Electronic Research Archive

ERA, 32(4): 2699–2727.
DOI: 10.3934/era.2024122
Received: 30 January 2024
Revised: 11 March 2024
Accepted: 18 March 2024
Published: 01 April 2024

https://www.aimspress.com/journal/ERA

Research article

Adaptive fractional physical information neural network based on PQI scheme for solving time-fractional partial differential equations

Ziqing Yang[1], Ruiping Niu[1], Miaomiao Chen[2,3], Hongen Jia[1,4,*] and Shengli Li[4,5]

[1] College of Mathematics, Taiyuan University of Technology, Taiyuan, China
[2] College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, Taiyuan, China
[3] Department of Mathematics, Jinzhong University, Jinzhong, China
[4] Shanxi Key Laboratory for Intelligent Optimization Computing and Blockchain Technology, Taiyuan, China
[5] College of Mathematics and Statistics, Taiyuan Normal University, Jinzhong, China

* **Correspondence:** Email: jiahongen@aliyun.com; Tel: +8618635179159.

Abstract: In this paper, an accurate fractional physical information neural network with an adaptive learning rate (adaptive-fPINN-PQI) was first proposed for solving fractional partial differential equations. First, piecewise quadratic interpolation (PQI) in the sense of the Hadamard finite-part integral was introduced in the neural network to discretize the time-fractional derivative in the Caputo sense. Second, the adaptive learning rate residual network was constructed to keep the network from being stuck in the locally optimal solution, which automatically adjusts the weights of different loss terms, significantly balancing their gradients. Additionally, different from the traditional physical information neural networks, this neural network employs a new composite activation function based on the principle of Fourier transform instead of a single activation function, which significantly enhances the network's accuracy. Finally, numerous time-fractional diffusion and time-fractional phase-field equations were solved using the proposed adaptive-fPINN-PQI to demonstrate its high precision and efficiency.

Keywords: adaptive learning rate; time-fractional partial differential equations; physical information neural network; composite activation function

1. Introduction

In recent years, the search for high-precision numerical solutions of differential equations has been the focus of scholars' research [1–3]. The non-locality and memory of fractional calculus make it better to describe complex practical problems; so, many scholars have done a lot of research on the problem and applied it to various practical mathematical physics problems [4–9]. Since obtaining analytical solutions of fractional differential equations is a challenging task, numerical methods for solving fractional differential equations have drawn extensive attention. At present, a series of effective numerical methods have been developed for solving fractional differential equations, such as the finite difference method [10,11], the finite element method [12,13], the spectral method [14,15], and the meshfree methods (MFMs) [20–22]. In [16], it was introduced the finite difference method to deal with time and space problems. A first-order finite difference (L1) of the time discretization scheme was presented by Langlands and Henry [17]. Lin et al. [18] introduced an improved numerical method that uses a finite difference scheme for time discretization and the Legendre spectral method for space discretization. Deng [19] studied the finite element method (FEM) of fractional Fokker–Planck equation. In recent years, MFMs have become increasingly popular in solving practical engineering problems by reducing the dependence on grids [20–22]. Recently, Li et al. [23] proposed a new meshless numerical method called the finite integral method (FIM) to approximate one-dimensional problems. The FIM of Hadamard fractional derivative has received extensive attention [24]. Li et al. [25] proved that the FIM has high stability, efficiency, and accuracy when solving multi-dimensional problems.

Deep learning has grown in popularity as a numerical technique for solving integer-order partial differential equations. Numerous models based on data-driven deep neural networks (DNNs) have been proposed for partial differential equations [26–32], which only utilize information from observed data. Wang et al. [33] combined the residual neural network with the data-driven method to solve the high-dimensional problem efficiently. Yohai et al. [34] introduced a data-driven discretization scheme that uses neural networks to estimate spatial derivatives and perform end-to-end optimization of spatial derivatives. However, data-driven networks are unable to solve the issue with only physical constraints since they need a large amount of labeled data. Given the limitations of data-driven networks, one kind of neural network incorporating physical information has been developed. The networks have demonstrated powerful capabilities in solving partial differential equations and have received extensive attention [35–40]. At present, neural network technologies for solving classical partial differential equations are developing and showing promising results in practical applications [41–48]. Lin and Chen [49] proposed two physics-informed neural network (PINN) schemes based on the Miura transform. They introduced Miura transform constraints into neural networks to solve nonlinear partial differential equations and achieve unsupervised learning. Pu and Chen [50] obtained the data-driven vector local waves of the Manakov system with initial and boundary conditions by improving the output function and the number of physical constraints of the physical information neural network and using the local adaptive activation function of neurons and the slope recovery term. Because the automatic differentiation function used in the PINNs cannot be implemented for solving fractional-order problems, the construction of a stable and efficient neural network for solving fractional-order partial differential equations has become a hot research topic in deep learning. Fractional PINNs (fPINNs) [51] used the L1 scheme to discretize the time-fractional derivative. Rostami [52] implemented a series expansion as a substitute solution, enabling more accurate resolutions of high-order linear fractional partial differential equations (FPDEs). Wang et al. [53] combined fractional-order physical information

with the nonlinear function of neural networks to improve the modeling and solving effect of temporal fractional-order phase-field models.

The classical first-order finite difference (L1) scheme for discretizing the fractional derivative exhibited good stability and numerical accuracy. However, due to its low convergence order, it may require more grid nodes to obtain the same numerical precision as higher-order difference schemes. The piecewise quadratic interpolation (PQI) in the sense of the Hadamard finite partial integral proposed by Li et al. [54] has higher convergence, which can improve the precision of the predicted solutions and enhance the efficiency of the model.

In this paper, the fractional derivative of time is discretized using PQI rather than L1 as in fPINN. The higher-order discretization scheme causes the residual loss of the governing equation to decrease rapidly, resulting in a more accurate solution. In order to address the gradient imbalance of different loss terms in PINN and fPINN, the adaptive learning rate is proposed in the loss function, which can effectively keep the network from failing in the local optimal solution and significantly enhance its trainability, convergence, and accuracy. Furthermore, a new composite activation function is designed, greatly improving the precision of the destruction solution.

The rest of the paper is organized as follows. In Section 2, the precise format of the fractional differential equation is given, including the governing equation, initial condition, and boundary conditions. In Section 3, the techniques used to construct the proposed adaptive-fPINN-PQI framework are introduced in detail, including the formula of PQI discretization scheme, the principle and formula of the adaptive learning rate, the architecture of the network, and the components of the composite activation function. In Section 4, intensive numerical examples are conducted to demonstrate the excellent performance in accuracy and efficiency of the adaptive-fPINN-PQI. Some conclusions about the adaptive-fPINN-PQI are drawn in Section 5.

2. Time-fractional equation

In this paper, we consider the following form of time-fractional partial differential equation:

$$ {}_0^C D_t^\alpha u(\mathbb{X},t) - Lu(\mathbb{X},t) = f(\mathbb{X},t), \quad \mathbb{X} \in \Omega, t \in [0,\mathrm{T}], \tag{2.1} $$

where ${}_0^C D_t^\alpha u(\mathbb{X},t)$ is the fractional derivative of $u(\mathbb{X},t)$ with respect to time t under the Caputo definition, α is the order of the derivative, $\alpha \in [0,1]$. L represents a differential operator.

The initial condition can be written as

$$ u(\mathbb{X},0) = g_1(\mathbb{X}), \quad \mathbb{X} \in \Omega, \tag{2.2} $$

where $g_1(\mathbb{X})$ is a given function of the space.

The boundary condition can be expressed in detail as follows

$$ u(\mathbb{X},t) = g_2(\mathbb{X}), \quad \mathbb{X} \in \partial\Omega, t \in [0,\mathrm{T}], \tag{2.3} $$

where $\partial\Omega$ is the boundary of Ω and $g_2(\mathbb{X})$ is given for the well-posed time-fractional partial differential equations.

3. Methodology

Physics-informed neural networks (PINNs) [35] incorporate the partial difference equations into the loss function as a penalty term, enabling the network to learn the hidden information in the physical equations. However, for partial difference equations, any order derivative of space and time can be easily derived using automatic differentiation of networks, whereas the fractional derivative is not feasible in calculation. Therefore, a variety of difference schemes are tried in PINNs to approximate the time-fractional differential operators.

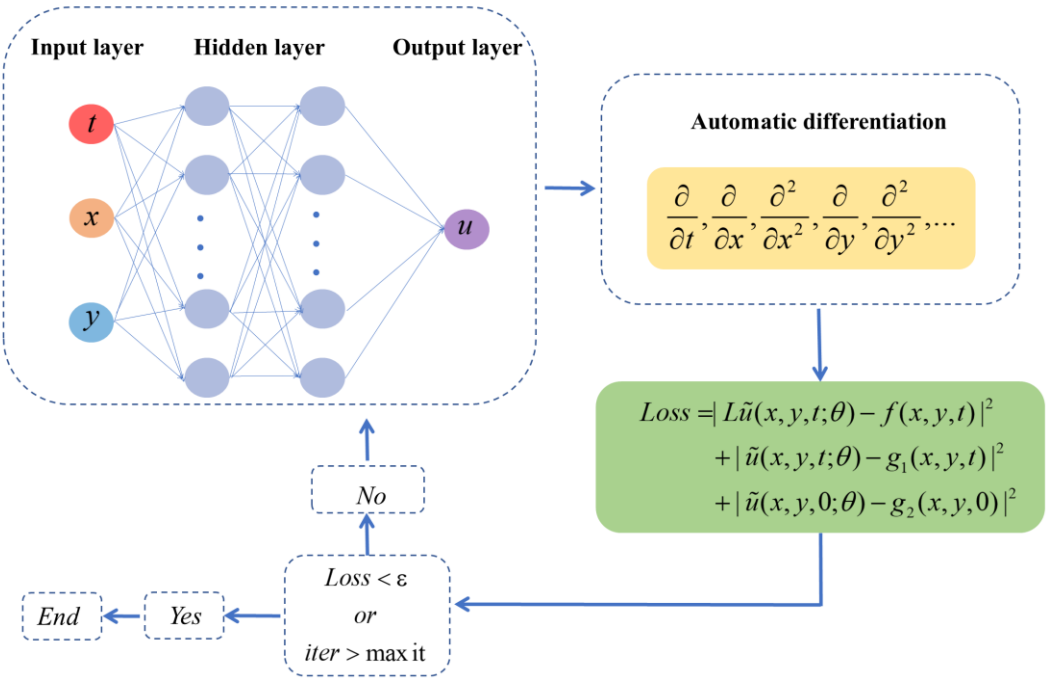


Figure 1. Network structure of PINNs [35].

3.1. Time discretization

The fractional derivative is defined in Caputo sense by [58]

$$ {}_0^cD_t^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-x)^{-\alpha} f'(t)dx, \ 0 < \alpha < 1, t > 0, \tag{3.1} $$

where Γ represents the Gamma function and α is the time-fractional derivative.

