

基于紧支撑径向基函数的物理信息神经网络求解偏微分方程

王樂天¹, 贾宏恩¹, 张 扬², 范振宙¹

¹太原理工大学数学学院, 山西 太原

²临汾市人民医院超声科, 山西 临汾

收稿日期: 2026年2月19日; 录用日期: 2026年3月13日; 发布日期: 2026年3月24日

摘 要

物理信息神经网络(PINN)作为一种融合物理先验知识与深度学习的新型计算范式,在偏微分方程数值求解领域展现出广阔的应用前景。然而,传统PINN采用全连接神经网络作为逼近器,存在谱偏置问题,难以有效捕捉解高频特征和局部变化。针对这一局限性,本文提出一种基于紧支撑径向基函数的单层物理信息神经网络方法(CSPINN),该方法以紧支撑Wendland函数为基础构建局部化的基函数表示,通过将支撑半径和中心点位置设置为可学习参数,在训练过程中自适应地调整网络的局部感受野,增强网络对解函数高频分量与局部细节的捕捉能力。为增强方法的鲁棒性,本文提出均匀初始化策略,将各中心点的支撑半径初始化为在计算域特征尺度范围内均匀分布的随机值。通过多种类型偏微分方程的数值实验,验证了CSPINN方法的有效性和精度优势。

关键词

物理信息神经网络, 紧支撑径向基函数, Wendland函数, 偏微分方程, 自适应支撑半径

Compactly Supported Physics-Informed Neural Network for Solving Partial Differential Equations

Letian Wang¹, Hong'en Jia¹, Yang Zhang², Zhenzhou Fan¹

¹School of Mathematics, Taiyuan University of Technology, Taiyuan Shanxi

²Department of Ultrasound, Linfen People's Hospital, Linfen Shanxi

Received: February 19, 2026; accepted: March 13, 2026; published: March 24, 2026

文章引用: 王樂天, 贾宏恩, 张扬, 范振宙. 基于紧支撑径向基函数的物理信息神经网络求解偏微分方程[J]. 应用数学进展, 2026, 15(3): 478-490. DOI: 10.12677/aam.2026.153120

Abstract

Physics-Informed Neural Networks (PINNs) have emerged as a promising paradigm for solving partial differential equations by integrating physical laws into deep learning. However, standard PINNs using multi-layer perceptrons suffer from spectral bias, making it difficult to capture high-frequency features. This paper proposes a Compactly Supported Physics-Informed Neural Network (CSPINN) based on Wendland's compactly supported radial basis functions. By treating support radii and center locations as learnable parameters, CSPINN adaptively adjusts the local receptive fields during training. A uniform initialization strategy is proposed to enhance robustness. The support radii of each center point are initialized as random values that are uniformly distributed within the characteristic scale range of the computational domain. Numerical experiments on various PDEs demonstrate the effectiveness and accuracy advantages of CSPINN.

Keywords

Physics-Informed Neural Networks, Compactly Supported Radial Basis Functions, Wendland Functions, Partial Differential Equations, Adaptive Support Radius

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

偏微分方程是描述自然界和工程领域中各种物理现象的基本数学工具，广泛应用于流体力学、固体力学、电磁学、热传导、量子物理等诸多学科。对于复杂的偏微分方程，通常难以求得解析解，因此往往需要借助数值方法来获取近似解。传统的数值方法如有限差分法(FDM)、有限元法(FEM)和有限体积法(FVM)虽然经过数十年的发展已趋于成熟，但在处理复杂几何区域、高维问题和多尺度现象时仍面临算法复杂、计算量大、维数灾难等挑战。

无网格方法的出现为上述问题提供了新的解决思路。其中，径向基函数(RBF)方法凭借其数学形式简洁、实现方便、对空间维数不敏感等优点，成为无网格方法中最具代表性的分支之一[1][2]。RBF方法的核心思想是利用径向对称的基函数对未知解进行插值或逼近，通过满足偏微分方程和边界条件建立代数方程组求解。根据基函数的支撑范围，RBF可分为全局支撑(如多二次曲面、高斯函数)和紧支撑(如Wendland函数、Wu函数)两大类。

全局支撑RBF方法虽然在光滑问题上能够达到谱精度，但由于每个基函数对全域都有贡献，导致插值矩阵为稠密矩阵，计算复杂度高达 $O(N^3)$ ，严重制约了其在大规模问题中的应用。相比之下，紧支撑径向基函数(CS-RBFs)通过将基函数的影响限制在有限半径内，使得插值矩阵具有稀疏性，大幅降低了计算和存储开销。Wendland于1995年首次系统性地构造了一类具有任意光滑阶的正定紧支撑RBF，奠定了CS-RBF研究的理论基础[3]。

然而，CS-RBF方法在实际应用中面临一个关键性难题：支撑半径的选取。支撑半径过小会导致插值矩阵条件数过大甚至奇异，数值解不稳定；支撑半径过大则会削弱稀疏性优势，增加计算成本[1]。最优支撑半径往往与问题的具体特征(解的光滑性、节点分布、边界条件等)密切相关，传统的参数选择方法如交叉验证、条件数优化等计算开销大且缺乏普适性[2]。这一问题长期以来制约着CS-RBF方法的工程应用。

近年来,深度学习技术的蓬勃发展为科学计算带来了新的机遇。Raissi 等人于 2019 年提出的物理信息神经网络(PINNs)开创性地将深度神经网络与物理方程约束相结合[4],通过最小化方程残差和边界条件误差来训练网络参数,实现了无需标注数据的偏微分方程求解。PINN 避免了传统数值方法中的网格生成和矩阵组装过程,展现出处理高维问题和复杂几何的潜力。为便于研究应用,一些深度学习库如 DeepXDE 已被开发,用于支持 PINNs 求解各种微分方程,包括正问题和逆问题,并处理复杂几何域和多种 PDE 类型[5]。

