采用 RAR-D 自适应采样求解椭圆界面问题的基于 PINN 的 Kolmogorov-Arnold 网络

Zijuan Xin[a]、Chenyao Wang[a]、Feng Shi[a]、Yizhong Sun[b,*]

[a]中国深圳哈尔滨工业大学理学院[b]中国香港香港浸会大学数学系

arXiv:2602.01876v1 [math.NA] 2 Feb 2026

摘要

物理信息神经网络（PINN）已成为求解偏微分方程（PDE）的常用且强大的框架。它利用神经网络近似解，并将偏微分方程约束、边界条件和界面跳跃条件直接纳入损失函数。然而，现有的大多数 PINN 方法都基于多层感知机（MLP），要达到较高精度，可能需要较大的网络规模和大量训练，尤其是在处理复杂界面问题时。本文提出一种基于 Kolmogorov-Arnold 网络（KAN）的新型 PINN 架构。KAN 在激活函数的选择上更加灵活，并且能够以更少的参数表示函数。具体而言，我们引入一种双 KAN 结构，在不同子域上耦合两个 KAN，并显式施加界面条件。为进一步提高训练效率和收敛速度，我们结合 RAR-D 自适应采样策略，动态优化训练点。椭圆界面问题的数值实验表明，该方法在整个计算域内得到的误差分布更加均匀；这说明，与标准 PINN 相比，基于 PINN 的 KAN 能以显著更小的网络规模和更快的收敛速度达到更高的精度。

关键词： 椭圆界面问题、物理信息神经网络、Kolmogorov–Arnold 网络、双网络、自适应采样

1. 引言

椭圆界面问题在计算科学与工程中发挥着核心作用，应用于多相流 [1], 电磁学 [2], 以及非均质材料中的热传导 [3]。典型模型包含不连续系数和界面跳跃条件，这些因素可能导致解或其通量在嵌入界面两侧出现不连续性。除非对界面进行细致处理，否则这些非光滑特征会使利用标准离散方法获得高精度结果变得困难。包括有限元 [4], 浸入式方法 [5] 和无网格方法 [6], 在内的传统方法已取得良好表现。近年来的进展

*通讯作者（电子邮箱：bill950204@126.com）。

在界面非拟合离散方法和高效求解器方面的进展进一步凸显了对稳健界面处理的持续关注 [7]。然而，对于复杂的界面几何，它们通常需要进行网格构造或在界面附近采用特殊处理，这可能增加实现复杂度和计算成本，并可能影响稳定性。

近年来，深度学习已用于构建偏微分方程（PDE）的无网格求解器。物理信息神经网络（PINN） [8, 9] 利用神经网络逼近解，并通过损失函数施加偏微分方程和边界条件，同时借助自动微分和 GPU 加速实现高效训练。该框架适用于不规则区域，也能减轻网格生成的负担。受区域分解的启发，deepDDM [10, 11] 在各子域中分别训练网络，并通过界面约束将其耦合起来，以求解椭圆问题。相关的基于分解的 PINN 变体包括 cPINN [12], ，它在子域之间施加通量守恒；XPINN [13], ，它将分解思想拓展至更广泛的非线性偏微分方程场景；以及 I-PINN [14], ，它针对界面问题显式纳入界面跳跃条件。

尽管取得了这些进展，标准 PINN 在求解椭圆界面问题时仍面临挑战。现有的大多数 PINN 类方法都采用多层感知机（MLP）[15] 作为基本网络模型。尽管多层感知机具有通用逼近性质 [16], ，但椭圆界面问题的解通常在界面附近不光滑，难以高效解析剧烈的导数／通量跳跃。实际应用中，界面附近的误差和残差往往较大，因此要达到高精度可能需要使用规模较大的网络并进行大量训练。PINN 用于不连续问题计算时也有类似发现：除非引入额外机制，否则非光滑结构可能导致显著的训练困难 [17]。此外，PINN 的性能很大程度上取决于配点的布置：均匀采样可能会在残差最显著的界面附近布置过少的点，从而减缓收敛并造成误差分布不均。

因此，针对界面问题，有两种自然的改进思路：(i) 使用更灵活的网络骨干，以更少的参数更好地拟合由界面主导的解特征；(ii) 使用自适应采样，在残差表明精度不足的位置布置更多训练点。在网络骨干方面，受 Kolmogorov–Arnold 表示定理 [19, 20], 启发的 Kolmogorov–Arnold 网络（KAN） [18], 为多层感知机提供了一种替代方案。不同于依赖固定激活函数的多层感知机，KAN 在网络边上使用可学习的一维函数（通常由 B 样条表示） [18], ，使非线性部分能够在训练过程中进行调整。KAN 已应用于算子学习和物理信息建模，包括基于 KAN 的 DeepONet [21] 以及 KAN 相关的 PINN 框架[22, 23, 24]。在自适应采样方面，基于残差的策略已得到系统研究，并被证明能够提升具有尖锐或局部特征问题的鲁棒性[25, 26, 27]。特别是，基于残差的多样性自适应细化（RAR-D） [25] 会细化高残差区域，同时保持对整个区域的覆盖；当较大的残差集中在界面附近时，这一方法尤为适用。

在本研究中，我们提出了一种基于 PINN 的 KAN，采用双网络区域分解

结构，并结合 RAR-D 自适应采样求解椭圆界面问题。计算域被划分为多个子域，每个子域均由专门的 KAN 表示。通过在损失函数中显式施加界面跳跃条件，将各子域网络耦合起来。我们进一步引入 RAR-D [25]，根据不断变化的残差分布自适应更新配点，从而提高收敛速度并减少界面主导的误差。数值实验表明，与标准的基于 MLP 的 PINN 相比，所提出的基于 PINN 的 KAN 能够实现更均匀的误差分布，并以更小的网络规模和更快的收敛速度达到更高的精度。

本文结构安排如下。第 2 节介绍符号表示和一个椭圆界面模型问题，并回顾一种基线双网络 PINN 公式，该公式采用标准 MLP 主干网络来施加偏微分方程、边界条件和界面跳跃条件。第 3 节介绍我们提出的基于 PINN 的 KAN 框架及其在子域之间的双网络耦合，随后介绍 RAR-D 自适应采样策略，以提高界面主导区域中的训练效率。第 4 节给出若干数值算例，用于展示和比较基于 PINN 的 KAN 与标准物理信息神经网络的精度和可解释性。

2. 模型问题与用于界面问题的基线双网络 PINN

2.1. 椭圆界面问题

首先，我们给出模型问题及相应的界面条件。假设 Ω 是 R^2 中的一个有界区域，其边界为 $\partial\Omega$，界面为Γ，由此 Ω 被分解为两个互不重叠的子域 Ω_1 和 Ω_2。本文采用的椭圆界面问题模型可描述为

$$\begin{aligned} -\nabla\cdot(a_i\nabla u_i) &= f_i, \quad \text{in } \Omega_i,\ i=1,2,\\ [\![a\nabla u\cdot\mathbf{n}]\!] &= \psi, \quad \text{on } \Gamma,\\ [\![u]\!] &= \varphi, \quad \text{on } \Gamma,\\ u_i &= g_i, \quad \text{on } \partial\Omega_i\backslash\Gamma, \end{aligned} \tag{1}$$

其中，$u_i = u_i(x,\ y)$ 是 Ω_i 中的解，f_i 表示源项，$\mathbf{n}$ 表示 Γ 上的单位法向量。我们假设 $\partial\Omega$ 和 Γ 均为 Lipschitz 连续。u 跨越界面的跳跃量表示为 $[\![u]\!] := u|_{\Omega_2} - u|_{\Omega_1}$。系数

$$a(\mathbf{x}) = \begin{cases} a_1(\mathbf{x}), & \text{if } \mathbf{x}\in\Omega_1,\\ a_2(\mathbf{x}), & \text{if } \mathbf{x}\in\Omega_2, \end{cases} \tag{2}$$

是分片空间函数。

为以无网格方式求解 (1)，我们使用神经网络近似各子域的解，并通过物理信息最小二乘损失来施加控制方程、边界条件和界面跳跃条件。为便于说明并为后续比较做准备，我们首先概述一种标准的双 MLP PINN 公式，用于求解椭圆界面问题，并将其作为数值实验中的基准。

