PDE 问题的物理信息神经网络：综述 Physics-informed neural networks for PDE problems: a comprehensive review

Kuang Luo 等 翻译：Gemini Agent

2025 年

Abstract

摘要

随着“科学人工智能”(AI for Science）的持续发展，物理信息神经网络（PINN）已成为科学计算和深度学习领域的一种变革性方法，为求解偏微分方程（PDE）和其他复杂物理系统提供了一个鲁棒且灵活的框架。通过将物理定律直接嵌入神经网络的架构中，PINN 能够整合领域特定的知识，确保模型在拟合现有数据的同时遵守已知的物理规律。在本文中，我们全面综述了 PINN 在广泛的 PDE 问题中的最新进展和应用。特别关注了 PINN 的架构、PINN 的数据重采样方法、损失函数和激活函数、特征嵌入方法等。此外，还讨论了 PINN 的潜在未来方向和预期演变。我们旨在为 PDE 问题的 PINN 提供有价值的见解，希望能鼓励这一充满希望的领域的进一步探索和研究。

关键词：科学人工智能 · 物理信息神经网络 · 深度学习 · 偏微分方程（PDE）

Contents

1 引言 **3**
1.1 物理信息神经网络 (PINN) 基础 . . . 5
1.1.1 问题表述 . . . 5
1.1.2 PINN 的性能评估 . . . 6
1.2 论文组织和贡献 . . . 7

2 用于 PDE 问题的 PINN 求解器 **7**
2.1 PINN 的架构 . . . 7
2.1.1 多层感知机 (MLP) . . . 9
2.1.2 卷积神经网络 (CNN) . . . 10
2.1.3 循环神经网络 (RNN) . . . 10
2.1.4 生成对抗网络 (GAN) . . . 12
2.1.5 Kolmogorov—Arnold 网络 (KAN) . . . 12
2.1.6 Transformer . . . 14
2.1.7 PINN 的其他架构 . . . 15
2.1.8 区域分解 . . . 16
2.1.9 PINN 的激活函数 . . . 18
2.2 PINN 的自适应数据重采样 . . . 18
2.3 PINN 的损失函数和优化 . . . 20
2.3.1 损失重加权 . . . 20
2.3.2 新型损失函数 . . . 21
2.4 特征嵌入和增强 . . . 22

3 应用 **23**
3.1 流体动力学 . . . 23
3.2 固体力学 . . . 23
3.3 电磁学和光学 . . . 24

4 软件 **24**
4.1 DeepXDE 25
4.2 IDRLnet 25
4.3 NeuroDiffEq 25
4.4 SciANN 26
4.5 TensorDiffEq 26

5 挑战与展望 **26**
5.1 PINN 面临的挑战 26
5.2 算子学习 27

6 结论 **27**

1 引言

近几十年来，计算机技术的进步深刻地改变了科学研究的面貌。传统上，研究人员依靠理论推导和实验验证来研究自然现象。然而，计算方法的出现引入了各种数值模拟技术，使研究人员能够更深入地洞察复杂的现实世界系统。这些基于计算机的方法促进了对以前使用传统方法难以研究的复杂现象的探索和分析。因此，计算模拟已成为科学工具箱中必不可少的工具，增强了我们对不同学科的理解。

微分方程 (DE) 在各种科学和工程学科 (Boussange 等 2023；Mallikarjunaiah 2023；Namaki 等 2023；Soldatenko 和 Yusupov 2017；Rodkina 和 Kelly 2011）中发挥着至关重要的作用，因为它们能够对涉及多个变量函数的复杂现象进行建模。它们的重要性源于许多物理过程，如热传导、流体动力学和波传播，都可以用捕捉函数及其导数之间关系的 PDE 来描述。根据自变量的数量，微分方程可分为以下两类（Taylor 等 2023）：常微分方程（ODE）仅涉及一个自变量及其导数，

$$\frac{du}{dx} + u = 0. \tag{1}$$

偏微分方程（PDE）涉及多个自变量及其偏导数，

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2}. \tag{2}$$

与代数方程不同，微分方程表达了未知函数与其导数之间的方程关系。

在实际应用中，PDE 的求解在航空航天工程等领域变得极其重要，它们被用于优化飞机设计和分析结构周围的气流。此外，在气象学中，PDE 在天气预报模型中起着至关重要的作用，通过模拟大气动力学实现准确的预测。有效求解 PDE 的能力是各个学科推进技术和提高预测能力的基础。PDE 的数学理论已得到显著发展，产生了各种用于分析和求解 PDE 的方法。为了求解一些简化的偏微分方程，可以使用通用算子（Melchers 等 2023）。目前，求解偏微分方程较常用的方法有有限元法（FEM）（Liu 等 2022；Chernyshenko 和 Olshanskii 2015）、有限体积法（FVM）（Dick 2009）、基于粒子的方法（Oñate 和 Owen 2011）和有限单元法（FCM）（Kollmannsberger 等 2019）。FEM 已被证明是一种方便且准确的方法，利用了大量的计算资源（Innerberger 和 Praetorius 2023）。然而，多重迭代求解的实用性是有限的。为了增强 PDE 求解的能力，研究人员探索了样条函数的使用（Kolman 等 2017；Qin 等 2019）。同样，FEM 对 PDE 解进行离散化和数值近似，使得引入在表达 PDE 方面效率更高的傅里叶变换或拉普拉斯变换变得更加可行（Mugler 和 Scott 1988；El-Ajou 2021）（图 1）。

最近，深度学习的显著进步导致了计算机视觉和自然语言处理等各个领域的突破性发展。此外，深度学习对科学计算也产生了深远的影响。物理信息机器学习（PIML）作为一种结合传统机器学习技术与领域特定物理知识来解决复杂科学和工程问题的方法应运而生。与仅依赖数据的传统机器学习方法不同，PIML 将物理定律和约束整合到学习过程中，允许模型不仅从数据中学习，还可以从已知的控制方程、边界条件和其他物理原理中学习。这种混合方法增强了模型的泛化能力，特别是在数据稀缺、嘈杂或不完整的情况下。PIML 在各个领域发挥着越来越重要的作用，包括优化（Mowlavi 和 Nabi 2023；Hwang 等 2022；Zhang 等 2022b）、流体（Cai 等 2021；Di Leoni 等 2023；Zhang 等 2023；Ranade 等 2021；Sun 等 2020；Hanna 等 2022）、控制论（Schiassi 等 2022；Patel 等 2023；Zheng 等 2023；Faria 等 2024）、气候（Li 等 2024；Bonev 等 2023）、力学（Tripura 和 Chakraborty 2023；Wang 等 2023；Diao 等 2023；Le-Duc 等 2024）、物理学（Pan 等 2024；Liu 等 2024a；Jeong 等 2024；Jo 等 2024）、量子计算（Norambuena 等 2024；Xiao 等 2024；Sedykh 等 2024）等。

基于深度学习求解 PDE 问题的方法自 20 世纪 90 年代以来一直被持续研究（Dissanayake 和 Phan-Thien 1994；Lagaris 等 1998；Kumar 和 Yadav 2011；Blechschmidt 和 Ernst 2021；Hafiz 等 2024）。特别是，物理信息神经网络（PINN）作为处理正向和反向 PDE 问题的一种有效方法已获得突出的地位（Raissi 等 2019）。通过将控制 PDE 整合到损失函数中，PINN 利用自动微分来指导训练过程，确保学习到的模型遵守物理定律。这种无网格方法提供了极大的灵活性，使 PINN 能够在单个框架内无缝地结合基于物理的约束和观测数据。这种能力使得 PINN 在传统 PDE 求解器可能难以应对的场景中特别具有优势，例如高维问题或具有复杂几何形状的问题。此外，融合数据与物理模型的能力增强了预测的准确性和鲁棒性，从而实现更可靠的

solution is limited. To enhance the capability of PDE solving, researchers have explored the use of spline functions (Kolman et al. 2017; Qin et al. 2019). Similarly, FEM discretizes and numerically approximates the PDE solution, making the introduction of Fourier transforms or Laplace transforms with higher efficiency in expressing PDEs more feasible (Mugler and Scott 1988; El-Ajou 2021) (Fig. 1).

Recently, remarkable progress in deep learning has led to groundbreaking advancements in various fields, such as computer vision and natural language processing. Additionally, deep learning has had a profound impact on scientific computing. Physics-informed machine learning (PIML) has emerged as an approach that combines traditional machine learning techniques with domain-specific physical knowledge to solve complex scientific and engineering problems. Unlike conventional machine learning methods that rely solely on data, PIML integrates physical laws and constraints into the learning process, allowing models to learn not only from data but also from known governing equations, boundary conditions, and other physical principles. This hybrid approach enhances the model's ability to generalize, especially in scenarios where data is scarce, noisy, or incomplete. PIML has played an increasingly crucial role across various fields, including optimization (Mowlavi and Nabi 2023; Hwang et al. 2022; Zhang et al. 2022b), fluid (Cai et al. 2021; Di Leoni et al. 2023; Zhang et al. 2023; Ranade et al. 2021; Sun et al. 2020; Hanna et al. 2022), cybernetics (Schiassi et al. 2022; Patel et al. 2023; Zheng et al. 2023; Faria et al. 2024), climate (Li et al. 2024; Bonev et al. 2023), mechanics (Tripura and Chakraborty 2023; Wang et al. 2023; Diao et al. 2023; Le-Duc et al. 2024), physics (Pan et al. 2024; Liu et al. 2024a; Jeong et al. 2024; Jo et al. 2024), quantum computing (Norambuena et al. 2024; Xiao et al. 2024; Sedykh et al. 2024), etc.

Methods based on deep learning for solving PDE problems have been continuously studied since 1990 s (Dissanayake and Phan-Thien 1994; Lagaris et al. 1998; Kumar and Yadav 2011; Blechschmidt and Ernst 2021; Hafiz et al. 2024). Particularly, physics-informed neural networks (PINNs) have gained prominence as an efficient method for tackling both forward and inverse PDE problems (Raissi et al. 2019). By integrating the governing PDEs into

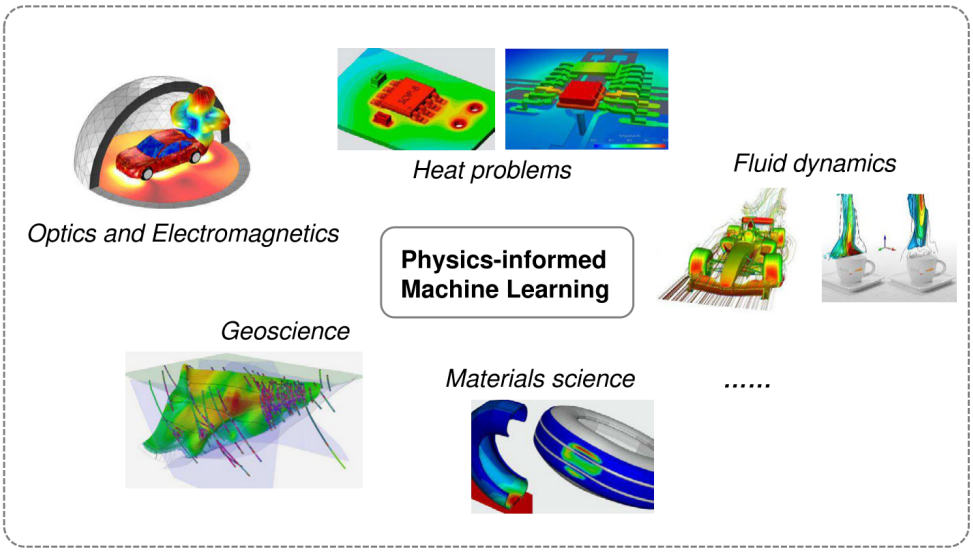

Fig. 1 Various applications of physics-informed machine learning

Figure 1: 物理信息机器学习的各种应用

模拟和解决方案。随着研究的继续，PINN 的多功能性可能会扩展，进一步巩固其在解决各个领域具有挑战性的 PDE 问题中的作用（表 1）。

1.1 物理信息神经网络 (PINN) 基础

1.1.1 问题表述

PINN 是一类将通常由 PDE 描述的领域特定知识整合到神经网络训练过程中的机器学习方法。PINN 背后的核心思想是在神经网络训练期间将物理系统的控制方程作为约束强制执行（图 2）。

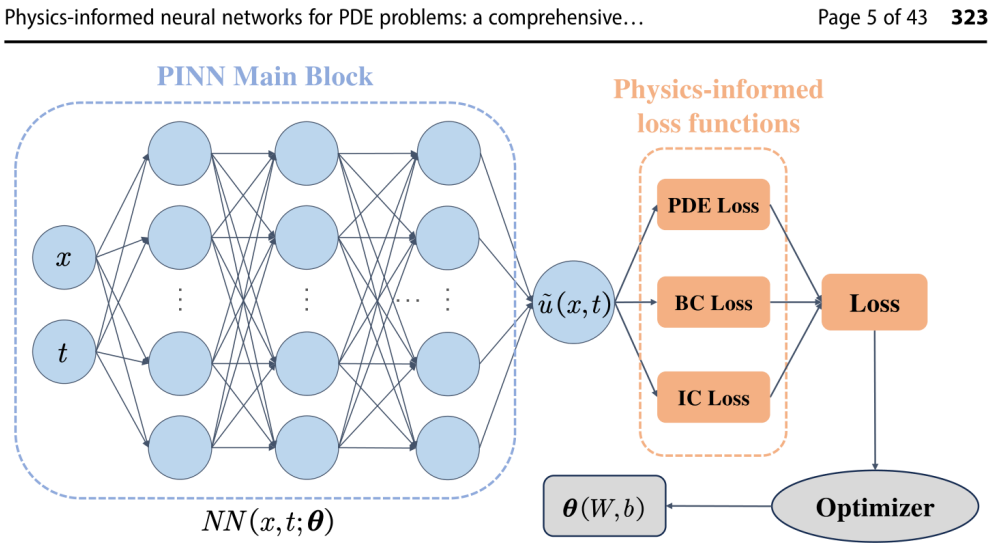

Fig. 2 The framework of PINNs

Figure 2: PINN 的框架

在求解 PDE 约束优化问题的背景下，解 $u(x,t)$ 代表状态变量，其中 x 表示空间坐标，t 表示时间。相关的 PDE 通常可以写为：

$$
\begin{aligned}
f\left(x,t,\frac{\partial}{\partial x}u,\frac{\partial}{\partial t}u,\lambda\right)=0, \quad x\in\Omega, t\in[0,T],\\
u(x,0)=h(x), \quad x\in\Omega,\\
u(x,t)=g(x,t), \quad x\in\partial\Omega, t\in[0,T], (3)
\end{aligned}
\tag{3}
$$