然而,标准 PINN 通常采用多层感知机(MLP)作为函数逼近器,存在明显的谱偏置问题——网络倾向于先学习低频成分,对高频特征的捕捉能力较弱。这导致 PINN 在求解具有尖锐梯度、激波、边界层等局部特征的问题时表现欠佳[6]。为克服这一缺陷,研究者们提出了多种改进策略,包括傅里叶特征嵌入、自适应激活函数、多尺度网络架构等[7] [8]。借助 RBF 函数(如高斯函数)天然具备的局部化,将 RBF 融入 PINN 架构[9],增强网络对解函数高频分量与局部细节的捕捉能力,在求解具有奇异性或多尺度特性的物理问题时展现出独特潜力。

基于上述背景,本文提出一种基于紧支撑径向基函数的物理信息神经网络方法(CSPINN)。CSPINN 以 Wendland 紧支撑函数为基础,将支撑半径 δ 和中心点位置 c 作为可学习参数,与权重系数 w 一起通过反向传播进行优化。这种设计使得网络能够在训练过程中自适应地调整各基函数的影响范围,无需预先确定最优支撑半径。为进一步增强方法的鲁棒性,本文采用均匀初始化策略,将初始支撑半径设为计算域尺度范围内的均匀分布随机值,为网络提供多样化的初始感受野配置。

2. 物理信息神经网络

物理信息神经网络是一种将物理定律(如偏微分方程)直接嵌入神经网络训练过程的框架。它通过最小化损失函数来求解 PDE 的近似解,同时整合观测数据和物理约束。核心在于将物理知识转化为可优化的约束,从而在数据稀缺时提升模型的泛化能力。

PINN 使用一个全连接的前馈神经网络作为解的逼近器:

$$\hat{u}(x, t; \theta) = NN(x, t; \theta) \quad (1)$$

其中: θ 是网络参数,包括权重 w 和偏置 b ,网络结构通常为多层感知机(MLP),激活函数如 \tanh 或 ReLU,以确保光滑性和非线性表达能力。输入是时空坐标 (x, t) ,输出是近似解 \hat{u} 。通过自动微分,可以计算 \hat{u} 的任意阶导数,由深度学习框架(如 PyTorch 或 TensorFlow)支持的链式法则实现。

PINN 的核心是构建一个复合损失函数,将物理残差和数据匹配相结合。总损失函数通常为:

$$L(\theta) = \lambda_R L_R(\theta) + \lambda_B L_B(\theta) + \lambda_I L_I(\theta) + \lambda_D L_D(\theta) \quad (2)$$

包含残差损失 $L_R(\theta)$,边界条件损失 $L_B(\theta)$,初始条件损失 $L_I(\theta)$ 和数据损失 $L_D(\theta)$,这允许在数据稀缺时强调物理约束(纯物理驱动),或在数据丰富时结合两者(数据-物理混合),训练 PINN 相当于求解一个非凸优化问题:

$$\theta^* = \arg \min_{\theta} L(\theta) \quad (3)$$

使用梯度下降算法,如 Adam 优化器。梯度通过自动微分计算:

$$\nabla_{\theta} L = \nabla_{\theta} (\lambda_R L_R + \lambda_B L_B + \lambda_I L_I + \lambda_D L_D) \quad (4)$$

每个迭代中前向传播计算 \hat{u} 和残差。反向传播更新 θ ,收敛后, $\hat{u}(x, t; \theta)$ 即为 PDE 的近似解。

PINN 的优势在于其无网格特性,适用于高维问题,整合不确定性并且高效处理非线性 PDE。但损失

函数可能不平衡, 导致谱偏差, 即高频模式收敛慢。

3. 基于紧支撑径向基函数的物理信息神经网络

3.1. 紧支撑径向基函数

径向基函数是一类仅依赖于自变量到某固定点(中心点)距离的函数。设中心点为 $c \in \mathbb{R}^d$, 支撑半径为 $\delta > 0$, 则径向基函数可表示为 $\varphi\left(\frac{\|x-c\|}{\delta}\right)$, 其中 $\|\cdot\|$ 为欧氏范数。紧支撑径向基函数的特点是在支撑域外取值为零, 即当 $\|x-c\| > \delta$ 时, $\varphi\left(\frac{\|x-c\|}{\delta}\right) = 0$ 。

Wendland 于 1995 年通过对截断幂函数进行积分运算, 系统地构造了一类具有最小多项式阶数且保证正定性的紧支撑 RBF。Wendland 函数族记为 $\varphi_{d,k}$, 其中 d 为空间维数, k 为光滑性参数。函数的一般构造形式为:

$$\varphi_{d,k} = I^k \varphi_{d,0}, \tag{5}$$

其中 I 为积分算子, 基础函数 $\varphi_{d,k}(r) = (1-r)_+^l p(r)$, $l = [d/2] + k + 1$, $(\cdot)_+$ 表示截断函数, 即 $(1-r)_+ = \max(1-r, 0)$ 。通过 k 次积分运算, 可得到 C^{2k} 连续的 Wendland 函数。

本文采用 C^4 连续 Wendland 函数($d=3, k=2$), 定义为:

$$\varphi(r) = (1-r)_+^6 (35r^2 + 18r + 3), \tag{6}$$

该函数在 $r=0$ 处达到最大值 $\varphi(0)=3$, 在 $r=1$ 处光滑地衰减到零, 并且在 $r \geq 1$ 时恒等于零, C^4 连续性保证了函数的导数连续。

图 1 展示了三种典型径向基函数的二维可视化及其在 $y=0$ 处的横截面。这其中包括紧支撑函数和全局支撑函数的比较。每个子图的颜色映射表示函数值的大小, x 和 y 轴为空间坐标。

- (a) Wendland 函数(紧支撑型): 该函数具有紧凑支持, 即仅在有限半径内非零(图中圆形虚线区域), 超出该范围函数值为零。这有助于提高计算效率并避免远距离影响。
- (b) Gaussian 函数(全局支撑型): 该函数在整个空间域内非零, 但值随距离增加而指数衰减, 呈现出平滑的钟形分布, 适用于需要全局光滑性的应用。
- (c) Inverse Multiquadric 函数(全局支撑型): 类似于 Gaussian, 但衰减较慢, 函数值在中心较高并向外渐减, 常用于处理噪声数据以获得更稳定的插值结果。
- (d) 在 $y=0$ 处的横截面: 该子图比较了三种函数沿 x 轴的剖面曲线, 此比较突显了不同函数的支承范围和衰减特性对数值模拟的影响。

