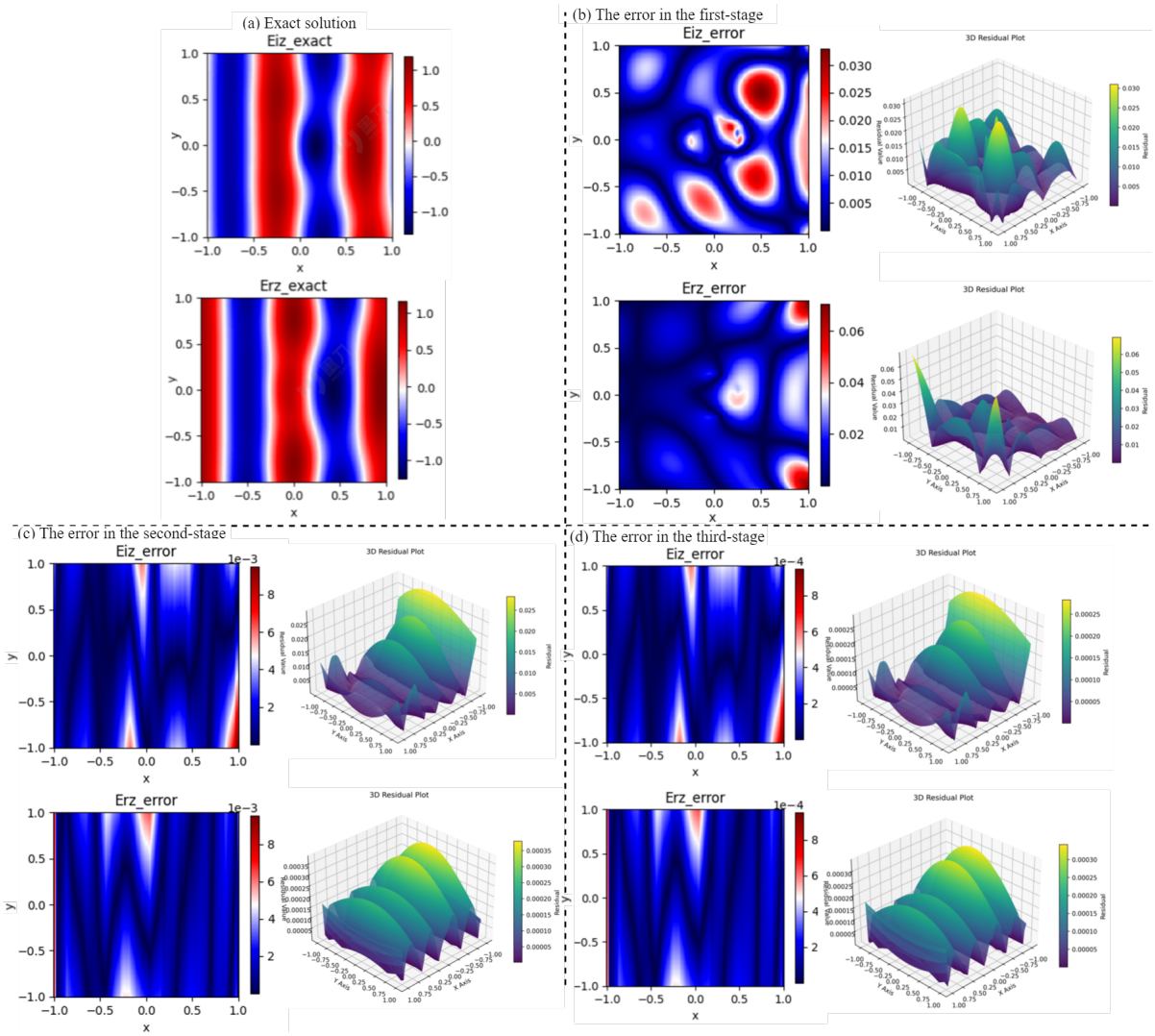


Figure 9: SI-MSPINNs for Helmholtz equation($\epsilon = 1.5$).

- Compared with PINNs, both models effectively overcome their spectral bias issues. Specifically, SI-MSPINNs extracts dominant spectral patterns of residuals through discrete Fourier transform to guide network initialization, enabling accurate capture of high-frequency physical features. This allows the network to more efficiently adapt to high-frequency components when learning PDE residuals. RFF-MSPINNs, on the other hand, dynamically adjusts the frequency sampling distribution of random Fourier features based on residual power spectral density, prioritizing focus on high-energy physical modes. It breaks through the limitations of traditional PINNs, such as inefficient derivative calculation caused by automatic differentiation and insufficient ability to learn high-frequency features.

- Compared with MSNNs, SI-MSPINNs and RFF-MSPINNs further address their core defects in 2D problems. Although MSNNs alleviate spectral bias through multistage residual learning, their convergence efficiency decreases significantly in high-dimensional scenarios, and spectral initialization is susceptible to spectral aliasing, leading the network to fall into local optima. In contrast, SI-MSPINNs adopts a spectrum-informed hierarchical initialization strategy, enabling each stage of the network to specifically correct the residuals of the preceding cumulative solution. Combined with the gradual enhancement of the physical constraint weights, it achieves a progressive error reduction while ensuring physical consistency. RFF-MSPINNs replaces frequency sorting with probability sampling, avoiding the computational burden of sorting spectral features in high dimensions. It enables the network focus on key frequencies while maintaining randomness, resulting in faster convergence and stronger high-dimensional scalability. Results on solving Burgers equation and parameterized Helmholtz equation show the effectiveness of the proposed methods.

