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

汇报人：Gemini Agent

基于 Luo 等 (2025) 的论文

November 27, 2025

目录

1 背景与数理基础

2 网络架构演进

3 关键优化方法

4 应用与未来

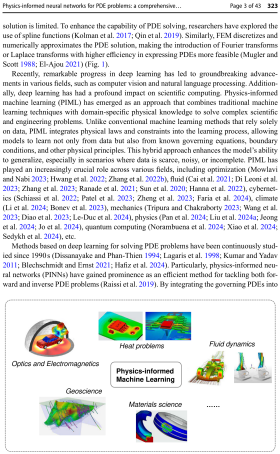
科学计算的新范式：AI for Science

传统数值方法 (FEM/FVM)

- **优点**：理论成熟，精度高。
- **痛点**：网格生成复杂，高维问题“维数灾难”，反问题求解困难。

物理信息机器学习 (PIML)

- **核心**：数据 + 物理定律 (PDEs)。
- **优势**：无网格 (Mesh–free)，易于处理反问题，融合观测数据。



Physics-informed neural networks for PDE problems: a comprehensive… Page 3 of 43 323

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 (Schinosi 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 1990s (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

图 1：物理信息机器学习在各领域的应用

PINN 的核心数理逻辑

问题：求解 u(x, t) 使得

$$f(x, t, \partial_x u, \partial_t u, \lambda) = 0$$

损失函数 $\mathcal{L} = \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{data}}$

- **物理损失** (残差): $\mathcal{L}_f = \frac{1}{N_f} \sum |f(x_f, t, \lambda)|^2$
- **边界/初始条件**: $\mathcal{L}_{b/i} = \frac{1}{N_{b/i}} \sum |\tilde{u} - g|^2$
- **数据损失** (可选): $\mathcal{L}_{\text{data}} = \frac{1}{N_d} \sum |\tilde{u} - u_{\text{obs}}|^2$

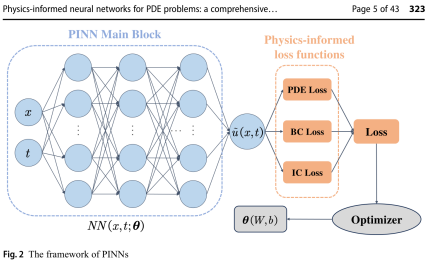


图 2：PINN 框架

PINN 架构概览

integrated the knowledge of PDEs along with their boundary and initial conditions, while simultaneously training multiple neural networks.

However, these early methods were limited by the constraints of neural network structures and algorithms, which significantly hindered their ability to handle more complex problems.

In recent years, with the rapid development of artificial intelligence technology, high-performance computing hardware such as GPUs and TPUs have emerged, and deep learning frameworks like TensorFlow (Abadi et al. 2016) and PyTorch (Paszke et al. 2019) have been launched and continuously improved. The application of deep learning in solving PDEs has made further progress. In 2019, Raissi et al. (2019) proposed the PINN model, which pioneered a new paradigm for solving PDEs using neural networks. By integrating the PDE and its initial and boundary conditions into the loss function of the neural network, and using automatic differentiation and backpropagation to guide the training process, the model can seamlessly combine physics-based constraints with observational data within a unified framework.

Figure 3 illustrates the timeline of neural network-based methods for solving PDEs. It provides a clear and concise visual representation of the evolution of these methods, emphasizing the central and transformative role that neural networks now play in modern PDE solving approaches. The timeline highlights the progressive integration of neural networks, demonstrating their growing influence and effectiveness in addressing the complexities of PDEs (Table 2).

2.1 Architectures for PINN

Various architectures tailored to address specific challenges have emerged in PINNs. These different PINN architectures represent innovative adaptations of traditional neural network structures, specifically designed to enhance performance in solving PDE problems. This section explores and compares various PINN architectures, highlighting their unique design principles, computational efficiencies, and applicability across diverse domains. The main architectures covered in PINNs mainly include Multi-layer perceptron (MLP) (Hornik et al. 1989), Convolutional neural networks (CNN) (Krizhevsky et al. 2012; Gu et al. 2018), Recurrent neural networks (RNN) (Giles et al. 1994; Schuster and Paliwal 1997), and others such as Generative adversarial networks (GAN) (Goodfellow et al. 2014), Transformer (Vaswani et al. 2017).

