

A residual-based adaptive refinement physics-informed neural networks (RAR-PINNs) method for fifth-order KdV equation

Shi-Fang Tian(田十方), Ya-Xuan Yu(于亚璇), and Biao Li(李彪)
Citation: Chin. Phys. B, 2026, 35 (**5**): 050201. DOI: 10.1088/1674-1056/ae1458
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Modified PINN approach integrating conservation laws for efficient multi-stage training of coupled nonlinear systems

Jie Deng(邓婕) and Lijia Han(韩励佳)
Chin. Phys. B, 2026, 35 (**4**): 040202. DOI: 10.1088/1674-1056/ae030b

LT-PINNs: Physics-informed neural networks based on Laplace transform for solving Caputo-type fractional partial differential equations

Ruibo Zhang(张瑞波), Fengjun Li(李风军), and Jianqiang Liu(刘建强)
Chin. Phys. B, 2026, 35 (**3**): 030201. DOI: 10.1088/1674-1056/adf827

Mutation detection and fast identification of switching system based on data-driven method

Zhonghua Zhang(张钟化), Wei Xu(徐伟), and Yi Song(宋怡)
Chin. Phys. B, 2023, 32 (**5**): 050201. DOI: 10.1088/1674-1056/acb9f5

Sparse identification method of extracting hybrid energy harvesting system from observed data

Ya-Hui Sun(孙亚辉), Yuan-Hui Zeng(曾远辉), and Yong-Ge Yang(杨勇歌)
Chin. Phys. B, 2022, 31 (**12**): 120203. DOI: 10.1088/1674-1056/ac9cbf

Data-driven parity-time-symmetric vector rogue wave solutions of multi-component nonlinear Schrödinger equation

Li-Jun Chang(常莉君), Yi-Fan Mo(莫一凡), Li-Ming Ling(凌黎明), and De-Lu Zeng(曾德炉)
Chin. Phys. B, 2022, 31 (**6**): 060201. DOI: 10.1088/1674-1056/ac4cc5

A residual-based adaptive refinement physics-informed neural networks (RAR-PINNs) method for fifth-order KdV equation

Shi-Fang Tian(田十方), Ya-Xuan Yu(于亚璇), and Biao Li(李彪)†

School of Mathematics and Statistics, Ningbo University, Ningbo 315211, *China*

(Received 17 September 2025; revised manuscript received 15 October 2025; accepted manuscript online 17 October 2025)

In the field of nonlinear partial differential equations (PDEs), the fifth-order Korteweg–De Vries (KdV) equation serves as a fundamental model with significant physical implications, extending the classical KdV framework through the incorporation of high-order spatial derivatives to capture strong dispersion effects. However, the inherent nonlinearity and complexity of this PDE present substantial challenges for obtaining accurate numerical solutions. To address these issues, this paper proposes a residual-based adaptive refinement physics-informed neural networks (RAR-PINNs) method. This approach synergizes the nonlinear approximation capability of PINNs with a residual-driven adaptive sampling strategy. By dynamically redistributing training points according to the magnitude of the PDE residuals, RAR-PINNs effectively concentrate computational resources on "critical regions", such as soliton peaks and high-gradient zones, where errors are predominant. Furthermore, we construct a composite physics-informed loss function that incorporates initial and boundary conditions, PDE residuals, and, in an enhanced variant, energy conservation laws, to further improve solution fidelity. Numerical experiments on two variants of the fifth-order KdV equation demonstrate that RAR-PINNs significantly outperform conventional PINNs in terms of both accuracy (reducing relative errors by one to two orders of magnitude) and computational efficiency. The conservation-law-enhanced version of the model yields even higher precision, underscoring the efficacy and robustness of the proposed method. This study establishes a powerful deep learning framework for tackling complex PDEs with sharp or singular solution structures.

Keywords: data-driven, PINNs, residual-based adaptive refinement, fifth-order KdV equation

PACS: 02.30.Ik, 02.30.Jr, 05.45.–a

DOI: 10.1088/1674-1056/ae1458 **CSTR:** 32038.14.CPB.ae1458

1. Introduction

In the field of nonlinear partial differential equations (PDEs), the fifth-order Korteweg–De Vries (KdV) equation,[1–4] as a significant class of PDEs, has garnered widespread attention in recent years. Serving as a higher-order extension of the classical KdV equation, it embodies rich physical background and implications. While the classical KdV equation primarily models weakly nonlinear and weakly dispersive long-wave propagation in shallow water, the fifth-order KdV equation incorporates higher-order spatial derivatives, enabling more accurate descriptions of physical processes with strong dispersion effects. It also maintains a close relationship with the classical KdV equation in one spatial and one temporal dimension.

A general form of the fifth-order KdV equation can be written as

$$u_t + \beta u u_{xxx} + \gamma u_{xxxxx} = \frac{\partial}{\partial x} f(u, u_x, u_{xx}), \tag{1}$$

where f is a smooth function, and α, β, γ are real constants with $\gamma \neq 0$. Besides this broad formulation, several specific forms are also commonly studied, such as

$$u_t + \alpha u^2 u_x + \beta u_x u_{xx} + \beta u u_{xxx} + \gamma u_{xxxxx} = 0,$$

which more directly capture particular physical phenomena.

Unlike integrable KdV systems, the fifth-order KdV equation supports a variety of soliton structures. Various methods have been employed to investigate its solution properties. For instance, the variational approach has been used to establish the existence of ground state traveling wave solutions and to analyze their stability.[5,6] By applying Wu Wenjun's characteristic set method[7] in conjunction with computer algebra systems, solitary and soliton solutions for a class of generalized fifth-order KdV equations have been derived, leading to refinements and improvements of prior results. Nevertheless, the inherent nonlinearity and complexity of the equation continue to pose substantial challenges for obtaining accurate solutions.