The interval $[0, T]$ is divided uniformly into nodes $0 = t_0 < t_1 < t_2 < \ldots < t_{2s} < t_{2s+1} < \ldots < t_{2N_T} = T$, in which s and N_T denote position integer. When $t = t_{2s}$, according to the first-order finite difference method in the sense of the Caputo fractional derivative, the fractional derivative can be evaluated as follows:

$$
\begin{aligned}
{}_0^cDf_{2s} &= \frac{1}{\Gamma(1-\alpha)}\sum_{i=0}^{2s-1}\int_{t_i}^{t_{i+1}}(t_{2s}-x)^{-\alpha}((f_{i+1}-f_i)\Delta t^{-1})dx \\
&= \frac{1}{\Gamma(2-\alpha)}\sum_{i=0}^{2s-1}((f_{i+1}-f_i)\Delta t^{-1})[(2s-i)^{1-\alpha}-(2s-i-1)^{1-\alpha}](\Delta t)^{1-\alpha} \\
&= \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)}\sum_{i=0}^{2s-1}(f_{2s-i}-f_{2s-i-1})[(i+1)^{1-\alpha}-i^{1-\alpha}] \\
&= \sum_{i=0}^{2s-1}\Delta t^{-\alpha}\psi_{i,2s}(f_{2s-i}-f_{2s-i-1})
\end{aligned}
\tag{3.2}
$$

where f_i represents $f(t_i)$, Δt is the size of time step, and

$$\psi_{i,2s} = \frac{1}{\Gamma(2-\alpha)}[(i+1)^{1-\alpha}-i^{1-\alpha}]$$

For time dependent $\hat{f}(\mathbb{X},t_{2s})\in\mathbb{R}^d$, Eq (3.2) can be further written as:

$${}_0^cD_t^{\alpha}\hat{f}(\mathbb{X},t_{2s}) = \Delta t^{-\alpha}\sum_{k=0}^{2s}\bar{\psi}_{k,2s}\hat{f}(\mathbb{X},t_{2s-k}), \tag{3.3}$$

where $\bar{\psi}_{k,2s}$ is evaluated using

$$\bar{\psi}_{k,2s} = \begin{cases} \psi_{0,2s}, & k=0, \\ \psi_{k,2s}-\psi_{k-1,2s}, & 0<k<2s, \\ \psi_{0,0}, & k=2s. \end{cases}$$

In the piecewise quadratic form, the Riemann-Liouville fractional derivative [58] is first used

$${}_0^cDf(t) = \frac{1}{\Gamma(1-\alpha)}\frac{d}{dt}\int_0^t(t-x)^{-\alpha}f(t)dx, 0<\alpha<1, t>0, \tag{3.4}$$

and in the sense of Hadamard finite-part integral, the fractional-order integral discretization scheme within the interval [a, b] can be defined [55]

$$
\begin{aligned}
&\int_a^b(t-\alpha)^{\gamma}f(t)dt \\
&= \sum_{k=0}^{\lfloor\gamma\rfloor-1}\frac{f^{(k)}(b-a)^{k+1-p}}{(k+1-\gamma)k!}+\int_a^b(t-\alpha)^{-\gamma}R_{\lfloor\gamma\rfloor-1}(t,a)dt, \gamma>1
\end{aligned}
\tag{3.5}
$$

in which $R_{\lfloor\gamma\rfloor-1}(t,a)$ is evaluated using

$$R_{\lfloor\gamma\rfloor-1}(t,a) = \frac{1}{\lfloor\gamma\rfloor-1!}\int_a^t(t-y)^{\lfloor\gamma\rfloor-1}f^{\lfloor\gamma\rfloor}(y)dy,$$

$\int$ is the Hadamard finite-part integral and $\lfloor\ \rfloor$ is the maximum operator. In Hadamard finite-part integral sense, the Riemann-Liouville fractional derivative in Eq (3.1) becomes [55]:

$$D_t^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int (t-x)^{-1-\alpha} f(t) dx. \tag{3.6}$$

Based on the PQI polynomial on Eq (3.2), we have

$$D_t^{\alpha} f(\mathbb{X}, t_{2s}) = \Delta t^{-\alpha} \sum_{k=0}^{2s} \nu_{k,2s} \hat{f}(\mathbb{X}, t_{2s-k}) + R_2^{2s}, \tag{3.7}$$

where $|R_2^{2s}| \leq C\Delta t^{3-\alpha} \max_{0<t<1} |\hat{\dot{f}}^{(3)}(\mathbb{X}, t_{2s})|$, $\hat{\dot{f}}^{(3)}$ is the third derivative, and C is a finite constant. The weights $\nu_{k,2s}$, k = 0, 1, 2, ..., 2s, s = 1, 2, ..., N_T have the form of

$$\nu_{k,2s} = \frac{\mathbb{W}_{k,2s}}{\Gamma(3-\alpha)}, k = 1, 2, ..., 2s, \tag{3.8}$$

where

$$\mathbb{W}_{k,2s} = \begin{cases} 2^{-\alpha}(2-\alpha), & \text{for } k = 0, \\ 2^{2-\alpha}(-\alpha), & \text{for } k = 1, \\ \frac{1}{2}W_0(2) + (2^{-\alpha})\alpha^2, & \text{for } k = 2, \\ -W_1(m), & \text{for } k = 2m-1, m = 2, 3, ..., s, \\ \frac{1}{2}(W_2(m) + W_0(m+1)), & \text{for } k = 2m, m = 2, 3, ..., s-1, \\ \frac{1}{2}W_2(s), & \text{for } k = 2s. \end{cases}$$

in which $W_i(k)$ and $V_i(k, \alpha)$ are evaluated separately using

$$\begin{aligned} W_i(k) = {} & V_i(k, \alpha)((2k)^{-\alpha} - (2k-2)^{-\alpha})(1-\alpha)(2-\alpha) \\ & - Z_i(k, \alpha)(-\alpha)(2-\alpha) \\ & + ((2k)^{2-\alpha} - (2k-2)^{2-\alpha})(-\alpha)(1-\alpha), i = 0, 1, 2 \end{aligned}$$

$$V_i(k, \alpha) = \begin{cases} (2k-1)(2k), & i = 0, \\ (2k-2)(2k), & i = 1, \\ (2k-1)(2k-2), & i = 2, \end{cases}$$

$$Z_i(k, \alpha) = \begin{cases} (4k-1)((2k)^{1-\alpha} - (2k-2)^{1-\alpha}), & i = 0, \\ (4k-2)((2k)^{1-\alpha} - (2k-4)^{1-\alpha}), & i = 1, \\ (4k-3)((2k)^{1-\alpha} - (2k-4)^{1-\alpha}), & i = 2. \end{cases}$$

When T is set as 1, we have $t_{2s} = \frac{2s}{2N_T} = \frac{s}{N_T}$. The Riemann-Liouville fractional derivative is transformed into the fractional derivative in Caputo sense.

$$ {}_0^cD_t^{\alpha}\hat{f}(\mathbb{X},t_{2s}) = D_t^{\alpha}(\hat{f}(\mathbb{X},t_{2s}) - \hat{f}(\mathbb{X},t_0)) = \Delta t^{-\alpha}\sum_{k=0}^{2s}\tilde{\nu}_{k,2s}\hat{f}(\mathbb{X},t_{2s-k}) + R_2^{2s}, \tag{3.9} $$

where

$$ \tilde{\nu}_{k,2s} = \begin{cases} \nu_{k,2s}, & k = 0,1,2,\ldots,2s-1, \\ \nu_{k,2s} - \dfrac{(2s)^{-\alpha}}{\Gamma(1-\alpha)}, & k = 2s. \end{cases} $$

3.2. The adaptive learning rate

In our work, the time-fractional differential operators are approximated using the PQI discrete form, which is then integrated into the loss function of fPINNs [51], whose network structure is shown in Figure 2.

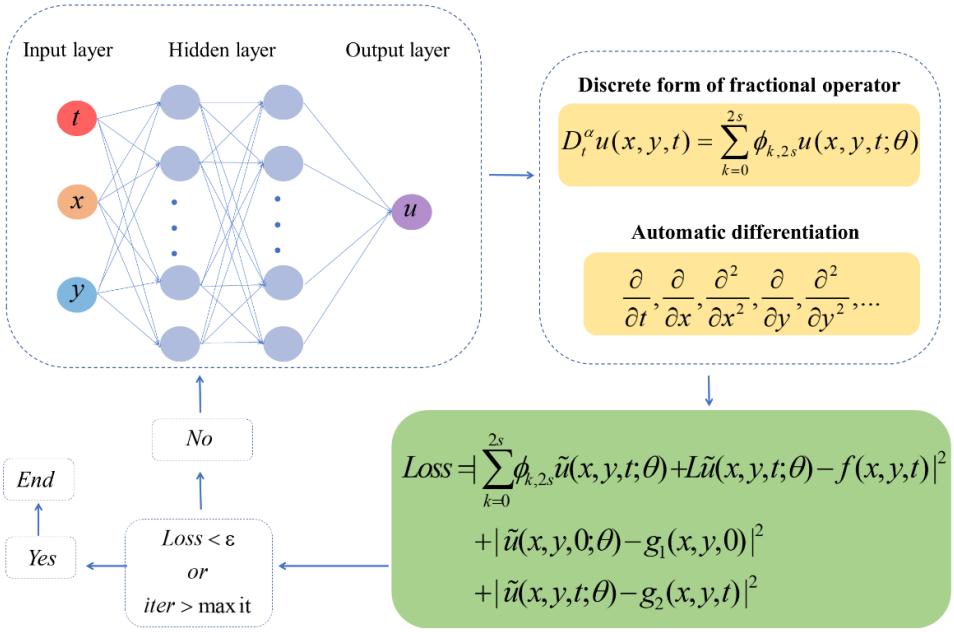


Figure 2. Schematic of fPINNs proposed in [51].

The PQI scheme is used to discretize the fractional derivative, which can be written as:

$$ D_t^{\alpha}u(\mathbb{X},t_{2s}) = \sum_{k=0}^{2s}\tilde{\nu}_{k,2s}u(\mathbb{X},t_{2s-k}), \tag{3.10} $$

where $\tilde{\nu}$ can be calculated using Eq (3.9).

Then, by substituting Eq (3.10) into Eq (2.1), we obtain

$$ \sum_{k=0}^{2s}\tilde{\nu}_{k,2s}u(\mathbb{X},t_{2s-k}) + Lu(\mathbb{X},t) = f(\mathbb{X},t), \quad \mathbb{X}\in\Omega, t\in[0,\mathrm{T}]. \tag{3.11} $$

Hence, the loss function is defined as the mean-squared error (MSE) of the residual loss term, boundary loss term, and initial loss term.

$$
\begin{aligned}
&L(\boldsymbol{\theta}) = L_r(\boldsymbol{\theta}) + \lambda_i L_{u_i}(\boldsymbol{\theta}) + \lambda_b L_{u_b}(\boldsymbol{\theta}),\\
&L_r(\boldsymbol{\theta}) = \frac{1}{N_f}\sum_{i=1}^{N_f} |\sum_{k=0}^{2s} \tilde{v}_{k,2s}\tilde{u}(\mathbb{X}_i, t_{2s-k}) - f_i|^2,\\
&L_{u_i}(\boldsymbol{\theta}) = \frac{1}{N_0}\sum_{i=1}^{N_0} |\tilde{u}(\mathbb{X}_i, 0) - g_{1_i}|^2,\\
&L_{u_b}(\boldsymbol{\theta}) = \frac{1}{N_b}\sum_{i=1}^{N_b} |\tilde{u}(\mathbb{X}_i, t_i) - g_{2_i}|^2 .
\end{aligned}
\tag{3.12}
$$

where λ_b and λ_i represent the adaptive learning rate of the boundary loss term and the initial loss term, respectively, and $\boldsymbol{\theta}$ is the network parameters including the weights and biases, which are updated and iterated using the gradient descent method:

$$
\theta_{n+1} = \theta_n - \eta\nabla_\theta L(\theta) = \theta_n - \eta(\nabla_\theta L_r(\theta) + \nabla_\theta L_{u_i}(\theta) + \nabla_\theta L_{u_b}(\theta)), \tag{3.13}
$$

where η is the learning rate, $\nabla_\theta L_r(\theta), \nabla_\theta L_{u_i}(\theta), \nabla_\theta L_{u_b}(\theta)$ denotes the gradient value of the residual loss term, boundary loss term, and initial loss item.

The coefficients of different loss terms for both standard PINNs and fPINNs are often fixed at 1, as shown in Figures 1 and 2. However, due to the gradient imbalance of different loss terms, the network easily falls into the local optimum because it pays too much attention to the boundary condition, initial condition, or governing equation. To illustrate the issue, we conduct a two-dimensional time-fractional diffusion equation given in Example 2 as an example, which is solved by the fPINNs. The loss function of the network is shown in Figure 3(a), where the loss value of the initial loss term decreases greatly, while the loss values of the boundary loss and residual loss terms remain steady, which results in sluggish convergence or deviation from the optimal solution due to the fixed learning rate.