其中 ∂ 表示微分算子，$\lambda=[\lambda_1,\lambda_2,\cdots,\lambda_n]$ 表示 PDE 的参数，f 表示 PDE 的残差。Ω 和 $\partial\Omega$ 分别表示求解域及其边界。初始条件由 $h(x)$ 给出，边界条件为 $g(x,t)$。

利用神经网络 $\tilde{u}(x,t;\theta)$ 来近似解 $u(x,t)$，其中 θ 表示神经网络参数。为了将此 PDE 整合到神经网络训练中，我们可以定义一个由两项组成的损失函数：强制神经网络拟合可用数据的数据驱动项，以及强制 PDE 约束的物理信息项。损失函数可以公式化为：

$$
\mathcal{L}=\mathcal{L}_{\text{physics}}+\mathcal{L}_{\text{data}}, \tag{4}
$$

这里，$\mathcal{L}_{\text{physics}}$ 强制执行如下 PDE 约束：

$$
\mathcal{L}_{\text{physics}}=w_f\mathcal{L}_f(\theta;\mathcal{T}_f)+w_b\mathcal{L}_b(\theta;\mathcal{T}_b)+w_i\mathcal{L}_i(\theta;\mathcal{T}_i), \tag{5}
$$

Table 1: 本文中使用的符号

符号	描述
u	PDE 问题的未知解
x	系统域中的空间坐标
t	时间变量，通常用于时变问题
λ	由 PDE 控制的物理系统参数
θ	神经网络参数
Ω	物理系统的域
$\mathcal{L}_f$	残差损失
$\mathcal{L}_b$	边界条件损失
$\mathcal{L}_i$	初始条件损失

其中

$$\begin{aligned}\mathcal{L}_f(\theta;\mathcal{T}_f) &= \frac{1}{|\mathcal{T}_f|}\sum_{x_f\in\mathcal{T}_f}|f(x_f,t,\lambda)|^2,\\ \mathcal{L}_b(\theta;\mathcal{T}_b) &= \frac{1}{|\mathcal{T}_b|}\sum_{x_b\in\mathcal{T}_b}|\tilde{u}(x_b,t;\theta)-g(x_b,t)|^2,\\ \mathcal{L}_i(\theta;\mathcal{T}_i) &= \frac{1}{|\mathcal{T}_i|}\sum_{x_i\in\mathcal{T}_i}|\tilde{u}(x_i,0;\theta)-h(x_i)|^2,\end{aligned} \tag{6}$$

其中 w_f、w_b 和 w_i 是权重；$\mathcal{T}_f$、$\mathcal{T}_b$、$\mathcal{T}_i$ 分别是域内部、边界条件和初始条件的样本集。$\mathcal{L}_{\text{data}}$ 衡量预测数据与观测数据之间的误差：

$$\mathcal{L}_{\text{data}} = \frac{1}{|\mathcal{T}_d|}\sum_{x_d\in\mathcal{T}_d}|\tilde{u}(x_d,t;\theta)-u(x_d,t)|^2, \tag{7}$$

其中 $\{u(x_d,t)\}_{x_d\in\mathcal{T}_d}$ 表示已知的观测数据。通过使用梯度下降等技术最小化此组合损失函数，PINN 可以直接从数据中学习系统的底层物理机制，使其成为解决反问题和发现复杂物理系统中隐藏模式的有力工具。

1.1.2 PINN 的性能评估

PINN 的评估涉及几个关键指标，用于评估解的准确性以及网络遵守控制方程中嵌入的底层物理原理的程度。这些指标对于确定 PINN 在求解复杂 PDE 时的有效性至关重要，同时确保解保持物理一致性并遵守边界和初始条件。下面，我们详细阐述评估 PINN 性能时常用的关键评估指标。

- **L^2 相对误差**
  L^2 相对误差衡量向量空间中的相对误差。在数学上，PINN 预测值 $\tilde{u}(x,t;\theta)$ 与真实解 $u(x,t)$ 之间的 L^2 相对误差可以表示如下：

$$\text{L2 error} = \frac{\|\tilde{u}(x,t;\theta)-u(x,t)\|_2}{\|u(x,t)\|_2} = \frac{\sqrt{\sum_{i=1}^N|\tilde{u}(x_i,t_i;\theta)-u(x_i,t_i)|^2}}{\sqrt{\sum_{i=1}^N|u(x_i,t_i)|^2}}, \tag{8}$$

  其中 $\|\cdot\|_2$ 表示 L^2 范数。

- **均方根误差 (RMSE)**
  RMSE 作为量化预测误差离散度的关键指标。它是回归和机器学习模型中常用的指标，用于评估预测的准确性。在数学上，RMSE 定义为：

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^N|\tilde{u}(x_i,t_i;\theta)-u(x_i,t_i)|^2} \tag{9}$$

- **PDE 残差**
  PDE 残差衡量预测解满足 PDE 的程度。它表示模型的解在空间和时间上的给定点偏离满足控制方程的程度。在数学上，对于形式为 $\mathcal{L}(u(x,t)) = f(x,t)$ 的一般 PDE，其中 $\mathcal{L}$ 是微分算子，$f(x,t)$ 是 PDE 的源项。PDE 残差表示如下：

$$\text{PDE residual} = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{L}(\tilde{u}(x_i,t_i;\theta)) - f(x_i,t_i))^2. \tag{10}$$

1.2 论文组织和贡献

本文的其余部分结构如下。第 2 节对现有的 PINN 基 PDE 求解器文献进行了全面分析。第 3 节展示了 PINN 的应用，第 4 节介绍了目前公开用于执行 PINN 的工具和代码。第 5 节讨论了该领域的挑战和一些可能的未来方向。最后，第 6 节总结了全文。

本综述旨在提供 PINN 的全面概述，重点介绍其理论基础、应用和进展。这项工作的主要贡献如下：

- **PINN 简介**：本文清晰地介绍了 PINN 的基本原理，解释了它们如何将深度学习技术与物理定律相结合来解决 PDE 问题。
- **关键方法论讨论**：综述深入探讨了 PINN 中的各种方法论，包括架构、自适应数据重采样方法、损失函数设计、特征嵌入等。
- **应用回顾**：本文调查了 PINN 在流体动力学、材料科学和生物学等多个领域的广泛应用，展示了它们在应对现实世界挑战方面的多功能性。
- **最新进展**：本文总结了 PINN 研究的最新进展，包括计算精度和效率的提高、泛化能力。
- **新兴范式和未来方向**：重点介绍了 PINN 与算子学习等新兴范式的融合。此外，本文讨论了 PINN 目前面临的挑战，并提出了未来研究的潜在方向，以提高其适用性和性能。

2 用于 PDE 问题的 PINN 求解器

早在 20 世纪，一些研究人员就开始探索应用机器学习方法来求解偏微分方程。其概念是利用神经网络强大的近似能力，使用时间和空间位置作为输入。通过网络的映射，最终可以获得 PDE 的解。1994 年，Dissanayake 和 Phan-Thien (1994) 是采用基于神经网络的方法求解 PDE 的先驱。他们利用神经网络的“通用近似器”特性，通过配点法将求解 PDE 的数值问题转化为无约束最小化问题。

随后，Lagaris 等人 (1998) 引入了一种人工神经网络模型，用于求解具有各种边界条件的 PDE。这种方法在解的准确性、泛化能力和所需参数数量方面显示出巨大的潜力。Monterola 和 Saloma (2003) 引入了一种基于无监督神经网络的方法来求解非线性薛定谔方程，并提供了误差的上限估计。Aarts 和 van der Veer (2001) 开发了具有特定结构的神经网络，该结构整合了 PDE 的知识及其边界和初始条件，同时训练多个神经网络。

然而，这些早期方法受到神经网络结构和算法限制的制约，这极大地阻碍了它们处理更复杂问题的能力。

近年来，随着人工智能技术的快速发展，GPU 和 TPU 等高性能计算硬件的出现，以及 TensorFlow (Abadi 等 2016) 和 PyTorch (Paszke 等 2019) 等深度学习框架的推出和不断改进，深度学习在求解 PDE 中的应用取得了进一步的进展。2019 年，Raissi 等人 (2019) 提出了 PINN 模型，开创了使用神经网络求解 PDE 的新范式。通过将 PDE 及其初始和边界条件整合到神经网络的损失函数中，并使用自动微分和反向传播来指导训练过程，该模型可以在统一的框架内无缝地结合基于物理的约束和观测数据。

图 3 展示了基于神经网络求解 PDE 方法的时间线。它清晰简洁地直观展示了这些方法的演变，强调了神经网络现在在现代 PDE 求解方法中发挥的核心和变革性作用。时间线突出了神经网络的逐步整合，展示了它们在解决 PDE 复杂性方面日益增长的影响力和有效性（表 2）。

Table 2: PDE 问题的 PINN 求解器概述

方法		代表性工作
PINN 的架构	MLP	vanilla PINN (Raissi et al. 2019; He et al. 2020)
	CNN	PhyGeoNet (Gao et al. 2021; Fang 2022), Spline-PINN (Wandel et al. 2022), PICNN (Lei et al. 2023), f-PICNN (Yuan et al. 2024)
	RNN	PhyCRNet (Ren et al. 2022; Mavi et al. 2023)
	GAN	PI-GANs (Yang et al. 2020; Gao and Ng 2022)
	KAN	Wang et al. (2024c), Shukla et al. (2024), Rigas et al. (2024)
	Transformer	PIT (Santos et al. 2023), PINNsFormer (Zhao et al. 2024), Transolver (Wu et al. 2024)
	PINN 的其他架构	NAS-PINN (Wang and Zhong 2024), SPINN (Cho et al. 2022), sparabreak PirateNets (Wang et al. 2024b)
区域分解		XPINNs (Karniadakis and Em 2020), cPINNs (Jagtap et al. 2020a), (Wu et al. 2022)
PINN 的激活函数		Jagtap et al. (2020b), PIKFNNs (Fu et al. 2024), PAFs (Abbasi and Andersen 2024; Zeng et al. 2022)
自适应数据重采样		RAR (Lu et al. 2021b; Nabian et al. 2021), RAR-D (Wu et al. 2023), R3 sampling (Daw et al. 2023) DAS-PINNs (Tang et al. 2023), DMIS (Yang et al. 2023), GAS (Jiao et al. 2024) PINNACLE (Lau et al. 2024), AAS (Tang et al. 2024)
PINN 的损失函数	损失重加权	Wang et al. (2021c, 2022a), lbPINNs (Xiang et al. 2022) SelectNet (Gu et al. 2021), SA-PINNs (McClenny and Braga-Neto 2023), LA-PINN (Song et al. 2024)
	新型损失函数	gPINNs (Yu et al. 2022; Wang et al. 2022b, 2024a,d)
特征嵌入和增强		Wang et al. (2021a), Li et al. (2024a, 2024c) PD-PINN (Peng et al. 2020), sf-PINN (Wong et al. 2024), DaPINN (Guan et al. 2023)

2.1 PINN 的架构

各种为应对特定挑战而量身定制的架构已在 PINN 中涌现。这些不同的 PINN 架构代表了传统神经网络结构的创新改编，专门设计用于增强求解 PDE 问题的性能。本节探讨并比较了各种 PINN 架构，重点介绍了它们独特的设计原则、计算效率以及在不同领域的适用性。PINN 涵盖的主要架构主要包括多层感知机（MLP）（Hornik 等 1989）、卷积神经网络（CNN）（Krizhevsky 等 2012；Gu 等 2018）、循环神经网络（RNN）（Giles 等 1994；Schuster 和 Paliwal 1997），以及其他如生成对抗网络（GAN）（Goodfellow 等 2014）、Transformer（Vaswani 等 2017）。

2.1.1 多层感知机 (MLP)

多层感知机（MLP）（Hornik 等 1989）是人工神经网络中的一种基础架构，由相互连接的神经元层组成，可以通过迭代调整权重和使用激活函数来近似复杂的非线性函数。其理论基础的核心是通用近似定理（Cybenko 1989；Chen 和 Chen 1995），该定理指出，具有适当激活函数和有限数量神经元的单隐层 MLP 可以近似任何连续函数。MLP 的输入由表示输入特征的向量 x 组成（图 4）。

x 的每个元素对应于输入数据的一个特征。对于每个隐藏层 l，其中 $l = 1, 2, extdots, L$，计算：

$$\begin{aligned} z^{(l)} &= W^{(l)} a^{(l-1)} + b^{(l)}, \\ a^{(l)} &= \sigma^{(l)}(z^{(l)}), \end{aligned} \tag{11}$$

其中权重矩阵 W 和偏置向量 b 是参数，$\sigma()$ 是非线性激活函数。输出层计算最终的预测向量，

$$y = W^{(L+1)} a^{(L)} + b^{(L+1)}. \tag{12}$$

全连接前馈网络（FNN）架构（也称为 MLP）的特征，例如层数和每层神经元数，通常是针对不同的 PDE 问题根据经验确定的。在 vanilla PINN（Raissi 等 2019）中，使用了不同类型的