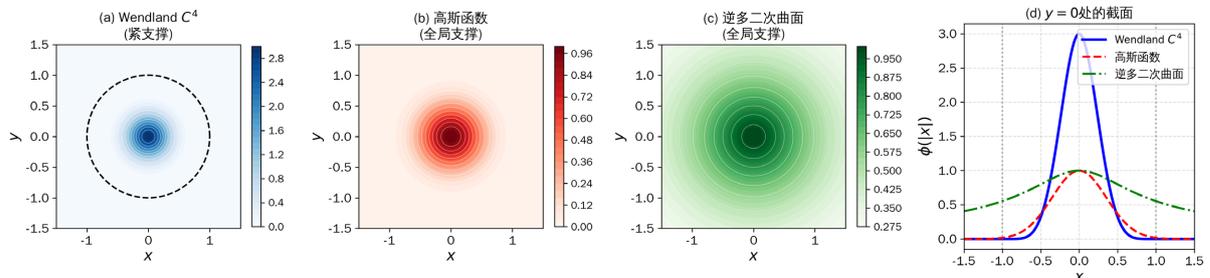


Figure 1. Comparison of RBF Functions
图 1. RBF 函数对比

Wendland 函数具有多种关键性质，首先是正定性，由 Wendland 函数构造的插值矩阵是正定的，保证了插值问题的唯一可解性；其次为紧支撑性，基函数仅在有限区域内非零，导致插值矩阵稀疏，计算效率高；同时基函数光滑性可控，可通过参数 k 可以获得任意阶的光滑性；除此之外，多项式阶数最小，在保证正定性的前提下，Wendland 函数的多项式阶数是最小的，这有利于减少计算量。

3.2. 网络架构

CSPINN 采用基于紧支撑径向基函数的单层网络结构对偏微分方程的解进行逼近。设网络包含 K 个基函数单元，第 i 个单元的中心点位置为 $c_i \in \mathbb{R}^d$ ，支撑半径为 $\delta_i > 0$ ，权重系数为 $w_i \in \mathbb{R}$ 。对于输入坐标 $x \in \mathbb{R}^d$ ，网络输出定义为：

$$u_\theta(x) = \sum_{i=1}^k w_i \Phi(x; c_i, \delta_i), \tag{7}$$

其中 $\theta = \{w, c, \delta\}$ 为全体可学习参数，基函数 Φ 定义为 Wendland 函数：

$$\Phi(x; c_i, \delta_i) = \varphi\left(\frac{\|x - c_i\|}{\delta_i}\right), \tag{8}$$

在该表示形式下，每个基函数的影响范围由其支撑半径 δ_i 控制：当 $\|x - c_i\| > \delta_i$ 时， $\Phi(x; c_i, \delta_i) = 0$ 。这种局部化特性意味着对于给定的点 x ，只有支撑域覆盖该点的基函数对输出有贡献，从而实现了计算的稀疏性。

值得注意的是，CSPINN 的逼近形式与传统 RBF 配点法在形式上相似，但有本质区别：传统方法通过求解线性方程组确定权重系数，中心点和支撑半径通常预先固定；而 CSPINN 将所有参数统一纳入梯度下降优化框架，通过损失函数驱动实现学习。

图 2 展示了 CSPINN 的整体网络架构。网络由三个主要部分组成：输入层、紧支撑径向基函数层和输出层。在图 1 左侧，计算域 Ω 内分布着基函数中心点 c_i （以十字标记表示），每个中心点对应一个支撑域 δ_i （以虚线圆表示）。配点用于计算方程残差和边界条件误差。值得注意的是，不同中心点的支撑半径各不相同，这体现了均匀初始化策略的核心思想。

网络的前向传播过程如下：输入坐标 (x, t) 首先进入 CSRBF 层，该层包含 K 个紧支撑 Wendland 基函数 $\Phi_1, \Phi_2, \dots, \Phi_K$ ，每个基函数根据输入点与其中心点的距离计算激活值，当距离超过支撑半径时激活值为零。随后，各基函数的输出经过可学习权重的线性组合，得到网络预测值。

在损失计算阶段，利用自动微分技术计算预测值对输入变量的各阶偏导数，进而构建方程残差损失 L_r 和边界条件损失 L_b 。对于一般形式的偏微分方程边值问题：

$$\mathcal{L}u(x) = f(x), x \in \Omega, \tag{9}$$

$$\mathcal{B}u(x) = g(x), x \in \partial\Omega, \tag{10}$$

$$u(x, 0) = u_o(x) \tag{11}$$

其中 $\Omega \subset \mathbb{R}^d$ 为计算域， $\partial\Omega$ 为边界， \mathcal{L} 为微分算子， \mathcal{B} 为边界条件算子， u 为待求解的未知函数， f ， g 和 $u_o(x)$ 为已知函数。

用 $u_\theta(x)$ 逼近真解 $u(x)$ ，其中 θ 表示网络参数。网络参数通过最小化以下复合损失函数来确定：网络参数通过最小化以下复合损失函数来确定：

$$L(\theta) = \lambda_r L_r(\theta) + \lambda_b L_b(\theta) + \lambda_l L_l(\theta), \tag{12}$$

其中残差损失 $L_r(\theta)$ ，边界条件损失 $L_b(\theta)$ ，初始条件损失 $L_l(\theta)$ 分别定义为：

$$L_R(\theta) = \frac{1}{N_R} \sum_{i=1}^{N_R} |\mathcal{L}\hat{u}(\mathbf{x}_i) - f(\mathbf{x}_i)|^2, \quad (13)$$

$$L_B(\theta) = \frac{1}{N_B} \sum_{j=1}^{N_B} |\mathcal{B}\hat{u}(\mathbf{x}_j) - g(\mathbf{x}_j)|^2, \quad (14)$$

$$L_I(\theta) = \frac{1}{N_I} \sum_{k=1}^{N_I} |\hat{u}(\mathbf{x}_k) - u_0(\mathbf{x}_k)|^2, \quad (15)$$