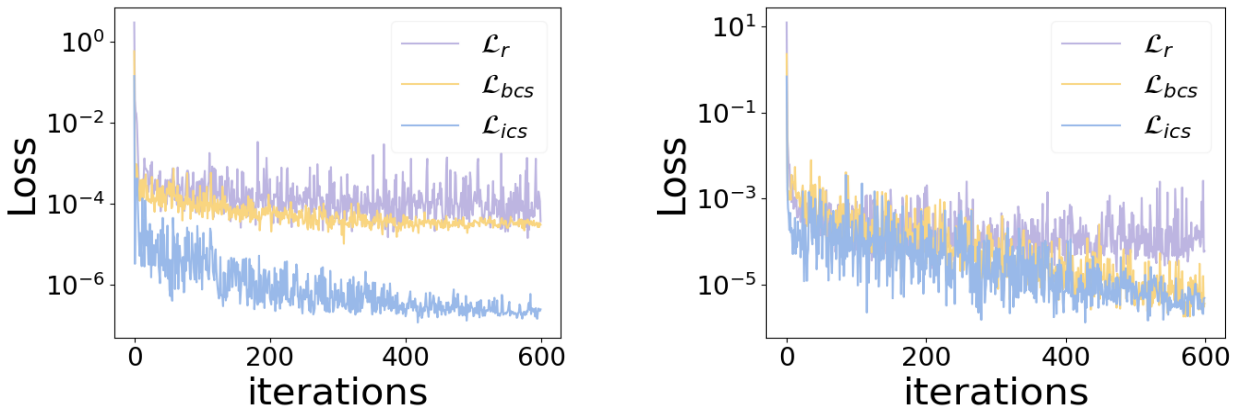


Figure 3. (a) Two-dimensional time-fractional diffusion equation in Example 2 is solved by fPINNs with the fixed weights of different loss terms: L_r represents residual loss and L_{bcs} and L_{ics} indicate boundary loss and initial loss, respectively; (b) two-dimensional time-fractional diffusion equation in Example 2 is solved by fPINNs with the weights with adaptive learning rates of different loss terms.

Hence, we introduce an adaptive weight parameter to mitigate the gradient imbalance between $\nabla_\theta L_r(\theta)$, $\nabla_\theta L_{u_i}(\theta)$ and $\nabla_\theta L_{u_b}(\theta)$, which is defined as

$$\forall \varepsilon > 0, |\nabla_\theta L_r(\theta) - \lambda_{u_i} \nabla_\theta L_{u_i}(\theta) - \lambda_{u_b} \nabla_\theta L_{u_b}(\theta)| < \varepsilon, \tag{3.14}$$

and

$$\lambda_{u_i} = \frac{\overline{|\nabla_\theta L_r(\theta)|}}{\overline{|\nabla_\theta L_{u_i}(\theta)|}}, \ \lambda_{u_b} = \frac{\overline{|\nabla_\theta L_r(\theta)|}}{\overline{|\nabla_\theta L_{u_b}(\theta)|}} \tag{3.15}$$

where $\overline{|\nabla_\theta L_r(\theta)|}$, $\overline{|\nabla_\theta L_{u_i}(\theta)|}$ and $\overline{|\nabla_\theta L_{u_b}(\theta)|}$ denote the gradient mean of residual loss term, boundary loss term, and initial loss term with respect to parameters $\boldsymbol{\theta}$, respectively. When $\lambda_{u_i} = \frac{\max|\nabla_\theta L_r(\theta)|}{\overline{|\nabla_\theta L_{u_i}(\theta)|}}$ and $\lambda_{u_b} = \frac{\max|\nabla_\theta L_r(\theta)|}{\overline{|\nabla_\theta L_{u_b}(\theta)|}}$, it is more conducive to training.

Dynamic weights are assigned to the boundary loss term and initial loss term adaptively according to gradient statistics, so they can alleviate the imbalance gradient of these terms to ensure that the network pays attention to all the loss terms in backpropagation. We introduce the idea of adaptive learning rate into fPINNs and apply it to solve the 2D time-fractional diffusion equation, as shown in Figure 3(b). Through the comparison between Figure 3(a),(b), it is evident that the loss value of the boundary loss item decreases obviously due to the adaptive learning rate.

3.3. The composite activation function

Tanh activation function is widely used by neural networks to solve partial differential equations, and its effectiveness has been proved in many experiments. In the experiments of this paper, we find that the swish activation function also has excellent performance in solving partial differential equations. Using Example 2 of 2D time-fractional diffusion equation as an example, a different activation function is imposed in the neural network, whose results are recorded in Table 1 and Figure 4. Swish activation function has more accuracy than the other three activation functions, the second being tanh(x).

Table 1. Comparison of the relative errors of the prediction solutions solved by the adaptive fPINN based on PQI with various traditional activation functions.

σ	**sigmoid**	**tanh**	**softplus**	**swish**
Relative error	2.04×10^{-3}	$\mathbf{4.78\times10^{-4}}$	6.78×10^{-4}	$\mathbf{5.18\times10^{-5}}$

Considering the excellent performance of swish activation function, we combine it with a novel triangular activation function proposed by authors in [56], which is defined as follows

$$\sigma(x) = m * (\sin x + \cos x) \tag{3.16}$$

where $m \in (0,1)$ is used to adjust the output range. Through numerous numerical examples, m=0.5 has proven to be the most effective.

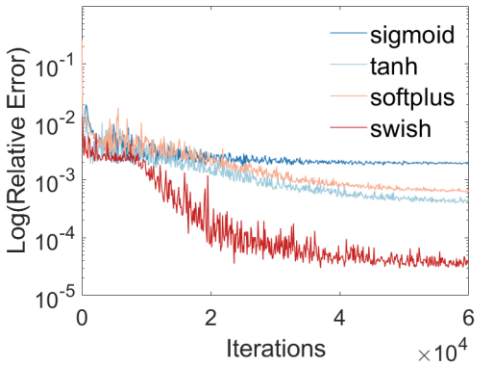


Figure 4. Relative errors of the network solutions under traditional activation functions for solving the 2D time-fractional diffusion equation.

The well-known Fourier series expansion is that any periodic function can be decomposed into a combination of simple sin and cos functions with different frequencies and amplitudes. Therefore, the neural network $\Im(x)$ can be considered as a function of a series of triangular function expansion and combination, namely

$$\Im(x) = \sum_{n=1}^{N} a(x;\theta) B_n(x;\theta) \tag{3.17}$$

$B_n(x;\theta)$ represents triangular functions that served as the activation function for the first hidden layer, and $a(x;\theta)$ represents a set of Fourier coefficients derived from the remaining hidden layers. For the neural network, the simpler the learning content, the better the learning impact. The novel triangular activation function simplifies a complex function to facilitate machine learning by functioning as the Fourier transform.

We create a composite activation function by combining the newly proposed triangular activation function with the typical activation function, allowing the benefits of different activation functions to be fully utilized. It is evident from the results in Table 1 that tanh and swish perform better than the others. Therefore, the composite activation function employs the proposed triangular activation function in the first hidden layer and tanh activation function or swish activation function in the remaining hidden layers.

In this activation function setting, the 2D time-fractional diffusion equation is also conducted to compare the efficiency of the composite function activation along with the single triangular activation function. Each group of experiments was calculated 10 times in a 3-200-200-200-200-200-200-1 neural network. The number of iterations for each operation was 60,000 times. The experimental results come from the average of 10 test results, which are shown in Table 2. The best combination is the triangular activation function and the swish activation function, which is chosen for the following numerical examples.

Table 2. Comparison of the relative errors of the prediction solutions solved by the adaptive fPINN based on PQI using different activation function combinations.

σ	**tanh**	**tanh+triangular**	**swish**	**swish+triangular**
Relative error	4.78×10^{-4}	1.18×10^{-4}	5.18×10^{-5}	$\mathbf{1.94\times10^{-5}}$

3.4. The architecture of adaptive-fPINN-PQI

ResNet structure [57], shown in Figure 5, can efficiently resolve the problem of gradient disappearance in the traditional fully connected neural network (DNN) used in PINNs. In this paper, the ResNet block replaces DNN beginning with the second hidden layer to guarantee the backpropagation efficiently.

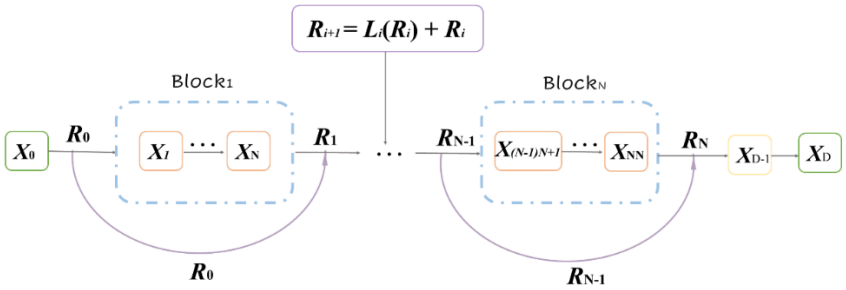


Figure 5. Architecture of ResNet block.

In conclusion, an adaptive-fPINN-PQI algorithm is put forward that makes use of PQI scheme and the composite activation function, together with the Resnet blocks, whose principle is shown in Figure 6 and flowchart is shown in Algorithm 1.

Algorithm 1 Adaptive fPINN neural network with residual blocks (adaptive-fPINN-PQI)

Input: Space and time variables of PDEs.
Output: The latent solution $\tilde{u}$.

1: **discretization:** The fractional derivative is discretized into piecewise quadratic interpolation scheme.
2: **initialization:** Initialize the parameters $\boldsymbol{\theta}$. Then, through the K step of the gradient descent algorithm, the parameters are updated as:
3: **for** n = 1, 2, 3, ..., K **do**
4: i: Read the loss function L ($\boldsymbol{\theta}$).
5: ii: Computer $\overline{\lambda_i} = \dfrac{\max_\theta\{|\nabla_\theta L_r(\theta_n)|\}}{|\nabla_\theta L_i(\theta_n)|}, i = 1, 2, ..., M.$
6: iii: Update the parameters λ_i using a weighted average of the form $\lambda_i = (1-\alpha)\lambda_i + \alpha\lambda_i$, The hyper-parameter is set as α = 0.9.
7: iv: Updating the parameters $\boldsymbol{\theta}$ with adaptive-fPINN-PQI.
8: **end for**

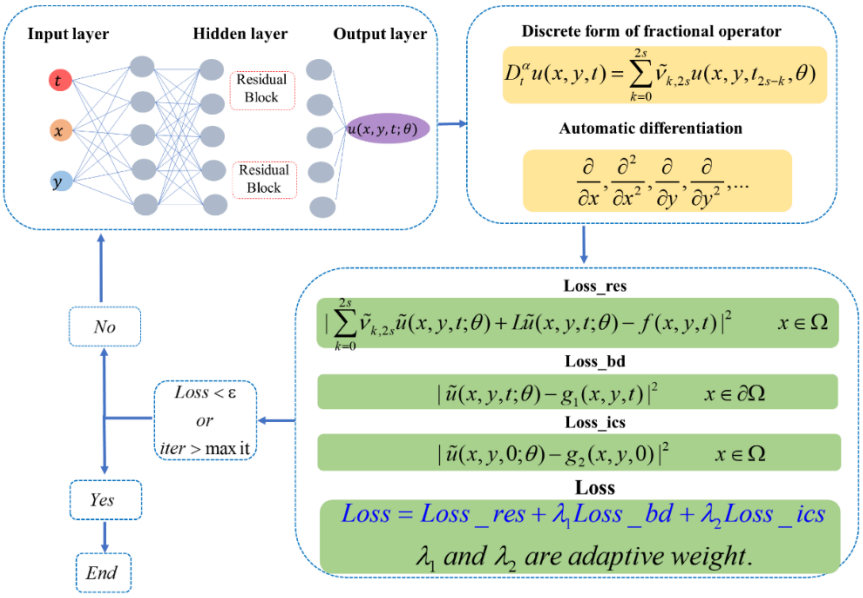


Figure 6. Architecture of adaptive-fPINN-PQI for time-fractional equations.

4. Numerical examples and results

In this section, intensive time-fractional diffusion equations and time-fractional AC equations are conducted to verify the effectiveness of adaptive-fPINN-PQI. In order to evaluate the performance of our method, we consider the absolute error $\|\tilde{\mathbf{u}}-\mathbf{u}\|_1$, and the relative error is expressed as $\frac{\|\tilde{\mathbf{u}}-\mathbf{u}\|_2}{\|\mathbf{u}\|_2}$, where $\tilde{\mathbf{u}}$ and $\mathbf{u}$ denote the approximated and the exact solutions, respectively.

All numerical experiments in this paper are carried out on AMD Ryzen 9 5900HS with Radeon Graphics 3.30 GHz, RTX3060, memory 16.0 GB, and Windows 11 system. TensorFlow (version 2.13.0) is applied for programming utilizing its automatic distinguishing property, and the Adam algorithm is used to optimize the loss function. Xavier is employed for the initialization. The learning rate, the number of hidden layers, the number of neurons, and the number of iterations of the neural network are fixed at 1×10^{-3}, 5200, and 6×10^{4}, respectively. In the numerical examples, we will compare the results of the proposed adaptive-fPINN-PQI with other models. To objectively quantify the accuracy and stability of the neural networks, each model is calculated 10 times and the mean of the errors is then obtained. The models involved in all numerical studies are presented in the following.