2.2. 用于椭圆界面问题的双 *MLP* 物理信息神经网络

深度神经网络（DNN）是基于多层神经元结构的模型，具有强大的数据处理和表征能力。DNN 通常由多个隐藏层构成，其中给定层的每个神经元都与前一层和后一层的神经元相连。通过设置加权连接和非线性激活函数，DNN 能够对复杂的映射关系进行建模。DNN 在图像识别和自然语言处理等多个领域展现出卓越的性能，已成为现代人工智能研究的核心方法[28]。具体而言，对于一个具有 k 层的神经网络，其输出可写为:

$$N^k = \mathbf{W}^k \sigma(N^{k-1}(\mathbf{x})) + \mathbf{b}^k, \quad 2 \leq k \leq L, \tag{3}$$

其中$\mathbf{x}$是神经网络的输入变量， σ 表示激活函数。最后一层 $N^L(\mathbf{x}) = \mathbf{W}^L N^{L-1}(\mathbf{x}) + \mathbf{b}^L$采用恒等激活函数。令 $\theta = \{\mathbf{W}^k, \mathbf{b}^k\} \in \mathrm{V}$ 表示所有权重和偏置的集合，其中 V 表示参数空间。

然而，传统神经网络方法使用单个 DNN 在整个区域 Ω 上近似解 $u(\mathbf{x})$，这通常会导致优化过程缓慢，且难以达到足够的精度。根据界面将 Ω 划分为两个互不相交的子域 Ω_1 和 Ω_2, 后，我们在 Ω_1 和 Ω_2 中分别采用两个独立的 DNN 结构来近似 $u|_{\Omega_1}$ 和 $u|_{\Omega_2}$，记为 $u_i(\mathbf{x})$ $i = 1$，2，$\mathbf{x} \in \Omega_i$。因此，我们有

$$u_i(\mathbf{x}) \approx U_{i,N}(\mathbf{x}, \theta) = N_i^L \circ N_i^{L-1} \circ \cdots \circ N_i^2 \circ N_i^1(\mathbf{x}), \quad i = 1, 2. \tag{4}$$

在 [8], 中，作者提出了一种深度神经网络来近似偏微分方程的解，称为 u 网络。随后应用自动微分来获得方程的微分算子，从而得到一个能够捕捉方程物理信息的 f 网络。接着，基于最小二乘原理构造边界函数和内部损失函数。

确定神经网络的输入后，我们需要利用给定的边界条件和方程信息来构造损失函数。通常会采用最小二乘法，并结合自动微分技术 [29]。求解椭圆界面问题时，我们使用随机拉丁超立方体方法提取数据点，并需要分别建立两个独立的物理信息神经网络来近似u_1 和 u_2。椭圆界面问题的总损失写为

$$\mathcal{L}(\theta) = L_{\Omega_1} + L_{\Omega_2} + L_\Gamma + L_{\partial\Omega_1} + L_{\partial\Omega_2}, \tag{5}$$

其中， L_{Ω_i} ($i = 1$，2) 表示控制方程残差相对于子域 Ω_i 的损失，而 $L_{\partial\Omega_i}$ 和 L_Γ 表示与

相应区域 Ω_i 的边界及界面条件相关的损失，即

$$L_{\Omega_i} = \frac{1}{N_i}\sum_{j=1}^{N_i}\left|-\nabla\cdot(a_i\nabla u_i(\mathbf{x}^j;\theta_i)) - f_i(\mathbf{x}^j)\right|^2, \quad i=1,2, \tag{6}$$

$$L_\Gamma = \frac{1}{N_\Gamma}\sum_{j=1}^{N_\Gamma}\left|[\![u(\mathbf{x}_\Gamma^j;\theta)]\!] - \varphi\right|^2 + \frac{1}{N_\Gamma}\sum_{j=1}^{N_\Gamma}\left|[\![a\nabla u(\mathbf{x}_\Gamma^j;\theta)\cdot\mathbf{n}]\!] - \psi\right|^2, \tag{7}$$

$$L_{\partial\Omega_i} = \frac{1}{N_{\partial\Omega_i}}\sum_{j=1}^{N_{\partial\Omega_i}}\left|u_i(\mathbf{x}_{\partial\Omega_i}^j;\theta_i) - g_i(\mathbf{x}_{\partial\Omega_i}^j)\right|^2, \quad i=1,2. \tag{8}$$

其中，$\{\mathbf{x}^j\}_{j=1}^{N_i}$ 表示在子域 Ω_i (i = 1,2) 内随机采样的点。$\{\mathbf{x}_\Gamma^j\}_{j=1}^{N_\Gamma}$ 和 $\{\mathbf{x}_{\partial\Omega_i}^j\}_{j=1}^{N_{\partial\Omega_i}}$分别表示界面点和边界点。

为求解椭圆界面问题，我们采用双PINNs架构来近似解 u_i(**x**)。用于求解椭圆界面问题的双PINNs结构如图 1 所示。其基本思想是使用两个神经网络来近似椭圆偏微分方程的解，其中每个网络负责问题的一个子域。这些网络通过界面条件耦合，从而确保解及其通量在界面两侧均连续。

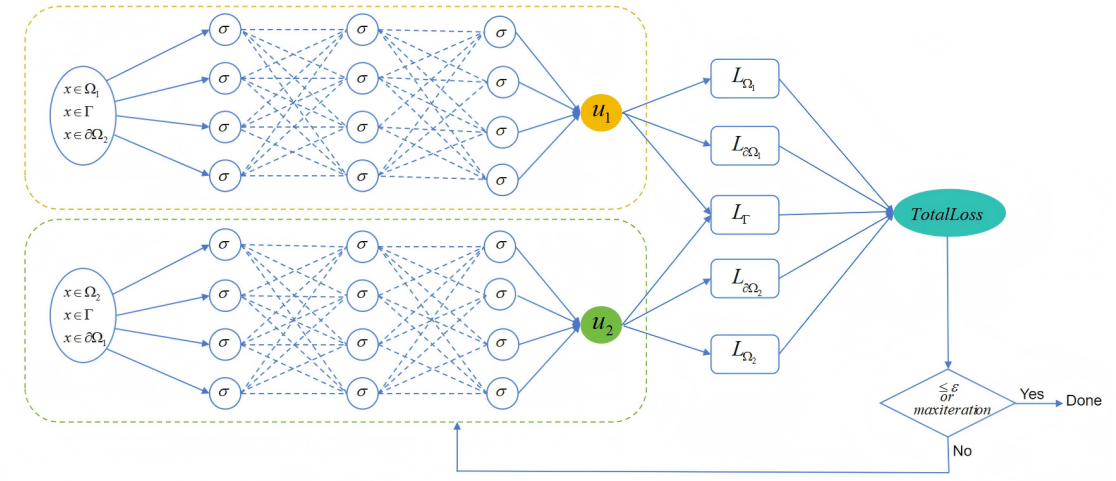


图 1：用于求解椭圆界面问题的双PINNs结构。

为实现这一目标，构造了一个物理信息损失函数，该函数基于给定偏微分方程的残差以及界面和边界条件，如下所示：

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\mathcal{L}(\theta), \tag{9}$$

其中， θ^* 是总损失 $\mathcal{L}(\theta)$的极小化解。

具体而言，残差项通过将神经网络的输出代入偏微分方程得出，而界面项则确保解及其法向导数在界面处连续。边界项则确保计算域边界上满足给定的狄利克雷或诺伊曼条件。

上述双 MLP 物理信息神经网络为椭圆界面问题提供了一个标准基线。然而，在实验中我们发现，要准确解析界面诱发的非光滑特征，可能需要较大的网络容量，并在界面附近谨慎布置采样点。这促使我们在下一节提出基于 PINN 的 KAN 表述和自适应采样策略。

3. 基于 PINN 的 KAN 与 RAR-D 自适应采样

在第 2 节的双物理信息神经网络表述基础上，我们现在介绍所提出的方法。核心思路是在每个子域中以 Kolmogorov–Arnold 网络（KAN）替代 MLP 主干，并通过显式的界面损失耦合各子域网络。随后，我们引入一种基于 RAR-D 的自适应配置点策略，以提高残差集中在界面附近时的训练效率。

3.1. Kolmogorov-Arnold 网络

Kolmogorov–Arnold 表示定理指出，连续多变量函数可以表示为连续单变量函数的叠加[19, 20]。受该定理启发，Kolmogorov–Arnold 网络（KANs）通过可学习的单变量函数对每条边进行参数化，通常使用 B 样条实现[18]。与传统架构相比，这种设计增强了网络的表达能力，使其能够以更少的层或参数逼近复杂的多变量函数。