2.1.1 Multilayer perceptron (MLP)

The multilayer perceptron (MLP) (Hornik et al. 1989) is a fundamental architecture within artificial neural networks, consisting of layers of interconnected neurons that can approximate complex non-linear functions by adjusting weights iteratively and using activation functions. At the core of its theoretical foundation lies the Universal Approximation Theorem (Cybenko 1989; Chen and Chen 1995), which states that a single hidden layer MLP can approximate any continuous function with suitable activation functions and a finite number of neurons. The input to the MLP consists of a vector $\boldsymbol{x}$ representing the input features (Fig. 4). Each element of $\boldsymbol{x}$ corresponds to a feature of the input data. For each hidden layer l, where $l = 1, 2, \ldots, L$, compute:

$$\begin{aligned} \boldsymbol{z}^{(l)} &= \boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)}, \\ \boldsymbol{a}^{(l)} &= \sigma^{(l)}(\boldsymbol{z}^{(l)}), \end{aligned} \tag{11}$$

where the weight matrix $\boldsymbol{W}$ and the bias vector $\boldsymbol{b}$ are parameters and $\sigma()$ is the non-linear activation function. The output layer computes the final prediction vector,

$$\boldsymbol{y} = \boldsymbol{W}^{(L+1)}\boldsymbol{a}^{(L)} + \boldsymbol{b}^{(L+1)}. \tag{12}$$

The characteristics of the fully connected feed-forward networks (FNN) architecture, also known as MLP, such as the number of layers and neurons per layer, are typically determined empirically for different PDE problems. In the vanilla PINN (Raissi et al. 2019), different typologies of FNN are ultilized, for example, a 5 layer FNN with 100 neurons per layer for the Schrodinger equation, a 4 layer FNN and 200 neurons per layer for the Allen–Cahn equation. In He et al. (2020), the effect of neural network size on prediction accuracy is investigated. It indicates that employing excessively small or large sizes for PINNs can lead

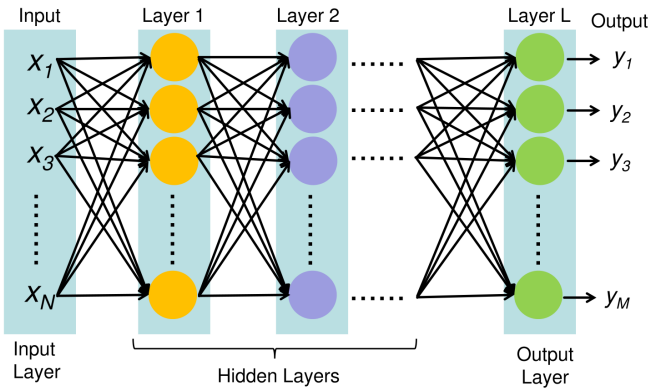

Fig. 4 The framework of MLP

Figure 3: MLP 的框架

FNN，例如，用于薛定谔方程的每层 100 个神经元的 5 层 FNN，用于 Allen—Cahn 方程的每层 200 个神经元的 4 层 FNN。在 He 等 (2020) 中，研究了神经网络大小对预测精度的影响。这表明，为 PINN 使用过小或过大的尺寸会导致性能下降。由于 PINN 的黑盒性质，关于神经网络大小如何影响各种问题的性能的理论分析仍未得到探索。解决这个问题至关重要但也具有挑战性。

2.1.2 卷积神经网络（CNN）

卷积神经网络（CNN）(Li 等 2022）是一种专门设计用于处理具有网格状拓扑结构的数据（特别是图像数据）的深度学习模型。CNN 的核心在于使用卷积层和池化层来提取和压缩特征，使其在图像分类、目标检测和图像分割等任务中非常有效。卷积层通过卷积操作（即应用一组可学习的滤波器）提取局部特征，而池化层通过下采样操作减少特征图的空间维度，从而降低计算复杂度并防止过拟合。CNN 的成功很大程度上归功于它们自动学习图像中分层特征的能力，从低级特征（如边缘）到高级特征（如物体部分）。

卷积操作涉及在输入数据上滑动可学习的滤波器（或卷积核），并计算滤波器与输入数据之

间的点积，描述如下：

$$M = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \tag{13}$$

其中 I 是输入图像，K 是卷积核，M 是输出特征图，$*$ 表示卷积操作。然后池化层减少特征图的空间维度，降低计算复杂度并防止过拟合，

$$P(i, j) = \max_{0 \le m < p} \max_{0 \le n < p} M(s \cdot i + m, s \cdot j + n) \tag{14}$$

其中 P 是池化输出，M 是特征图，p 是池化窗口大小，s 是步长。

许多研究人员探索了将 CNN 架构集成到 PINN 中。Gao 等 (2021) 提出了 PhyGeoNet，这是一种物理信息几何自适应 CNN，用于求解不规则域上的参数化稳态 PDE。它使用椭圆坐标映射将不规则几何形状转换为规则几何形状，从而能够应用标准 CNN 架构。PhyGeoNet 可以在没有标记数据的情况下学习热传导、Navier–Stokes 和泊松方程的解。在 Fang (2022) 中，介绍了一种用于求解 PDE 的高效混合 PINN。它结合了 CNN 和有限体积法的思想，使用局部拟合方法来近似微分算子，而不是自动微分。数值实验验证了其正确性和效率，在计算时间和精度方面均显示出优于传统 PINN 的显著优势。Wandel 等 (2022) 介绍了 Spline-PINN，它使用 Hermite 样条核结合了 PINN 和 CNN 的优点。它实现了适合 CNN 的基于网格的表示的连续插值，允许使用物理信息损失在没有预计算数据的情况下进行训练。Spline-PINN 提供快速、准确且可推广到未见域的解，正如在 Navier-Stokes 和阻尼波动方程中所演示的那样。Lei 等 (2023) 提出了一种专门设计用于求解球面上 PDE 的物理信息 CNN (PICNN)。基于深度学习和球谐函数分析的最新进展，作者建立了 Sobolev 范数中 PICNN 近似误差的严格理论界限。他们开发了一种新颖的定位复杂性分析来证明 PICNN 的快速收敛速度。数值实验突出了 PICNN 在球面域上求解 PDE 的有效性。在 Yuan 等 (2024) 中，设计了一种基于 CNN 的 f-PICNN 用于求解时空域中的 PDE。利用非线性卷积单元堆栈和记忆机制，f-PICNN 可以处理具有非线性、剧烈梯度和多尺度特征的 PDE。该网络是无监督的，不需要标记数据。f-PICNN 的功效在各种基准 PDE 中得到了证明，展示了其准确近似解的能力（图 5）。

2.1.3 循环神经网络 (RNN)

循环神经网络（RNN）（Lipton 等 2015）设计用于处理序列数据，其特点是具有循环连接，可以捕捉数据中的时间依赖性。RNN 跨时间步共享权重并引入隐藏状态，使网络能够记忆和利用过去的信息。这种架构擅长诸如时间序列预测、语言建模和序列标注等任务。RNN 的理论基础包括其递归性质，允许信息通过序列传播。尽管在长序列训练期间面临梯度消失和梯度爆炸等挑战，但 RNN 及其变体 [如 LSTM (Hochreiter 和 Schmidhuber 1997; Yu 等 2019) 和 GRU (Chung 等 2014)] 在解决这些问题方面取得了重大进展。

RNN 的基本结构包括输入层、隐藏层和输出层，隐藏层具有循环连接。这些循环连接允许根据当前输入和上一时间步的隐藏状态更新隐藏状态。隐藏状态更新公式如下：

$$h_t = f(W_{xh} x_t + W_{hh} h_{t-1} + b_h), \tag{15}$$

其中 h_t 是时间步 t 的隐藏状态，x_t 是时间步 t 的输入，W_{xh} 是从输入到隐藏层的权重矩阵，W_{hh} 是从隐藏层到隐藏层的权重矩阵，b_h 是隐藏层的偏置，f 是激活函数，通常是 tanh 或 ReLU。输出计算可以公式化为

$$y_t = g(W_{hy} h_t + b_y), \tag{16}$$

其中 y_t 是时间步 t 的输出，W_{hy} 是从隐藏层到输出层的权重矩阵，b_y 是输出层的偏置，g 是激活函数（图 6）。

Ren 等 (2022) 提出了 PhyCRNet，这是一种新颖的物理信息卷积循环学习架构，旨在无需标记数据即可求解时空 PDE。PhyCRNet 利用编码器-解码器卷积 LSTM 网络进行有效的空间特征提取和时间演化。提出的 PhyCRNet 在各种基准问题上进行了评估，证明了其准确求解 PDE 的能力，并优于现有的 PINN 方法。Mavi 等 (2023) 提出了一种具有非局部相互作用的无监督 CNN-LSTM 架构，用于求解 PDE。该网络利用非局部近场动力学微分算子 (PDDO) 作为

CNN (PICNN) specifically designed for solving PDEs on the sphere. Building on recent advancements in deep learning and spherical harmonic analysis, the authors establish rigorous theoretical bounds for the approximation error of PICNNs in Sobolev norms. They develop a novel localization complexity analysis to demonstrate fast convergence rates for PICNNs. The numerical experiments highlights the PICNN's effectiveness in solving PDEs on spherical domains. In Yuan et al. (2024), a CNN based f-PICNN is designed for solving PDEs in space-time domains. Leveraging a stack of nonlinear convolutional units and a memory mechanism, f-PICNN can handle PDEs with nonlinearity, drastic gradients, and multiscale characteristics. The network is unsupervised and does not require labeled data. The efficacy of f-PICNN is demonstrated in various benchmark PDEs, showcasing its capability to accurately approximate solutions (Fig. 5).

2.1.3 Recurrent neural networks (RNN)

Recurrent Neural Networks (RNNs) (Lipton et al. 2015) are designed for processing sequential data and are characterized by their recurrent connections, which capture temporal dependencies in the data. RNNs share weights across time steps and introduce hidden states, enabling the network to remember and utilize past information. This architecture excels in tasks such as time series prediction, language modeling, and sequence labeling. The theoretical foundation of RNNs includes their recursive nature, allowing information to propagate through the sequence. Despite facing challenges like vanishing and exploding gradients during training on long sequences, RNNs and their variants [such as LSTM (Hochreiter and Schmidhuber 1997; Yu et al. 2019) and GRU (Chung et al. 2014)] have made significant progress in addressing these issues.

The basic structure of an RNN consists of an input layer, a hidden layer, and an output layer, with recurrent connections in the hidden layer. These recurrent connections allow the

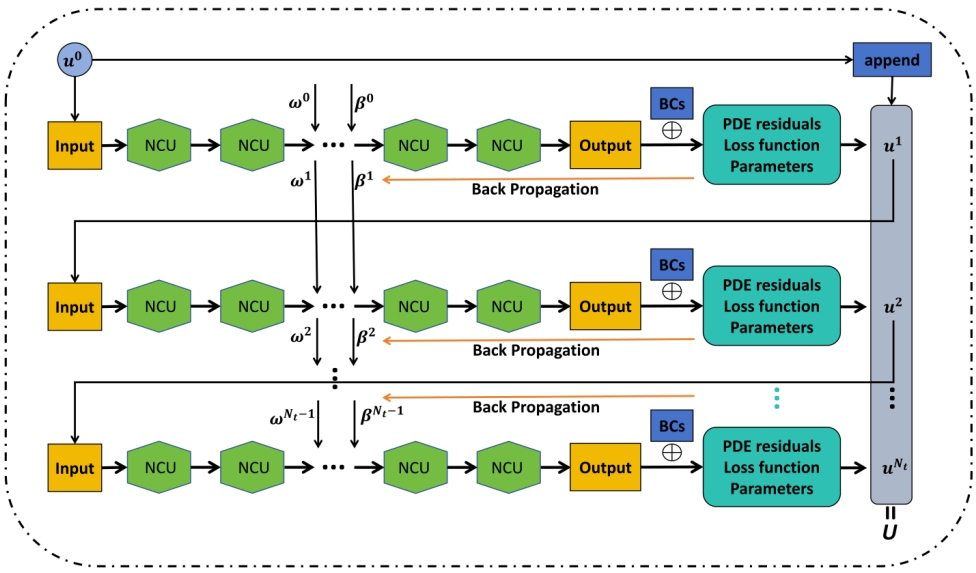

Fig. 5 The neural network architecture of the f-PICNN (Yuan et al. 2024) for time-dependent PDEs, with

Figure 4: 用于时变 PDE 的 f-PICNN (Yuan 等 2024) 神经网络架构，具有非线性卷积单元 (NCU)

hidden state to be updated based on the current input and the hidden state from the previous time step. The hidden state update formula is as follows,

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{15}$$

where h_t is the hidden state at time step t, x_t is the input at time step t, W_{xh} is the weight matrix from input to hidden layer, W_{hh} is the weight matrix from hidden layer to hidden layer, b_h is the bias of the hidden layer, and f is the activation function, typically tanh or ReLU. The output calculation can be formulated as

$$y_t = g(W_{hy}h_t + b_y), \tag{16}$$

where y_t is the output at time step t W_{hy} is the weight matrix from hidden layer to output layer b_y is the bias of the output layer g is the activation function (Fig. 6).

Ren et al. (2022) proposed PhyCRNet, a novel physics-informed convolutional-recurrent learning architecture designed to solve spatiotemporal PDEs without requiring labeled data. PhyCRNet utilizes an encoder-decoder convolutional LSTM network for efficient spatial feature extraction and temporal evolution. The proposed PhyCRNet is evaluated on various benchmark problems, demonstrating its ability to accurately solve PDEs and outperforms existing PINN approaches. Mavi et al. (2023) proposed an unsupervised CNN-LSTM architecture with nonlocal interactions for solving PDEs. The network evaluates derivatives of the field variable by utilizing the nonlocal peridynamic differential operator (PDDO) as a convolutional filter. The architecture captures time dynamics in a reduced latent space through encoder-decoder layers with a Convolutional LSTM layer in between. The physics is enforced at the network's output and in the latent space. Benchmark PDEs demonstrate the training performance and extrapolation capability of this novel architecture.