• **fPINN[51]:** fPINNs are an improvement of PINNs, using the mathematical discrete method of fractional derivatives to make up for the technical problem that PINNs cannot calculate fractional derivatives by automatic differentiation. The loss function in the fPINNs consists of residual loss, boundary loss, and initial loss, in which their weights are set at 1. The fractional derivative is discretized using the first-order finite difference scheme. Tanh is used as the activation function.

• **adaptive-fPINN:** This model introduces Resnet structure and assigns dynamic weights to different loss terms in the loss function according to the gradient statistics to alleviate the gradient imbalance. The activation function is also changed from a single tanh to a coincidence activation function. The first layer of this composite activation function is a triangular activation function ($0.5(\sin x+\cos x)$), and the remaining layers are swish functions. The other settings are the same as the fPINN model.

• **fPINN-PQI:** In this model, the PQI is introduced for the time-fractional derivative. The activation function uses the same composite activation function as the adaptive fPINN. The rest of the configuration is the same as the fPINN model.

• **adaptive-fPINN-PQI:** Compared with the adaptive-fPINN model, the only adjustment of this model is that the PQI scheme replaces the L1 scheme as the discretization method of the time-fractional derivative.

4.1. Time-fractional diffusion equations

$$\begin{aligned} {}_0^C D_t^\alpha u(\mathbb{X},t) - \Delta u(\mathbb{X},t) &= f(\mathbb{X},t), \quad \mathbb{X}\in\Omega,\ t\in[0,\mathrm{T}], \\ u(\mathbb{X},0) &= g_1(\mathbb{X}), \quad \mathbb{X}\in\Omega, \\ u(\mathbb{X},t) &= g_2(\mathbb{X}), \quad \mathbb{X}\in\partial\Omega,\ t\in[0,\mathrm{T}], \end{aligned} \tag{4.1}$$

where Δ represents the Laplacian operator, $\Omega=[0,1]^d$, $d=1,2,3$ and $\partial\Omega$ is the boundary of Ω, T is set as 1.

4.1.1. Example 1. 1D time-fractional diffusion equation

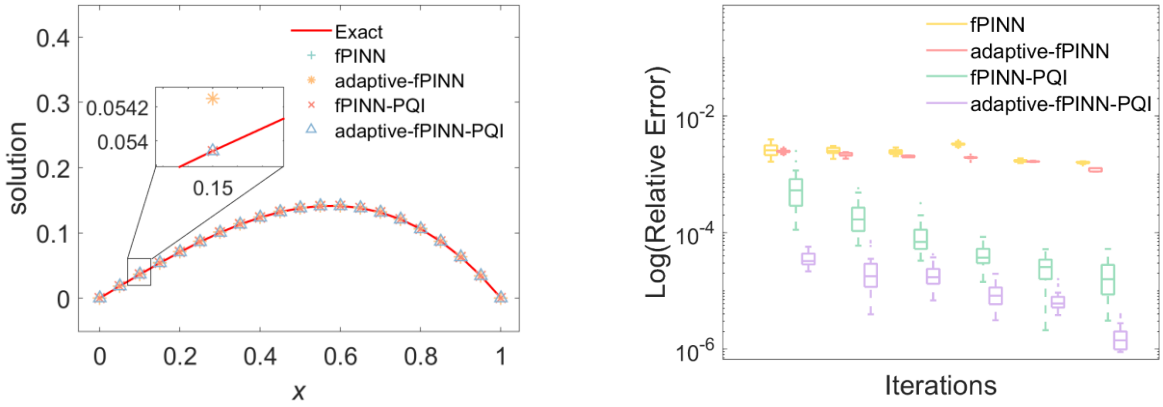
Consider a 1D time-fractional diffusion equation having the form of Eq (4.1) with the fabricated analytical solution $u(x,t)=t^4(x^2-x)$. We can derive the forcing term formula with the form:

$$f(x,t_i)=\frac{\Gamma(5)}{\Gamma(5-\alpha)}t^{4-\alpha}(x^2-x)-2t^4 \tag{4.2}$$

First, Table 3 examines the relative errors of the solutions obtained by different neural networks under different α with the time interval of 0.01. It is obviously shown that 1) in general, the proposed adaptive model produces more accurate results than other counterparts using the same settings; 2) when α is 0.8, the accuracy of the networks using PQI scheme is higher than that of the networks using L1 scheme whether adaptive learning rate is employed or not; and 3) when α is 0.2, the accuracy of the L1 networks is superior to that of the PQI networks.

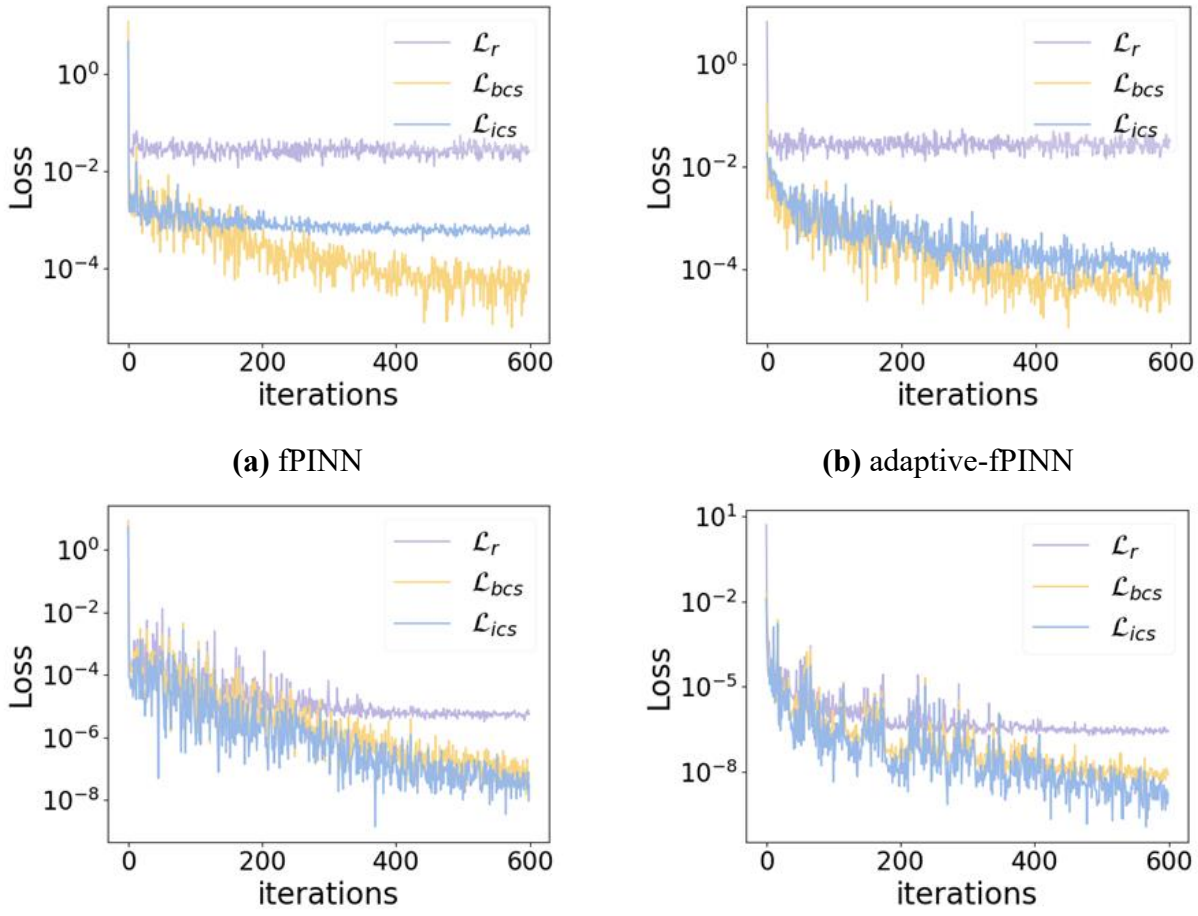
Figure 7(a) displays the distributions of the solutions obtained by the fPINN, adaptive-fPINN, fPINN-PQI, and the proposed adaptive-fPINN-PQI when α is 0.8, Δt is 0.01, and T is 1. It is clearly found that the predicted solutions of the networks utilizing the PQI discretization format accurately are in better agreement with the exact solution compared to the L1 discretization format. Figure 7(b) is the box diagram of the relative errors of these predicted solutions, which further shows the performance of these neural networks. It is obvious that the new model clearly outperforms both in terms of convergence and accuracy.

Then, the evolution process of the loss function in adaptive-fPINN-PQI, as well as those of the other models, is analyzed in Figure 8. When comparing Figure 8(a),(c), it is not difficult to find that the residual loss under the PQI discrete scheme decreases significantly, which is probably reduced by three orders of magnitude. This phenomenon is consistent with the theory of the higher order difference property of the PQI scheme, which provides effective experimental evidence and theoretical basis for the higher accuracy of our model. In addition, the initial loss term in adaptive-fPINN has a clear decreasing trend due to the prominent advantage of adaptive learning rate, observed in Figure 8(a),(b).



(a) **(b)**

Figure 7. (a) Exact solution and the predicted solutions obtained by various neural models for 1D time-fractional diffusion equation at T = 1 after 60,000 iterations; (b) box diagram of the relative errors of the solutions by various neural models at $\alpha = 0.8$.



(a) fPINN **(b)** adaptive-fPINN

(c) fPINN-PQI **(d)** adaptive-fPINN-PQI

Figure 8. Evolution process of the loss function in fPINN, adaptive-fPINN, fPINN-PQI, and adaptive-fPINN-PQI.

Table 3. The relative error between the predicted and exact solutions obtained using fPINN, adaptive-fPINN, fPINN-PQI, and adaptive fPINN-PQI for the 1D time-fractional diffusion equation at t = 1.

α	fPINN	adaptive-fPINN	fPINN-PQI	adaptive-fPINN-PQI
0.2	1.71×10^{-3}	$\mathbf{1.33\times10^{-3}}$	2.30×10^{-3}	1.90×10^{-3}
0.5	1.79×10^{-3}	1.23×10^{-3}	1.16×10^{-4}	$\mathbf{6.73\times10^{-5}}$
0.8	1.78×10^{-3}	1.15×10^{-3}	1.06×10^{-5}	$\mathbf{1.57\times10^{-6}}$

4.1.2. Example 2. 2D time-fractional diffusion equation

Consider a 2D time-fractional diffusion equation defined in Eq (4.1) with the fabricated solution $u(x,y,t)=t^4(x^2-x)(y^2-y)$. The forcing term formula can be deduced as follows:

$$f(x,t_i)=\frac{\Gamma(5)}{\Gamma(5-\alpha)}t^{4-\alpha}(x^2-x)(y^2-y)-2t^4(y^2-y)-2t^4(x^2-x) \tag{4.3}$$

First, the solution error obtained using the proposed neural frame, together with those of the other three models for the 2D time-fractional diffusion equation, is listed in Table 4. It is seen that the adaptive-fPINN-PQI model stands out clearly, where the PQI discrete format also has a significant impact on the solution accuracy for this problem.

Table 4. Relative error between the predicted and the exact solutions obtained using fPINN, adaptive-fPINN, fPINN-PQI, and adaptive-fPINN-PQI for the 2D time-fractional diffusion equation at t = 1.

α	fPINN	adaptive-fPINN	fPINN-PQI	adaptive-fPINN-PQI
0.2	2.16×10^{-3}	$\mathbf{3.76\times10^{-4}}$	2.81×10^{-3}	2.18×10^{-3}
0.5	2.18×10^{-3}	4.70×10^{-4}	8.43×10^{-5}	$\mathbf{2.39\times10^{-5}}$
0.8	2.02×10^{-3}	3.24×10^{-4}	6.61×10^{-5}	$\mathbf{1.94\times10^{-5}}$

To more intuitively reflect the performance of each model, the abstract point-wise error distributions of the four models are plotted in Figure 9. We can observe that the abstract point-wise error of the proposed network is the smallest among the four networks. In the problem domain, the proposed adaptive-fPINN-PQI agrees well with the exact solution, and its abstract error is clearly controlled below 10^{-7} in contrast to the other models, which perform poorly and have large errors distributed throughout the region. On the boundary of the problem domain, the abstract error of our model is still controlled in 10^{-6}, despite the fact that its accuracy has decreased, whereas those of some other models even approach 10^{-4}.