具体而言，柯尔莫哥洛夫–阿诺德定理保证，对于任意连续函数 $f:\ [0,1]^n \to \mathbb{R}$，都存在连续单变量函数 Φ_q 和$\phi_{q,\ p}$ 使得

$$f\left(x_1, x_2, \ldots, x_n\right) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}\left(x_p\right) \right). \tag{10}$$

KANs 通过引入 KANLinear 层来实现这种分解。这些层在训练过程中学习单变量函数，从而推广了神经网络中的标准线性层。

一般而言，KAN 网络由 L 层组合而成，即给定输入向量$\mathbf{x} \in \mathbb{R}^{n_0}$，KAN 的输出为

$$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}^L \circ \mathbf{\Phi}^{L-1} \circ \cdots \circ \mathbf{\Phi}^2 \circ \mathbf{\Phi}^1)(\mathbf{x}), \tag{11}$$

其中， $\mathbf{\Phi}^k(2 \leq k \leq L)$ 是与第 k 层 KAN 层相对应的函数矩阵。每一层都基于单变量函数进行变换，其定义为

$$\mathbf{\Phi}^k\left(\mathbf{x}^{(k)}\right) = \begin{pmatrix} \phi_{k,1,1}(\cdot) & \cdots & \phi_{k,1,n_k}(\cdot) \\ \phi_{k,2,1}(\cdot) & \cdots & \phi_{k,2,n_k}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{k,n_{k+1},1}(\cdot) & \cdots & \phi_{k,n_{k+1},n_k}(\cdot) \end{pmatrix} (\mathbf{x}^{(k)}), \tag{12}$$

其中， n_k 是第 k 层的输入节点数，$\phi_{k、i、j}$ 是第 k 层的单变量激活函数，在网络的计算图中连接其第 i 个输入节点和第 j 个输出节点。显然，一层的输出节点充当后续层的输入节点。然而，对于 Kolmogorov–Arnold 网络（KANs），「层」的概念并非由节点集合定义，而是由连接节点的边定义，激活函数就位于这些边上。具体而言，对于具有 n_k 个输入节点和 n_{k+1} 个输出节点的层，单变量激活函数的总数等于乘积 $n_k \cdot n_{k+1}$。KAN 的架构可以用一系列整数完全指定，其中每对相邻项分别表示某一层的输入维度和输出维度。一般而言，一个具有 L 层的 KAN 的架构写作 $[n_0, n_1, \ldots, n_L]$。

在 KAN 的原始实现中，单变量激活函数表示为

$$\phi(x) = c_r r(x) + c_B B(x), \tag{13}$$

here

$$r(x) = \frac{x}{1 + \exp(-x)} \tag{14}$$

是一个残差式激活函数，而

$$B\left(x\right) = \sum_{i=1}^{G+m} c_i B_i\left(x\right) \tag{15}$$

是在具有 G 个区间的网格上定义的 m 阶 B样条激活函数。给定 G 和 m 后，一组样条基函数 $\{B_i\}_{i=1}^{G+m}$ 被唯一确定，从而构成式（13）所示的激活函数，该函数取决于网格区间 G。参数 c_r， c_B 和 $\sum_{i=1}^{G+m} c_i$ 均为可训练参数，因此 KAN 中的激活函数并非固定不变，这与多层感知机不同。这种构造使激活函数能够在训练过程中适应问题的具体特征，相较于 ReLU 或 Tanh 等传统固定激活函数具有显著优势。图2示意性地描述了用于求解此类界面问题的、具有结构 [2,3,3,3,1] 的双 KAN 框架。

考虑到解在界面两侧的不连续性，我们提出了一种双KAN架构，分别独立地近似界面两侧的解

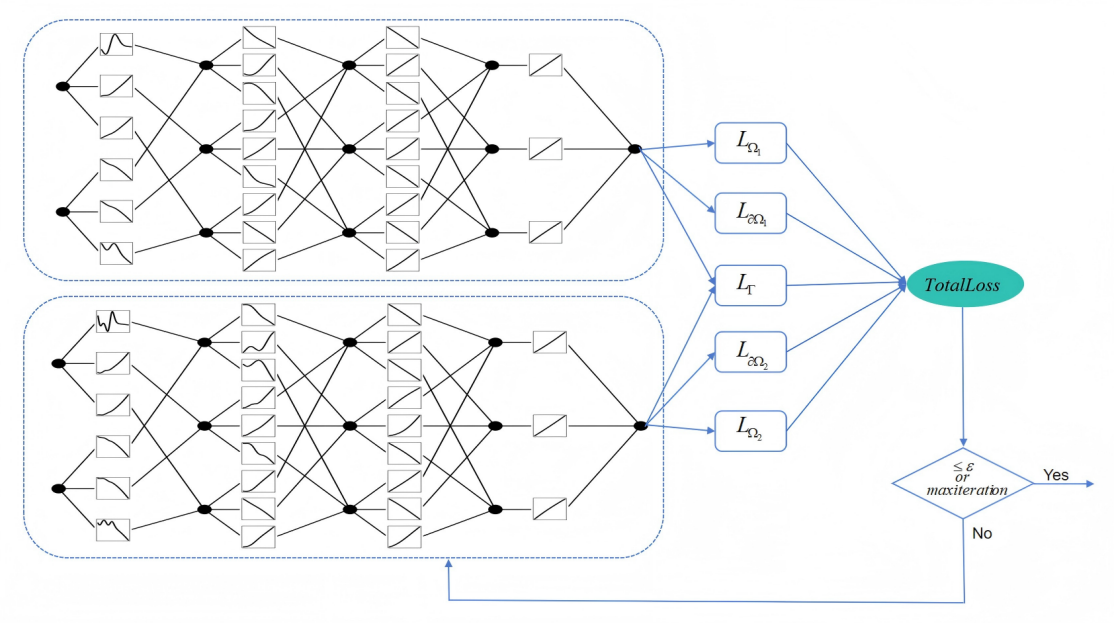


图2：双 KAN 的结构示意图 [2,3,3,3,1]。

以求解椭圆界面问题。具体而言，我们在 Ω_1 和 Ω_2 中构建两个独立的 KAN 结构，分别近似$u|_{\Omega_1}$ 和 $u|_{\Omega_2}$ ，并构造为u_i（$\mathbf{x}$），$\mathbf{x} \in \Omega_i$，$i = 1$，2，即

$$u_i(\mathbf{x}) \approx \mathrm{KAN}_{i,\mathbf{\Phi}}(\mathbf{x}) = (\mathbf{\Phi}_i^L \circ \mathbf{\Phi}_i^{L-1} \circ \cdots \circ \mathbf{\Phi}_i^2 \circ \mathbf{\Phi}_i^1)(\mathbf{x}), \quad i = 1, 2. \tag{16}$$

在椭圆界面问题中，训练残差在空间上的分布通常高度不均匀。在一些常见情形下，这一现象尤为明显，例如：(i) 系数对比度较大 a_1，a_2；(ii) 跳跃数据较强或随空间变化（φ，ψ）；以及 (iii) 曲面界面处的局部解特征沿 Γ快速变化。在这些情况下，均匀配点可能会在残差占主导的界面邻域分配过少的点，从而导致收敛缓慢，并产生界面局部误差。因此，我们采用残差驱动自适应采样，并使用下文所述的 RAR-D 策略。

3.2. 基于残差的多样性自适应细化（*RAR-D*）

文献 [25]提出了 RAR-D 自适应采样，以提升物理信息神经网络（PINN）的性能。该方法旨在根据训练过程中的偏微分方程残差动态调整残差点的分布，从而以更少的残差点实现更高效的训练，提升物理信息神经网络的精度。同时，本文将 RAR-D 自适应采样方法应用于 KAN 网络。

密度函数 p(x)的概念被设计为与偏微分方程的界面残差成正比，并且所有残差均基于该密度函数重新采样。因此，新生成的残差点按照

该密度函数分布。对于任意点$\mathbf{x}$，我们首先计算偏微分方程残差 $\varpi^i(\mathbf{x}) = |-\nabla \cdot (a_i \nabla u_i(x^j; \theta)) - f_i|$，然后计算概率为

$$p(\mathbf{x}) \propto \frac{\varpi^k(\mathbf{x})}{\mathbb{E}\left[\varpi^k(\mathbf{x})\right]} + c. \tag{17}$$