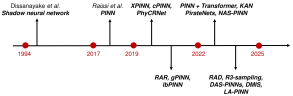


Fig. 3 The timeline of neural network based methods for solving PDEs

从简单的 MLP 到结合 KAN、Transformer 的复杂架构

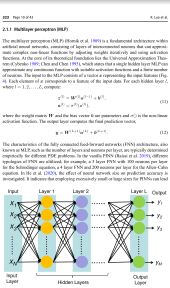
主流架构对比 (1)

MLP (多层感知机)

- 基础架构，通用近似定理。
- **局限**：频谱偏差，难捉高频。

CNN (卷积神经网络)

- **适用**：规则网格，图像类数据。
- **变体**：PhyGeoNet, f−PICNN。



223 Page 10 of 41 X. Lui et al.

2.1.1 Multilayer perceptron (MLP)

The multilayer perceptron (MLP) (Hornik et al. 1989) is a fundamental architecture within machine learning and neural networks. It is a feed-forward network that approximates a target complex non-linear function 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 $\mathbf{x}$ representing the input features (Fig. 4). Each element of $\mathbf{x}$ corresponds to a feature of the input data. For each hidden layer l, where $l = 1, 2, \ldots, L$, compute:

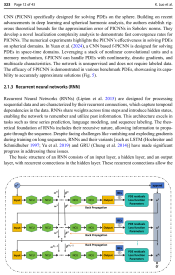
$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)},$$
$$\mathbf{a}^{(l)} = \sigma^{(l)}(\mathbf{z}^{(l)}), \tag{1.1}$$

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

$$\mathbf{y} = \mathbf{W}^{(L+1)}\mathbf{a}^{(L)} + \mathbf{b}^{(L+1)}. \tag{1.2}$$

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 utilized, for example, a 5 layer PIN with 100 neurons per layer for the Schrödinger equation, a 4 layer FNN and 200 neurons per layer for the Allen Cahn equation. In Hu et al. (2024), 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



223 Page 12 of 41 X. Lui et al.

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 highlight the PICNN's effectiveness in solving PDEs on various spherical data, including the shallow water equations and the Allen-Cahn equation on PDEs in spacetime domains. Leveraging a stack of nonlinear convolutional units and a memory mechanism, f-PICNN can handle PDEs with nonlinearity, chaotic 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, $\mathbf{h}_t$, that act as memory units to retain information from previous inputs. This capability makes them suitable for time series prediction, language modeling, and sequence labeling. The theoretical foundations 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.

In a basic RNN cell, the input 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

图 5：f−PICNN 架构

主流架构对比 (2)

RNN (循环神经网络)

- **适用**：时间依赖问题，序列建模。
- **PhyCRNet**：CNN + ConvLSTM。

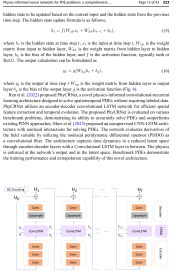


Fig.8 The framework of PhyCRNet (Ren et al. 2022)

Transformer

- **优势**：Attention 捕捉长程依赖。
- **PINNsFormer**：Wavelet + Attention。

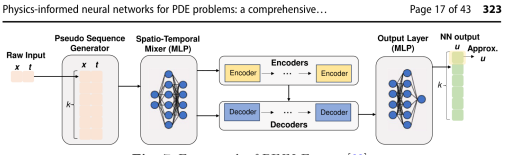


Fig. 7 Framework of PINNsFormer (Zhao et al. 2024)

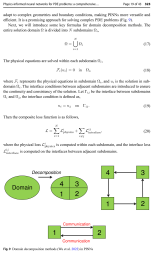
前沿架构：KAN 与区域分解

Kolmogorov–Arnold Networks (KAN)

- 激活函数在**边**上。
- **公式**：$f(x) = \sum \Phi_q(\sum \phi_{q,p}(x_p))$
- **优势**：高维函数更高效，可解释性。

区域分解 (Domain Decomposition)

- **XPINNs / cPINNs**
- 将复杂域划分为子域，并行求解。
- 通过界面条件拼接。



Physics-informed neural networks for PDE problems: a comprehensive ... Page 10 of 30 **A10**

adapt to complex geometries and boundary conditions, making PINNs more versatile and efficient. It is a promising approach for solving complex PDE problems (Fig. 3).

Next, we will introduce the idea of XPINNs and other 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,

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

where $\mathcal{F}_i$ represents the physical equations in subdomain Ω_i and u_i is the solution in sub-domain Ω_i. The interface conditions between adjacent subdomains are introduced to ensure the continuity and consistency of the solutions. 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,

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{physics_i} + \sum_{i,j=1}^{N} \mathcal{L}_{interface_{ij}} \tag{20}$$

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

Fig. 3 Domain decomposition methods (Ma et al. 2023) for PINNs

痛点与解决方案概览

收敛慢 / 陷入局部极小	→ 自适应重采样 (RAR, GAS, HA)
多目标损失不平衡 (梯度病态)	→ 损失重加权 (Loss Reweighting, NTK)
高频/多尺度问题 (频谱偏差)	→ 特征嵌入 (Fourier) + 维度增强 (DaPINN)