这里 $\{\mathbf{x}_i\}$ 、 $\{\mathbf{x}_j\}$ 和 $\{\mathbf{x}_k\}$ 分别为域内配点，边界配点和初始条件配点， N_R 、 N_B 和 N_I 为相应的配点数目， λ_R 、 λ_B 和 λ_I 为平衡系数。微分算子 \mathcal{L} 通过自动微分技术计算。CSPINN 的全体可学习参数包括权重系数、中心点位置和支撑半径三类，其中支撑半径采用均匀初始化策略，为网络提供多尺度的初始感受野配置。

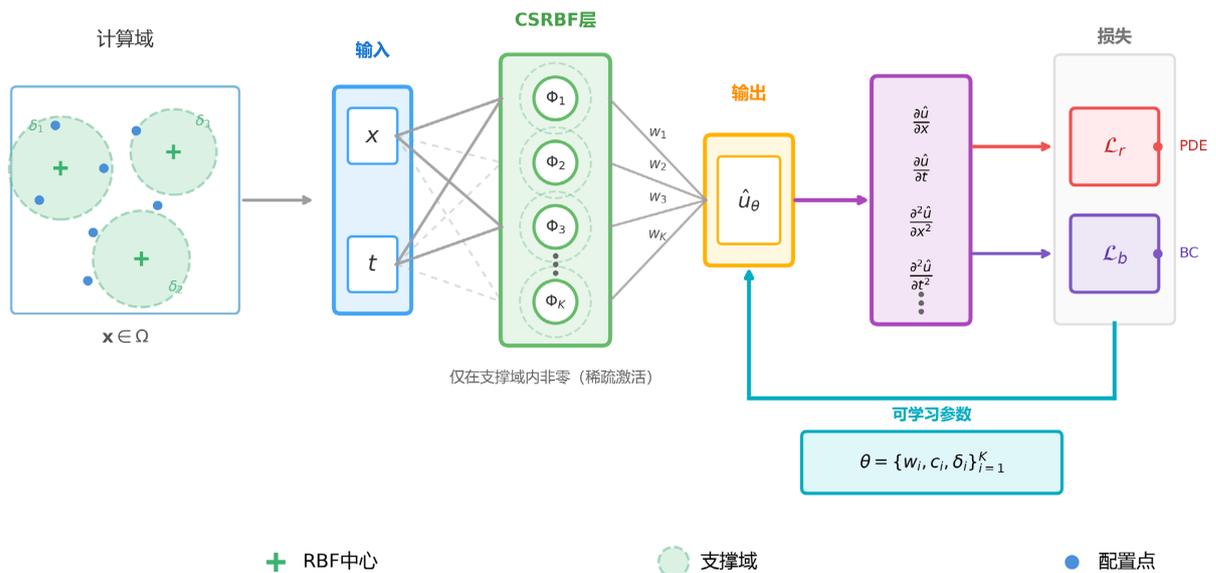


Figure 2. The CSPINN architecture
图 2. CSPINN 网络结构

3.3. 参数初始化策略

神经网络的训练效果在很大程度上受初始化策略的影响。对于 CSPINN 的三类参数，本文采用以下初始化方案：

(1) 中心点位置初始化

中心点位置 c 初始化为计算域 Ω 内的均匀分布随机点，这种初始化保证了基函数在计算域内的均匀覆盖。

(2) 权重系数初始化

权重系数 w 采用 Xavier 均匀初始化。

(3) 支撑半径初始化——均匀初始化策略

支撑半径的初始化是 CSPINN 方法的关键创新点。传统 CS-RBF 方法通常需要针对具体问题精心选择支撑半径，这一过程耗时且缺乏普适性。本文提出均匀初始化策略，将各中心点的支撑半径初始化为计算域特征尺度范围内的均匀分布随机值：

$$\delta_i \sim U(\delta_{\min}, \delta_{\max}), \delta_{\min} = 0.1L, \delta_{\max} = 1.0L, \tag{16}$$

这里 L 为计算域的特征尺度(如边长或对角线长度)。均匀初始化策略的核心思想是为网络提供多样化的初始感受野配置,使得不同基函数具有不同的初始影响范围,覆盖从局部到全局的多种尺度。这种多样性为后续的梯度优化提供了更丰富的探索空间。

均匀初始化策略具有多种优势,包括无需预先确定最优支撑半径,简化了方法的应用流程;多样化的初始配置,增强了网络对不同尺度特征的适应能力;通过训练可以自适应地调整到适合具体问题的支撑半径分布。

4. 数值结果

本节通过两个典型偏微分方程算例,对本文所构建的网络模型进行验证和评估。为了进一步检验其性能,将 CSPINN 方法与传统 CSBRF 数值方法、PINN 进行了对比分析。所有实验均采用 PyTorch 2.0 深度学习框架实现[10]。为保证实验的可重复性,网络采用 Adam 优化器[11],所有随机种子均设置为 42,主要实验参数设置如表 1:

Table 1. Experimental parameter settings

表 1. 实验参数设置

参数	扩散方程	泊松方程	Burgers 方程
基函数数目	500	1000	1000
域内配点数	500	1000	1000
边界配点数	100	200	200
训练轮数	30000	30000	30000
学习率	0.001	0.001	0.001
初始支撑半径	U (0.1L, 1.0L)	U (0.1L, 1.0L)	U (0.1L, 1.0L)

数值精度采用相对 L^2 误差进行评估:

$$\mathcal{E} = \frac{\|u_\theta - u_{\text{exact}}\|_2}{\|u_{\text{exact}}\|_2} \tag{17}$$