这里，$k \geq 0$ 和 $c \geq 0$ 是两个超参数。$\mathbb{E}\left[\varpi^k(\mathbf{x})\right]$ 可以通过数值积分（例如蒙特卡罗积分）进行近似。具体的自适应采样算法如算法 1 所示。

算法 1 RAR-D [25]

步骤 1： 使用拉丁超立方采样方法生成初始采样点 $\mathcal{T}$；

步骤 2： 训练网络模型若干次迭代；**步骤 3：** 通过对偏微分方程进行自动微分，计算初始采样点的残差；

步骤 4： 使用公式 (17) 中的密度函数 $p(\mathbf{x})$ 选择新的采样点，以替换初始残差点：

1. 计算 $p(\mathbf{x})$ 在 $\mathcal{T}$ 中各点处的值；2. 归一化后的概率密度函数 $p(\mathbf{x})$ 定义了新的概率密度函数 $\tilde{p}(\mathbf{x}) = {}^{p}{}_{(\mathbf{x})\ A}$，其归一化常数为 $A = \sum_{\mathbf{x}\in\mathcal{T}} p(\mathbf{x})$；3. 根据 $\tilde{p}(\mathbf{x})$，更新训练集 $\mathcal{T}$。 ——

步骤 5： 继续训练，直到总迭代次数达到上限或残差收敛。

End

4. 数值实验

本节介绍了若干用于求解二维椭圆界面问题的数值实验。首先，构造了具有解析解的数值算例，以验证并比较物理信息神经网络和 KAN 的准确性与有效性。通过改变 KAN 的神经元数量和 G 区间数，我们研究了这些因素对椭圆界面问题求解的影响，从而评估 KAN 的内部结构对解的精度的影响。随后，我们构造了一个在界面处存在不连续性的解析解，并比较了两种不同网络架构得到的数值结果。接着，我们设计了一种自适应采样算法，并将其与物理信息神经网络和 KAN 相结合，以展示自适应采样对求解椭圆界面问题的影响。将这两种模型与自适应采样算法相结合后的结果分别称为“PINNs-A”和“KANs-A”。

为评估泛化能力，我们在由 N_{test} 个随机采样点构成的 Ω 内独立测试集上评估训练后的网络，并报告最大绝对误差和相对 ℓ_2误差，其定义如下:

$$|u-\hat{u}|_{\infty}=\max_{1\leq j\leq N_{\text{test}}}|u(\mathbf{x}^j)-\hat{u}(\mathbf{x}^j)|,\quad \|u-\hat{u}\|_{\ell_2}=\sqrt{\frac{\sum_{j=1}^{N_{\text{test}}}|u(\mathbf{x}^j)-\hat{u}(\mathbf{x}^j)|^2}{\sum_{j=1}^{N_{\text{test}}}|u(\mathbf{x}^j)|^2}}. \tag{18}$$

其中$\hat{u}$ 为数值解函数。在下文的数值实验中，e_{Ω_1}、 e_{Ω_2}、 e_{Γ}、 $e_{\partial\Omega_1}$ 和 $e_{\partial\Omega_2}$ 分别表示 Ω_1,Ω_2，界面 Γ 和边界 $\partial\Omega_1$,$\partial\Omega_2$，上的 ℓ_2- 误差。

4.1. 连续界面

例4.1。 作为第一个例子，我们考虑正方形区域 $\Omega=[-1,1]\times[-1,1]$上的简单泊松方程，其具有圆形界面 $\Gamma=\{\mathbf{x}:\ x^2+y^2=0.25\}$。该界面将区域分为两个子区域 $\Omega_1=\{\mathbf{x}:\ x^2+y^2\leq 0.25\}$ 和 $\Omega_2=\Omega\backslash\Omega_1$。系数取为 $a_1=a_2=1$，解析解选为

$$u(\mathbf{x})=\begin{cases}1 & \text{in } \Omega_1,\\ 1-\log_2\sqrt{x^2+y^2} & \text{in } \Omega_2.\end{cases} \tag{19}$$

将上述精确解代入式(1)，即可得到源项 f_i、边界数据 g_i和跳跃数据 ψ,φ。

物理信息神经网络和 Kolmogorov-Arnold 网络均用于求解该问题。我们采用具有 3 层、每层 20 个神经元的双PINN，以及具有 3 层、每层 3 个神经元的双 KAN。在 KAN 中，我们使用 $G=10$ 来控制样条函数的精细程度。此外，采用拉丁超立方采样，分别在区域 Ω_1中取 $N_1=200$，在区域 Ω_2中取 $N_2=500$，在界面 Γ上取 $N_\Gamma=300$，并在边界 $\partial\Omega_2$, 上取 $N_{\partial\Omega_2}=800$。

首先，图 3(a) 展示了解析解的等高线图，图 3(b)–(c) 分别给出了物理信息神经网络和 KAN 得到的近似解。相应的绝对误差解分别绘于图 3(d)–(e)。我们可以看到，所提出的物理信息神经网络和 KAN 均能通过逼近界面问题的解来捕捉界面；与此同时，从误差图可以看出，与物理信息神经网络相比，KAN 的逼近性能显著更优。我们还在图 4 中绘制了物理信息神经网络和 KAN 解的损失函数演化情况，这表明，与物理信息神经网络相比，KAN 使用更小的网络即可加速损失函数的收敛。

图 5 展示了耦合 Kolmogorov-Arnold 网络训练后的激活函数，初始激活函数可参见图 2，以供比较。可以观察到，训练过程中每一层的基函数都会进行自适应调整，从而形成多样化的非线性表示能力。这一机制使网络在保持相对小规模的同时，仍具备强大的表达能力和较高的收敛效率。