自适应采样 (Adaptive Resampling)

- **核心思想**：在 PDE 残差大的区域增加采样点。
- **方法**：
  - **RAR**：基于残差添加点。
  - **GAS**：高斯混合分布采样。
  - **HA (Hybrid Adaptive)**：结合随机性与残差导向。
- **效果**：显著降低 L2 误差 (见图 10)。

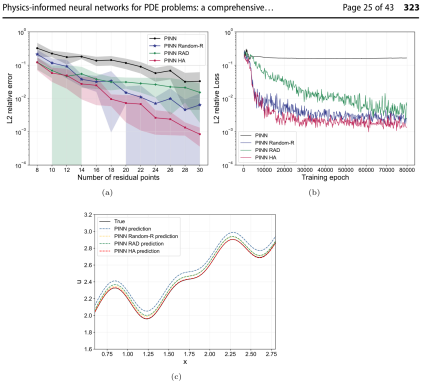


Fig. 10 The performance of our proposed HA sampling method for 1D Poisson equation

图 10：HA 采样方法在泊松方程上的性能提升

特征嵌入与维度增强

特征嵌入 (Feature Embedding)

- 解决 MLP 的频谱偏差。
- **傅里叶特征**：映射输入 $x \to [\sin(2\pi Bx), \cos(2\pi Bx)]$。

维度增强 (DaPINN)

- 人为扩展网络输入维度。
- 提取更丰富特征，提高精度。

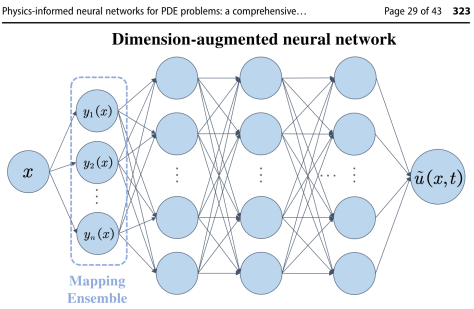


Fig. 11 The framework of dimension-augmented PINN (DaPINN) (Guan et al. 2023)

图 11：DaPINN 框架示意图

DaPINN 实验效果

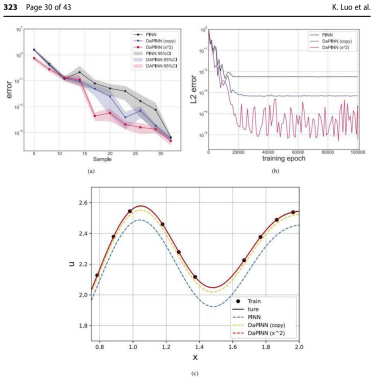


Fig. 12 The performance of DaPINN (Guan et al. 2023) for 1D Poisson equation

图 12：1D 泊松方程 (DaPINN vs PINN)

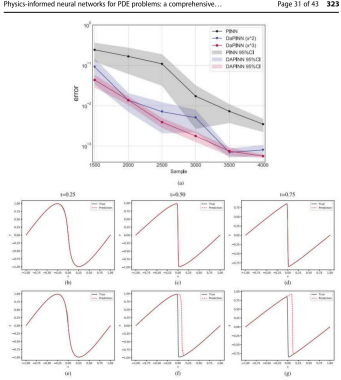


Fig. 13 The performance of DaPINN (Guan et al. 2023) for Burgers equation

图 13：Burgers 方程 (DaPINN vs PINN)

典型应用场景

- **流体动力学**
  - 求解 Navier–Stokes 方程。
  - 根据稀疏观测数据重建流场 (如 PIV 实验)。
  - 无需完整边界条件即可求解反问题。
- **固体力学**
  - 弹性、塑性、断裂力学模拟。
  - 材料参数反演。
- **光学与电磁学**
  - 求解 Maxwell 方程组。
  - 超材料设计与逆散射问题。

挑战与未来展望

当前挑战

1. **优化困难**：非凸优化，梯度不平衡。
2. **计算效率**：训练时间长，收敛精度瓶颈 ($10^{-3} \sim 10^{-4}$)。
3. **高频/多尺度**：传统网络难以捕捉。

未来方向：算子学习 (Operator Learning)

- 从学习“函数”转向学习“算子”。
- **DeepONet / FNO**。
- 一次训练，求解一类方程（不同初始条件/参数）。
- 泛化能力更强，推理速度更快。

总结

- **PINN 是连接数据驱动与物理定律的桥梁。**
- **架构创新** (KAN, Transformer) 不断突破精度极限。
- **优化策略** (自适应采样, 损失平衡) 解决了训练难题。
- **算子学习**代表了下一代科学计算的发展方向。

谢谢!
Q & A