Recent advances in deep learning have opened new pathways for addressing PDEs. The physics-informed neural networks (PINNs) framework, introduced by Raissi *et al.*, integrates the expressive power of neural networks with physical constraints derived from conservation laws. By incorporating equation residuals, along with initial and boundary conditions, into the loss function, PINNs ensure that the solution conforms to the underlying physics during training. This approach allows the network to learn solution functions directly without reliance on spatial discretization. Owing to their in-

†Corresponding author. E-mail: libiao@nbu.edu.cn

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

herent flexibility in handling high-dimensional problems and complex boundaries, PINNs[8] have been successfully applied to classical systems such as the KdV equation and the nonlinear Schrödinger equation.

However, when addressing equations involving nonlinear or multi-order derivative coupling, physics-informed neural networks (PINNs) often encounter performance bottlenecks due to suboptimal sampling point distributions. Uniform or random sampling strategies frequently fail to concentrate on "critical regions" exhibiting large residuals, such as shock wave fronts or solitary wave peaks, leading to inefficient allocation of computational resources and insufficient solution accuracy.

To mitigate this limitation, several enhanced strategies have been proposed. For instance, Jagtap *et al.*[9,10] applied an improved PINNs model to simulate an inverse problem in supersonic flow. Chen and collaborators introduced a modified PINNs framework to obtain data-driven localized wave solutions for the N-double pole solutions of the nonlocal Hirota equation under nonzero boundary conditions,[11] and further simulated vector localized waves in the Manakov system[12] using upgraded PINNs. Yan *et al.*[13–15] developed an enhanced PINN approach by refining the loss function and incorporating first-order derivative boundary constraint terms, successfully predicting peakon and periodic solutions of Camassa–Holm-like equations. Wen *et al.*[16] developed an extended physics-informed neural networks with twin subnets (EPINN-TS) to predict seven types of vector solitons and discover model parameters for the coupled mixed derivative nonlinear Schrödinger equation. Guo *et al.*[17] proposed a temporal causality-based adaptive sampling method (TCAS-PINN), which effectively enhanced the model's capability to solve time-evolution problems. To further improve accuracy, Zhou *et al.*[18] developed a multi-distributed sampling strategy and successfully applied it to obtain exact solutions such as optical solitons.

Building upon standard PINNs, Li's team proposed gradient-optimized PINNs,[19,20] mixed-training PINNs, and prior-information hybrid training PINNs,[21,22] all contributing to improved accuracy of learned solutions. The physically-informed adaptive sampling approach (exemplified by PIP-INNs) constructs hybrid training domains by enriching sampling points within steep regions of the solution inside initial-boundary (IB) condition areas. This enables more comprehensive learning in high-gradient zones. Meanwhile, integrating physical constraints, such as energy conservation laws, into the loss function further elevates solution precision. These methods draw inspiration from adaptive mesh refinement techniques, effectively strengthening PINNs' capacity to capture complex solution structures through targeted computational resource allocation.

The residual-based adaptive refinement (RAR) method[23] emerges as a solution to these issues. Its core idea is to dynamically adjust sampling point distribution during training based on PDE residual magnitudes: after each iteration, new sampling points are added in regions with large residuals, enabling the model to capture critical local details more effectively and improving solution accuracy and convergence speed. Compared to traditional fixed sampling, RAR provides a more flexible and efficient data-driven approach for PINNs. Xu *at el.*[24] proposes a parallel hard-constraint physics-informed neural network (phPINN) that effectively addresses both forward and inverse problems of the nonlinear Schrödinger–Maxwell–Bloch (NLS–MB) equation by integrating a parallel fully-connected neural network (PFNN) architecture with the RAR method.

Building on this background, this paper focuses on solving the fifth-order KdV equation and proposes a PINNs method incorporating the RAR sampling strategy (RAR-PINNs). The innovation of RAR-PINNs in the following aspects: The original model[23] proposed a general RAR framework, which primarily offers a universal structure applicable to various types of PDEs, without specific optimization for particular high-order PDEs. In contrast, RAR-PINNs constructs a physics-informed loss function that incorporates high-order derivative constraints, specifically addressing higher-order terms such as u_{xxxxx} in the fifth-order KdV equation. Furthermore, it introduces the energy conservation law as an additional constraint into the mean squared error (MSE).

The structure of this paper is as follows: (i) a detailed introduction to the physical background and mathematical model of the fifth-order KdV equation; (ii) the proposal of the RAR-PINNs implementation framework, including the specific workflow of the RAR adaptive sampling strategy and its integration with PINNs; (iii) validation of the proposed method's accuracy and ability to simulate complex solution structures through numerical experiments, with comparison to traditional PINNs; (iv) summary of research conclusions and prospects for the method's application in broader nonlinear evolution equations.

2. Establishing RAR-PINNs model

Residual-based adaptive refinement (RAR) is an adaptive sampling strategy proposed to address the accuracy deficiency of physics-informed neural networks (PINNs) in solving PDEs. Its core objective is to dynamically adjust the distribution of training points, allowing the neural network to allocate more computational resources to "critical regions" of the solution (such as regions with sharp gradients or large residuals), thereby improving solution accuracy while maintaining efficiency. The following describes the RAR algorithm for the model (see Algorithm 1).