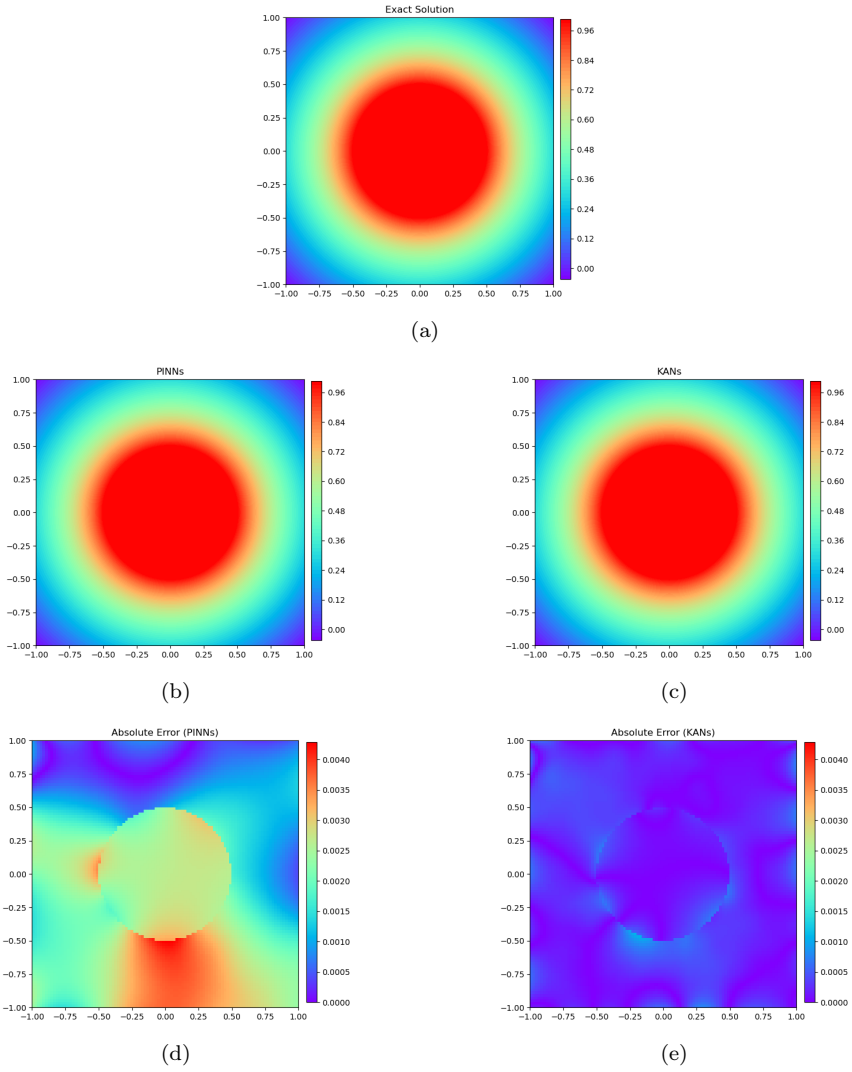


图3：例4.1的等高线图：精确解(a)、物理信息神经网络(b)和KAN(c)的近似解，以及物理信息神经网络(d)和KAN(e)的绝对误差。

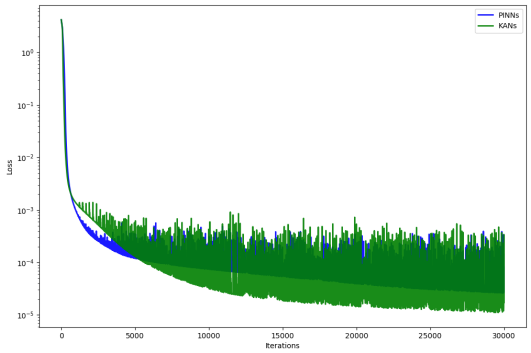


图4：例4.1中损失误差的演化。

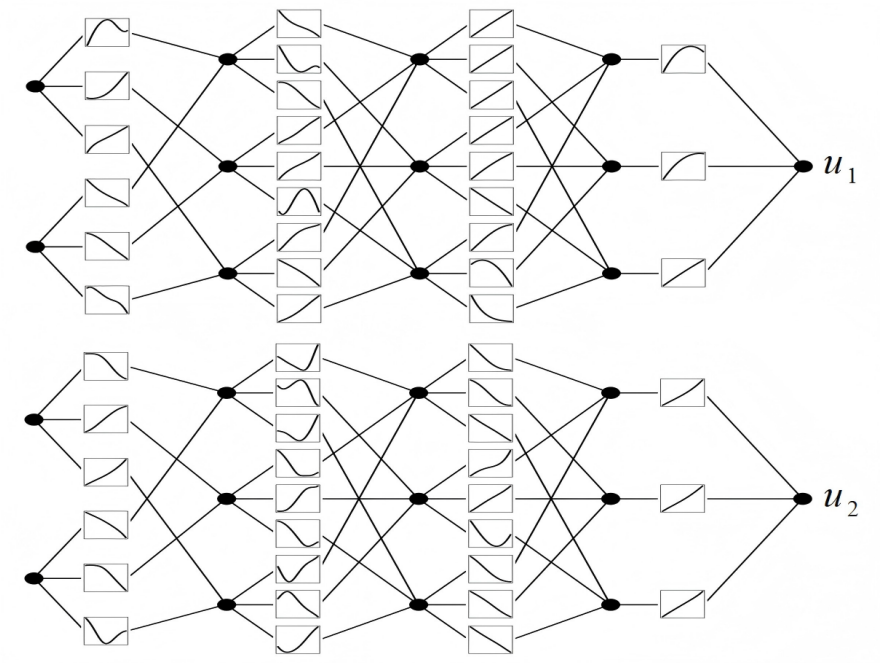


图 5：例4.1 中双 KAN 训练后的激活函数可视化。

KAN 的性能取决于网络架构和网格区间G。为此，表 1 展示了不同神经元数量下的相对 ℓ_2 误差（ $G = 10$ ），表 2 则列出了固定神经元=5 数量、采用不同 G区间数量时 KAN 的结果。结果表明，当固定一个参数并改变另一个参数时，无论是神经元数量还是 G区间数量，都不会显著影响计算误差。因此，可以选择合适的网络深度和宽度，以实现良好的精度和效率。

在自适应采样阶段，物理信息神经网络和 KAN 网络均使用 Adam 训练 40000 步。我们生成 5000 个初始采样点进行 20000 步训练，随后每训练 2000 步进行一次自适应重采样（RAR-D）。这里， $k = 2$ 和 $c= 0$。在相同的初始采样条件下，我们分别对物理信息神经网络和 KAN 进行训练，并采用 RAR-D 自适应算法，然后将计算结果列于表 3。从表中可以得出，两种网络的自适应采样方法都能有效提高 ℓ_2-范数下的计算精度。更重要的是，可以看出 KAN 第二行中的每个误差都远小于物理信息神经网络第一行中的相应误差。此外，从图 3(d)-(e) 可以看出，KAN 的计算误差分布比物理信息神经网络的误差分布均匀得多。这些观察结果清楚地表明，在求解椭圆界面问题时，KAN 的性能优于物理信息神经网络。

表 1：例4.1中不同神经元数量的 KANs 误差比较。

神经元数量	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$
3	1.456×10^{-4}	5.194×10^{-4}	3.402×10^{-4}	3.022×10^{-3}
5	2.565×10^{-4}	3.982×10^{-4}	3.064×10^{-4}	1.124×10^{-3}
7	2.074×10^{-4}	7.310×10^{-4}	2.587×10^{-4}	1.485×10^{-3}

表 2：例4.1中 KANs 在不同 G区间下的误差比较。

G 区间	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$
5	5.232×10^{-4}	9.185×10^{-4}	1.002×10^{-3}	2.421×10^{-3}
10	2.565×10^{-4}	3.982×10^{-4}	3.064×10^{-4}	1.124×10^{-3}
15	4.298×10^{-5}	3.281×10^{-4}	2.343×10^{-4}	1.677×10^{-3}

表 3：例4.1中采用相同初始采样和 RAR-D 自适应采样训练时的误差比较。

网络	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$	$\lvert u - \hat{u}\rvert_\infty$
物理信息神经网络	9.790×10^{-4}	3.280×10^{-3}	1.046×10^{-3}	8.297×10^{-3}	3.693×10^{-3}
KANs	2.040×10^{-4}	5.024×10^{-4}	2.889×10^{-4}	1.156×10^{-3}	8.079×10^{-4}
PINNs-A	5.634×10^{-4}	8.883×10^{-4}	6.641×10^{-4}	2.842×10^{-3}	1.075×10^{-3}
KANs-A	1.035×10^{-4}	1.927×10^{-4}	1.680×10^{-4}	6.512×10^{-4}	3.437×10^{-4}

4.2. 不连续界面

例4.2。 本例采用一种更复杂的界面，其参数化形式为

$$\begin{cases} x(\theta) = (a + b\cos(m\theta))\sin(n\theta)\cos(\theta), \\ y(\theta) = (a + b\cos(m\theta))\sin(n\theta)\sin(\theta), \end{cases} \tag{20}$$

其中 $\theta \in [0, 2\pi]$、 $a = b = 0.40178$、 $m = 2$ 和 $n = 6$。在区域 $\Omega = [-1, 1] \times [-1, 1]$ 中，Ω_1 和 Ω_2 分别定义为界面 Γ内侧和外侧的区域。系数 a_i 和解如下

$$a = \begin{cases} \frac{x^2 - y^2 + 3}{7}, & \text{in } \Omega_1, \\ \frac{2 + xy}{5}, & \text{in } \Omega_2, \end{cases} \qquad u(\mathbf{x}) = \begin{cases} \sin(x + y) + \cos(x + y) + 1, & \text{in } \Omega_1, \\ x + y + 1, & \text{in } \Omega_2. \end{cases}$$

该问题解析解的等高线图见图6(a)。为求解例4.2，构建了一个包含3层、每层20个神经元的双PINN，以及一个包含3层、每层3个神经元的双 KAN。两个网络均选择$N_1 = 300$， $N_2 = 500$， $N_\Gamma = 300$， $N_{\partial\Omega_2} = 800$ ，并在

KAN中选择{v11}。与例4.1类似，我们在图6(b)-(c)中展示了PINN和KAN的解的等高线图，并在图6(d)-(e)中展示了相应的绝对误差。此外，图7展示了PINN和KAN的损失函数演变情况，表明KAN比PINN收敛更快。

表4和表5列出了KAN随神经元数量或 G 区间变化时的性能。从这两张表来看，就 L^2范数误差而言，3个神经元适用于带有 $G=5$的KAN，且此类KAN对 G 区间规模的敏感度低得多，因此在当前情形下 $G=5$ 已足够。

至于自适应采样，物理信息神经网络和KAN这两种网络均使用Adam训练30000步。在Adam训练阶段，我们首先生成5000个残差点并训练20000步，随后每1000次迭代应用一次自适应重采样（RAR-D）。此处， $k=2$ 和 $c=1$。表6给出了物理信息神经网络与KAN及其自适应性能在 L^2误差意义下的比较。