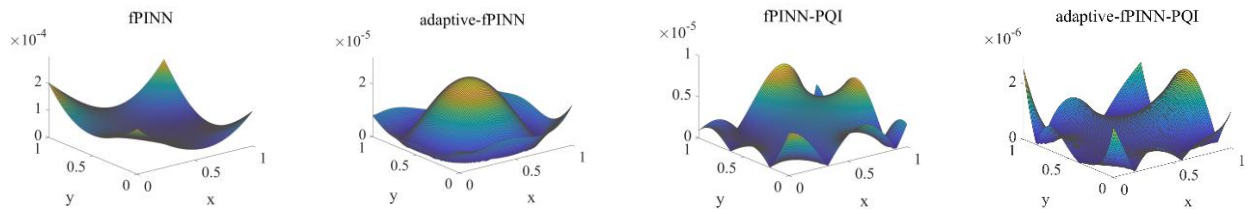


Figure 9. Point-wise error between the exact and predicted solutions of the various neural networks under the condition of $\alpha = 0.8$.

We also investigated the convergence performance of four models in Figure 10. The maximum and minimum values from the 10 experiments are represented by the shadow boundaries, while the average is represented by the solid line. It demonstrates that the accuracy of adaptive-fPINN-PQI is better than that of the other models. After 30,000 iterations, the shadow width of these models gradually decreases, indicating that the network prediction is more and more stable.

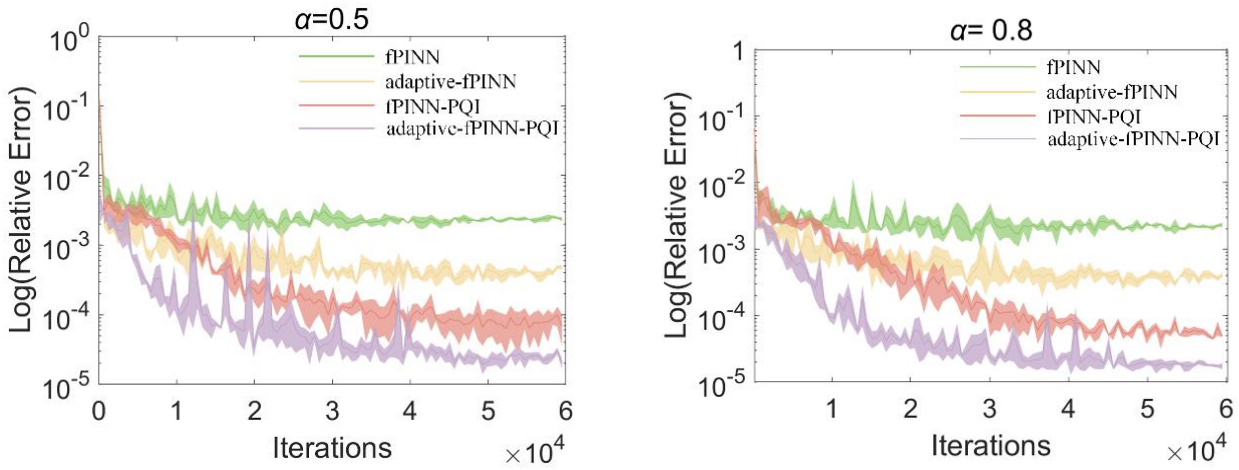


Figure 10. Relative errors of the solutions obtained using the four neural networks.

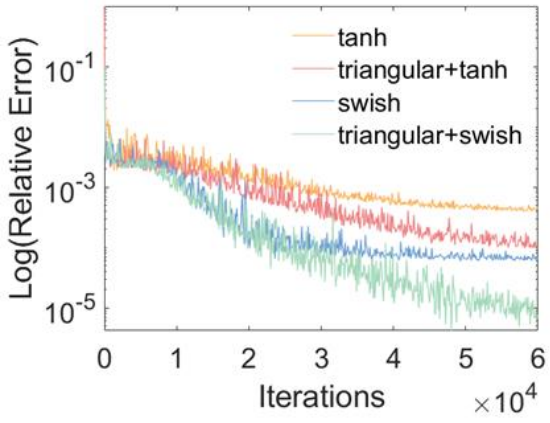


Figure 11. Relative error of adaptive-fPINN-PQI model under various activation functions.

Finally, the effectiveness of the newly proposed composite activation function is discussed for

the 2D time-fractional diffusion equation. Figure 11 plots the relative error of adaptive-fPINN-PQI utilizing various activation functions. In comparison to the two traditional activation functions, tanh and swish, the proposed composite triangular activation function has a significant impact on enhancing the calculation accuracy. Although the result using the suggested composite activation function exhibits some fluctuation in error, the maximum error remains one or two orders smaller than its counterparts, and the fluctuation becomes less pronounced as the number of iterations increases.

4.1.3. Example 3. 3D time-fractional diffusion equation

Consider a 3D time-fractional diffusion equation defined in Eq (4.1) with the fabricated exact solution $u(x,\mathrm{y},z,t)=t^4(x^2-x)(y^2-y)(z^2-z)$ and the forcing term can be written as follows:

$$\begin{aligned} f(x,t_i)=\frac{\Gamma(5)}{\Gamma(5-\alpha)}t^{4-\alpha}(x^2-x)(y^2-y)(z^2-z)-2t^4(y^2-y)(z^2-z)\\ -2t^4(x^2-x)(z^2-z)-2t^4(x^2-x)(y^2-y) \end{aligned} \tag{4.4}$$

Table 5 presents the relative error of the adaptive-fPINN-PQI for the 3D time-fractional diffusion equation defined above, in comparison with the other three models. Among these neural frameworks, adaptive-fPINN-PQI turns out to be the most accurate, proving once more how well the proposed model performs in terms of precision.

Table 5. Relative error between the predicted and exact solutions obtained using fPINN, adaptive-fPINN, fPINN-PQI, and adaptive-fPINN-PQI for the 3D time-fractional diffusion equation at t = 1.

α	**fPINN**	**adaptive-fPINN**	**fPINN-PQI**	**adaptive-fPINN-PQI**
0.2	5.45×10^{-3}	$\mathbf{3.19\times10^{-4}}$	6.74×10^{-4}	6.6×10^{-4}
0.5	3.24×10^{-3}	8.49×10^{-4}	2.34×10^{-4}	$\mathbf{9.51\times10^{-5}}$
0.8	3.13×10^{-3}	2.17×10^{-4}	1.52×10^{-4}	$\mathbf{7.35\times10^{-5}}$

Next, Figure 12 displays the distributions of the abstract errors obtained using the four models with α fixed at 0.8. A graph shows that the accuracy of the adaptive-fPINN-PQI model surpasses that of the competition models. Positions where the accuracy of these learning models is poor have one thing in common: the error is concentrated around the boundary, which is consistent with the way numerical computations work. However, the suggested approach is still effective in reducing the boundary error and narrowing the error range.

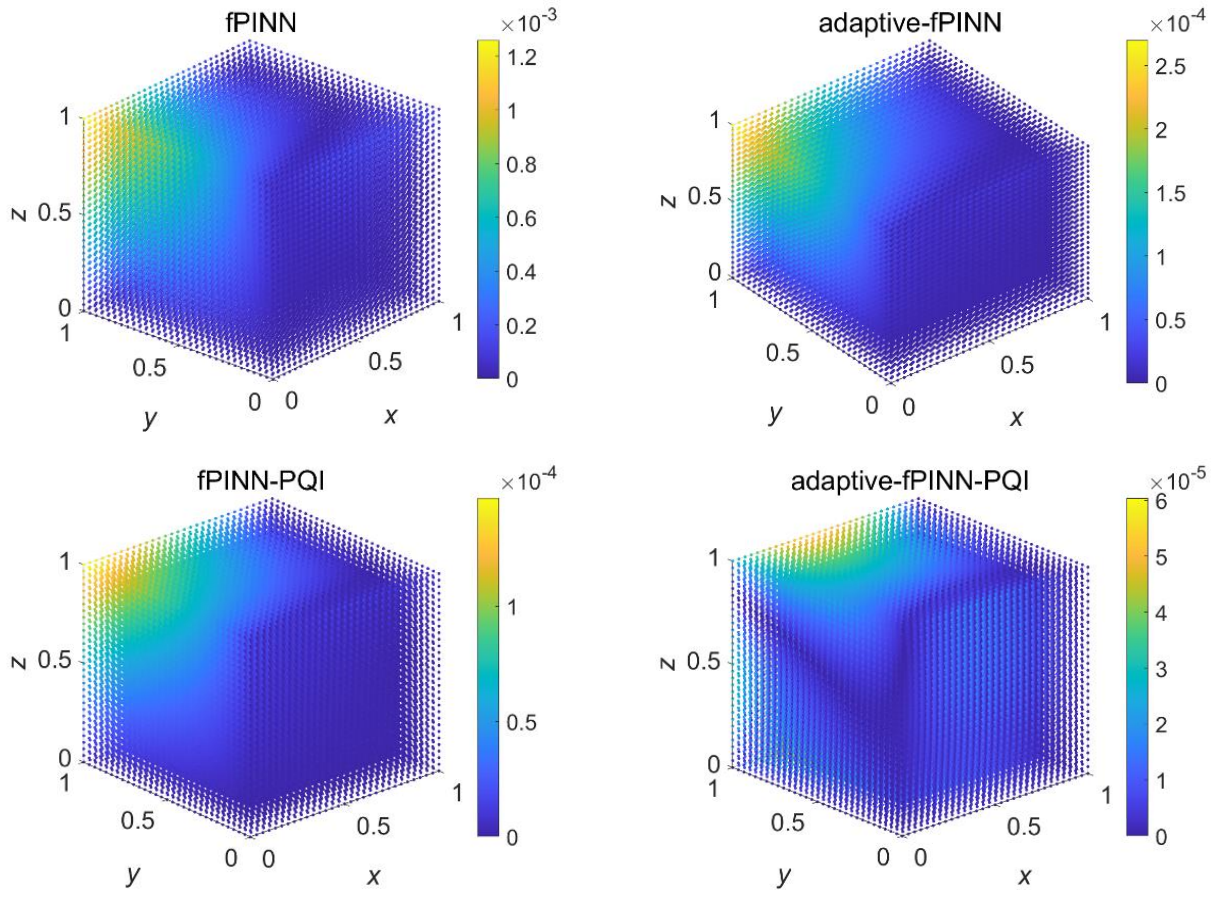
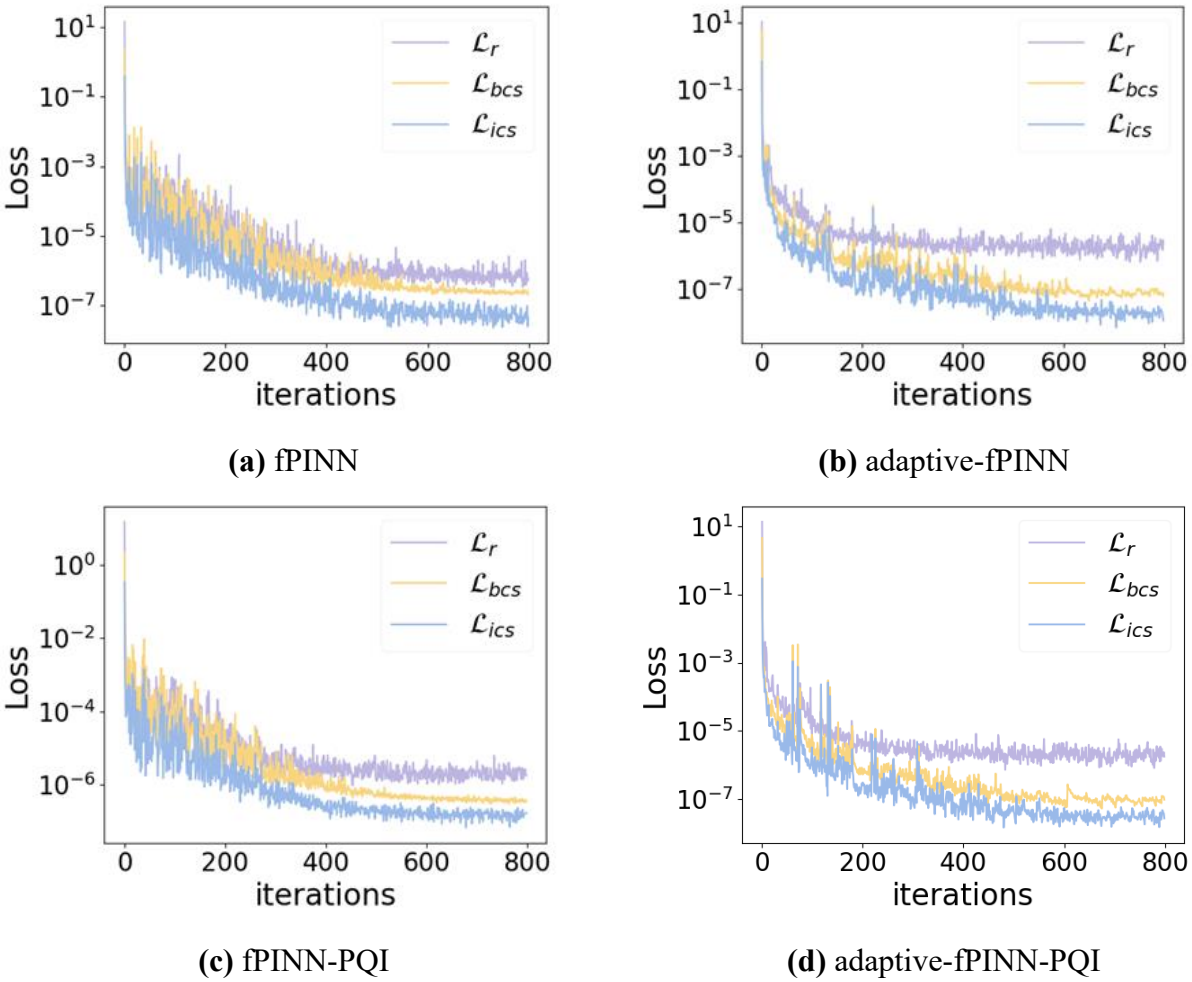


Figure 12. Distribution of the absolute errors of four distinct models for the 3D time-fractional diffusion equation.