Algorithm 1: Training algorithm for residual-based adaptive refinement physics-informed neural networks (RAR-PINNs)

Sampling domain $\Omega \subset \mathbb{R}^d$, Initial/boundary condition sets X_{ib} (merged X_{ic} and X_{bc}), residual threshold tol = 0.005, convergence threshold δ, maximum iteration count maxit

Initialization Generate initial sampling points for initial/boundary conditions: X_{ib} generate initial sampling points for PDE residuals: X_{pde} randomly initialize network parameters θ (*e.g.*, via Xavier method) set current iteration count $k = 0$

Iterative training while $k < maxit$ **do**

For each $(x,t) \in X_{\mathrm{pde}}$, compute PDE residual $r(x,t;\theta)$:

$$r(x,t;\theta) = u_t + \beta u u_{xxx} + \gamma u_{xxxxx} - \frac{\partial}{\partial x} f(u, u_x, u_{xx})$$

Compute maximum residual over Ω: max $_res = \max_{x\in\Omega} \|R(x)\|_2$, where $R(x)$ is the residual field ($T \times X \in [1.0, 2.0] \times [1, 25]$)

if max $_res < tol$ **then**

Set update flag: need_update = false (no RAR refinement) **else**

Set update flag: need_update = true (perform RAR refinement)

if *need_update* = *true* **then**

Screen high-residual subregions within Ω (based on $R(x)$) generate new sampling points in high-residual regions: X_{new} update residual sampling set: $X_{\mathrm{pde}} = X_{\mathrm{pde}} \cup X_{\mathrm{new}}$

Compute mix-training region loss L_{ib} (merged L_{ic} and L_{bc}):

$$L_{\mathrm{ib}} = \frac{1}{|X_{\mathrm{ic}}|} \sum_{(x_{\mathrm{i}},t_0)\in X_{\mathrm{ic}}} (\hat{u}(x_{\mathrm{i}},t_0;\theta) - h(x_{\mathrm{i}}))^2 + \frac{1}{|X_{\mathrm{bc}}|} \sum_{(x_{\mathrm{b}},t_{\mathrm{b}})\in X_{\mathrm{bc}}} (\hat{u}(x_{\mathrm{b}},t_{\mathrm{b}};\theta) - g(x_{\mathrm{b}},t_{\mathrm{b}}))^2$$

Compute PDE residual loss L_{pde}:

$$L_{\mathrm{pde}} = \frac{1}{|X_{\mathrm{pde}}|} \sum_{(x,t)\in X_{\mathrm{pde}}} (r(x,t;\theta))^2$$

Compute total loss: $L_{\mathrm{total}} = L_{\mathrm{ib}} + L_{\mathrm{pde}}$

Update θ via gradient descent (*e.g.*, Adam optimizer) to minimize L_{total} increment teration count: $k = k + 1$

if $L_{\mathrm{total}} < \delta$ or $k = maxit$ **then**

Break the iterative loop

Output trained RAR-PINNs model: $\hat{u}(x,t;\theta)$

2.1. Core idea of RAR method

The accuracy of PINNs highly depends on the distribution of training points: if the solution changes dramatically in local regions (such as near shock waves, boundary layers, or singular points), uniformly sampled training points often fail to capture these details, causing the PDE residual (deviation between predicted and true solutions) to increase significantly in these regions. The RAR method adaptively adds sampling points in regions with large residuals by monitoring the spatial distribution of PDE residuals in real-time, enabling the neural network to focus more on complex features of the solution, ultimately achieving the effect of "on-demand allocation" of computational resources.

2.1.1. Neural network architecture

Using multilayer perceptron (MLP) as the base model, with input being spatiotemporal coordinates ((x,t) ,and output being the predicted value of physical quantity $\hat{u}(x,t;\theta)$). The network structure is:

(i) Input layer: 2 neurons (corresponding to x, t);

(ii) Hidden layers: L layers, each containing N neurons, activation function using tanh or ReLU (enhancing nonlinear fitting capability);

(iii) Output layer: 1 neuron (corresponding to $\hat{u}$).

2.1.2. Loss function construction

The total loss function of PINNs consists of boundary condition loss, initial condition loss, and PDE residual loss, where the PDE residual loss is directly related to the residual calculation of RAR method:

(I) Boundary condition loss L_{bc}

Measures deviation between predicted values and boundary conditions:

$$L_{\mathrm{bc}} = \frac{1}{|X_{\mathrm{bc}}|} \sum_{(x_{\mathrm{b}},t_{\mathrm{b}})\in X_{\mathrm{bc}}} \left(\hat{u}\left(x_{\mathrm{b}},t_{\mathrm{b}};\theta\right) - g\left(x_{\mathrm{b}},t_{\mathrm{b}}\right)\right)^2,$$

where $|X_{\mathrm{bc}}|$ is the number of boundary points, $g(x_{\mathrm{b}},t_{\mathrm{b}})$ is the known boundary value.

(II) Initial condition loss L_{ic}

Measures deviation between predicted values and initial conditions:

$$L_{\mathrm{ic}} = \frac{1}{|X_{\mathrm{ic}}|} \sum_{(x_{\mathrm{i}},t_0)\in X_{\mathrm{ic}}} \left(\hat{u}\left(x_{\mathrm{i}},t_0;\theta\right) - h\left(x_{\mathrm{i}}\right)\right)^2,$$

where $|X_{\mathrm{ic}}|$ is the number of initial points, $h(x_{\mathrm{i}})$ is the known initial value.

(III) Mix-training region (initial condition and boundary condition) loss L_{ib}

$$L_{\text{ib}} = \frac{1}{|X_{\text{ic}}|}\sum_{(x_{\text{i}},t_0)\in X_{\text{ic}}}(\hat{u}(x_{\text{i}},t_0;\theta)-h(x_{\text{i}}))^2 + \frac{1}{|X_{\text{bc}}|}\sum_{(x_{\text{b}},t_{\text{b}})\in X_{\text{bc}}}(\hat{u}(x_{\text{b}},t_{\text{b}};\theta)-g(x_{\text{b}},t_{\text{b}}))^2.$$

(IV) PDE residual loss L_{pde}

Measures how well the predicted solution satisfies the PDE, calculated based on the dynamically updated residual points X_{pde} by RAR:

$$L_{\text{pde}} = \frac{1}{|X_{\text{pde}}|}\sum_{(x,t)\in X_{\text{pde}}}(r(x,t;\theta))^2,$$

where $r(x,t;\theta)$ is the PDE residual defined in the RAR method.