4.1. 扩散方程

本节考虑具有源项的一维扩散方程。该类方程广泛应用于热传导、物质扩散及金融工程等领域,是检验数值方法有效性的经典基准问题。

考虑如下扩散方程:

$$\frac{\partial y}{\partial t} = \frac{\partial^2 y}{\partial x^2} + e^t (\sin(\pi x) + \pi^2 \sin(\pi x)), \quad x \in [-1, 1], \quad t \in [0, 1], \tag{18}$$

初始条件定义为:

$$y(x, 0) = \sin(\pi x), \tag{19}$$

以及 Dirichlet 边界条件:

$$y(-1, t) = y(1, t) = 0, \tag{20}$$

该问题的解析解为:

$$y(x,t) = e^t \sin(\pi x) \quad (21)$$

传统 CSRBF 方法的精度对支撑半径的选择比较敏感。表 2 给出了支撑半径从 0.5 到 2.5 变化时的相对 L2 误差，可以看出，最优支撑半径与最差情况之间的误差相差近两个数量级。从 PINN 和 CSPINN 的结果可以看出，通过利用深度学习的自学习能力，本文实现了支撑半径的网络自适应优化。同时，所采用的紧支撑径向基函数增强了模型的非线性表达能力，相较于 PINN 中基于线性插值与激活函数的表达方式表现更优。

Table 2. The relative L2 error of different methods for solving the Diffusion equation
表 2. 不同方法求解扩散方程的相对 L2 误差

CSRBF	不同支撑半径	0.5	1.0	1.5	2	2.5
		4.27e-02	6.79e-04	4.82e-04	9.61e-04	1.08e-03
PINN				5.86e-04		
CSPINN				6.26e-05		

图 3 绘制了 PINN 和 CSPINN 预测结果随着时间的变化。这两个网络的结果都得到相对较高的精度。与 PINN 相比，本文提出的模型实现了约一个量级的精度提升，表明 CSPINN 具有更精确地表达函数的能力。此外，模型在时间步进过程中展现了优异的数值稳定性，预测精度未发生明显退化，在此方面优于求解同类瞬态问题的传统有限差分格式。

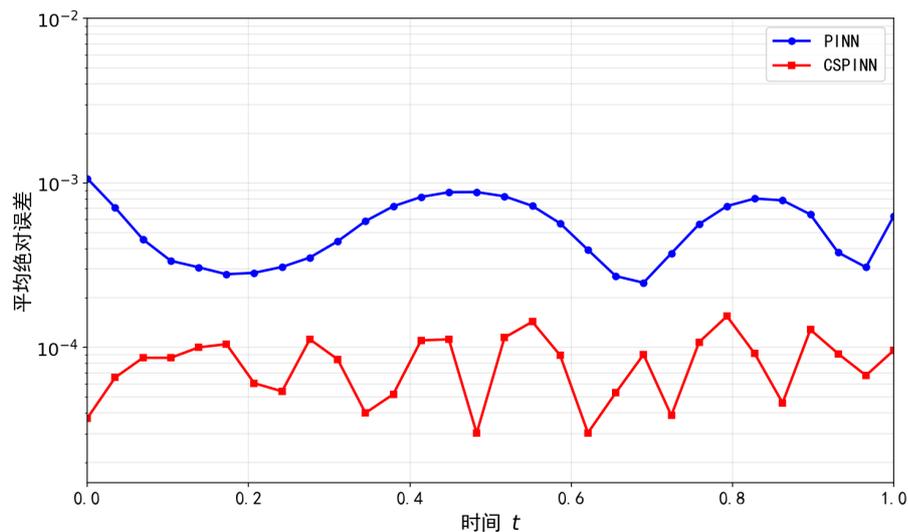


Figure 3. The distribution of the average absolute error of the PINN and CSPINN networks along the time axis

图 3. PINN 和 CSPINN 网络的平均绝对误差沿时间轴分布

为直观展示所提方法的自适应特性，图 4 给出了训练过程中基函数分布的演化情况。初始时刻(图 4(a))，500 个紧支撑径向基函数中心的均匀网格分布于整个时空计算域，支撑半径 δ 随机均匀初始化。

经过训练后(图 4(b))，可以观察到两个显著的自适应调整现象：

(1) 中心位置的自适应迁移：基函数中心从初始的均匀分布演化为向高误差区域聚集的非均匀分布。从图中可以看出，中心点在边界附近以及大时间区域呈现明显的聚集趋势，这些区域正是扩散方程解变化剧烈、数值求解难度较大的位置。

(2) 支撑半径的自适应收缩：训练后基函数的支撑半径整体呈现缩小趋势，平均缩减约 20%。更重要的是，半径的缩小与局部误差呈正相关——在高误差区域，散点颜色明显偏紫(对应较小的 δ 值)，表明模型自动在这些区域采用了更精细的局部逼近；而在低误差区域，基函数保持较大的支撑半径以提高计算效率。