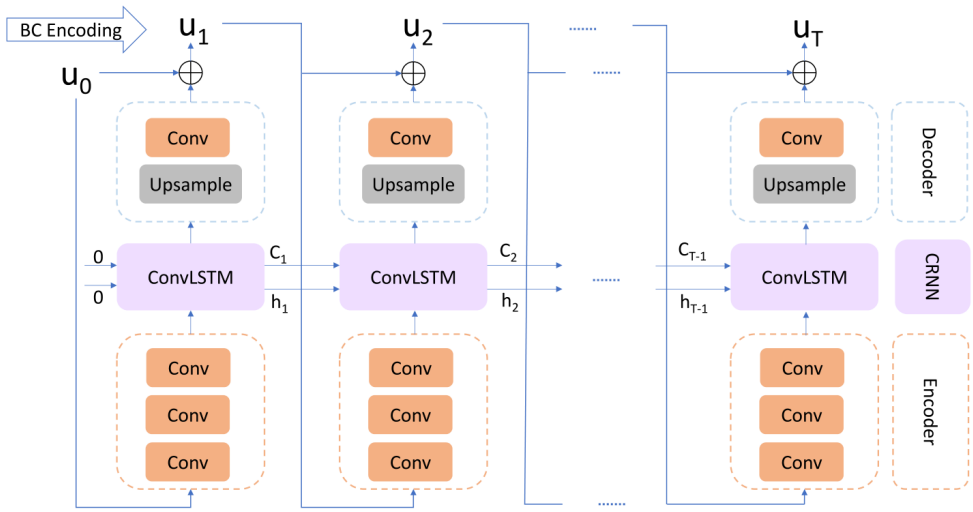

Fig. 6 The framework of PhyCRNet (Ren et al. 2022)

Figure 5: PhyCRNet (Ren 等 2022) 的框架

卷积滤波器来评估场变量的导数。该架构通过中间带有卷积 LSTM 层的编码器-解码器层在降维潜在空间中捕获时间动态。物理约束在网络的输出和潜在空间中强制执行。基准 PDE 证明了这种新颖架构的训练性能和外推能力。

2.1.4 生成对抗网络 (GAN)

GAN 是一类深度生成模型，旨在生成类似于给定真实数据集的新的合成数据样本。该架构由两个神经网络组成：生成器和判别器，它们通过对抗过程同时进行训练。生成器创建数据样本，目的是欺骗判别器使其相信这些样本是真实的，而判别器的目标是区分真实样本和合成样本。这种竞争框架促进了日益逼真的数据的生成（Creswell 等 2018）。

GAN 框架已在一些工作中集成到 PINN 中。Yang 等 (2020) 提出了物理信息 GAN (PI-GANs)，用于基于有限的离散测量值求解正向、反向和混合随机问题。PI-GANs 以随机微分方程 (SDE) 的形式编码物理定律。PI-GANs 使用三个生成器，即使在输入噪声与目标过程的有效维数之间存在不匹配的情况下，也能表现出对随机过程的良好近似。Gao 和 Ng (2022) 考察了一种应用 Wasserstein 生成对抗网络 (WGAN) 量化 PDE 解中不确定性的物理信息方法。该方法在对抗网络的判别器内利用 groupsort 激活函数，采用网络生成器来学习基于初始/边界数据的 PDE 解中存在的不确定性。理论发现表明，在温和条件下，如果样本数量充足，计算出的生成器的泛化误差往往会高概率地趋向于网络的近似误差。数值实验验证了这些理论见解，证明了即使在噪声占主导地位时，所提出的模型也能够检测不确定性并近似精确解。

2.1.5 Kolmogorov—Arnold 网络 (KAN)

Kolmogorov—Arnold 网络 (KAN) (Liu 等 2024b) 建立在 Kolmogorov—Arnold 表示定理 (KART) 之上。该定理规定，任何多元连续函数都可以分解为单变量连续函数加上加法运算的有限组合。据此，KAN 将高维函数建模为单变量函数的和。通过包含可学习的边缘激活函数（即权重），KAN 提高了可解释性和准确性。用数学术语来说，对于平滑函数 $f:[0,1]^n \to \mathbb{R}$，可以写成：

$$f(x) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right), \tag{17}$$

其中 $x = (x_1, \ldots, x_n)$ 表示输入向量，Φ_q 和 $\phi_{q,p}$ 表示单变量函数。

相比之下，广泛采用的 MLP 使用通用近似定理 (UAT) 作为其理论基础，通常将激活函数放置在神经元（即节点）处。然后每个神经元接收输入并通过激活函数（如 ReLU、Sigmoid 或 Tanh）产生输出。符号上，MLP 可以表示为：

$$\mathrm{MLP}(x) = (W_L \sigma(W_{L-1} \ldots \sigma(W_1 x + b_1) \ldots) + b_L), \tag{18}$$

其中 W_l 和 b_l 分别是第 l 层的权重矩阵和偏置，σ 是激活函数。这种以节点为中心的方案通常被认为是不透明的，因为内部工作原理并不直观易懂。

另一方面，KAN 将其激活函数定位在网络边缘而不是节点上。因此，每个单独的输入通过可学习的单变量映射（例如 B 样条函数），这些输出随后由加法节点组合。一般的 KAN 可以写成：

$$\mathrm{KAN}(x) = (\Phi_{L-1} \circ \cdots \circ \Phi_1 \circ \Phi_0)(x), \tag{19}$$

其中 Φ_l 表示第 l 层的变换。具体来说，

$$x_{l+1} = [\phi_{l,1,1}(x_l), \ldots, \phi_{l,n_{l+1},n_l}(x_l)], \tag{20}$$

其中每个 $\phi_{l,j,i}$ 表示第 l 层从节点 i 到节点 j 的单变量激活。这种结构提高了透明度，使研究人员能够观察单变量函数是如何配置的以及数据如何流经网络。

由于 KAN 将高维任务分解为较小的单变量部分，因此非常适合复杂、非线性的 PDE 问题。与传统的 MLP 相比，KAN 允许更清晰地可视化其内部函数组合，从而更容易改进模型的特定部分以获得更好的 PDE 解。此外，KAN 可以从其学习到的表示中推导出符号公式，有助于验

证解析 PDE 解。因此，KAN 不仅仅是一个解决问题的框架，还是理论洞察和教育目的的宝贵工具。

最近，一些基于 KAN 的 PINN 架构被提出，显示出巨大的潜力。Wang 等 (2024c) 提出了 Kolmogorov–Arnold 信息神经网络 (KINN)。通过系统的数值实验，证明了 KINN 在求解各种 PDE（包括涉及多尺度、奇异性、应力集中、非线性超弹性和异质属性的 PDE）时，在准确性和收敛速度方面优于基于 MLP 的 PINN。作者还强调了 KINN 在处理复杂几何形状方面的潜力，并提出了未来的研究方向，例如结合有限元方法的网格自适应技术来增强 KINN 的能力。它强调了 KAN 彻底改变 PDE 人工智能的潜力。Shukla 等 (2024) 介绍了 KAN 的修改版本，称为 Chebyshev KAN，并在准确性、效率和鲁棒性方面与传统 MLP 进行了比较。该研究在各种基准测试（包括计算物理中的正向和反向问题）上系统地评估了这些模型。结果表明，虽然基于 B 样条的原始 KAN 在准确性和效率方面表现不佳，但修改后的 Chebyshev KAN 表现出与 MLP 相当的性能，尽管在某些条件下存在潜在的发散问题。该论文通过信息瓶颈理论的视角进一步探讨了这些模型的学习动态，揭示了对其训练过程的见解。在 Rigas 等 (2024) 中，提出了一种物理信息 Kolmogorov-Arnold 网络的自适应训练方案，其中包括基于残差的注意力 (RBA)、配点的自适应重采样以及缓解 KAN 中网格更新相关挑战的自适应状态转换方法等技术。通过对各种 PDE 的比较实验，该研究表明 PIKAN 在准确性和收敛速度方面显著优于传统的基于 MLP 的 PINN，特别是对于涉及多尺度动力学、奇异性和异质材料的问题。

2.1.6 Transformer

Transformer (Vaswani 等 2017) 架构代表了 NLP 领域的重大突破，它通过自注意力机制实现了输入序列的并行处理，从而取代了 RNN。这一创新对于有效解决长距离依赖关系至关重要，克服了序列模型固有的计算限制。Transformer 的核心是自注意力模块，它允许序列中的每个位置关注上一层的所有其他位置，从而在单个操作中捕获全局上下文。该模型遵循编码器-解码器结构，利用堆叠的自注意力层和前馈网络。编码器处理输入序列的上下文，而解码器利用此上下文生成输出序列。集成了位置编码以传达有关序列顺序的信息。Transformer 在 BERT、GPT 和 T5 等模型的开发中发挥了关键作用，这些模型在机器翻译、文本摘要和问答等任务中树立了新标准。这些进步突显了 Transformer 架构在重塑现代 NLP 方面的变革性影响。

Santos 等 (2023) 介绍了物理信息 Transformer (PIT) 模型，该模型基于 Transformer 架构用于学习 PDE 的解算子。PIT 利用注意力机制来捕捉初始条件和查询点之间的关系，从而提高模型的泛化能力。它在处理输入域和查询域的离散化时保持不变性，同时通过交叉注意力块实现交互。与已建立的基于物理的 DeepONet (Lu 等 2021a) 相比，PIT 在多个方面显示出显著改进，即使在训练资源有限的情况下也能在高维问题中表现出有效的性能。PIT 还提供了调整每个训练实例的查询和输入样本数量以及离散化输入函数的灵活性——这是 DeepONet 等其他算子学习模型不支持的功能。与 PINN 网络中的算子学习相比，PIT 解决了现有解决方案性能不佳的问题，并消除了对初始和边界条件进行假设的需要。基于 Transformer 的架构包括编码器和解码器。为了减少与 PINN 损失相关的自动划分成本，PIT 避免了对查询和输入点的自注意力。作者还将 PIT 与传统的基于物理的 DeepONet 以及 Wang 等 (2021c) 的改进模型进行了比较。这些模型在一维 Burgers 方程和二维热方程上进行了测试。结果表明，虽然 PIT 的计算成本与 DeepONet 相似，但其求解性能得到了显著提高（图 7）。

此外，Zhao 等 (2024) 提出了 PINNsFormer，这是一种基于 Transformer 的架构，利用多头注意力机制捕捉时间依赖性，增强了 PDE 解的泛化能力和准确性。PINNsFormer 将逐点输入转换为伪序列，使模型能够学习时间依赖性并更有效地传播初始条件约束。一项关键创新是引入了一种新的激活函数 Wavelet，它预见了深度神经网络中的傅里叶分解，提高了近似能力。该框架还支持各种学习方案，例如神经切线核 (NTK)，以进一步提高性能。实证结果表明，PINNsFormer 在泛化和准确性方面优于传统的 PINN，特别是在传统模型失效的场景中。它在分布外数据和高维 PDE 的预测精度方面表现出显著提高。值得注意的是，PINNsFormer 降低了总损失项，表明优化效率更高。损失景观分析揭示了一个更平滑的景观，局部最小值更少，表明更容易收敛到全局最小值。Wu 等 (2024) 介绍了 Transolver，这是一个旨在高效准确地求解复杂几何形状上的 PDE 的模型。Transolver 利用 Transformer 架构，结合了 PhysicsAttention，这是一种自适应机制，将离散化域分割成可学习的切片。这些切片将具有相似物理状态的网格点分组，使模型能够以线性计算复杂度捕捉复杂的物理相关性。该架构用 PhysicsAttention 取

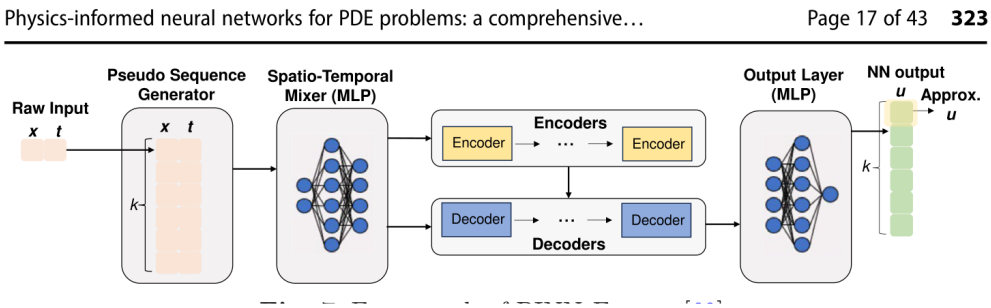

Fig. 7 Framework of PINNsFormer (Zhao et al. 2024)

Figure 6: PINNsFormer (Zhao 等 2024) 的框架

代了标准注意力，处理编码离散化域内相互作用的物理感知 token。对六个基准和大规模工业模拟的广泛实验证实了 Transolver 始终如一的最先进性能，相对增益为 22

2.1.7 PINN 的其他架构

除了上述讨论的神经网络框架外，还开发了各种新颖的 PINN 架构（图 8）。

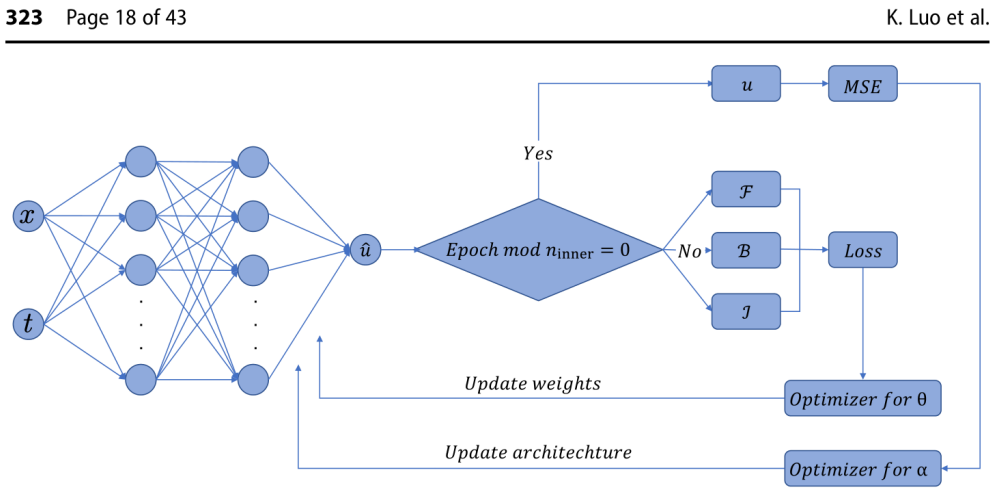

Fig. 8 Framework of PINN with Neural architecture search (NAS-PINN) (Wang and Zhong 2024)