For the specific fifth-order KdV equation, the residual is expressed as

$$r(x,t;\theta) = u_t + \beta u u_{xxx} + \gamma u_{xxxxx} - \frac{\partial}{\partial x}f(u,u_x,u_{xx}).$$

(V) Total loss function

$$L_{\text{total}} = L_{\text{ib}} + L_{\text{pde}}.$$

2.1.3. RAR-PINNs training process

The entire RAR-PINNs training process should be divided into two stages: preliminary experiments with PINNs and the RAR-based adaptive sampling point update, with the specific procedure as shown in (Fig. 1).

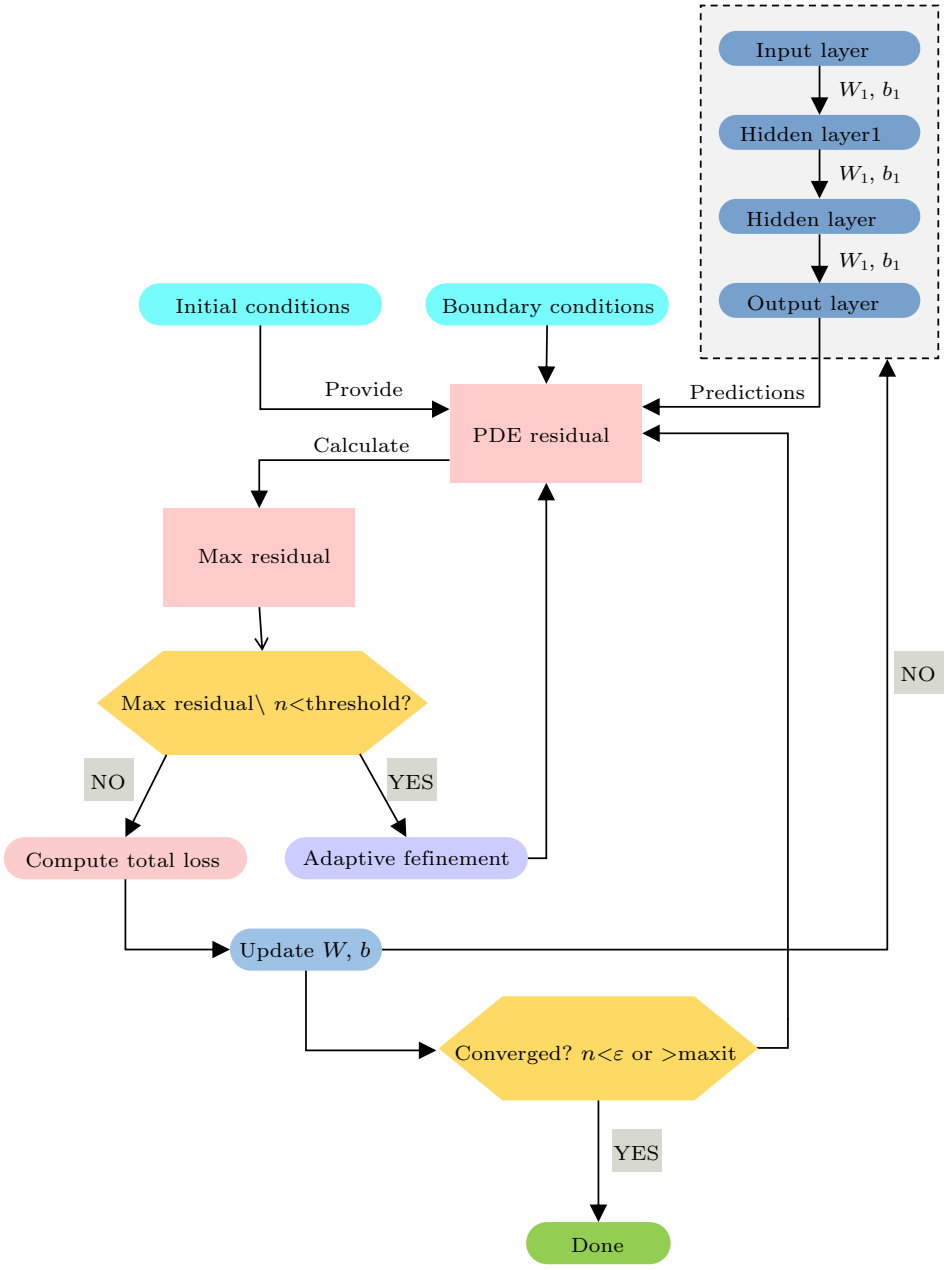


Fig. 1. New residual-based adaptive refinement physics-informed neural networks (RAR-PINNs) model.

2.2. Conservation law

To enhance the accuracy of numerical results simulated by RAR-PINNs (see Algorithm 1), we add constraints to the loss function $Loss_{\text{total}}$ by introducing the local conservation law.

In this work, the local conservation law is formulated as

$$u_t + \left[\left(\frac{\alpha}{3}u^3 + \beta u u_{xx} + \gamma u_{xxxx}\right)\right]_x = 0.$$

We define the conservation residual $c(x,t,\theta)$ as

$$c(x,t,\theta) = u_t + \left[\left(\frac{\alpha}{3}u^3 + \beta u u_{xx} + \gamma u_{xxxx}\right)\right]_x$$

At this point, the conservation law error $Loss_{\text{c}}$ is as follows:

$$Loss_{\text{c}} = \frac{1}{|X_{\text{c}}|}\sum_{(x_{\text{c}},t_v)\in X_{\text{c}}}(c(x,t,\theta))^2,$$

where $X_{\text{c}} = X_{\text{pde}}$.