To investigate the learning capability of adaptive-fPINN-PQI, we plot its loss function against the number of iterations in Figure 13, along with those of the other models. Generally, all four algorithms demonstrate excellent learning efficiency when forecasting the solution of the 3D time-fractional diffusion equation. When compared to L1 and PQI schemes, it is clear that the neural network using PQI significantly reduces the residual loss. Besides, the adaptive learning rate obviously balances the gradient differences of various loss terms, which come from the narrowed shadow width in Figure 13(b),(d).

Furthermore, we conducted a comparative analysis of the activation function for the 3D time-fractional diffusion equation, which is displayed in Figure 14. Among various activation functions, we noted that our composite activation functions exhibited superior convergence and stability. Especially after 5000 iterations, the conclusion becomes clearer.



(a) fPINN **(b)** adaptive-fPINN

(c) fPINN-PQI **(d)** adaptive-fPINN-PQI

Figure 13. Evolution process of the loss function in fPINN, adaptive-fPINN, fPINN-PQI, and adaptive-fPINN-PQI for the 3D time-fractional diffusion equation.

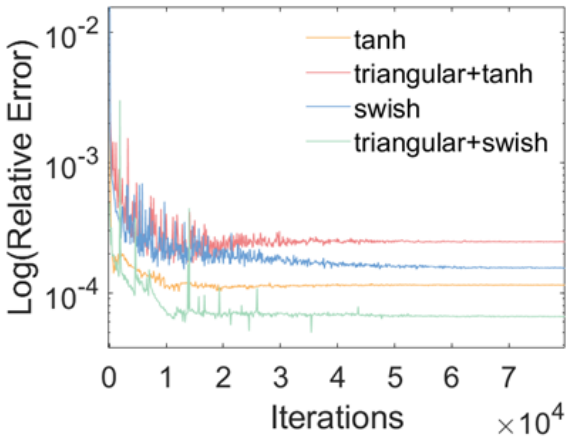


Figure 14. Relative error of adaptive-fPINN-PQI model under various activation functions for the 3D time-fractional diffusion equation.

4.2. Time-fractional AC equation

$$\begin{aligned} D_t^{\alpha} u(\mathbb{X},t) - \varepsilon^2 \Delta u - u + u^3 &= f(\mathbb{X},t), \quad \mathbb{X} \in \Omega, t \in [0,\mathrm{T}], \\ u(\mathbb{X},0) &= g_1(\mathbb{X}), \quad \mathbb{X} \in \Omega, \\ u(\mathbb{X},t) &= g_2(\mathbb{X}), \quad \mathbb{X} \in \partial\Omega, t \in [0,\mathrm{T}]. \end{aligned} \tag{4.5}$$

where Δ represents the Laplacian operator, $\Omega = [0,1]^d$, $d = 1,2$ and $\partial\Omega$ is the boundary of Ω, besides we fix $\varepsilon = 1, \mathrm{T} = 1$.

4.2.1. Example 1. 1D time-fractional AC equation

Consider the 1D time-fractional AC equation having the form of Eq (4.5) with the fabricated exact solution $u(x,t) = (1+t^2)\sin(2\pi x)$ and the forcing term

$$\begin{aligned} f(x,t) = \frac{\Gamma(3)}{\Gamma(3-\alpha)} t^{2-\alpha} \sin(2\pi x) + 4\pi^2 (1+t^2)\sin(2\pi x) \\ -(1+t^2)\sin(2\pi x) + [(1+t^2)\sin(2\pi x)]^3 \end{aligned} \tag{4.6}$$

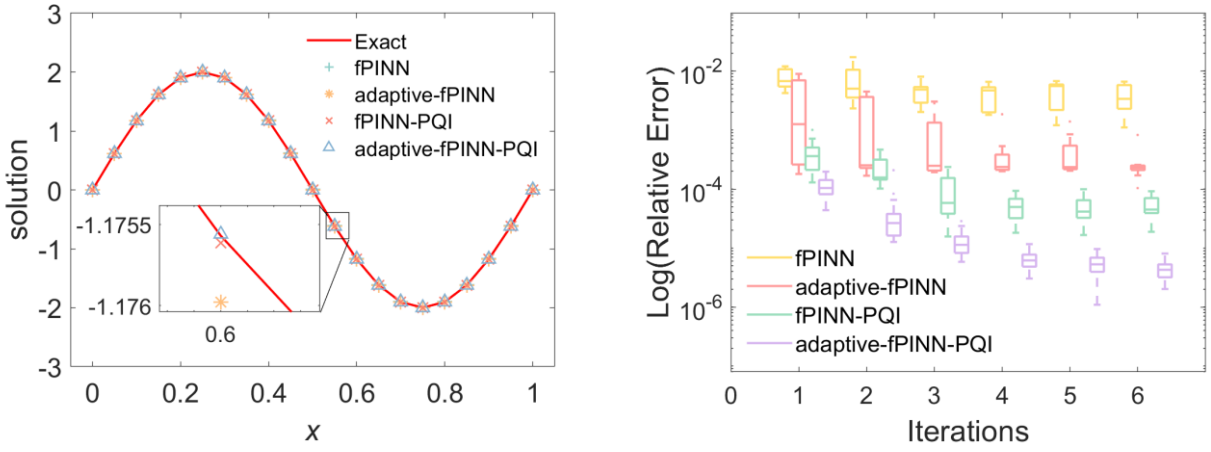
A comparative investigation on the accuracy of four neural models solving the 1D time-fractional AC equation is summarized in Table 6. We observed that the accuracy of adaptive-fPINN-PQI and fPINN-PQI is one to two orders of magnitude higher than the accuracy of fPINN and adaptive-fPINN, which is attributed to the higher-order discrete format. Additionally, the adaptive learning rate in loss function causes an order of magnitude increase in the precision of adaptive-fPINN-PQI.

Table 6. Comparison of the relative errors of the prediction solutions using four networks for the 1D time-fractional AC equation when t = 1.

α	**fPINN**	**adaptive-fPINN**	**fPINN-PQI**	**adaptive-fPINN-PQI**
0.2	5.95×10^{-3}	2.36×10^{-4}	1.65×10^{-4}	$\mathbf{4.34\times10^{-5}}$
0.5	3.94×10^{-3}	2.46×10^{-4}	6.85×10^{-5}	$\mathbf{7.26\times10^{-6}}$
0.8	3.85×10^{-3}	2.38×10^{-4}	7.14×10^{-5}	$\mathbf{4.32\times10^{-6}}$

The accuracy and stability of the prediction solutions of the four deep learning networks are examined in Figure 15, where the distribution of the solutions is plotted in Figure 15(a) and its box diagram is plotted in Figure 15(b). In terms of precision, the adaptive-fPINN-PQI model exhibits the best agreement with the exact solution, followed by fPINN-PQI, and fPINN and adaptive-fPINN perform worse. It can be observed from the box diagram that the adaptive-fPINN-PQI model has the shortest box length among its competitors, indicating that the proposed network is the most stable.

Experiments are then carried out for the 1D time-fractional AC equation to verify the effectiveness of the composite activation function proposed in our study. The results are presented in Figure 16, which shows that the composite activation function offers superior prediction accuracy, better convergence, and greater stability than the other competitors.



(a) **(b)**

Figure 15. (a) Difference between the exact solution and the predicted solution of the 1D time-fractional AC equation obtained by various models at t = 1 after 60,000 iterations; **(b)** box diagram of the relative errors of the solutions by various models at $\alpha = 0.8$.

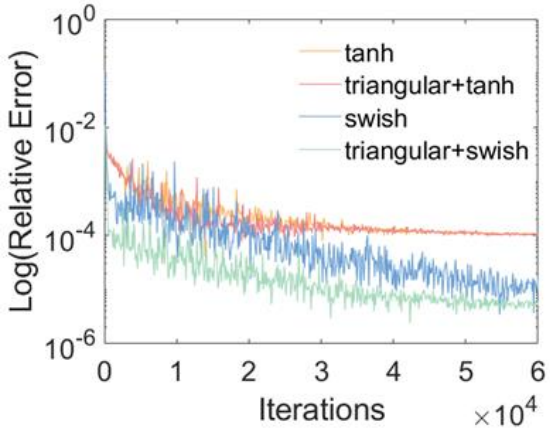


Figure 16. Relative error of adaptive-fPINN-PQI model using various activation functions for the 1D time-fractional AC equation.

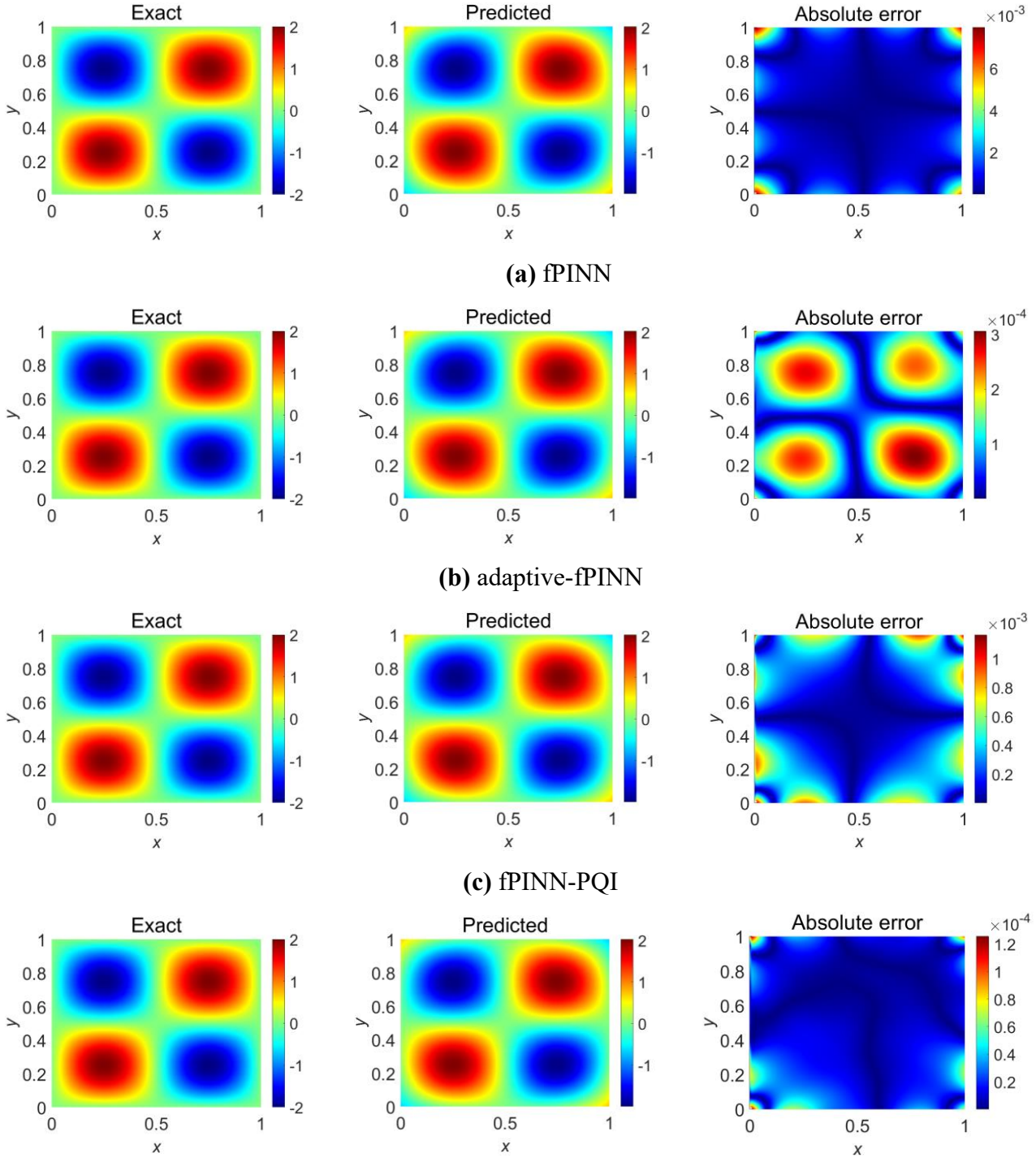
4.2.2. Example 2. 2D time-fractional AC equation