显然，自适应算法对两种网络均表现出更好的性能，且在相同初始采样条件下KAN优于物理信息神经网络。

表4：例4.2中不同神经元数量的 Kolmogorov-Arnold 网络误差比较

神经元数量	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$
3	6.865×10^{-5}	5.646×10^{-5}	7.124×10^{-5}	8.841×10^{-5}
5	2.779×10^{-4}	2.832×10^{-4}	2.714×10^{-4}	2.714×10^{-4}
7	1.051×10^{-4}	8.843×10^{-5}	1.022×10^{-4}	7.650×10^{-5}

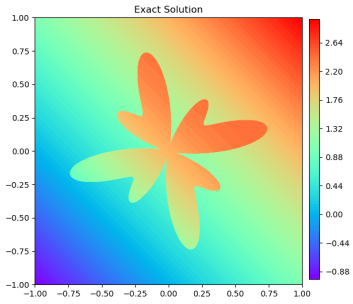
表5：例4.2中 Kolmogorov-Arnold 网络在不同 G区间下的误差比较。

G **区间**	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$
5	3.459×10^{-5}	2.892×10^{-5}	3.028×10^{-5}	3.453×10^{-5}
10	2.779×10^{-4}	2.832×10^{-5}	2.714×10^{-4}	2.714×10^{-5}
15	9.490×10^{-5}	1.031×10^{-4}	8.375×10^{-5}	1.236×10^{-4}

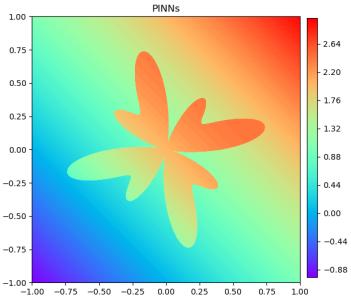
表6：例4.2中采用相同初始采样进行训练与采用 RAR-D 自适应采样进行训练的误差比较。

网络	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$	$\lvert u-\hat{u}\rvert_\infty$
物理信息神经网络	2.935×10^{-4}	4.671×10^{-4}	2.767×10^{-4}	4.188×10^{-4}	1.077×10^{-3}
KANs	5.357×10^{-5}	8.319×10^{-5}	5.255×10^{-5}	7.743×10^{-5}	2.251×10^{-4}
PINNs-A	1.012×10^{-4}	1.022×10^{-4}	1.031×10^{-4}	9.879×10^{-5}	4.225×10^{-4}
KANs-A	3.648×10^{-5}	2.177×10^{-5}	3.679×10^{-5}	3.146×10^{-5}	1.512×10^{-4}

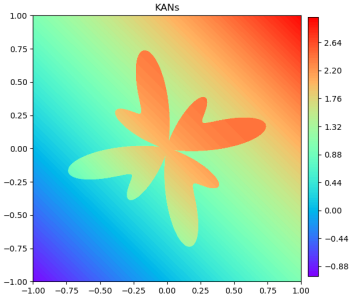
下面，我们再给出两个界面/边界几何复杂的椭圆界面问题示例。



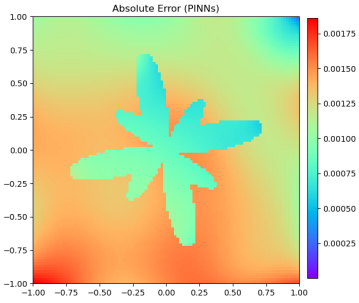
(a)



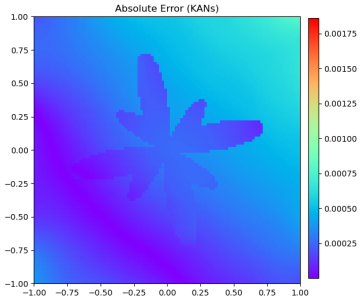
(b)



(c)



(d)



(e)

图6：例4.2的等高线图，包括精确解(a)、物理信息神经网络(b)和Kolmogorov-Arnold 网络(c)得到的近似解，以及物理信息神经网络(d)和Kolmogorov-Arnold 网络(e)得到的绝对误差。

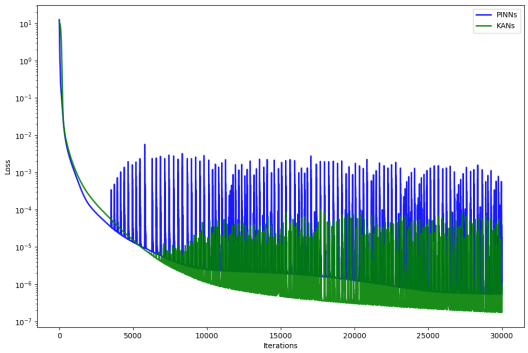


图7：例4.2的损失误差演化。

例4.3。 计算区域由两个高度不规则的非凸子域组成，其界面定义为

$$r = 0.6 + 0.216\sin(3\theta) + 0.096\cos(2\theta) + 0.24\cos(5\theta), \tag{21}$$

边界在极坐标下表示为

$$r = 0.14\sin(4\theta) + 0.12\cos(6\theta) + 0.09\cos(5\theta), \tag{22}$$

其中 $\theta \in [0, 2\pi]$。在区域 $\Omega = [-2,2]\times[-2,2]$ 上，Ω_1 和 Ω_2 分别定义为 Γ 内部和外部的区域。系数 a_i 和解定义如下

$$a = \begin{cases} x^2+y^2, & \text{in } \Omega_1, \\ xy, & \text{in } \Omega_2, \end{cases} \qquad u(\mathbf{x}) = \begin{cases} \cos(x+y), & \text{in } \Omega_1, \\ \sin(x+y), & \text{in } \Omega_2. \end{cases} \tag{23}$$

例4.4。 考虑一个内半径和外半径分别为 $r_{in} = 0.151$ 和 $r_{out} = 0.911$ 的环形区域，其中嵌入的星形界面 γ 由以下水平集函数描述：

$$\phi(x,y) = \sqrt{x^2+y^2} - r_0\left(1 + \sum_{k=1}^{3}\beta_k\cos\left(n_k\left(\arctan\left(\frac{y}{x}\right) - \theta_k\right)\right)\right), \tag{24}$$

and

$$r_0 = 0.483, \quad \begin{pmatrix} n_1 \\ \beta_1 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0.1 \\ 0.5 \end{pmatrix}, \quad \begin{pmatrix} n_2 \\ \beta_2 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 4 \\ -0.1 \\ 1.8 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} n_3 \\ \beta_3 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 0.15 \\ 0 \end{pmatrix}.$$

其中 $\theta \in [0, 2\pi]$。区域 $\Omega = [-1,1]\times[-1,1]$ 被该界面划分为 Ω_1 和 Ω_2，，其中系数 a_i 和解定义为

$$a = \begin{cases} 10\left(1 + \frac{1}{5}\cos(2\pi(x+y))\sin(2\pi(x-y))\right), & \text{in } \Omega_1, \\ 1, & \text{in } \Omega_2, \end{cases} \tag{25}$$

$$u(\mathbf{x}) = \begin{cases} \sin(2x)\cos(2y), & \text{in } \Omega_1, \\ \left(16\left(\frac{y-x}{3}\right)^5 - 20\left(\frac{y-x}{3}\right)^3 + 5\left(\frac{y-x}{3}\right)\right)\log(x+y+3), & \text{in } \Omega_2. \end{cases} \tag{26}$$

对于这两个例子，我们采用相同的网络设置，即双PINN包含3层，每层20个神经元；双 KAN 包含3层，每层5个神经元。例4.3和例4.4的解析解、数值解/误差解的等高线图以及损失误差的演变分别如图8–11所示。两个网络均选取$N_1 = 300$, $N_2 = 500$, $N_\Gamma = 300$，以及 $N_{\partial\Omega_2} = 800$（例4.3）和300（例4.4），并在 KAN 中选取 $G = 5$。