After incorporating the conservation law, the new loss function $Loss_{\text{total}}$ becomes:

$$\begin{aligned} Loss_{\text{total}} &= Loss_{\text{ib}} + Loss_{\text{c}} + Loss_{\text{pde}} \\ &= \frac{1}{|X_{\text{ic}}|}\sum_{(x_{\text{i}},t_0)\in X_{\text{ic}}}(\hat{u}(x_{\text{i}},t_0;\theta) - h(x_{\text{i}}))^2 \\ &\quad + \frac{1}{|X_{\text{bc}}|}\sum_{(x_{\text{b}},t_{\text{b}})\in X_{\text{bc}}}(\hat{u}(x_{\text{b}},t_{\text{b}};\theta) - g(x_{\text{b}},t_{\text{b}}))^2 \\ &\quad + \frac{1}{|X_{\text{c}}|}\sum_{(x_{\text{c}},t_v)\in X_{\text{c}}}(c(x,t,\theta))^2 \\ &\quad + \frac{1}{|X_{\text{pde}}|}\sum_{(x,t)\in X_{\text{pde}}}(r(x,t;\theta))^2. \end{aligned}$$

3. Numerical experiment

We employ deep neural networks with 4 to 6 hidden layers and 32 to 64 neurons per layer, using the hyperbolic tangent (tanh) as the activation function. The training process consists of the following stages: initially, the Adam optimizer (with default learning rate) is applied for 10^4 iterations, followed by the L-BFGS optimizer for another 10^4 iterations. After updating the sampling regions using the RAR adaptive method, a final fine-tuning is performed with the Adam optimizer (default learning rate) for an additional 2×10^4 iterations. Network parameters are initialized via the Xavier method. The sample sizes are configured as follows: initial PDE residual sampling points $N_{\text{f}} = 3000$, RAR-updated PDE residual sampling points $N_{\text{f}}^* = 1.8\times10^4$, and Dirichlet boundary data points $N_{\text{IB}} = 200$.

The relative error (relative L_2 norm) is defined as:

$$\text{Relative error} = \frac{\|\sqrt{\text{error}_i}\|_2}{\|u(x_{\text{i}},t)\|_2} = \frac{\sqrt{\sum_{i=1}^{N}[\hat{u}_i - u(x_{\text{i}},t)]^2}}{\sqrt{\sum_{i=1}^{N}[u(x_{\text{i}},t)]^2}},$$

where u_i is the predicted solution and $u(x_{\text{i}},t)$ is the exact solution.

3.1. RAR-PINNs

We consider the first type fifth-order KdV equation with Dirichlet boundary conditions ($r = 15$, $\beta = 15$, $\alpha = 45$)

$$\begin{cases} u_t + 45u^2u_x + 15uu_{xxx} + 15u_xu_{xx} + u_{xxxx} = 0, \\ u(x,t) = I(x,t), \quad x\in\Omega, \\ u(x,t) = \mathcal{B}(x,t), \quad (x,t)\in\partial\Omega\times T, \end{cases} \tag{2}$$

where $u(x,t)$ depends on space x and time t.

Firstly, we take the soliton solution of the first fifth-order KdV equation[25,26] as an example, when $T\in[1.0,2.0], X\in[1,25]$. The three-dimensional (3D) diagram (Fig. 2) of the soliton solution is described in the following

$$u = 6\cdot\frac{\partial^2\log(f)}{\partial x^2} + 1, \tag{3}$$

where

$$\begin{aligned} f &= 1 + \exp(\eta_1) + \exp(\eta_2) + \exp(\eta_3) \\ &\quad + a_{12}\exp(\eta_1)\exp(\eta_2) + a_{13}\exp(\eta_1)\exp(\eta_3) \\ &\quad + a_{23}\exp(\eta_2)\exp(\eta_3) \\ &\quad + a_{123}\exp(\eta_1)\exp(\eta_2)\exp(\eta_3), \\ \eta_i(t) &= -\int\left(p_i^3q(t)r(t) + \frac{6p_iq(t)^2r(t)}{C_0} + \frac{C_0r(t)}{30}\right)\mathrm{d}t \\ &\quad (i = 1,2,3), \\ a_{12} &= \frac{(p_1-p_2)^2(18q(t)+C_0(p_1^2-p_1p_2+p_2^2))}{(p_1+p_2)^2(18q(t)+C_0(p_1^2+p_1p_2+p_2^2))}, \\ a_{13} &= \frac{(p_1-p_3)^2(18q(t)+C_0(p_1^2-p_1p_3+p_3^2))}{(p_1+p_3)^2(18q(t)+C_0(p_1^2+p_1p_3+p_3^2))}, \\ a_{23} &= \frac{(p_2-p_3)^2(18q(t)+C_0(p_2^2-p_2p_3+p_3^2))}{(p_2+p_3)^2(18q(t)+C_0(p_2^2+p_2p_3+p_3^2))}, \\ a_{123} &= a_{12}\cdot a_{13}\cdot a_{23}, \end{aligned}$$

where $C_0 = 6$, $C_1 = 1$, $q(t) = 1$, $p_1 = 1.04$, $p_2 = 1.005$, $p_3 = 1.34$, $\xi_1 = 1.05$, $\xi_2 = 2.09$, $\xi_3 = 3.21$.