Consider a 2D time-fractional AC equation having the form of Eq (4.5) with the fabricated exact solution $u(x,t) = (1+t^2)\sin(2\pi x)\sin(2\pi y)$, the force term $f(x, y, t_i)$ can be obtained:

$$f(x,y,t) = \frac{\Gamma(3)}{\Gamma(3-\alpha)} t^{2-\alpha}\sin(2\pi x)\sin(2\pi y) + 8\pi^2(1+t^2)\sin(2\pi x)\sin(2\pi y) - (1+t^2)\sin(2\pi x)\sin(2\pi y) + [(1+t^2)\sin(2\pi x)\sin(2\pi y)]^3 \tag{4.7}$$

The relative errors of the different neural networks' solutions are summarized in Table 7. It shows clearly that adaptive-fPINN-PQI produces much more accurate solutions, which are one or two orders

of magnitude higher than other methods.



(a) fPINN

(b) adaptive-fPINN

(c) fPINN-PQI

(d) adaptive-fPINN-PQI

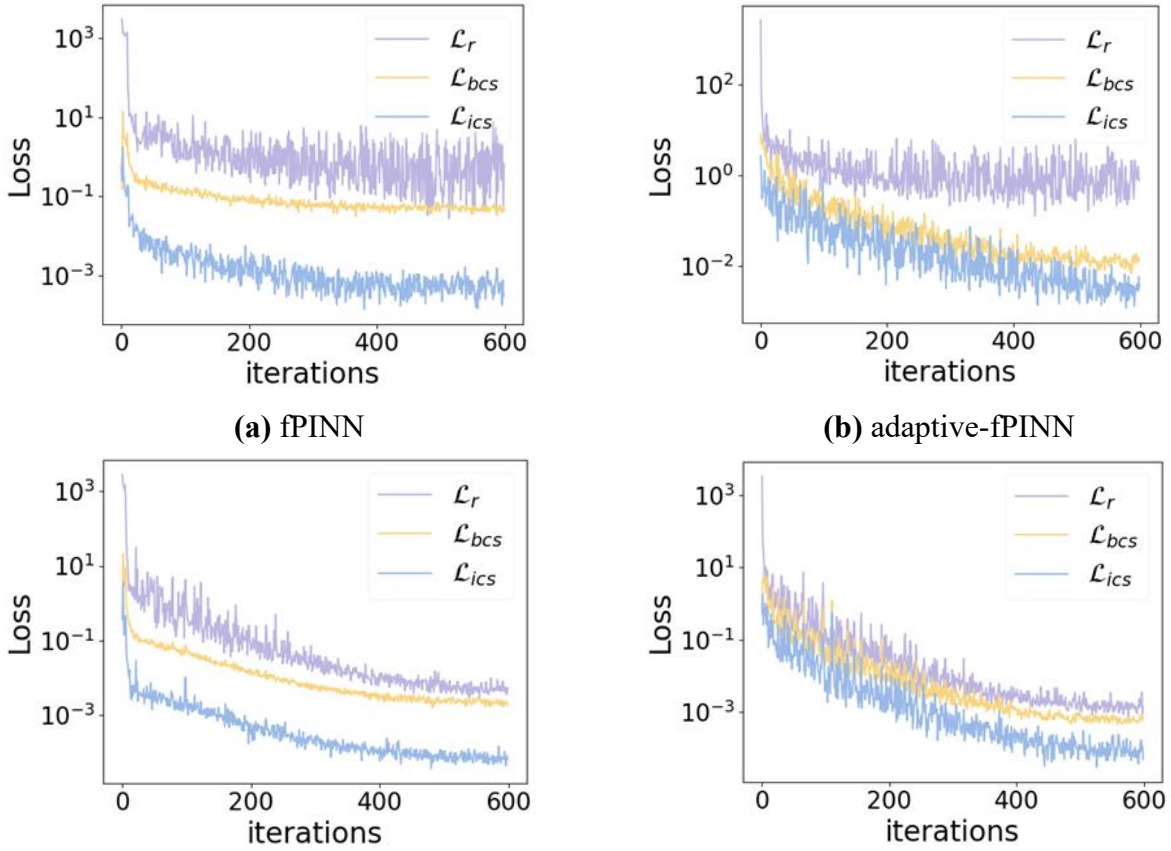
Figure 17. Exact solution versus the predicted solution by training different algorithm models after 60,000 iterations of gradient descent under the condition of t = 1 for the 2D time-fractional AC equation.

Table 7. Comparison of the relative errors of the solutions obtained using four neural networks for the 2D time-fractional AC equation after 60,000 iterations.

α	fPINN	adaptive-fPINN	fPINN-PQI	adaptive-fPINN-PQI
0.2	1.04×10^{-3}	1.65×10^{-4}	3.22×10^{-4}	$\mathbf{6.20\times10^{-5}}$
0.5	8.84×10^{-4}	1.60×10^{-4}	3.31×10^{-4}	$\mathbf{2.91\times10^{-5}}$
0.8	1.25×10^{-3}	1.53×10^{-4}	3.21×10^{-4}	$\mathbf{1.85\times10^{-5}}$

Then, the distributions of the predicated solutions are plotted in Figure 17 to further validate the performance of our neural network. In terms of the error in accuracy and distribution, the solution of adaptive-fPINN-PQI is the best, followed closely by that of adaptive-fPINN.

Figure 18 shows the trend of the loss values during the training process of four neural networks. We can find that the PQI discrete scheme has a significant effect on reducing the residual loss terms. In addition, the PQI format accelerates the convergence of the loss function compared to the format of L1, allowing the neural network to learn the governing equation and boundary information more thoroughly, thereby improving the predictive accuracy.



(a) fPINN **(b)** adaptive-fPINN

(c) fPINN-PQI **(d)** adaptive-fPINN-PQI

Figure 18. Loss function versus training iterations using different networks with the network frame 3-200-200- 200-200-200-1 for the 2D time-fractional AC equation.

A shadow graph of the relative error of 10 experimental results versus the number of iterations

for the 2D fractional AC equation is depicted in Figure 19. The suggested adaptive-fPINN-PQI model outperforms the other three models regardless of α value. The PQI discrete scheme overcomes the issue in which the residual loss term does not decrease, and the adaptive learning rate balances the gradient of different loss terms, resulting in the high precision of adaptive-fPINN-PQI.

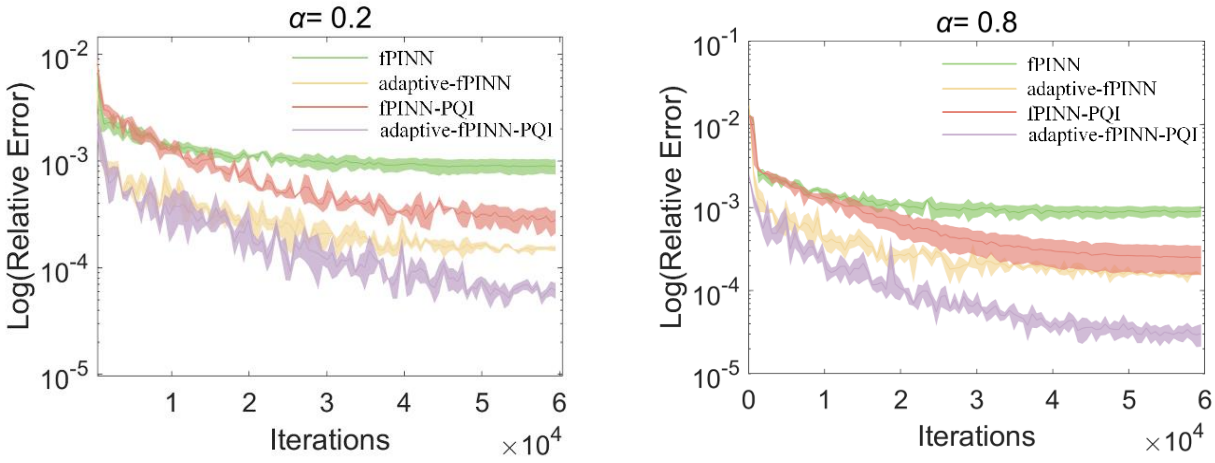


Figure 19. Relative errors of the solutions obtained by different models.

5. Conclusions

In this paper, an adaptive-fPINN-PQI network is built for solving time-fractional partial differential equations. The adaptive learning rate balances the interaction of different loss terms in order to solve the local optimal problem in the classical PINN and fPINN. The L1 scheme in fPINN is replaced by the higher-order scheme of PQI in our neural network, substantially reducing the residual loss term. Additionally, a composite function made up of the triangular activation function and swish activation function is more efficient than the tanh function. The numerical examples of time-fractional diffusion equations and time-fractional AC equations lead to the following conclusions:

- The proposed neural network outperforms fPINN in terms of precision, with a potential improvement of 1–2 orders of magnitude.
- The accuracy of the prediction solutions remains steady with the increase of the dimension of the problems.
- Adaptive-fPINN-PQI performs effectively with various time-fractional partial differential equations.

However, the model proposed in this paper is not as effective as the traditional fPINN model for the diffusion problem when the α value is small, while this issue does not arise for the phase-field problem. Therefore, it is the common goal of researchers to find a model with wider applicability and higher accuracy.

In future work, we will further research the fractional derivatives under different definitions, especially to solve the Hadamard-type fractional problems. A breakthrough for fractional-order problems is to find a higher-order convergent discrete format, which is a challenge for many mathematical researchers. In addition, developing a more widely applicable machine learning model for solving fractional-order problems is the focus of our future work.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

In this paper, the work was supported by the National Youth Science Foundation of China (Grant No. 12102283) and the Youth Science Research Foundation of Shanxi Province (Grant No. 20210302124159).

Competing Interests

No potential conflict of interest was reported by the authors.