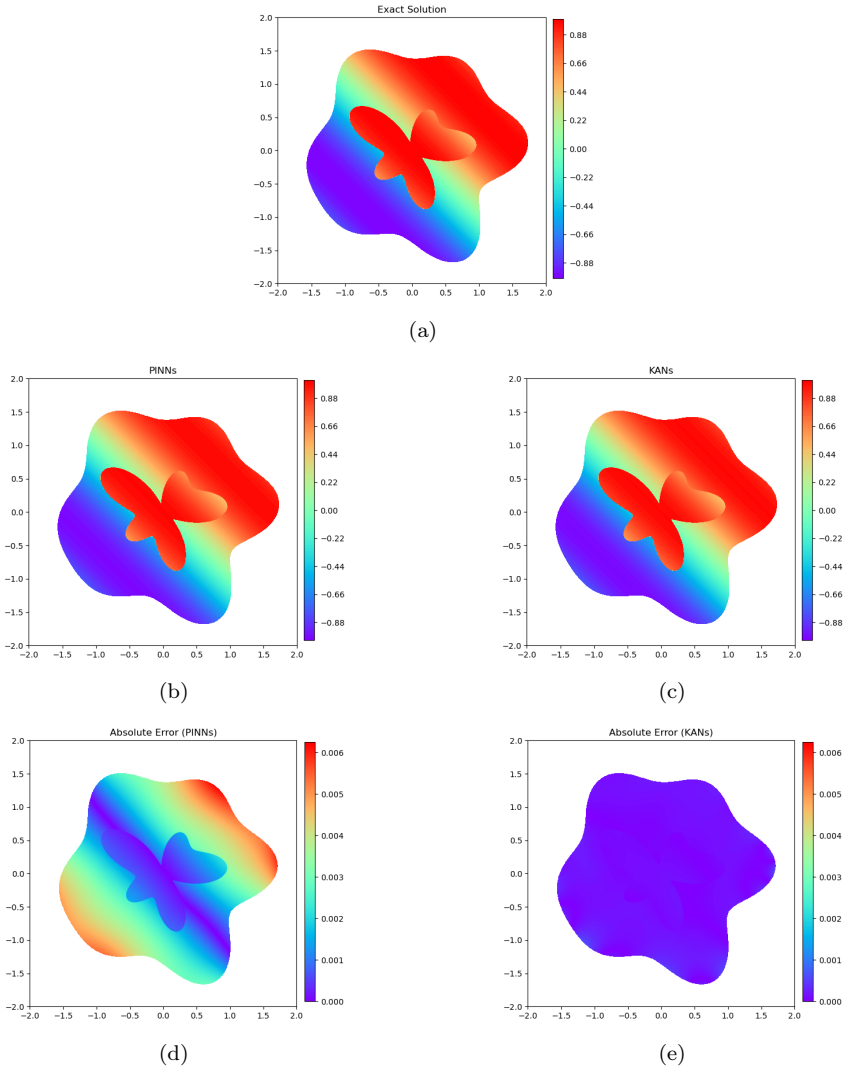


图8：例4.3的等高线图，包括精确解(a)、物理信息神经网络(b)和Kolmogorov-Arnold 网络(c)的近似解，以及物理信息神经网络(d)和Kolmogorov-Arnold 网络(e)的绝对误差。

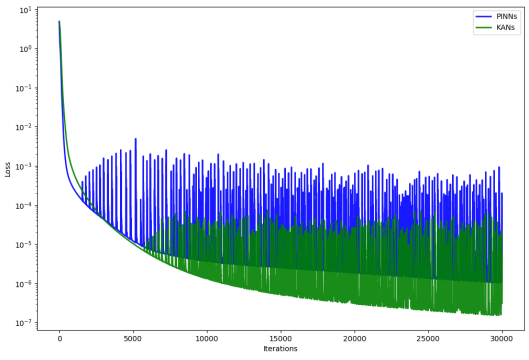


图9：例4.3中损失误差的演化。

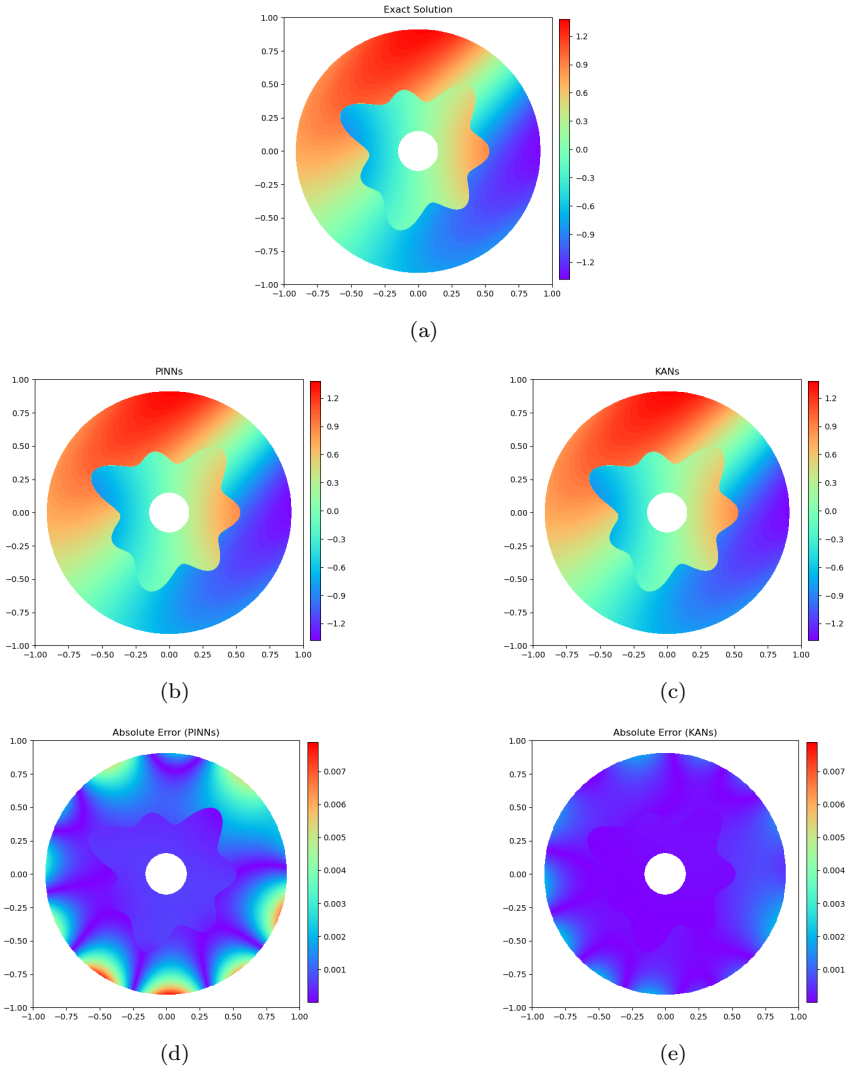


图10：例4.4的等高线图，包括精确解(a)、物理信息神经网络(b)和Kolmogorov-Arnold 网络(c)的近似解，以及物理信息神经网络(d)和Kolmogorov-Arnold 网络(e)的绝对误差。

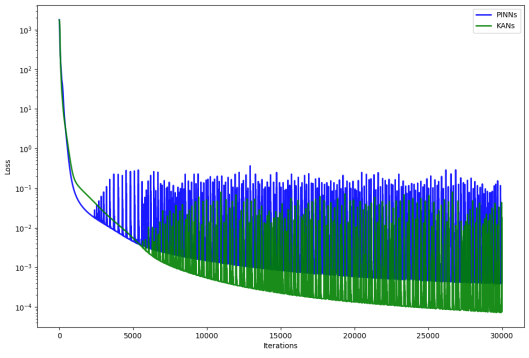


图11：例4.4中损失误差的演化。

表7：例4.3中使用相同初始采样和 RAR-D 自适应采样训练的误差对比

网络	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_2}$	$\|u-\hat{u}\|_\infty$
物理信息神经网络	1.409×10^{-3}	3.766×10^{-3}	1.527×10^{-3}	2.843×10^{-3}	3.568×10^{-3}
KANs	1.725×10^{-4}	1.789×10^{-4}	2.599×10^{-4}	2.567×10^{-4}	5.645×10^{-4}
PINNs-A	6.813×10^{-4}	1.689×10^{-3}	8.352×10^{-4}	1.384×10^{-3}	1.815×10^{-3}
KANs-A	4.618×10^{-5}	3.748×10^{-5}	5.695×10^{-5}	6.805×10^{-5}	1.676×10^{-4}