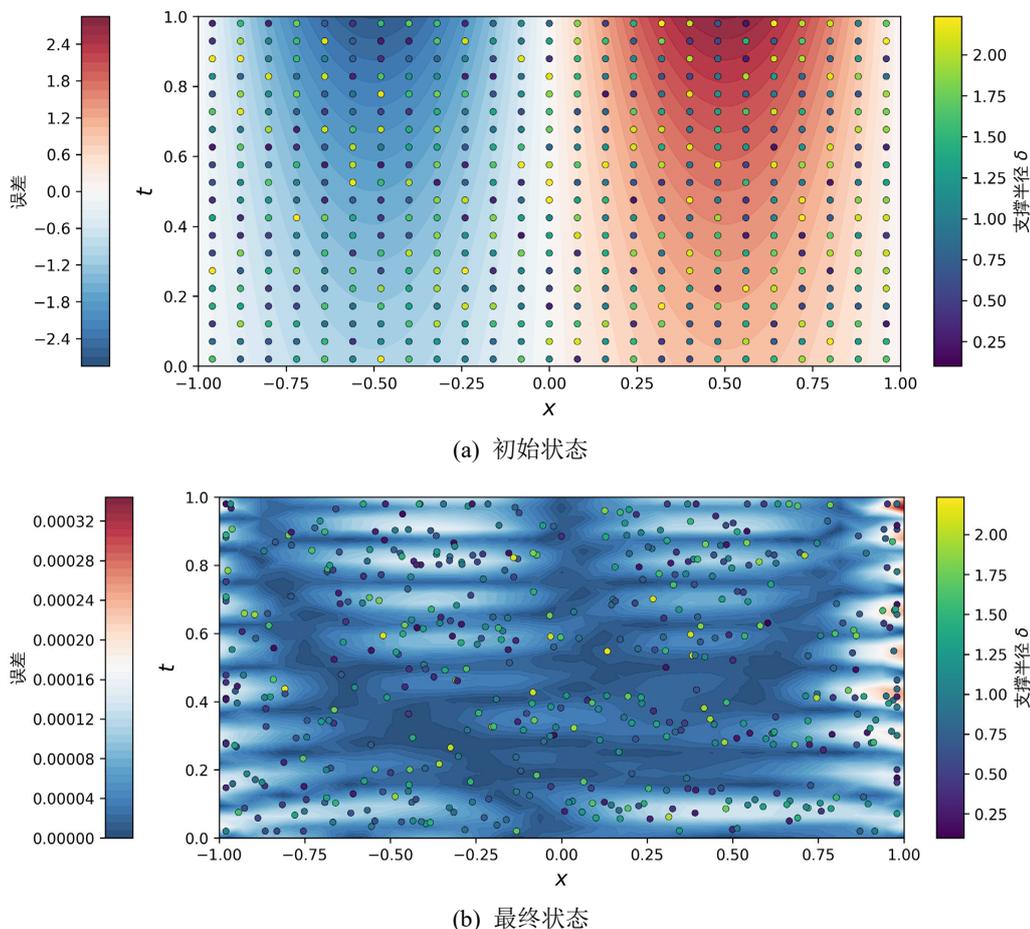


Figure 4. Initial and trained distribution of RBF centers with adaptive support radii
 图 4. RBF 中心点与自适应支撑半径的初始和训练后分布

4.2. 泊松方程

本节考虑一个二维 Poisson 方程，该方程常用于描述静电势、热分布和流体力学中的压力场等物理现象，是检验数值方法在捕捉多频率成分方面的经典测试问题。该算例的源项包含低频和高频分量，旨在评估 CSPINN 对解函数中高频特征的捕捉能力。

考虑如下 Poisson 方程：

$$-\Delta u = f(x, y), \quad \Omega = [0, 1]^2 \tag{22}$$

其中源项定义为：

$$f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y) + 2(8\pi)^2 \sin(8\pi x) \sin(8\pi y) \tag{23}$$

Dirichlet 边界条件：

$$u(x, y) = 0 \quad (24)$$

该问题的解析解为:

$$u(x, y) = \sin(\pi x)\sin(\pi y) + \sin(8\pi x)\sin(8\pi y) \quad (25)$$

该解函数结合了低频($\sin(\pi x)\sin(\pi y)$)和高频($\sin(8\pi x)\sin(8\pi y)$)分量,能够有效检验网络对谱偏置的鲁棒性。

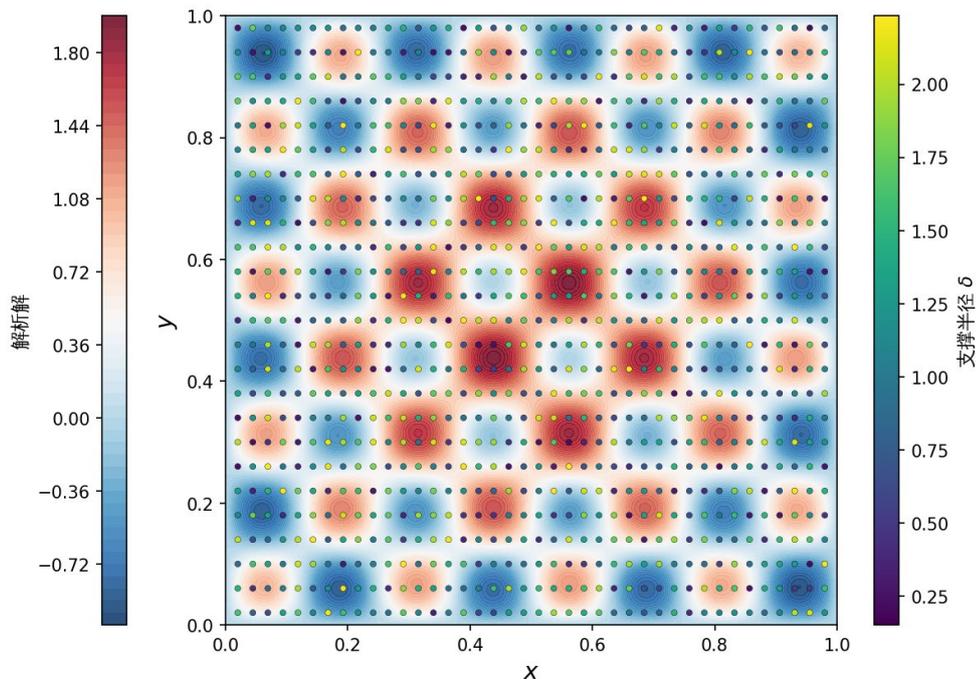
表 3 展示了不同方法在求解该 Poisson 方程时的相对 L2 误差。对于传统 CSRBF 方法,支撑半径的选择显著影响精度,最优值在 0.9 附近,但整体误差较高。PINN 方法受谱偏置影响误差较大。CSPINN 通过自适应支撑半径和局部化基函数,实现了最低误差,证明了其在捕捉多尺度特征方面的优势。

Table 3. The relative L2 error of different methods for solving the Poisson equation

表 3. 不同方法求解泊松方程的相对 L2 误差

CSRBF	不同支撑半径	0.3	0.6	0.9	1.2	1.5
		1.11e-02	6.52e-03	3.51e-03	8.89e-03	2.40e-03
PINN				3.12e-02		
CSPINN				6.27e-04		