Figure 7: 具有神经架构搜索 (NAS-PINN) 的 PINN 框架 (Wang 和 Zhong 2024)

神经架构搜索 (NAS) 已被用于优化 PINN。例如，Wang 等 (2022c) 介绍了 Auto-PINN，这是一种用于优化 PINN 超参数的系统化自动化方法。Auto-PINN 采用神经架构搜索 (NAS) 技术来避免手动或穷举搜索与 PINN 相关的超参数空间。通过使用标准 PDE 基准进行一系列预实验，他们探索了 PINN 中的结构-性能关系。他们发现不同的超参数可以解耦，并且 PINN 的训练损失函数可以作为有效的搜索目标。在 Wang 和 Zhong (2024) 中，提出了一种新颖的框架 NAS-PINN，它也利用 NAS 来优化 PINN 的设计。通过将搜索空间放宽到连续域并利用掩码来实现不同形状的张量加法，NAS-PINN 采用了双层优化过程。内层循环优化神经网络的权重和偏置，而外层循环更新架构参数。这种方法自动化了最佳神经网络架构的选择，提高了 PINN 的效率和准确性。通过广泛的实验，证明了 NAS-PINN 实现了优于手动设计架构的性能，特别是在处理复杂和不规则域时。

此外，Cho 等 (2022) 介绍了可分离 PINN (SPINN)，这是一种旨在减轻求解 PDE 时遇到的维数灾难的创新架构。SPINN 利用前向模式自动微分显着减少计算开销，按轴逐个处理输入而不是逐点处理。这种方法减少了所需的网络前向传递次数，使计算和内存成本随网格分辨率的增长更加缓慢，从而增强了可扩展性。实验结果表明，SPINN 大幅减少了训练运行时间，同时在各种 PDE 系统中实现了与标准 PINN 相当的精度。这一进步使 SPINN 成为高效处理高维 PDE 问题的有希望的解决方案。Wang 等 (2024b) 提出了一种新颖的 PINN 架构 PirateNets，旨在提高训练稳定性和效率。PirateNets 利用自适应残差连接，使网络能够从浅层模型开始并在训练期间逐步加深。这种创新方法缓解了与传统 MLP 相关的问题，传统 MLP 通常难以进行导数训练和不稳定地最小化 PDE 残差损失。自适应初始化方案允许网络编码特定于底层 PDE 系统的合适归纳偏置，从而提高模型性能。

2.1.8 区域分解

区域分解是一种用于将复杂求解域划分为多个较小子域以提高计算效率和求解精度的技术。它在 PINN 中的应用特别适合处理大规模和复杂的问题。通过将整个求解域分解为几个子域，可以在每个子域中独立求解问题，并且并行计算可以加快求解过程。在相邻子域之间引入界面条件以确保解的连续性和一致性。区域分解方法可以灵活地适应复杂的几何形状和边界条件，使 PINN 更加通用和高效。它是求解复杂 PDE 问题的一种有前途的方法（图 9）。

接下来，我们将介绍区域分解方法的一些关键公式。整个求解域 Ω 分为 N 个子域 Ω_i，

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \tag{21}$$

物理方程在每个子域 Ω_i 内求解，

$$\mathcal{F}_i(u_i) = 0 \quad \text{in} \quad \Omega_i, \tag{22}$$

其中 $\mathcal{F}_i$ 表示子域 Ω_i 中的物理方程，u_i 是子域 Ω_i 中的解。引入相邻子域之间的界面条件以确保解的连续性和一致性。令 Γ_{ij} 为子域 Ω_i 和 Ω_j 之间的界面，界面条件定义为，

$$u_i = u_j \quad \text{on} \quad \Gamma_{ij}. \tag{23}$$

那么复合损失函数如下，

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}^i_{\text{physics}} + \sum_{i \neq j} \mathcal{L}^{ij}_{\text{interface}}, \tag{24}$$

其中物理损失 $\mathcal{L}^i_{\text{physics}}$ 在每个子域内计算，界面损失 $\mathcal{L}^{ij}_{\text{interface}}$ 在相邻子域之间的界面上计算。

在 Karniadakis 和 Em (2020) 中，提出了扩展 PINN (XPINNs)，这是一种用于在复杂几何域上求解非线性 PDE 的广义时空区域分解方法。XPINNs 可以应用于任何类型的 PDE，并允许在空间和时间上进行任意区域分解，从而促进高效的并行计算。在每个子域内，部署一个具有最佳选择的超参数的单独神经网络。这使得在具有复杂解的子域中使用更深的网络，而在较简单的区域中使用较浅的网络。Jagtap 等 (2020a) 提出了用于求解非线性守恒律的离散域上的守恒 PINN (cPINNs)。它将计算域划分为离散子域，并在每个子域内应用 PINN，同时强制跨界面的通量连续性以确保全局守恒。cPINN 框架在选择优化算法和训练参数（如残差点的数量和位置、激活函数以及网络宽度和深度）方面提供了额外的灵活性。同样，Wu 等 (2022) 提出了一种基于区域分解的改进深度神经网络方法。它可以解决 PINN 由于使用单个神经网络和梯度病态而在处理大规模复杂问题时的局限性。

2.1.9 PINN 的激活函数

非线性激活函数在 PINN 中至关重要，因为它们使网络能够学习和表示 PDE 中固有的复杂非线性关系。常见的选择包括双曲正切 (tanh)、sigmoid。最近的研究探索了专门的激活函数，如正弦 (sin) 函数。

adapt to complex geometries and boundary conditions, making PINNs more versatile and efficient. It is a promising approach for solving complex PDE problems (Fig. 9).

Next, we will introduce some key formulas for domain decomposition methods. The entire solution domain Ω is divided into N subdomains Ω_i,

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \tag{17}$$

The physical equations are solved within each subdomain Ω_i,

$$\mathcal{F}_i(u_i) = 0 \quad \text{in} \quad \Omega_i, \tag{18}$$

where $\mathcal{F}_i$ represents the physical equations in subdomain Ω_i, and u_i is the solution in subdomain Ω_i. The interface conditions between adjacent subdomains are introduced to ensure the continuity and consistency of the solution. Let Γ_{ij} be the interface between subdomains Ω_i and Ω_j, the interface condition is defined as,

$$u_i = u_j \quad \text{on} \quad \Gamma_{ij}. \tag{19}$$

Then the composite loss function is as follows,

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}^i_{\text{physics}} + \sum_{i \neq j} \mathcal{L}^{ij}_{\text{interface}}, \tag{20}$$

where the physical loss $\mathcal{L}^i_{\text{physics}}$ is computed within each subdomain, and the interface loss $\mathcal{L}^{ij}_{\text{interface}}$ is computed on the interfaces between adjacent subdomains.

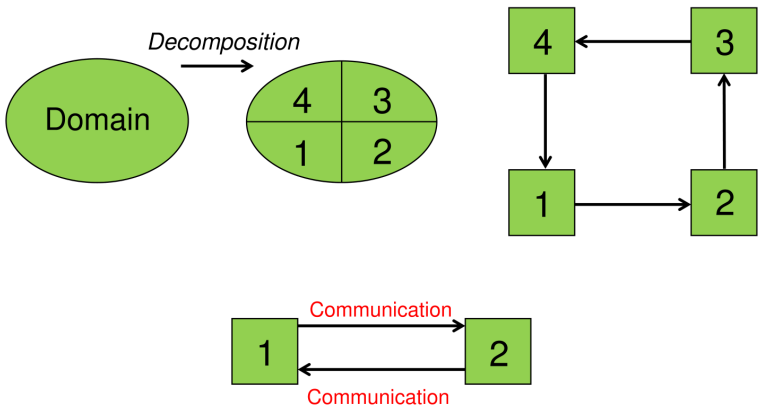

Fig. 9 Domain decomposition methods (Wu et al. 2022) in PINNs

Figure 8: PINN 中的区域分解方法 (Wu 等 2022)

已经为 PINN 开发了各种新颖的激活函数。例如，Jagtap 等 (2020b) 介绍了一种用于 PINN 的带有可缩放超参数的自适应激活函数，该函数在训练期间动态调整，从而优化网络的拓扑结构并显着提高收敛速度和求解精度。实证结果表明，自适应激活函数优于传统的固定激活函数，尤其是在早期训练阶段。该论文还提供了理论证明，即利用所提出方法的梯度下降算法不会被吸引到次优临界点或局部最小值。Fu 等 (2024) 介绍了物理信息核函数神经网络 (PIKFNNs)，旨在求解各种线性及特定非线性 PDE。PIKFNNs 采用单隐层浅层神经网络结构，使用物理信息核函数 (PIKFs) 作为自定义激活函数。这些 PIKF 包含 PDE 信息，如基本解、格林函数、调和函数和简化线性 PDE 的解，增强了网络准确近似解的能力。它表明，PIKFNNs 在准确性和效率方面比传统 PINN 有显著提高，特别是在处理不规则域和复杂几何形状方面。Abbasi 和 Andersen (2024) 提出了物理激活函数 (PAFs)，它是从所研究的物理系统中固有的数学表达式派生出来的，而不是仅依赖于 tanh 或 sigmoid 等标准激活函数。PAF 可以以显式或自适应方

式集成到神经网络中，后者允许自动确定 PAF 对每个神经元的相对影响。研究表明，PAF 显着提高了 PINN 预测的准确性，特别是对于训练外数据，并在保持准确性的同时将网络规模减少了多达 75

此外，Zeng 等 (2022) 介绍了创新的自适应技术，以提高深度神经网络 (DNN) 在求解高维 PDE 时的计算性能。作者提出了三个关键策略：自适应损失函数、自适应激活函数和自适应采样。这些方法旨在提高 DNN 的准确性和收敛速度，而不增加网络复杂性。自适应损失函数修改了传统的 L^2 范数，引入了一个参数 α，根据训练阶段转换损失函数。自适应激活函数改变跨层的激活，自适应采样策略将更多点分配给局部残差误差较高的区域。数值实验表明，这些技术显着优于传统方法，对于某些 50 维问题，相对误差达到 $O(10^{-4})$ 量级。该研究提供了一种有前途的自适应方法，以增强 DNN 在计算物理和工程中的性能。

2.2 PINN 的自适应数据重采样

PINN 的性能在很大程度上依赖于采样点的策略性选择，以有效地训练网络。这些点的位置或分布对于准确近似解至关重要。

在大多数 PINN 中，用于训练的收集点（通常称为残差点）是在训练开始时指定的，并且在训练过程中不会更改。PINN 中可以使用各种采样方法，例如均匀采样、随机采样、拉丁超立方采样 (LHS) (Stein 1987; Mckay 等 2000)、Halton 序列 (Halton 1960; Berblinger 和 Schlier 1991)、Hammersley 序列 (Wong 等 1997)、Sobol 序列 (Sobol'1967)。为了提高 PINN 的准确性，另一种方法是在每隔一定的迭代次数后选择一组新的残差点进行训练 (Lu 等 2021b)。甚至可以在不同时间使用不同的采样方法。

虽然固定采样策略因其简单性而被广泛使用，但它们并不总是最有效或最准确的方法，特别是对于具有复杂几何形状或解表现出急剧梯度的问题。最近，已经开发了自适应采样策略，根据网络训练的当前状态动态调整采样点的分布。这些自适应方法已显示出在较少残差点的情况下显着提高 PINN 的准确性。Lu 等 (2021b) 提出了 PINN 的第一个自适应采样，即基于残差的自适应细化 (RAR) 方法，该方法在 PDE 残差较大的位置添加新的残差点。在 Nabian 等 (2021) 中，介绍了一种提高 PINN 训练效率的重要性采样方法。通过根据与损失函数成比例的分布对配点进行采样，该方法加速了收敛并提高了训练效果。该技术易于实现，不引入新的超参数，并通过数值例子证明可以显着提高计算效率。Wu 等 (2023) 提供了 PINN 采样策略的综合研究，检查了非自适应和自适应重采样方法。它提出了两种新的自适应非均匀采样技术：基于残差的自适应分布 (RAD) 和具有分布的基于残差的自适应细化 (RAR-D)。这些方法根据设计的概率密度函数 (PDF) 动态调整残差点的分布，以提高 PINN 的训练效率和准确性。在 Daw 等 (2023) 中，提出了一种新颖的保留-重采样-释放 (R3) 采样算法，以解决 PINN 中的传播失败问题。它假设 PINN 的成功训练取决于解从初始/边界条件到内部点的有效传播。以不平衡 PDE 残差场为特征的传播失败会导致 PINN 收敛到平凡解。R3 算法在高 PDE 残差区域逐渐累积配点，以最小的计算开销减轻这些失败。