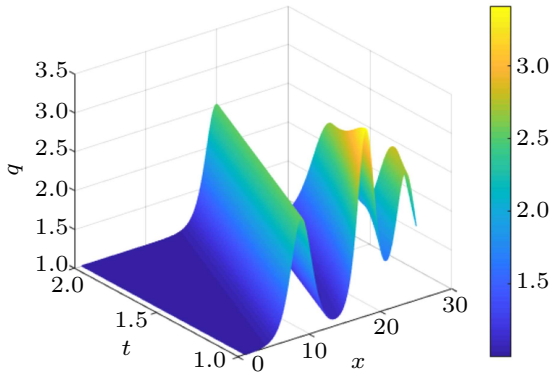


Fig. 2. Soliton solution of first type fifth-order KdV equation (2).

It is worth noting that we choose a neural networks with 6 hidden layers and 64 neurons in each layer, and add some training points in the sharp area for the new sampling area as the prior information.

We use the PINNs and RAR-PINNs, combined with the training conditions and optimization methods described above, to conduct numerical experiments.

From Table 1, we can obtain the experimental results of the prediction performance of PINNs and RAR-PINNs for the soliton solution under all neural networks conditions. Through Table 1, we find that RAR-PINNs are superior to PINNs when the number of neurons in the hidden layer and each layer is the same. RAR-PINNs not only have high accuracy, but also take less time. It is worth noting that in order to better and more clearly observe the performance of the model RAR-PINNs, we present the updated RAR sampling area and absolute error of the RAR-PINNs model in Fig. 3.

Table 1. Numerical results of solition solution of first type fifth-order KdV equation (2) simulated by PINNs and RAR-PINNs.

Neuron networks	PINNs		RAR-PINNs	
Neurons | Layers	$u(x,t)$	Time (s)	$u(x,t)$	Time (s)
32 | 4	4.3×10^{-1}	30500.1	7.1907×10^{-4}	23240.95
32 | 6	3.5×10^{-1}	35001.6	4.2481×10^{-4}	30052
64 | 4	4.5×10^{-1}	39023	4.4402×10^{-4}	33080
64 | 6	4.2×10^{-1}	44321	2.9335×10^{-4}	41268

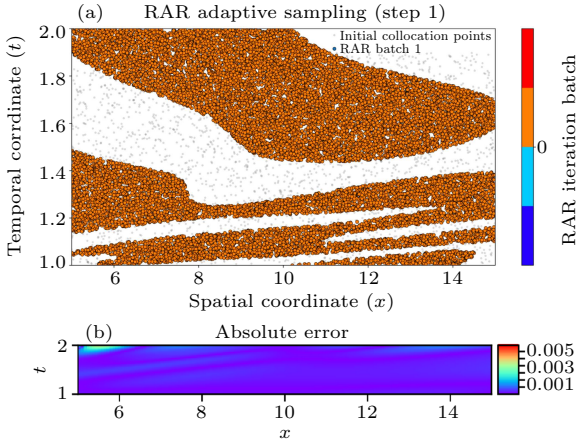


Fig. 3. (a) New sampling area, (b) absolute error of the soliton solution (3) simulated by RAR-PINNs (Relative error is 2.9335×10^{-4}).

According Fig. 3(a), we observe that after applying the RAR adaptive method, the sampling region, initially distributed uniformly across the entire domain, becomes concentrated primarily near points with larger errors from the preliminary training phase.

For the second type fifth-order KdV equation ($r=5$, $\beta=5$, $\alpha=5$, $T\in[1.0,2.0]$, $X\in[1,25]$):

$$\begin{cases} u_t+5u^2u_x+5uu_{xxx}+5u_xu_{xx}+u_{xxxxx}=0, \\ u(x,t)=\mathcal{I}(x,t), \quad x\in\Omega, \\ u(x,t)=\mathcal{B}(x,t), \quad (x,t)\in\partial\Omega\times T, \end{cases} \tag{4}$$

where $u(x,t)$ contains variables x and t, which are space and time variables, respectively.

The specific format and three-dimensional diagram (Fig. 4) of the soliton solution[25,26] are described in the following:

$$u=6\cdot\frac{\partial^2\log(f)}{\partial x^2}+1, \tag{5}$$

where

$$\begin{aligned} f &= 1+\exp(\eta_1)+\exp(\eta_2)+\exp(\eta_3) \\ &\quad +a_{12}\exp(\eta_1)\exp(\eta_2)+a_{13}\exp(\eta_1)\exp(\eta_3) \\ &\quad +a_{23}\exp(\eta_2)\exp(\eta_3) \\ &\quad +a_{123}\exp(\eta_1)\exp(\eta_2)\exp(\eta_3), \end{aligned} \tag{6}$$

$$\eta_i(t) = -\int\left(p_i^3q(t)r(t)+\frac{6p_iq(t)^2r(t)}{C_0}+\frac{C_0r(t)}{30}\right)\mathrm{d}t \quad (i=1,2,3), \tag{7}$$

$$a_{12}=\frac{(p_1-p_2)^2(18q(t)+C_0(p_1^2-p_1p_2+p_2^2)}{(p_1+p_2)^2(18q(t)+C_0(p_1^2+p_1p_2+p_2^2)},$$

$$a_{13}=\frac{(p_1-p_3)^2(18q(t)+C_0(p_1^2-p_1p_3+p_3^2)}{(p_1+p_3)^2(18q(t)+C_0(p_1^2+p_1p_3+p_3^2)},$$

$$a_{23}=\frac{(p_2-p_3)^2(18q(t)+C_0(p_2^2-p_2p_3+p_3^2)}{(p_2+p_3)^2(18q(t)+C_0(p_2^2+p_2p_3+p_3^2))},$$

$$a_{123}=a_{12}\cdot a_{13}\cdot a_{23},$$

where $C_0=2$, $C_1=1$, $q(t)=1$, $p_1=5$, $p_2=0.5$, $p_3=3.3$, $\xi_1=11.05$, $\xi_2=21.09$, $\xi_3=3.21$.