为直观展示所提方法的自适应特性,图 5 给出了求解 Poisson 方程过程中基函数参数的演化情况。如图 5(a)所示,训练初始阶段,1000 个基函数中心均匀分布于计算域内,支撑半径随机初始化。经过训练后(图 5(b)),可以观察到两个显著的自适应行为:其一,基函数中心点沿误差梯度方向迁移,在高频分量主导的区域形成明显聚集,表明模型能够自主识别解变化剧烈的区域并增强局部逼近能力;其二,支撑半径在高梯度高误差区域显著收缩,而在解相对平缓的区域保持较大值。这种协同机制使得本方法能够以有限的基函数数量有效捕捉多尺度特征,验证了自适应策略的有效性。



(a) 初始状态

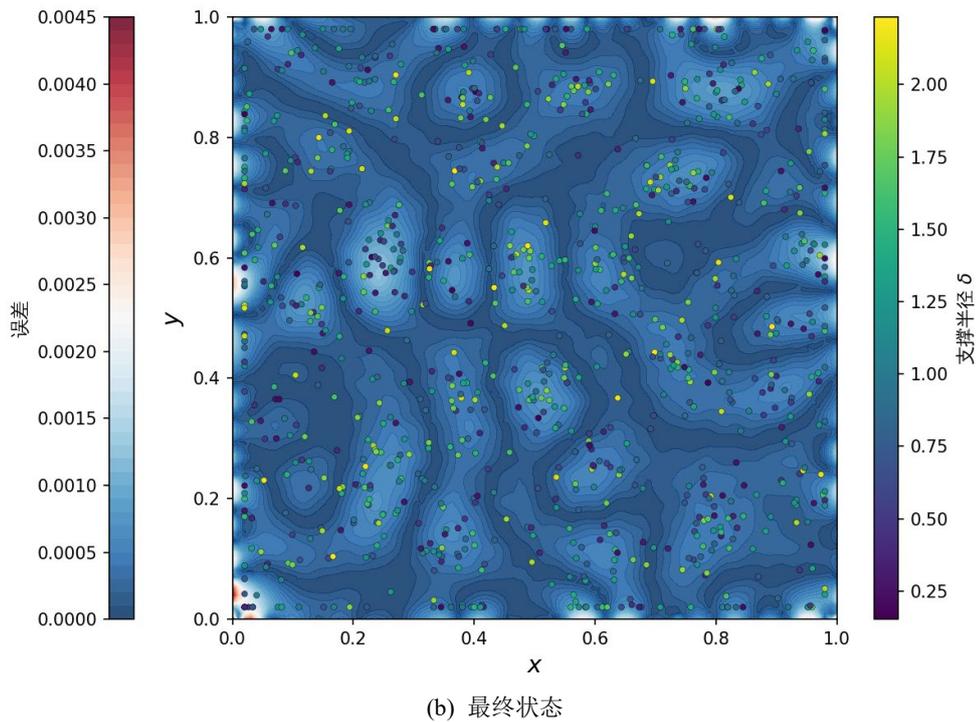


Figure 5. Initial and trained distribution of RBF centers with adaptive support radii
图 5. RBF 中心点与自适应支撑半径的初始和训练后分布

4.3. Burgers 方程

接下来，考虑一个瞬态的二维 Burgers 方程及边界条件和初始条件：

$$u_t + u(u_x + u_y) = 0.1(u_{xx} + u_{yy}), (x, y) \in [0, 1] \times [0, 1], t \in [0, 2], \quad (26)$$

$$u(0, x, y) = \frac{1}{1 + \exp\left(\frac{x+y}{0.2}\right)}, \quad (27)$$

$$u(t, x_b, y_b) = \frac{1}{1 + \exp\left(\frac{x_b + y_b - t}{0.2}\right)}, \quad (28)$$

该方程具有解析解：

$$u(t, x, y) = \frac{1}{1 + \exp\left(\frac{x+y-t}{0.2}\right)} \quad (29)$$

Table 4. The relative L2 error of different methods for solving the Burgers equation

表 4. 不同方法求解 Burgers 方程的相对 L2 误差

CSRBF	不同支撑半径	1.0	1.5	2	2.5	3
		2.18e-01	2.53e-01	2.45e-02	2.01e-01	1.67e-01
PINN			4.93e-03			
CSPINN			1.62e-03			

表 4 展示了不同方法的数值结果误差。在传统 CSRBF 方法中，支撑半径从 1.0 变化到 3.0，解的精度波动较为显著，且整体精度不高。相较之下，CSPINN 模型借助非线性插值函数和支撑半径的自适应优化机制，不仅显著降低了计算误差，而且有效克服了对经验支撑半径的依赖。

图 6 分别显示在 $t = 0, 0.5, 1.0, 1.5, 2.0$ 时刻的解析解、PINN 的预测解和 CSPINN 预测解的绝对误差。从图中可以看出，PINN 整体误差较大，CSPINN 预测误差整体小于 PINN 预测结果，精度优于 PINN。