此外，Tang 等 (2023) 提出了 DAS-PINNs，它使用深度生成模型通过生成新的配点来动态细化训练集。具体来说，该方法将残差视为概率密度函数，并使用称为 KRnet 的生成模型来采样与此分布一致的点——更多样本放置在残差较高的区域。Yang 等 (2023) 提出了基于动态网格的重要性采样 (DMIS)，以解决 PINN 训练中固有的计算效率低和收敛不稳定问题。DMIS 通过构建有效估计样本权重的动态三角网格，将重要性采样集成到训练过程中。对三个基准——薛定谔方程、粘性 Burgers 方程和 Korteweg-de Vries 方程的评估——证明了 DMIS 在提高收敛速度和准确性方面的优越性能。在 Jiao 等 (2024) 中，提出了一种用于 PINN 的基于高斯混合分布的自适应采样 (GAS) 方法。受增量学习的启发，作者提出了风险最小-最大框架，动态细化配点的采样。GAS 利用当前的残差信息生成高斯混合分布以采样额外的点，从而加速收敛并提高准确性。Lau 等 (2024) 介绍了 PINNACLE（PINN 自适应配点和实验点选择），这是一种旨在优化 PINN 所有类型训练点选择的算法。与以前仅关注配点或实验点的方法不同，PINNACLE 联合优化两者的选择，随着训练的进行调整配点类型的比例。Tang 等 (2024) 提出了对抗性自适应采样 (AAS)，这是一种将 PINN 与最优传输理论相结合的新颖方法。AAS 采用深度生成模型来调整训练集中随机样本的分布，确保神经网络引起的残差保持平滑的轮廓。通过将残差引起的分布与均匀分布之间的 Wasserstein 距离嵌入到损失函数中，AAS 最小化了随机样本引入的统计误差。所有这些方法都提高了网络训练的效率，从而显着增强了 PINN 在各种 PDE 问

题中的性能。

受上述研究的启发，我们提出了一种增强的混合自适应 PINN (Luo 等 2025)。它采用混合自适应 (HA) 采样方法，该方法确保采样点的随机性，同时也充分考虑训练过程中具有大 PDE 残差的点。HA 方法在训练期间结合了两种不同的重采样策略。表 3 和图 10 中展示了一些示例，证明了我们提出的方法优于最先进的重采样方法。

Table 3: 一维泊松方程的示例

方法	Epochs = 20000		Epochs = 40000	
	$N_r = 14$	$N_r = 26$	$N_r = 14$	$N_r = 26$
PINN	18.07% ± 3.70%	6.82% ± 4.11%	48.28% ± 36.22%	15.30% ± 13.95%
PINN Random-R	3.76% ± 2.02%	0.99% ± 1.37%	2.52% ± 1.89%	0.14% ± 0.06%
RAD (Wu et al. 2023)				
k = 1, c = 1	5.49% ± 4.53%	2.23% ± 2.54%	1.62% ± 1.28%	1.02% ± 1.53%
k = 1, c = 2	3.31% ± 1.57%	2.51% ± 2.81%	2.21% ± 1.40%	0.82% ± 0.82%
k = 2, c = 1	5.44% ± 4.95%	3.78% ± 2.30%	2.04% ± 2.06%	1.11% ± 0.65%
k = 2, c = 2	6.00% ± 5.49%	2.50% ± 2.44%	1.87% ± 1.59%	0.89% ± 1.01%
PINN HA (ours)	**2.73% ± 2.38%**	**0.24% ± 0.21%**	**1.28% ± 0.76%**	**0.11% ± 0.08%**

Table 4: *

比较方法的 L2 相对误差。实验重复十次，计算 L2 相对误差的平均值和标准差。N_r 表示残差点的数量。

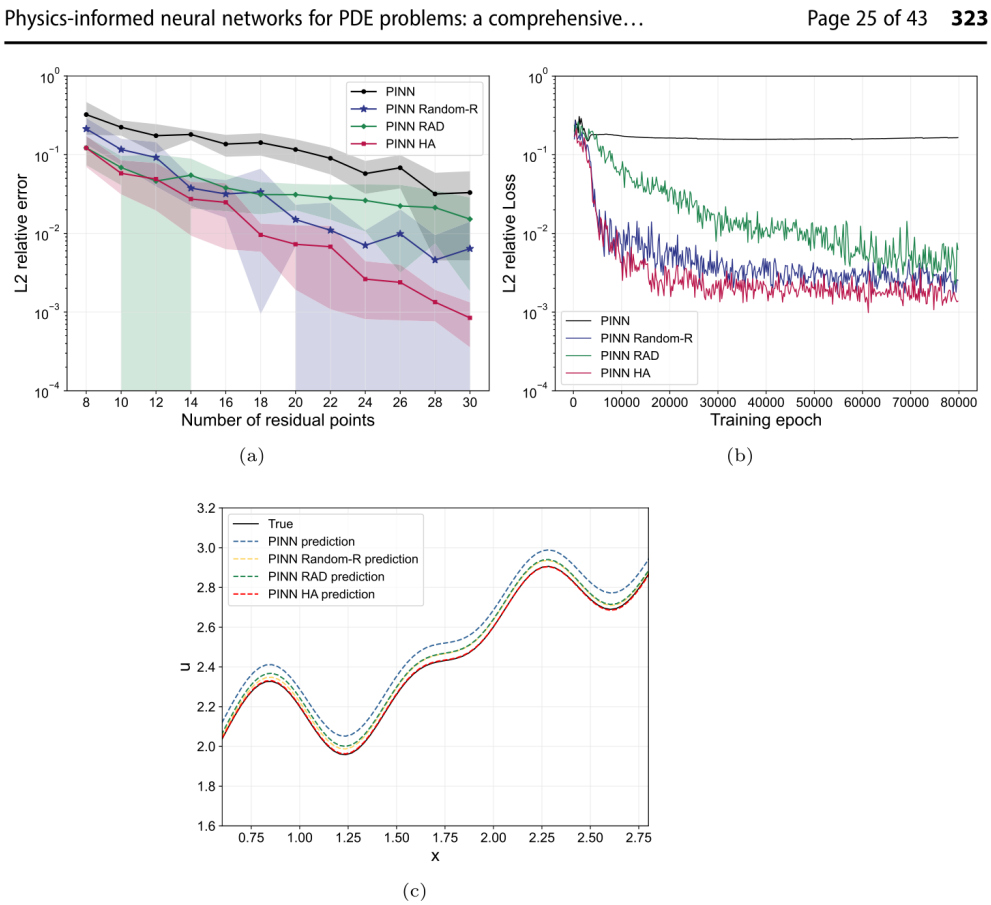

Fig. 10 The performance of our proposed HA sampling method for 1D Poisson equation

Figure 9: 我们要提出的 HA 采样方法在一维泊松方程上的性能

2.3 PINN 的损失函数和优化

在 PINN 中，损失函数在模型训练和性能中起着关键作用。损失函数影响收敛性和网络有效学习的能力。

2.3.1 损失重加权

在 PINN 中，PDE 损失、初始和边界条件损失通常加在一起并直接优化。然而，不同损失项的规模和收敛速度可能会有所不同。这些损失之间的不平衡会导致性能下降。为了缓解这些问题，已经提出了各种损失重加权方法，自适应地调整损失函数中不同分量的相对重要性。

Wang 等 (2021c) 检查了 PINN 中的梯度流病态，并提出了提高其预测精度的解决方案。它引入了一种学习率退火算法，利用梯度统计来平衡复合损失函数，并提出了一种对梯度问题更具弹性的新型神经网络架构。这些发展将 PINN 在一系列计算物理问题上的准确性提高 50-100 倍。它为训练受约束的神经网络提供了新的见解，并提供了缓解梯度流病态的实用方法。Wang 等 (2022a) 使用神经切线核 (NTK) 框架研究了 PINN 的训练动态。它推导了 PINN 的 NTK，并证明在某些条件下，在无限宽度限制下，它会在训练期间收敛到一个保持恒定的确定性核。这允许通过 NTK 的视角分析 PINN 训练动态，揭示不同损失分量收敛速度的差异。为了解决这些问题，它提出了一种新颖的梯度下降算法，利用 NTK 的特征值来平衡训练过程，从而提高 PINN 的稳定性和有效性。在 Xiang 等 (2022) 中，提出了一种自适应损失平衡 PINN (lbPINNs)，它使用自适应机制在训练期间动态调整不同损失项的权重。这是通过概率建模和最大似然估计实现的。它建立了一个输出为 u 的高斯概率模型。高斯似然定义为均值由 PINN 近似 $\hat{u}(x,t;\theta)$ 给出的高斯分布，以及不确定性参数 ε_d，

$$p(u|\hat{u}(x,t;\theta)) = N(\hat{u}(x,t;\theta),\varepsilon_d^2) \tag{25}$$

使用最大似然推断调整不确定性参数。为了实现这一点，最小化模型的负对数似然，

$$-\log p(\boldsymbol{u}|\hat{\boldsymbol{u}}(\boldsymbol{x},t;\theta)) \propto \frac{1}{2\varepsilon_d^2}\|\boldsymbol{u}-\hat{\boldsymbol{u}}(\boldsymbol{x},t;\theta)\|^2 + \log\varepsilon_d = \frac{1}{2\varepsilon_d^2}\mathcal{L}_{\text{data}}(\theta) + \log\varepsilon_d. \tag{26}$$

此外，假设我们的高斯概率模型的输出包括两个向量 u, g，每个都服从高斯分布，

$$p(u,g|\hat{u}(x,t;\theta)) = p(u|\hat{u}(x,t;\theta))\cdot p(g|\hat{u}(x,t;\theta)) = N(\hat{u}(x,t;\theta),\varepsilon_d^2)\cdot N(\hat{u}(x,t;\theta),\varepsilon_b^2). \tag{27}$$

因此，它建立了具有四个向量的多输出模型来定义复杂的损失函数。lbPINN 的损失函数可以公式化为，

$$\mathcal{L}(\epsilon;\theta;N) = \frac{1}{2\epsilon_f^2}\mathcal{L}_{\text{PDE}}(\theta;N_f) + \frac{1}{2\epsilon_b^2}\mathcal{L}_{\text{BC}}(\theta;N_b) + \frac{1}{2\epsilon_i^2}\mathcal{L}_{\text{IC}}(\theta;N_i) + \frac{1}{2\epsilon_d^2}\mathcal{L}_{\text{data}}(\theta;N_{\text{data}}) + \log\epsilon_f\epsilon_b\epsilon_i\epsilon_d, \tag{28}$$

其中 $\epsilon = \{\epsilon_f,\epsilon_b,\epsilon_i,\epsilon_d\}$ 描述了损失项的自适应权重。$\omega = \{\omega_f,\omega_b,\omega_i,\omega_d\}, \omega := \frac{1}{2\epsilon^2}$ 描述了损失项的总权重。尽管 lbPINN 在设计快速准确的科学机器学习技术方面取得了进展，但仍有一些问题需要解决，例如 (1) 在上述实验中，PDE 损失的自适应权重下降缓慢是一个普遍现象。因此，我们将对这一现象进行理论分析，以提高该方法的鲁棒性和可扩展性。(2) 值得探讨的是，这个概率模型是否是最合适的，并且在某些假设下，是否有可能与其它几种损失平衡算法建立统一的框架。此外，甚至可以在每个训练点调整每个损失项的不同权重。例如，Gu 等 (2021) 提出了 SelectNet，这是一种新颖的自步学习框架，旨在提高 PINN 的收敛性。通过结合自适应加权训练样本的选择网络，SelectNet 在训练的早期阶段优先考虑较简单的样本，并逐渐结合更具挑战性的样本。

此外，McClenny 和 Braga-Neto (2023) 介绍了自适应 PINN (SA-PINNs)，它采用完全可训练的自适应权重，对每个训练点施加单独的关注，使网络能够自主识别并专注于解的挑战性区域。自适应权重创建了一个软注意力掩码，随着相应损失的增加而增加。它还演示了如何使用高斯过程回归构建自适应权重的连续图，从而在传统梯度下降不足的情况下促进随机梯度下降的使用。实验表明，SA-PINNs 提高了 PDE 解的准确性和鲁棒性。Song 等 (2024) 引入了一种

名为损失注意力 PINN (LA-PINN) 的新颖架构，以解决普通 PINN 在难以拟合区域遭受收敛速度慢和精度差的问题。该架构通过为每个独立的损失分量分配一个损失注意力网络 (LAN) 来关注每个训练点的损失误差控制。通过将点平方误差 (SE) 馈送到 LAN，可以构建“注意力函数”来完成向不同点误差分配不同权重的任务。LA-PINN 架构采用了一种基于点误差的加权方法，利用主网络和 LAN 之间的对抗训练。负责预测解的主网络通过梯度下降最小化损失，而 LAN 通过梯度上升调整 SE 的权重。这种机制通过增加这些点的权重和 SE 更新梯度的增长率，确保更多关注难以拟合的点。通过这种动态加权方法，与普通 PINN 和其他最先进的方法相比，LA-PINN 表现出卓越的预测性能和更快的收敛速度。数值实验证实了 LA-PINN 在推进难以拟合点的收敛和实现求解 PDE 的高精度方面的有效性。

2.3.2 新型损失函数

除了上面讨论的损失重加权方法外，还开发了许多变体损失函数。

正则化是增强机器学习模型训练和泛化能力的重要技术。一些研究提出了 PINN 损失函数的各种正则化方法。Yu 等 (2022) 提出了梯度增强 PINN (gPINNs)，它将梯度信息集成到神经网络的损失函数中，从而增强了模型的性能。在传统的 PINN 中，主要目标是将 PDE 残差 f 约束为零，确保在每个点 x，$f(x)=0$。然而，由于残差 $f(x)$ 在任何点 x 均为零，因此可以推断其导数也应为零。因此，gPINNs 假设 PDE 的精确解足够平滑，使得 PDE 残差 $\nabla f(x)$ 的梯度存在，并建议通过强制 PDE 残差的导数也为零来增强 PINN。那么 gPINNs 的损失函数由下式给出，

$$\mathcal{L} = w_f\mathcal{L}_f + w_b\mathcal{L}_b + w_i\mathcal{L}_i + \sum_{i=1}^{d} w_{g_i}\mathcal{L}_{g_i}(\theta;\mathcal{T}_{g_i}), \tag{29}$$