References

1. Shallu, V. K. Kukreja, An improvised extrapolated collocation algorithm for solving Kuramoto–Sivashinsky equation, *Math. Methods Appl. Sci.*, **45** (2022), 1451–1467. https://doi.org/10.1002/mma.7865
2. V. K. Kukreja, An optimal B-spline collocation technique for numerical simulation of viscous coupled Burgers' equation, *Comput. Methods Differ. Equations*, **10** (2022), 1027–1045.
3. Shallu, V. K. Kukreja, An improvised collocation algorithm with specific end conditions for solving modified Burgers equation, *Numer. Methods Partial Differ. Equations*, **37** (2021), 874–896. https://doi.org/10.1002/num.22557
4. F. Y. Song, C. J. Xu, G. E. Karniadakis, A fractional phase-field model for two-phase flows with tunable sharpness: Algorithms and simulations, *Comput. Methods Appl. Mech. Eng.*, **305** (2016), 376–404. https://doi.org/10.1016/i.cma.2016.03.018
5. B. E. Treeby, B. T. Cox, Modeling power law absorption and dispersion for acoustic propagation using the fractional Laplacian, *J. Acoust. Soc. Am.*, **127** (2010), 2741–2748. https://doi.org/10.1121/1.3377056
6. S. Holm, S. P. Nasholm, Comparison of fractional wave equations for power-law attenuation in ultrasound and elastography, *Ultrasound. Med. Biol.*, **40** (2014), 695–703. https://doi.org/10.1016/j.ultrasmedbio.2013.09.033
7. T. Y. Zhu, J. M. Harris, Modeling acoustic wave propagation in heterogeneous attenuating media using decoupled fractional Laplacians, *Geophysics*, **79** (2014), T105–T116. https://doi.org/10.1190/geo2013-0245.1
8. W. Zhang, A. Capilnasiu, G. Sommer, G. A. Holzapfel, N. A. Nordsletten, An efficient and accurate method for modeling nonlinear fractional viscoelastic biomaterial, *Comput. Methods Appl. Mech. Eng.,* 2020, **362** (2020), 112834. https://doi.org/10.1016/j.cma.2020.112834
9. Q. W. Xu, Y. F. Xu, Quenching study of two-dimensional fractional reaction-diffusion equation from combustion process, *Comput. Math. Appl*, **78** (2019), 1490–1506. https://doi.org/10.1016/j.camwa.2019.04.006

10. K. A. Mustapha, K. M. Furati, O. M. Knio, O. P. Le Maître, A finite difference method for space fractional differential equations with variable diffusivity coefficient, *Commun. Appl. Math. Comput.*, **2** (2020), 671–688. https://link.springer.com/article/10.1007/s42967-020-00066-6
11. A. M. Vargas, Finite difference method for solving fractional differential equations at irregular meshes, *Math. Comput.*, **193** (2022), 204–216. https://doi.org/10.1016/j.matcom.2021.10.010
12. K. Nedaiasl, R. Dehbozorgi, Galerkin finite element method for nonlinear fractional differential equations, *Numer. Algorithms*, **88** (2021), 113–141. https://link.springer.com/article/10.1007/s11075-020-01032-2
13. F. H. Zeng, C. P. Li, F. W. Liu, I. Turner, The use of finite difference/element approaches for solving the time-fractional subdiffusion equation, *SIAM J. Sci. Comput.*, **35** (2013), A2976–A3000. https://epubs.siam.org/doi/10.1137/130910865
14. C. L. Wang, Z. Q. Wang, L. L. Wang, A spectral collocation method for nonlinear fractional boundary value problems with a Caputo derivative, *J. Sci. Comput.*, **76** (2018), 166–188. https://link.springer.com/article/10.1007/s10915-017-0616-3
15. C. P. Li, F. H. Zeng, F. W. Liu, Spectral approximations to the fractional integral and derivative, *Fractional Calculus Appl. Anal.*, **15** (2012), 383–406. https://link.springer.com/article/10.2478/s13540-012-0028-x
16. F. W. Liu, S. Shen, V. Anh, I. Turner, Analysis of a discrete non-markovian random walk approximation for the time fractional diffusion equation, *Aust. N. Z. Ind. Appl. Math. J.*, **46** (2004), C488–C504. https://doi.org/10.21914/anziamj.v46i0.973
17. T. A. M. Langlands, B. I. Henry, The accuracy and stability of an implicit solution method for the fractional diffusion equation, *J. Comput. Phys.*, **205** (2005), 719–736. https://doi.org/10.1016/j.jcp.2004.11.025
18. Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, *J. Comput. Phys.*, **225** (2007), 1533–1552. https://doi.org/10.1016/j.jcp.2007.02.001
19. W. Deng, Finite element method for the space and time fractional fokker-planck equation, *Soc. Ind. Appl. Math.*, **47** (2008), 204–216. https://epubs.siam.org/doi/abs/10.1137/080714130
20. Y. Yan, K. Pal, N. J. Ford, Higher order numerical methods for solving fractional differential equations, *BIT Numer. Math.*, **54** (2014), 555–584. https://link.springer.com/article/10.1007/s10543-013-0443-3
21. G. R. Liu, J. Zhang, K. Y. Lam, H. Li, G. Xu, Z. H. Zhong, et al., A gradient smoothing me-thod (GSM) with directional correction for solid mechanics problems, *Comput. Mech.*, **41** (2008), 457–472. https://link.springer.com/article/10.1007/s00466-007-0192-8
22. M. Li, P. H. Wen, Finite block method for transient heat conduction analysis in functionally graded media, *Int. J. Numer. Meth. Eng.*, **99** (2014), 372–390. https://doi.org/10.1002/nme.4693
23. M. Li, Y. C. Hon, T. Korakianitis, P. H. Wen, Finite integration method for nonlocal elastic bar under static and dynamic loads, *Eng. Anal. Bound. Elem.*, **37** (2013), 842–849. https://doi.org/10.1016/j.enganabound.2013.01.018
24. B. Ahmad, A. Alsaedi, S. K. Ntouyas, J. Tariboon, *Hadamard-Type Fractional Differential Equations, Inclusions and Inequalities*, Springer International Publishing, 2017.
25. M. Li, C. S. Chen, Y. C. Hon, P. H. Wen, Finite integration method for solving multi-dimensional partial differential equations, *Appl. Math. Modell.*, **39** (2015), 4979–4994. https://doi.org/10.1016/j.apm.2015.03.049

26. S. Brunton, J. Proctor, J. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Nat. Acad. Sci.*, **113** (2016), 3932–3937. https://doi.org/10.1073/pnas.1517384113
27. S. Brunton, B. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.*, **52** (2020), 477–508. https://doi.org/10.1146/annurev-fluid-010719-060214
28. W. S. Zha, W. Zhang, D. L. Li, Y. Xing, L. He, J. Q. Tan, Convolution-based model-solving method for three-dimensional, unsteady, partial differential equations, *Neural Comput.*, **34** (2022), 518–540. https://doi.org/10.1016/j.jcp.2019.108925
29. Z. C. Long, Y. P. Lu, B. Dong, PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network, *J. Comput. Phys.*, **399** (2019), 108925.
30. R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R. Gao, Deep learning and its applications to machine health monitoring, *Mech. Syst. Signal Process*, **115** (2019), 213–237. https://doi.org/10.48550/arXiv.1612.07640
31. Q. Wang, G. Zhang, C. Sun, N. Wu, High efficient load paths analysis with U index generated by deep learning, *Comput. Methods Appl. Mech. Eng.*, **344** (2019), 499–511. https://doi.org/10.1016/j.cma.2018.10.012
32. D. Finol, Y. Lu, V. Mahadevan, A. Srivastava, Deep convolutional neural networks for eigenvalue problems in mechanics, *Int. J. Numer. Methods Eng.*, **118** (2019), 258–275. https://doi.org/10.48550/arXiv.1801.05733
33. W. Zheng, F. T. Weng, J. L. Liu, K. Cao, M. Z. Hou, J. Wang, Numerical solution for high dimensional partial differential equations based on deep learning with residual learning and data-driven learning, *Int. J. Mach. Learn. Cybern.*, **12** (2021), 1839–1851. https://link.springer.com/article/10.1007/s13042-021-01277-w
34. B. Yohai, S. Hoyer, J. Hickey, M. P. Brenner, Learning data-driven discretizations for partial differential equations, *Proceed. Nat. Acad. Sci.*, **116** (2019), 201814058. https://doi.org/10.1073/pnas.1814058116
35. M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045
36. E. Kharazmi, Z. Zhang, G. Karniadakis, hp-VPINNs: Variational physics-informed neural networks with domain decomposition, *Comput. Methods Appl. Mech. Eng.*, **374** (2021), 113547. https://doi.org/10.1016/j.cma.2020.113547
37. G. Pang, M. D'Elia, M. Parks, M. Karniadakis, nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator, Algorithms and applications, *J. Comput. Phys.*, **422** (2020), 109760. https://doi.org/10.1016/j.jcp.2020.109760
38. A. Jagtap, G. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *Commun. Comput. Phys.*, **28** (2020), 2002–2041. https://doi.org/10.4208/cicp.oa-2020-0164
39. Q. Zheng, L. Zeng, G. Karniadakis, Physics-informed semantic inpainting: Application to geostatistical modeling, *J. Comput. Phys.*, **419** (2020), 109676. https://doi.org/10.1016/j.jcp.2020.109676
40. L. Yang, X. Meng, G. E. Karniadakis, B-PINNS: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, *J. Comput. Phys.*, **425** (2021), 109913. https://doi.org/10.1016/j.jcp.2020.109913

41. L. Guo, H. Wu, X. Yu, T. Zhou, Monte Carlo PINNS: Deep learning approach for forward and inverse problems involving high dimensional fractional partial differential equations, *Comput. Methods Appl. Mech. Eng.*, **400** (2022), 115523. https://doi.org/10.1016/j.cma.2022.115523
42. J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375** (2018), 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029
43. W. E, B. Yu, The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.*, **6** (2018), 1–12. https://link.springer.com/article/10.1007/s40304-018-0127-z
44. L. Lyu, Z. Zhang, M. Chen, J. Chen, MIM: A deep mixed residual method for solving high-order partial differential equations, *J. Comput. Phys.*, **452** (2022), 110930. https://doi.org/10.1016/j.jcp.2021.110930
45. L. Yang, X. Meng, G. E. Karniadakis, B-PINNS: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, *J. Comput. Phys.*, **425** (2021), 109913. https://doi.org/10.1016/j.jcp.2020.109913
46. J. Hou, Y. Li, S. Ying, Enhancing PINNs for solving PDEs via adaptive collocation point movement and adaptive loss weighting, *Nonlinear Dyn.*, **111** (2023), 15233–15261. https://link.springer.com/article/10.1007/s11071-023-08654-w
47. S. N. Lin, Y. Chen, A two-stage physics-informed neural network method based on conserved quantities and applications in localized wave solutions, *J. Comput. Phys.*, **457** (2022), 111053. https://doi.org/10.1016/j.jcp.2022.111053
48. J. C. Pu, Y. Chen, Complex dynamics on the one-dimensional quantum droplets via time piecewise PINNs, *Phys. D*, **454** (2023), 133851. https://doi.org/10.1016/j.physd.2023.133851
49. S. N. Lin, Y. Chen, Physics-informed neural network methods based on Miura transformations and discovery of new localized wave solutions, *Phys. D*, **445** (2023), 133629. https://doi.org/10.1016/j.physd.2022.133629
50. J. C. Pu, Y. Chen, Data-driven vector localized waves and parameters discovery for Manakov system using deep learning approach, *Chaos Solitons Fractals*, **160** (2022), 112182. https://doi.org/10.1016/j.chaos.2022.112182
51. G. Pang, L. Lu, G. Karniadakis, fPINNs: Fractional physics informed neural networks, SIAM J. *Sci. Comput*, **41** (2019), 2603–2626. https://doi.org/10.1137/18M1229845
52. F. Rostami, A. Jafarian, A new artificial neural network structure for solving high-order linear fractional differential equations, *Int. J. Comput. Math.*, **95** (2018), 528–539. https://doi.org/10.1080/00207160.2017.1291932
53. S. P. Wang, H. Zhang, X. Jiang, Fractional physics-informed neural networks for time-fractional phase field models, *Nonlinear Dyn.*, **110** (2022), 2715–2739. https://link.springer.com/article/10.1007/s11071-022-07746-3
54. Z. Li, Y. Yan, N. J. Ford, Error estimates of a high order numerical method for solving linear fractional differential equations, *Appl. Numer. Math.*, **114** (2017), 201–220. https://doi.org/10.1016/j.apnum.2016.04.0107
55. K. Diethelm, An algorithm for the numerical solution of differential equations of fractional order, *Electron. Trans. Numer. Anal.*, **5** (1997), 1–6.
56. M. Chen, R. Niu, W. Zheng, Adaptive multi-scale neural network with Resnet blocks for solving partial differential equations, *Nonlinear Dyn.*, **111** (2023), 6499–6518. https://link.springer.com/article/10.1007/s11071-022-08161-4

57. Z. Miao, Y. Chen, VC-PINN: Variable coefficient physics-informed neural network for forward and inverse problems of PDEs with variable coefficient, *Phys. D*, **456** (2023), 133945. https://doi.org/10.1016/j.physd.2023.133945
58. M. Abu-Shady, M. K. A. Kaabar, A generalized definition of fractional derivative with applications, *Math. Probl. Eng.*, (2021). https://doi.org/10.1155/2021/9444803

AIMS Press