表8：例4.4中使用相同初始采样和RAR-D 自适应采样进行训练时的误差比较。

网络	e_{Ω_1}	e_{Ω_2}	e_{Γ}	$e_{\partial\Omega_1}$	$e_{\partial\Omega_2}$	$\|u-\hat{u}\|_\infty$
PINNs	6.628×10^{-4}	2.821×10^{-3}	1.113×10^{-3}	1.516×10^{-3}	4.559×10^{-3}	9.426×10^{-3}
KANs	2.986×10^{-4}	7.623×10^{-4}	4.098×10^{-4}	7.298×10^{-4}	1.552×10^{-3}	2.709×10^{-3}
PINNs-A	2.131×10^{-4}	1.326×10^{-3}	6.364×10^{-4}	1.429×10^{-4}	3.295×10^{-3}	6.606×10^{-3}
KANs-A	8.515×10^{-5}	4.902×10^{-4}	1.522×10^{-4}	1.298×10^{-4}	8.365×10^{-4}	1.906×10^{-3}

选择 $k=2$ 和 $c=0$。在自适应采样设置中，物理信息神经网络和KAN网络均使用Adam重新训练40000步。由于例4.3的区域较大，我们生成10000个初始采样点，并先对模型训练20000步。随后，每经过2000次迭代便应用一次自适应重采样（RAR-D）（例4.3）。同样，对于例4.4，我们先生成5000个初始采样点，并对模型训练20000步，之后每经过2000次迭代便应用一次自适应重采样（RAR-D）。

表7和表8分别展示了例4.3和例4.4中采用相同初始采样进行训练，以及采用相应自适应采样进行训练时的相对 ℓ_2 误差。正如例4.2中的结论所示，我们可以清楚地看到，自适应算法提升了物理信息神经网络和KAN的性能，而KAN得到的近似结果优于物理信息神经网络。

5. 结论

本文聚焦于使用物理信息神经网络（PINNs）和Kolmogorov-Arnold网络（KANs）求解椭圆界面问题。我们为PINNs和KANs均采用了双网络框架。该网络在处理界面处的跳跃条件方面具有更大的灵活性。通过多次数值实验，我们证明，KANs在近似精度和误差分布方面表现更优。为进一步提高网络效率，我们将其与基于残差的多样性自适应细化（RAR-D）采样策略相结合。该组合可在训练过程中动态调整采样点，从而获得更好的网络性能。数值结果表明，KANs不仅能够获得更好的近似解

而且与PINNs相比仅需更小的模型规模，因此成为求解椭圆界面问题的一种有前景的替代方案。这表明，与基于传统多层感知机（MLPs）的神经网络相比，KANs在求解椭圆界面问题时具有更高的精度和更强的可解释性。

参考文献

[1] Hou T. Y., Li Z., Osher S. 和 Zhao H., 一种用于移动界面问题的混合方法及其在赫勒–肖流中的应用，计算物理学杂志，134 (2)，第 236–252 页，1997年。[2] Lin T. 和 Wang J., 一种用于离子光学建模的浸入有限元电场求解器，美国航空航天学会，2002年。[3] Ji H., Zhang Q., Wang Q. 和 Xie Y., 一种用于具有非齐次跳跃条件的椭圆界面问题的部分惩罚浸入有限元法，东亚应用数学杂志，8，第 1–23 页，2018年。[4] Ciarlet P.G., 《椭圆问题的有限元方法》，北荷兰出版社，1978年。[5] Ji H. 和 Li Z., 一种用于具有非齐次跳跃条件的各向异性椭圆界面问题的浸入有限元法 《科学计算杂志》，106 (27)，2026年。[6] Mu R., Song L. 和 Qin Q., 一种基于广义有限差分法的二维和三维各向异性椭圆界面问题无网格方法，工程分析与边界元，163，第 505–516 页，2024。[7] Chu H., Song Y., Ji H. 和 Cai Y., 用于椭圆界面问题浸入有限元离散的多重网格算法，《科学计算杂志》，98 (26)，2024。[8] Raissi M., Perdikaris P. 和 Karniadakis G. E., 物理信息神经网络：用于求解涉及非线性偏微分方程的正问题与逆问题的深度学习框架，计算物理学杂志，378，第 686–707 页，2019年。[9] Cuomo S., Schiano Di Cola V., Giampaolo F., Rozza G., Raissi M. 和 Piccialli F.,基于物理信息神经网络的科学机器学习：现状与未来，《科学计算杂志》，92 (88)，2022年。[10] Li W., Xiang X. 和 Xu Y., 深度区域分解方法：椭圆问题，数学、科学与机器学习，第 269–286 页，2020年。[11] Wu W., Feng X. 和 Xu H., 通过区域分解改进用于求解偏微分方程的深度神经网络，《科学计算杂志》，93 (20)，2022年。

[12] Jagtap A. D., Kharazmi E. 和 Karniadakis G. E., 离散域上的守恒型物理信息神经网络及其在守恒律中的应用：正问题与逆问题，《计算方法与应用力学和工程》，365，113028，2020年。[13] Jagtap A. D. 和 Karniadakis G. E., 扩展物理信息神经网络（*XPINNs*）：一种基于时空域分解的非线性偏微分方程通用深度学习框架，《计算物理通讯》，28 (5)，2020年。[14] Sarma A. K., Roy S., Annavarapu C., Roy P. 和 Jagannathan S., 界面*PINN*（*I-PINNs*）：一种用于界面问题的物理信息神经网络框架，《计算方法与应用力学和工程》，429，117135，2024。[15] Haykin S.,《神经网络：全面基础》，Prentice Hall PTR，1998年。[16] Hornik K., Stinchcombe M. 和 White H., 多层前馈网络是通用逼近器，《神经网络》，2 (5)，第 359–366 页，1989年。[17] Liu L., Liu S., Xie H., Xiong F., Yu T., Xiao M., Liu L. 和 Yong H., 利用物理信息神经网络计算不连续问题，《科学计算杂志》，98 (22)，2024年。[18] Liu Z., Wang Y., Vaidya S., Rühle F., Halverson J., Soljačić M., Hou T. Y. 和 Tegmark M., *KAN*：*Kolmogorov-Arnold*网络，arXiv:2404.19756，2024年。[19] Kolmogorov A.N., 通过较少变量的连续函数叠加表示多变量连续函数，美国数学学会，1961年。[20] Braun J. 和 Griebel M., *Kolmogorov*叠加定理的构造性证明，《构造逼近》，30，第 653–675 页，2009年。[21] Abueidda D. W., Pantidis P. 和 Mobasher M. E., *Deepokan*：一种基于*Kolmogorov Arnold*网络、用于力学问题的深度算子网络，《计算方法与应用力学和工程》，436，117699，2025年。[22] Wang Y., Sun J., Bai J., Anitescu C., Eshaghi M. S., Zhuang X., Rabczuk T. 和 Liu Y., *Kolmogorov Arnold*信息神经网络：一种基于*Kolmogorov-Arnold*网络、用于求解偏微分方程的物理信息深度学习框架，arXiv:2406.11045，2024年。[23] Howard A. A., Jacob B., Murphy S. H., Heinlein A. 和 Stinis P., 有限基*Kolmogorov-Arnold*网络：面向数据驱动和物理信息问题的区域分解，arXiv:2406.19662，2024年。[24]So C. C. 和 Yung S. P., 用于更准确、稳健和快速求解物理信息神经网络（*PINN*）的高阶*ReLU-KAN*（*HRKAN*），arXiv:2409.14248

，2024
年。

[25] Wu C., Zhu M., Tan Q., Kartha Y., 和 Lu L., 物理信息神经网络非自适应采样与基于残差的自适应采样综合研究，《计算方法与应用力学和工程》，403，115671，2023年。[26] Mao Z. 和 Meng X., 采用基于残差/梯度的自适应采样方法求解具有尖锐解的偏微分方程的物理信息神经网络，应用数学与力学，44 (7)，第 1069–1084 页，2023年。[27] Gao Z., Yan L., 和 Zhou T., 物理信息神经网络的失效信息驱动的自适应采样，SIAM科学计算杂志，45 (4)，第 A1971–A1994 页，2023年。[28] Samek W., Montavon G., Lapuschkin S., Anders C. J., 和 Müller K.-R., 深度神经网络及更多内容的解释：方法与应用综述，IEEE会刊，109 (3)，第 247–278 页，2021年。[29] Baydin A. G., Pearlmutter B. A., Radul A. A., 和 Siskind J. M., 机器学习中的自动微分：综述，机器学习研究杂志，18 (153)，第 1–43 页，2018年。