其中

$$\begin{aligned}
\mathcal{L}_f(\theta;\mathcal{T}_f) &= \frac{1}{|\mathcal{T}_f|}\sum_{x\in\mathcal{T}_f}\left|f\left(x;\frac{\partial\hat{u}}{\partial x_1},\ldots,\frac{\partial\hat{u}}{\partial x_d};\frac{\partial^2\hat{u}}{\partial x_1\partial x_1},\ldots,\frac{\partial^2\hat{u}}{\partial x_1\partial x_d};\ldots;\lambda\right)\right|^2,\\
\mathcal{L}_b(\theta;\mathcal{T}_b) &= \frac{1}{|\mathcal{T}_b|}\sum_{x\in\mathcal{T}_b}|\mathcal{B}(\hat{u},x)|^2,\\
\mathcal{L}_{g_i}(\theta;\mathcal{T}_{g_i}) &= \frac{1}{|\mathcal{T}_{g_i}|}\sum_{x\in\mathcal{T}_{g_i}}\left|\frac{\partial f}{\partial x_i}\right|^2,\\
\mathcal{L}_i(\theta,\lambda;\mathcal{T}_i) &= \frac{1}{|\mathcal{T}_i|}\sum_{x\in\mathcal{T}_i}|\hat{u}(x)-u(x)|^2,
\end{aligned} \tag{30}$$

其中 w_f, w_b, w_i 和 w_g 是权重。gPINNs 的优点包括提高了准确性、效率，并且更好地泛化到没有训练点的区域。它们通常需要较少的训练点即可达到与传统 PINN 相似或更好的精度，从而使其计算效率更高。通过利用梯度信息，gPINNs 具有更好的泛化能力，从而产生更鲁棒的解。当与 RAR 结合使用时，gPINNs 表现非常出色，特别是对于具有陡峭梯度解的 PDE。这使得 gPINNs 成为解决光学、流体力学、系统生物学和生物医学等各个领域中正向和反向 PDE 问题的有力工具。总体而言，gPINNs 代表了深度学习在 PDE 应用中的重大进步，提供了增强的性能和效率。在 Wang 等 (2024a) 中，专门设计了一个实用的 PINN 框架来解决具有多量级损失项的多尺度问题。所提出的方法通过实施分组正则化策略增强了传统的 PINN 方法，该策略归一化了不同损失分量的尺度。这确保了每一项对整体损失函数的贡献相等，从而提高了网络捕捉解的宏观和微观特征的能力。此外，该框架还结合了专门的神经网络架构，如傅里叶特征网络，以进一步提高准确性和效率。通过数值实验，证明了这种改进的 PINN 框架在求解多尺度 PDE 方面显着优于标准 PINN，使其成为应用在复杂多尺度现象中的宝贵工具。

此外，Wang 等 (2022b) 挑战了在训练 PINN 求解 PDE 时使用 L^2 损失的传统做法。它仔细审查了损失函数与学习解的准确性之间的关系，引入了 PDE 理论中的稳定性概念来分析当损失趋近于零时解的渐近行为。该研究聚焦于最优控制中关键的 Hamilton–Jacobi–Bellman (HJB) 方程，揭示了对于一般的 L^p 损失函数，只有当 p 足够大时才能实现稳定性。这一发现表明，无处不在的 L^2 损失对于 HJB 方程可能是次优的，主张使用 L^∞ 损失。该论文提出了一种新颖的

PINN 训练算法，该算法采用类似于对抗训练的最小-最大优化策略来最小化 L^∞ 损失，通过实证实验证明了改进的解精度。该研究为 PINN 的损失函数设计提供了重要的见解，特别是对于高维非线性 PDE。Wang 等 (2024d) 调查了不同损失函数对 PINN 训练的影响。它将损失函数分为两类：观测数据损失，直接约束和衡量模型输出；以及模型损失，包括来自控制方程和变分形式的信息，间接影响网络性能。它揭示了一个稳定的“损失跳跃”现象：当从数据损失过渡到包含不同阶数导数信息的模型损失时，神经网络解会发生与精确解的显着偏差。进一步的实验表明，这种现象源于神经网络在各种损失函数下的不同频率偏好。理论分析阐明了模型损失下的频率偏好，为 PINN 求解 PDE 的潜在机制提供了见解。这项工作为优化训练过程和提高 PINN 的准确性提供了宝贵的视角。

2.4 特征嵌入和增强

特征嵌入广泛应用于机器学习的各个领域，如自然语言处理 (NLP) (Patil 等 2023)、计算机视觉 (CV) (Kan 等 2019) 和推荐系统 (Zhang 等 2016)。感兴趣的特征（或变量）被映射到向量空间，算法可以在其中更容易地辨别关系和模式。NeRF (Mildenhall 等 2020) 是一个典型的例子，空间编码有效地表示和处理有关输入点位置的信息。这种编码将每个输入点的空间位置与颜色和辐射水平等其他相关细节相结合，从而提高了图像重建和渲染的质量。NeRF 采用正弦和余弦函数对三维空间中的坐标进行编码，如下所示，

$$\gamma(x,y,z) = (x,y,z,\sin(2^0x),\sin(2^0y),\sin(2^0z),\cos(2^0x),\cos(2^0y),\cos(2^0z),\ldots,\sin(2^{n-1}z),\cos(2^{n-1}z)) \tag{31}$$

同样，Tancik 等 (2020) 基于神经切线核 (NTK) 理论 (Jacot 等 2018) 证明了傅里叶特征映射可以有效地减轻基于坐标的 MLP 对低频的谱偏差。采用具有合适尺度的随机选择的傅里叶特征映射，

$$\gamma(v) = (a_1\cos(2\pi b_1^T v), a_1\sin(2\pi b_1^T v),\ldots,a_m\cos(2\pi b_m^T v), a_m\sin(2\pi b_m^T v))^T. \tag{32}$$

实验表明，这种方法提高了基于坐标的 MLP 在计算机视觉和图形学中各种低维任务上的性能（图 11）。

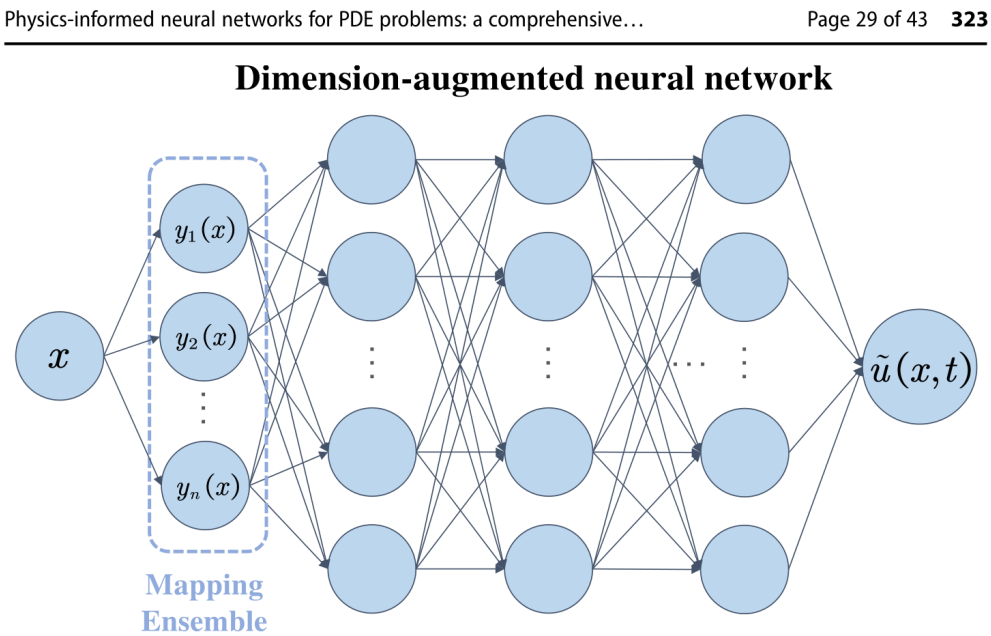

Fig. 11 The framework of dimension-augmented PINN (DaPINN) (Guan et al. 2023)

Figure 10: 维度增强 PINN (DaPINN) (Guan 等 2023) 的框架

受上述工作的启发，许多结合特征嵌入的新型 PINN 被提出来提高性能。PINN 的一个基本弱点是谱偏差 (Rahaman 等 2019)，这是全连接神经网络中公认的挑战，限制了它们有效学习高频函数的能力。在 Wang 等 (2021a) 中，介绍了一种利用这种时空和多尺度随机傅里叶特征的新型 PINN 架构。它阐明了这些坐标嵌入层如何增强 PINN 模型的鲁棒性和准确性，有效地解决高频和多尺度挑战。Li 等 (2024a, c) 也利用了类似的傅里叶特征映射方法。Peng 等 (2020) 提出了配备任务相关字典的先验字典 PINN (PD-PINN)。PD-PINN 通过有效地从字典中捕获特征，在任务中表现出增强的表示能力，从而在训练期间实现更快的收敛。此外，Wong 等 (2024) 提出了 sf-PINN，其中利用了输入的正弦映射。这种方法可以有效地增加输入梯度变异性，从而避免陷入欺骗性的局部最小值。使用 sf-PINN，模型精度可以提高几个数量级。

此外，一些计算机视觉 (CV) 任务中常用一些增强技术 (Perez 和 Wang 2017)，包括空间变换（翻转、旋转、裁剪等）、噪声注入、颜色变换等。同样，可以将更多特征插入 PINN 的输入向量中以提高模型的能力。我们提出了一种维度增强 PINN (DaPINN) (Guan 等 2023)，它系统地操纵网络的输入维度。DaPINN 的性能已在各种正向和反向 PDE 中进行了评估，包括泊松方程、热方程、扩散问题、Burgers 方程等。DaPINN 模型通过副本增强、幂级数增强和傅里叶级数增强提高了求解精度。例如，考虑以下具有 Dirichlet 边界条件 $u(x=0)=0$ 和 $u(x=\pi)=\pi$ 的一维泊松方程，

$$-\Delta u = \sum_{k=1}^{3} k\sin(kx) + 7\sin(7x) + 8\sin(8x), \quad x \in [0,\pi], \tag{33}$$

图 12 展示了我们提出的 DaPINN 在一维泊松方程上的性能，并与 PINN 进行了比较。Burgers 方程结果的另一个示例如图 13 所示。

DaPINN 模型通过扩展网络输入维度，优于 PINN，这使得神经网络能够提取更多信息特征。

3 应用

该领域的最新进展证明了 PINN 在广泛应用中的有效性。它标志着机器学习与计算物理学交叉领域的重大飞跃。下面，我们将讨论 PINN 正在产生影响的几个关键领域。

3.1 流体动力学

PINN 已成为流体力学中的一项关键技术，为求解控制流体运动的 Navier–Stokes 方程 (NSE) 提供了一种新颖的方法 (Cai 等 2021; Mao 等 2020)。物理学与数据的结合对于解决复杂的高维流体流动问题特别强大，在这些问题中，传统的计算流体力学 (CFD) 方法可能难以应对包含噪声数据或有效处理反问题的要求。

PINN 已成功应用于模拟钝体（如圆柱体）后面的三维尾流，它们可以从有限的观测数据重建完整的速度场和压力场。这种能力对于只有部分测量值可用的复杂流动的诊断目的特别有价值，例如在平面粒子图像测速 (PIV) 实验中。此外，PINN 在模拟高速可压缩流动（例如涉及激波的流动）方面发挥了重要作用。在这些场景中，传统的 CFD 方法通常依赖于精确的边界条件，而在实验设置中可能无法轻易获得这些条件。然而，PINN 可以利用密度梯度和表面压力测量值等可用数据来推断完整的流场。即使在缺乏完整的边界条件信息的情况下，这种方法也被证明是有效的，突出了 PINN 在解决不适定问题方面的灵活性。PINN 的应用也扩展到了生物医学流体力学。例如，在血流动力学 (Kissas 等 2020) 中，PINN 已被用于模拟血流动力学、估计壁面剪切应力并预测特定患者动脉网络中的脉搏波传播。

3.2 固体力学

在固体力学中，PINN 已被有效地应用于模拟线性弹性、弹塑性、超弹性和断裂力学。PINN 可以处理正向问题，预测给定输入的材料响应。例如，Rao 等 (2021) 使用 PINN 强形式算法求解了线性弹性动力学问题。在解决高度非均匀的正向问题时，与有限差分法 (FDM) 和有限元法

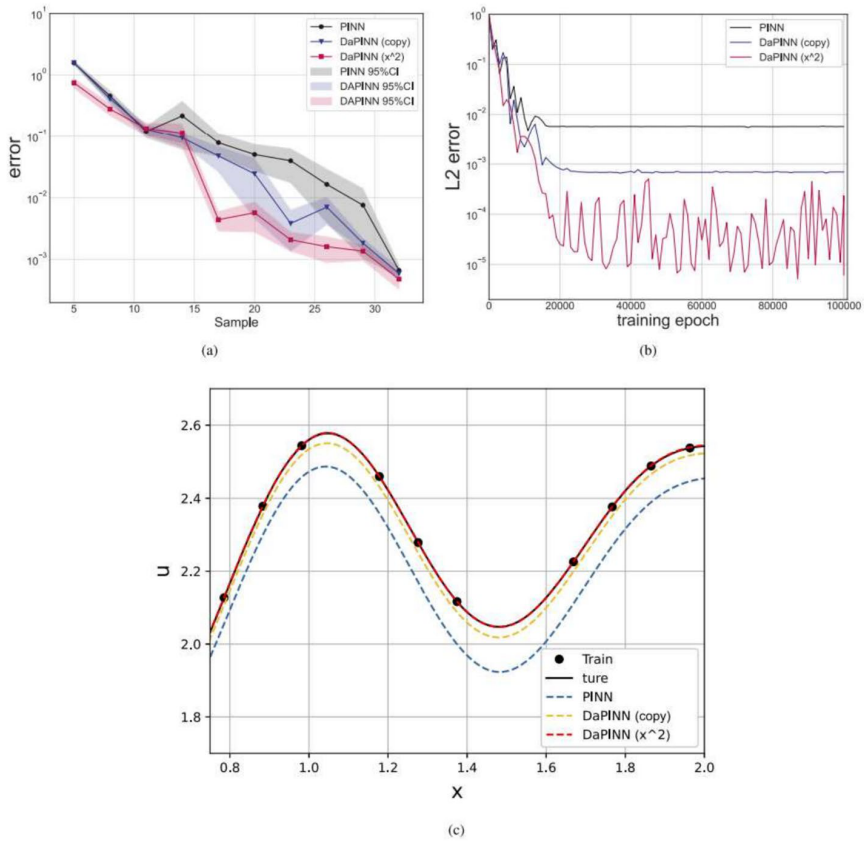

Fig. 12 The performance of DaPINN (Guan et al. 2023) for 1D Poisson equation

Figure 11: DaPINN (Guan 等 2023) 在一维泊松方程上的性能

(FEM) 相比，PINN 表现出卓越的精度和效率 (Guo 等 2022)。固体力学的反问题也可以得到解决，主要包括本构方程 (Haghighat 等 2021b) 和几何拓扑 (Zhang 等 2022a)。