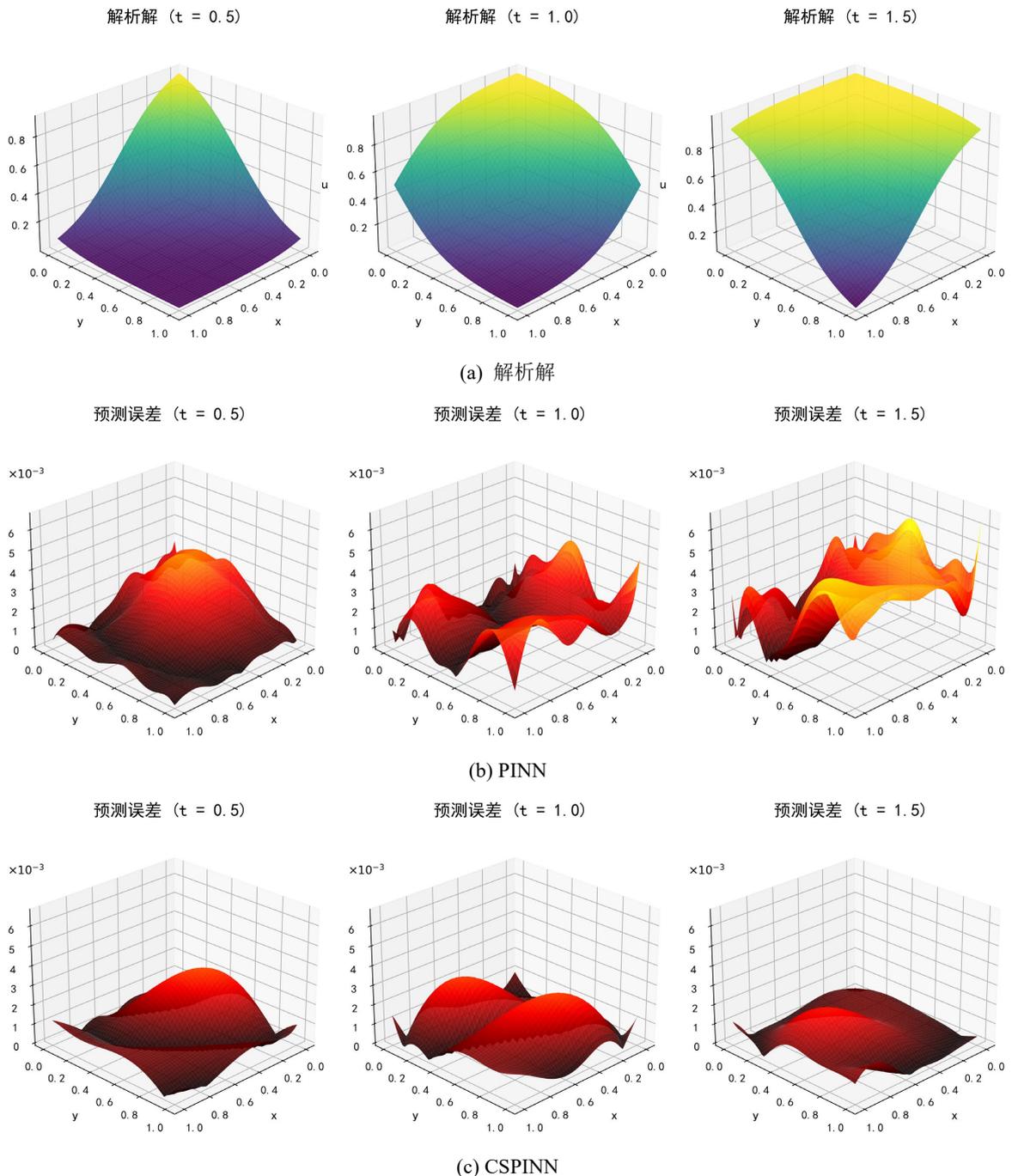


Figure 6. The analytical solution of the equation and the absolute error distribution of the PINN and CSPINN networks
图 6. 方程解析解以及 PINN 和 CSPINN 网络的绝对误差分布

5. 结论

本文提出了一种基于紧支撑径向基函数的物理信息神经网络方法 CSPINN, 用于求解偏微分方程。该方法以 C^4 连续的 Wendland 函数为基础构建局部化基函数表示, 并将支撑半径和中心点位置设置为可学习参数, 结合均匀初始化策略进行训练。CSPINN 具有以下优点: (1) CSPINN 相比标准 PINN 在计算精度上具有优势; (2) 均匀初始化策略无需参数调试; (3) 支撑半径的自适应学习机制能够有效优化网络结构。

未来的研究工作将从以下几个方向展开: (1) 探索更高效的初始化策略和训练技巧, 进一步提升方法的收敛速度和稳定性; (2) 研究支撑半径分布与问题特征之间的内在联系, 为参数选择提供理论指导; (3) 将 CSPINN 推广到时空耦合问题、高维问题和多物理场耦合问题; (4) 结合域分解策略实现大规模问题的并行求解。

基金项目

本研究受 2025 年度山西省高等学校科技创新计划项目(项目名称: 基于超声造影 - 细胞角蛋白的精确诊断胆囊息肉样病变的人工智能模型构建研究; 项目编号: 2025L041)资助。

参考文献

- [1] Fasshauer, G.E. (2007) Meshfree Approximation Methods with MATLAB. World Scientific. <https://doi.org/10.1142/6437>
- [2] Buhmann, M.D. (2003) Radial Basis Functions: Theory and Implementations. Cambridge University Press. <https://doi.org/10.1017/cbo9780511543241>
- [3] Wendland, H. (1995) Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree. *Advances in Computational Mathematics*, **4**, 389-396. <https://doi.org/10.1007/bf02123482>
- [4] Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019) 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. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [5] Lu, L., Meng, X., Mao, Z. and Karniadakis, G.E. (2021) DeepXDE: A Deep Learning Library for Solving Differential Equations. *SIAM Review*, **63**, 208-228. <https://doi.org/10.1137/19m1274067>
- [6] Rahaman, N., Baratin, A., Arpit, D., et al. (2019) On the Spectral Bias of Neural Networks. *International Conference on Machine Learning. PMLR, 2019*, Long Beach, 9-15 June 2019, 5301-5310.
- [7] Wang, S., Teng, Y. and Perdikaris, P. (2021) Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, **43**, A3055-A3081. <https://doi.org/10.1137/20m1318043>
- [8] Tancik, M., Srinivasan, P., Mildenhall, B., et al. (2020) Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *Advances in Neural Information Processing Systems*, **33**, 7537-7547.
- [9] Bai, J., Liu, G., Gupta, A., Alzubaidi, L., Feng, X. and Gu, Y. (2023) Physics-Informed Radial Basis Network (PIRBN): A Local Approximating Neural Network for Solving Nonlinear Partial Differential Equations. *Computer Methods in Applied Mechanics and Engineering*, **415**, Article ID: 116290. <https://doi.org/10.1016/j.cma.2023.116290>
- [10] Paszke, A., Gross, S., Massa, F., et al. (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, **32**, 8024-8035.
- [11] Kingma, D.P. and Ba, J. (2015) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980.