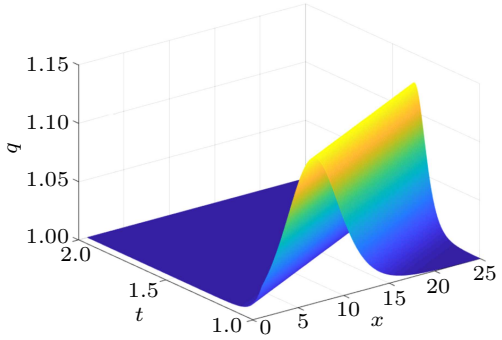


Fig. 4. Soliton solution of second type fifth-order KdV equation (4).

Table 2. Numerical results of solition solution of second type fifth-order KdV equation (4) simulated by PINNs and RAR-PINNs.

Neuron networks	PINNs		RAR-PINNs	
Neurons | Layers	$u(x,t)$	Time (s)	$u(x,t)$	Time (s)
32 | 4	1.42×10^{-1}	7001	2.37×10^{-3}	6106
32 | 6	3.45×10^{-1}	9404.2	1.24×10^{-3}	8787.2
64 | 4	2.34×10^{-1}	12034.4	2.14×10^{-3}	9945
64 | 6	6.34×10^{-1}	17345.5	9.2684×10^{-4}	15853

We follow the same experimental method mentioned above. The experimental results for the predictive performance of RAR-PINNs for soliton solution of second type fifth-order KdV equation under all neural network conditions are listed in Table 2. From Table 2, we also find that RAR-PINNs are still more accurate than PINNs. It is worth noting that in

order to better and more clearly observe the performance of the model RAR-PINNs, we present the updated RAR sampling area and absolute error of the RAR-PINNs model in Fig. 5.

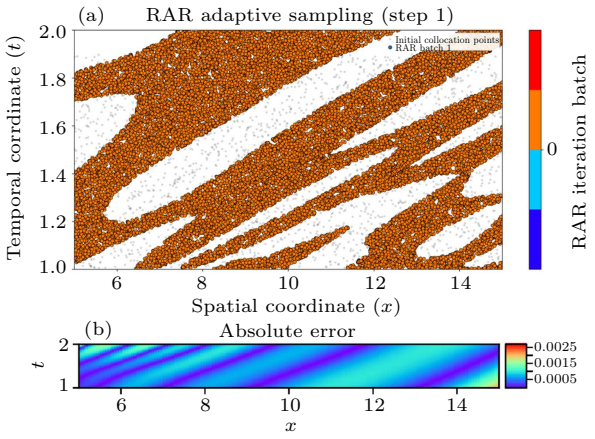


Fig. 5. (a) New sampling area, (b) absolute error of the soliton solution (5) simulated by RAR-PINNs (Relative error is 9.2684×10^{-3}).

According to the numerical results, we find that the performance of the RAR-PINNs model is still very good and the accuracy is improved again. In terms of time consumption, it is still better than that of PINNs. In conclusion, when simulating soliton solutions of these fifth-order KdV equations, the accuracy and performance of RAR-PINNs are better than PINNs.

3.2. New RAR-PINNs

We use PIPINNs with two conservation laws (where $\gamma=1$, $\alpha=45$, $\beta=15$; $\gamma=1$, $\alpha=5$, $\beta=5$) of energy to carry out numerical experiments on soliton solutions of two type fifth-order KdV equations, respectively. At this point, the conservation laws of the two fifth-order KdV equations (2) and (4) are as follows:

$$c_1 = u_t + \left[\left(\frac{45}{3}u^3 + 15uu_{xx} + u_{xxxx}\right)\right]_x, \tag{8}$$

$$c_2 = u_t + \left[\left(\frac{5}{3}u^3 + 5uu_{xx} + u_{xxxx}\right)\right]_x. \tag{9}$$

Through numerical experiments, we present the results in the following. To facilitate readers in observing the performance of the new RAR-PINNs, we still present the absolute error (Figs. 6 and 8) from the numerical experiments on the two fifth-order KdV equations and corresponding RAR sampling region (Figs. 7 and 9).

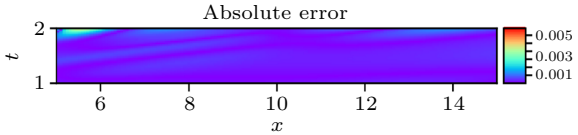


Fig. 6. Absolute error of soliton solution (3) of the first type fifth-order KdV equation (2).

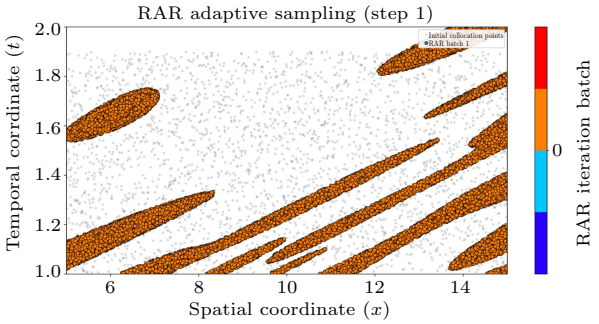


Fig. 7. RAR new sampling region.

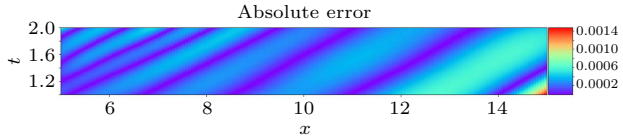


Fig. 8. Absolute error of soliton solution (5) of the second type fifth-order KdV equation (4).