The SI-MSPINNs and RFF-MSPINNs have shown higher accuracy in 2D problems. Our future work will be focused on breaking through the hierarchical spectrum initialization strategy for higher-dimenstion scenes, and improving the ability to capture local features in shock wave regions through the fusion of adaptive resampling [28] and gradient enhancement constraints [29]. At the same time, exploring the use of wavelet basis functions instead of Fourier basis to optimize the spectral representation of non-stationary signals such as turbulence, and combining quantum Fourier transform to accelerate high-dimensional spectral processing.

7. Acknowledgments

The work is partially supported by Natural Science Foundation of Sichuan Province (Grant No. 24NSFSC1366) and NSFC (Grant No. 12101511).

References

[1] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, in: Proceedings of the National Academy of Sciences, Vol. 116, 2019, pp. 15322–15327.

[2] H. Shang, J. Wang, G. Luo, Asymptotic self-similar blow-up profile for three-dimensional axisymmetric euler equations using neural networks, Physical Review Fluids 9 (2) (2024) 024703.

[3] W. Chen, Q. Wang, J. S. Hesthaven, C. Zhang, Physics-informed machine learning for reduced-order modeling of nonlinear problems, Journal of Computational Physics 446 (2021) 110666.

[4] J. Fu, D. Xiao, R. Fu, C. Li, C. Zhu, R. Arcucci, I. M. Navon, Physics-data combined machine learning for parametric reduced-order modelling of nonlinear dynamical systems in small-data regimes, Computer Methods in Applied Mechanics and Engineering 404 (2023) 115771.

[5] T. M. E. Brockherde, Vogt, By-passing the kohn-sham equations with machine learning, Nature Communications 1 (8) (2017) 872.

[6] J. M. Rahm, M. Hoffmann, P. O. Dral, Machine learning enables long time scale molecular photodynamics simulations, Nature Communications 15 (2024) 2097.

[7] A. Chattopadhyay, P. Hassanzadeh, D. Subramanian, Long-term instabilities of deep learning-based digital twins of the climate system: The cause and a solution, arXiv preprint arXiv:2304.07029 (2023).

[8] G. E. Karniadakis, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440.

[9] A. D. Rahaman, Baratin, On the spectral bias of neural networks, Proceedings of the 36th International Conference on Machine Learning 2 (2019) 5301–5310.

[10] L. Wang, X. Yan, T. Cheng, J. Yang, Overview frequency principle/spectral bias in deep learning, National Science Review 7 (1) (2020) 36–41.

[11] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 8571–8580.

[12] L. Wu, J. Ghaboussi, Y. Yuan, Y. Liu, Interpretable structural model error discovery using spectral bias-reduced neural networks, Engineering Applications of Artificial Intelligence 126 (2023) 107093.

[13] S. Rasp, M. S. Pritchard, P. Gentine, Machine learning for climate physics and simulations, Nature Communications 11 (1) (2020) 5482.

[14] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks for solving pdes, in: International Conference on Learning Representations (ICLR) Workshop, 2019.

[15] T. Udrescu, S.-M., Ai feynman: A physics-inspired method for symbolic regression, Science Advances 16 (6) (2020).

[16] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. Battaglia, Multi-stage neural networks: Function approximator of machine precision, Proceedings of the National Academy of Sciences 118 (32) (2021) e2105274118.

[17] M. A. Rahman, Z. Ross, K. Azizzadenesheli, Spectral bias analysis of physics-informed neural networks for hyperbolic pdes, Journal of Computational Physics 492 (2023) 112432.

[18] Q. Li, J. Lu, S. Cao, Spectrum-informed multistage neural network: Multiscale function approximator of machine precision, SIAM Journal on Scientific Computing 42 (5) (2020) A3241–A3264.

[19] S. Cao, Q. Li, J. Lu, Enhanced multistage neural networks for high-frequency approximation, Journal of Computational Physics 512 (2024) 113145.

[20] W. Y. Liu X, Adaptive multistage training for neural networks solving high-dimensional pdes, IEEE Transactions on Neural Networks and Learning Systems 2 (35) (2024) 1890–1903.

[21] R. Rahimi.A., Random features for large-scale kernel machines, Advances in Neural Information Processing Systems (NeurIPS) 20 (2007) 1177–1184.

[22] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, R. Ng, Fourier feature augmentation in physics-informed learning, Advances in Neural Information Processing Systems 33 (2020) 11248–11259.

[23] S. Mishra, R. Molinaro, Spectral prior guided deep learning for multiscale pdes with discontinuous coefficients, Journal of Scientific Computing 98 (2024) 23.

[24] L. Li, Spectrum-weighted random features for physics-informed neural networks, Neural Computing and Applications 36 (3) (2024) 1–18.

[25] W. C. Chew, Waves and fields in inhomogeneous media, IEEE Press (1995).

[26] Z. Yu, J. Chen, Data-driven spectral analysis of electromagnetic scattering using pinns, IEEE Transactions on Antennas and Propagation 72 (2) (2024) 1053–1064.

[27] W. C. Chew, Waves and fields in inhomogeneous media, in: Chapter 7: Scattering by Spheres and Cylinders, IEEE Press, 1995, pp. 289–294.

[28] M. K. Lu, Meng, Mdeepxde: A deep learning library for solving differential equations, SIAM Review 63 (1) (2021) 208–228.

[29] Z. Yu, J. Chen, J. Zhao, Gradient-enhanced physics-informed neural networks for forward and inverse pde problems, Computer Methods in Applied Mechanics and Engineering 428 (2024) 117064.