PINN 在固体力学中的潜力是巨大的，有望增强我们要对材料行为的理解和预测，从而在结构设计和材料科学中带来创新的解决方案。

3.3 电磁学和光学

PINN 已扩展到求解光学和电磁学中的问题，例如 3-D 亥姆霍兹方程和拟线性 PDE 算子。它们已被证明对这些领域的反问题有效，由于电磁场的复杂性质，这些问题可能特别具有挑战性 (Fang 和 Zhan 2020)。Chen 等 (2020a) 利用 PINN 解决光子超材料和纳米光学技术中的逆散射问题。无网格 PINN 已被应用于检索涉及许多相互作用的纳米结构以及多组分纳米粒子的许多有限尺寸散射系统的有效介电常数参数这一艰巨任务。

PINN 代表了解决流体动力学、固体力学、电磁学和光学等各个领域复杂问题的变革性方法。它们的多功能性和效率继续推动研究和应用，使其成为未来计算建模工作的关键组成部分。

4 软件

已经开发了许多开源工具和库来促进 PINN 的实施。这些工具抽象了构建 PINN 所涉及的大部分复杂性，使研究人员和从业者更容易将此方法应用于科学和工程中的各种问题。本节概述了一些广泛使用的、可用于实施 PINN 的公共框架和库（表 4）。

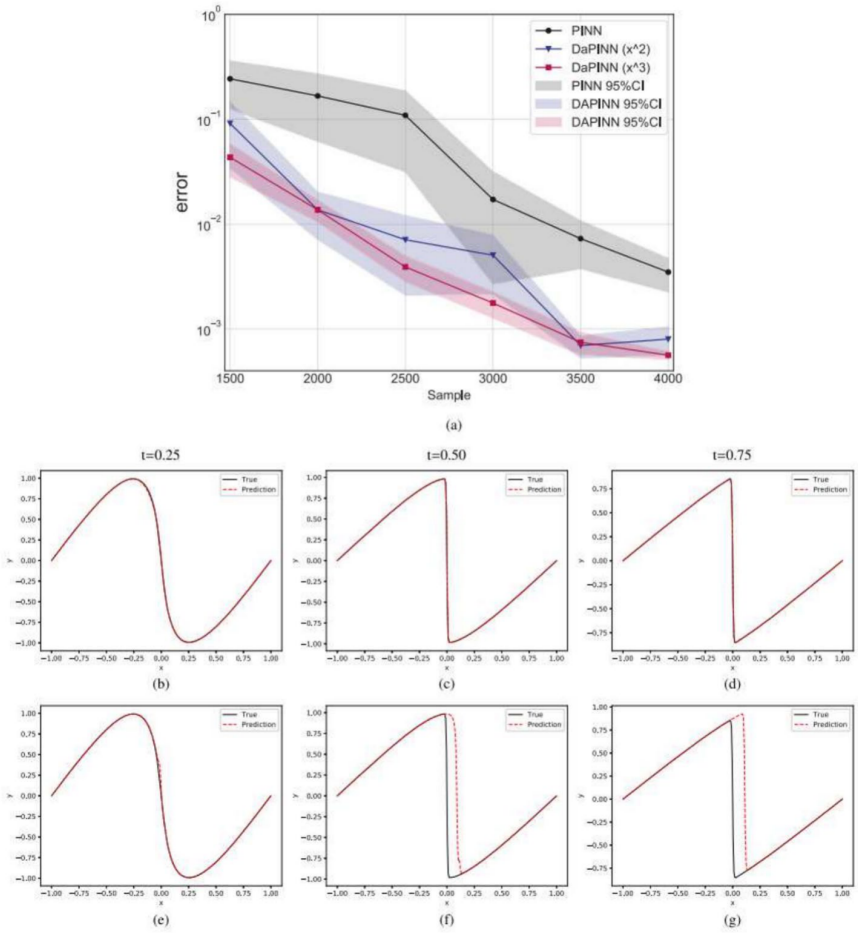

Fig. 13 The performance of DaPINN (Guan et al. 2023) for Burgers equation

Figure 12: DaPINN (Guan 等 2023) 在 Burgers 方程上的性能

4.1 DeepXDE

DeepXDE (Lu 等 2021b) 是一个专门设计用于使用 PINN 求解微分方程的开源 Python 库。它通过利用深度学习技术，为定义、训练和求解 ODE 和 PDE 提供了一个高级、易于使用的接口。该库与 TensorFlow 或 PyTorch 无缝集成，允许用户轻松地将神经网络应用于复杂的、现实世界的基于物理的问题。

4.2 IDRLnet

IDRLnet (Peng 等 2021) 是一个用于通过 PINN 系统地建模和求解问题的 Python 工具箱。它提供了一种结构化的方式来在 Python 中集成几何对象、数据源、人工神经网络、损失指标和优化器。

4.3 NeuroDiffEq

NeuroDiffEq (Chen 等 2020b; Liu 等 2025) 是一个开发的用于使用神经网络求解微分方程的开源 Python 库，特别是在 PyTorch 框架内。它利用人工神经网络的能力来近似常微分方程 (ODE) 和偏微分方程 (PDE) 的解，并服从指定的初始或边界条件。该库专为灵活性而设计，使研究人员和从业者能够解决各种用户定义的问题。此外，它促进了不仅连续而且可微的解的计

算，与有限差分和有限元方法等传统数值方法相比具有显著优势。

4.4 SciANN

SciANN (Haghighat 和 Juanes 2021a) 是专门为科学计算设计的高级神经网络 API。它建立在 Keras 和 TensorFlow 之上，使科学家能够轻松利用深度学习进行物理信息建模。SciANN 允许研究人员定义输入、输出和约束，从而能够将科学问题制定为深度学习任务。该框架对于求解 PDE 和模拟系统动力学特别有用，使其成为希望将深度学习技术应用于其研究的科学家和工程师的宝贵工具。

4.5 TensorDiffEq

TensorDiffEq (McClenny 等 2021) 是构建在 TensorFlow 之上的 Python 包，旨在为 PINN 提供可扩展且高效的求解器。它特别专注于 PINN 的可扩展求解，用于推理和反问题发现。TensorDiffEq 的独特之处在于它完全支持自适应 PINN 求解器，并且是唯一完全开源的多 GPU PINN 解决方案套件。

Table 5: 一些用于 PINN 的开源框架

框架名称	后端
DeepXDE	TensorFlow, PyTorch, JAX
IDRLnet	PyTorch
NeuroDiffEq	PyTorch
SciANN	TensorFlow
TensorDiffEq	Tensorflow

5 挑战与展望

PINN 已被证明是跨各种科学和工程学科的多功能工具，为求解传统方法可能难以应对的复杂 PDE 提供了一种鲁棒且灵活的方法。尽管取得了成功，但这方面的研究仍在继续解决理论问题和实际局限性，旨在提高 PINN 的可扩展性、可解释性和可靠性。在下一节中，我们将重点介绍与 PINN 相关的关键挑战，并提出未来研究的潜在方向。

5.1 PINN 面临的挑战

PINN 解决流体动力学、材料科学和生物力学等各个领域问题的能力证明了该方法弥合数据驱动模型与传统数值模拟之间差距的潜力。然而，尽管具有令人印象深刻的能力，PINN 仍然面临一些限制和挑战，阻碍了它们的广泛采用。PINN 由三个关键要素组成：(1) 近似底层函数的神经网络，(2) 整合物理定律的物理信息损失函数，以及 (3) 用于训练模型的优化方法。这些要素使 PINN 能够有效地将物理约束嵌入到学习过程中。然而，在实践中管理这些组件的复杂性通常会导致诸如收敛缓慢、对超参数敏感以及难以扩展到高维或多物理场问题等问题。

为了提高 PINN 目前约为 10^{-3} 的收敛精度，需要进行严格的分析来评估 PINN 的学习能力，重点关注泛化、鲁棒性以及物理系统复杂性所施加的限制等方面。一些研究提供了对 PINN 的理论见解 (Krishnapriyan 等 2021; Wang 等 2022a; Hu 等 2022)，但仍需对该主题进行更深入的研究。例如，Krishnapriyan 等 (2021) 确定了一个关键的“梯度病态”问题，该问题由于数值刚性而出现，导致反向传播期间的梯度不平衡。这种不平衡阻碍了网络准确满足 PDE 约束和边界条件的能力，通常导致次优解。

PINN 的关键问题之一是难以捕捉快速变化或高频解。传统的优化算法，如 Adam 或 L-BFGS，可能难以在这些场景中有效收敛，导致性能不佳或训练缓慢。此外，不同物理过程相互作用的多物理场问题引入了额外的复杂性，当前的优化技术可能无法很好地处理。为了解决这些挑战，开发为 PINN 量身定制的新优化算法至关重要。这些算法应通过适应问题的特定性质来增强训练过程的鲁棒性和效率。潜在的方法包括设计更复杂的损失函数、结合分层模型或利

用高级优化策略。这些进步对于提高 PINN 的可扩展性、准确性和可靠性至关重要，使其更适用于涉及高频动力学和多物理场相互作用的现实世界问题。

另一个挑战是当前的 PINN 无法有效处理嘈杂或不完整的数据。虽然 PINN 将物理定律整合到学习过程中，但它们仍然容易受到训练数据中错误和不一致的影响，这可能会影响预测的准确性。此外，对于多尺度和多物理场问题，PINN 的可扩展性仍然是一个挑战，因为不同物理模型的相互作用通常会导致模型复杂性增加。

此外，仍然需要开发有价值的实验数据集以及相应的物理模型和参数。高质量的数据集对于训练鲁棒且准确的 PINN 模型至关重要，但许多科学和工程领域缺乏可访问的、详细的实验数据，这限制了现实世界应用的潜力。此外，对可集成到 PINN 中的准确且计算高效的物理模型和参数的需求构成了另一个重大障碍。

5.2 算子学习

虽然 PINN 通过将物理定律纳入学习过程在解决复杂物理问题方面取得了显著成功，但仍然存在一些挑战阻碍了它们的全部潜力。PINN 的一个主要局限性是它们依赖于物理约束的手动公式化，这对于具有复杂或未知物理机制的问题来说可能变得繁琐且具有挑战性。此外，PINN 的性能对神经网络架构的选择、训练策略以及边界条件的处理高度敏感，这通常需要大量的调整。

近年来，已经向深度算子学习转变，这是一个有前途的范式，通过专注于直接学习控制物理系统的底层算子来扩展 PINN。与依赖于使用神经网络求解特定 PDE 或系统动力学的传统 PINN 不同，算子学习 (Lu 等 2021a; Wang 等 2021b) 旨在学习输入和输出函数（例如 PDE 的解）之间的映射，从而实现更具泛化能力的模型，无需显式的基于物理的约束即可应用于广泛的物理问题。从 PINN 到深度算子学习的转变为未来的研究提供了几个令人兴奋的途径。首先，开发能够有效学习高维算子映射的更鲁棒的训练算法至关重要。这些模型应设计用于处理现实世界应用中经常遇到的多尺度、多物理场问题的复杂性。此外，需要在算子学习中进一步研究数据驱动模型和基于物理模型的集成，特别是在可用数据稀疏或嘈杂的情况下。未来的研究还应探索算子学习模型的可解释性和可解释性，以确保它们在安全关键型工程领域中的部署。

总之，虽然 PINN 为物理信息学习奠定了基础，但向深度算子学习的转变为将该框架推广到更广泛的问题类别提供了令人兴奋的新可能性。在这个方向上的持续研究对于释放神经网络在科学计算和工程应用中的全部潜力至关重要。

6 结论

总之，PINN 已成为求解 PDE 的有力工具，通过将物理定律直接嵌入神经网络的损失函数中，从而弥合了传统数值方法与现代机器学习技术之间的差距。

这篇综述论文全面考察了 PINN 在广泛的科学和工程领域的进步和应用。强调了来自不同视角的各种创新 PINN，包括其架构、自适应重采样、损失和激活函数、特征嵌入和大型模型。尽管 PINN 取得了成功，但这篇综述也承认了正在面临的挑战，例如需要更大的批量大小、复杂的训练程序以及处理高维和非线性问题。未来的研究旨在解决这些问题，重点是增强 PINN 的可扩展性、可解释性和可靠性。

PINN 代表了机器学习和计算科学交叉领域的变革性技术。通过将物理知识直接嵌入机器学习模型，它们为求解跨多个学科的复杂 PDE 提供了强大的工具集。随着研究的进展，预计 PINN 将继续发展，解决现有的局限性并将其适用性扩展到新领域，从而为科学发现和工程实践的进步做出贡献。