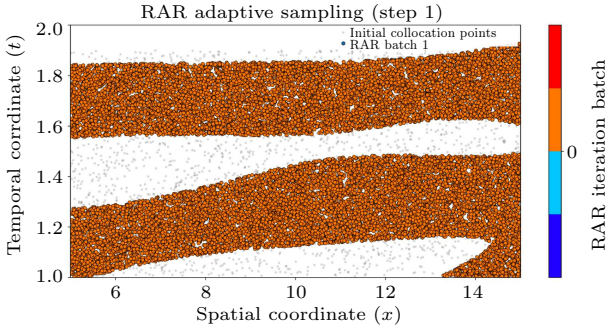


Fig. 9. RAR new sampling region.

From Tables 3 and 4, it is evident that when simulating soliton solutions of two-type fifth-order KdV equations, the RAR-PINNs with conservation laws (8) and (9) are more accurate than the original RAR-PINNs, and the new RAR-PINNs can obtain better accuracy over a slightly longer time than the original RAR-PINNs, which is very meaningful.

Table 3. Numerical results of soliton solution of the first type fifth-order KdV equation (2) simulated by the new RAR-PINNs.

Neuron networks	RAR-PINNs		New RAR-PINNs	
Neurons | Layers	$u(x,t)$	Time (s)	$u(x,t)$	Time (s)
64 | 6	2.9335×10^{-4}	41268	1.5378×10^{-4}	62342

Table 4. Numerical results of soliton solution of the second type fifth-order KdV equation (4) simulated by the new RAR-PINNs.

Neuron networks	RAR-PINNs		New RAR-PINNs	
Neurons | Layers	$u(x,t)$	Time (s)	$u(x,t)$	Time (s)
64 | 6	9.2684×10^{-4}	15853	3.4086×10^{-4}	19738

4. Discussion and conclusion

In this study, we introduced the RAR-PINNs framework, which integrates residual-based adaptive refinement with PINNs to effectively solve the fifth-order KdV equation. By dynamically concentrating sampling points in regions

with large PDE residuals, such as soliton peaks and high-solution-gradient areas, the proposed method overcomes key limitations of conventional PINNs related to inefficient sampling and poor error localization. Numerical experiments conducted on two types of fifth-order KdV equations confirm that RAR-PINNs achieve significantly higher accuracy than standard PINNs, with relative errors reduced by one to two orders of magnitude, while also reducing computational time in most scenarios.

Furthermore, the enhanced version of our model, which incorporates local conservation laws into the loss function, demonstrated additional improvements in solution accuracy. This confirms that embedding physical constraints not only enriches the learning paradigm but also endows the model with better generalization capability and mathematical consistency. The ability of RAR-PINNs to accurately capture complex soliton dynamics, even under varying network architectures and hyperparameters, highlights its robustness and adaptability.

In conclusion, RAR-PINNs represent a viable and efficient deep learning-based approach for solving challenging nonlinear evolution equations. By adaptively refining training sets based on residual feedback and incorporating physical priors, this framework offers a flexible and accurate alternative to traditional numerical methods, opening new avenues for computational modeling in applied mathematics and theoretical physics.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12175111 and 12235007), and the K. C. Wong Magna Fund in Ningbo University.

References

[1] Korteweg D J and de Vries G 1895 *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **39** 422
[2] Wang M L and Li M 2025 *Acta Phys. Sin.* **74** 150201 (in Chinese)
[3] Wazwaz A M 2016 *Journal of Taibah University for Science* **10** 469
[4] Zhang F, Hu Y R, Xin X P and Liu H Z 2022 *Computational and Applied Mathematics* **41** 139
[5] Liu N and Guo B 2021 *Anal. Appl.* **19** 575
[6] Niu Z J and Wang Z G 2021 *Journal of Applied Analysis & Computation* **11** 2866
[7] Wu W J 1978 *Scientia Sinica* **21** 159
[8] Raissi M, Perdikaris P and Karniadakis G E 2019 *J. Comput. Phys.* **378** 686
[9] Jagtap A D, Mao Z P, Adams N and Karniadakis G E 2022 *J. Comput. Phys.* **466** 111402
[10] Pang G F, Lu L and Karniadakis G E 2022 *SIAM J. Sci. Comput.* **41** 2603
[11] Peng W Q and Chen Y 2022 *Physica D* **435** 133274
[12] Pu J C and Chen Y 2022 *Chaos, Solitons and Fractals* **160** 112182
[13] Wang L and Yan Z Y 2021 *Phys. Lett. A* **404** 127408
[14] Wang L and Yan Z Y 2022 *Phys. Lett. A* **450** 128373
[15] Fang Y, Wu G Z, Wang Y Y and Dai C Q 2021 *Nonlinear Dyn.* **105** 603
[16] Wen X K, Jiang J H, Liu W and Dai C Q 2023 *Nonlinear Dyn.* **111** 13343
[17] Guo J, Wang H F, Gu S L and Hou C P 2024 *Chin. Phys. B* **33** 050701
[18] Zhou H S, Zhang Z Y, Liu M W, Qi F H and Liu W J 2025 *Chin. Phys. Lett.* **42** 070001
[19] Li J H and Li B 2022 *Nonlinear Dyn.* **107** 781
[20] Tian S F and Li B 2023 *Acta Phys. Sin.* **72** 100202 (in Chinese)
[21] Li J H and Li B 2022 *Chaos, Solitons and Fractals* **164** 112712
[22] Tian S F, Zhang Z and Li B 2024 *Chin. Phys. Lett.* **41** 030201
[23] Lu L, Meng X M, Mao Z P and Karniadakis G E 2021 *SIAM Review* **63** 10
[24] Xu S Y, Zhou Q and Liu W 2023 *Nonlinear Dyn.* **111** 18401
[25] Sun F W and Wang M 2013 *Mathematics in Practice and Theory* **43** 256
[26] Li W T, Chen Y P, Zhang Z, Yang X Y and Li B 2025 *Chin. Phys. Lett.* **42** 010202