Journal of Network Computing and Applications (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/jnca.2025.100110
ISSN 2371-9214 Vol. 10 Num. 1

An Adaptive Physics-Informed Neural Network by Sampling Alternately from Time and Space for Solving Spatiotemporal PDE

Hanya Wen[1], Jin Su[1,2,3,*]

[1]School of Science, Xi'an Polytechnic University, Xi'an, Shaanxi, 710048, China
[2]Key Laboratory of Functional Textile Material and Product, Ministry of Education, Xi'an Polytechnic University, Xi'an, 710048, Shaanxi, China
[3]Xi'an International Science and Technology Cooperation Base for Big Data Analysis and Algorithms, Xi'an, Shaanxi, 710048, China

Keywords: Physical Information Neural Network; Sampling Alternately from Time and Space; Adaptive Sampling

Abstract: In the past several years, Physics-Informed Neural Network (PINN) for solving partial differential equations (PDE) has an advance development, however, under the traditional sampling method, it is difficult for the network to accurately capture the changes of the solution in complex areas. For this reason, we propose a spatio-temporal collaborative sampling strategy of PINN for solving PDE, to optimize the layout of omni-directional sampling points. In our method, the time interval is first subdivided into multiple sub-intervals, and local optimization sampling is performed for each sub-interval. The entire procedures of sampling will be pulled out alternatively in two stages in each sub-interval: first, in the aspect of spatial adaptive sampling, we adopt a dynamic resampling strategy based on the dynamical training error of neural network, which can sensitively identify the changing region of the solution and automatically increase the sampling density in the region with dramatic changes to capture more details; Secondly, time dimension sampling was performed similarly. Numerical tests on the Schrödinger and heat-conduction PDE show over 40% faster convergence and a reduction in relative L2 compared to traditional PINN. This work presents a new approach for efficiently solving complex PDE with PINN.

1. Introduction

With the rapid development of deep learning technology, the PINN shows the potential to improve an accuracy in solving PDE [1][2]. Unlike traditional numerical methods like the finite element method, PINN combine the physical constraints of PDE into the loss function of the neural network, which could enable the model to directly estimate the solution of PDE in the continuous domain without the need for cumbersome grid partitioning and discretization. Waheed et al.[3] put forward the Kronecker neural network to address the issue of spectrum deviation during wave field computation; Roy et al.[4] proposed an Adaptive PINN for simulating transient diffusion problems

in heterogeneous media.

In terms of algorithm and theoretical research, McClenny et al.[5] developed an adaptive PINN to improve the theoretical system; Yu et al.[6] improved PINN to deal with forward and inverse PDE problems by gradient enhancement. In applications, Bai et al.[7] used the improved PINN in the financial field; Xiang et al.[8] employed adaptive loss balancing to optimize PINN training. Mattey et al.[9] proposed a sequential training method to solve a specific equation, Dong et al.[10] put to use local extreme learning machine and domain decomposition to solve PDE; Yan et al.[11] applied the boundary element method combined with nonlinear polarization curve modeling in electrochemistry. In addition, physical information machine learning is also utilized to head impact detection[12]. Zhang et al.[13]proposed a self-supervised mesh generation method called MeshKINN based on PINN and KAN. Cui et al.[14]proposed a method that combines PINN and LSTM to enhance the predictive ability of learning models by utilizing the physical properties of batteries. Zhang et al.[15]used a physical information-based neural network method with embedded conservation laws to solve the Hirota equation. Cao et al.[16]designed a physics informed interactive learning convolutional recursive network for spatiotemporal dynamics. Overall, these studies demonstrate the potential of PINN in multiple interdisciplinary fields.

When solving PDE using PINN, sampling is a key factor that connects physical laws with neural networks and affects the accuracy of the solution. Its density, distribution, and strategy directly determine whether the model can accurately capture the characteristics of the solution. PINN have met with a series of challenges in practical applications, such as the solution accuracy and convergence speed highly depending on data point distribution[17][18].

Under the traditional sampling strategy, it is difficult for the network to accurately capture the changes of the solution in complex areas, resulting in large local residuals[19][20]. Pienaar et al. used Hilbert curve[21]for sampling optimization, and Nabian et al.[22]effectively adjusted the sampling points of the PINN through importance sampling, and achieved effective training of the PINN through importance sampling. Luo et al.[23] introduced Residual-based Smote (RSmote), a local adaptive sampling technique that integrates imbalanced learning's oversampling to target high-residual regions in PINNs. This technique achieves comparable accuracy with reduced memory usage, especially for high-dimensional PDEs. Fang et al.[24] developed a PINN based on a mixture of Cartesian grid sampling and Latin hypercube sampling to solve forward and backward modified diffusion equations. This neural network solver can be generalized to other PDE. Anticev et al.[25]proposed SGM-PINN, a graph-based importance sampling framework to improve the training efficacy of PINN on parameterized problems, biasing sampling towards more important clusters allows smaller mini-batches and training datasets, improving training speed and accuracy. Guo et al.[26] proposed a temporal causality-based adaptive sampling method that dynamically determines sampling ratios by integrating PDE residual and temporal causality, addressing the underutilization of temporal information in sampling for time-dependent PDEs.

In order to solve the problem of low training efficiency caused by insufficient distribution of data points, we propose a new spatiotemporal collaborative adaptive resampling method to optimize the data distribution of PINN in PDE solutions. We divide the time domain into multiple consecutive subintervals and alternate between spatial resampling and temporal resampling within each sub interval. During the spatial resampling stage, a fixed time point is used to increase the sampling density in areas with significant changes in the solution based on the residual probability density function; During the time resampling process, fixed spatial points are used, and sampling points are dynamically increased during periods of rapid physical quantity changes. We combine spatiotemporal co sampling to adaptively adjust the distribution of sampling points and avoid error accumulation. Unlike static importance sampling, our method can dynamically adjust the sampling strategy based on the current training state of the model, effectively capturing the local variation

characteristics of the solution. Finally, numerical experiments were conducted on typical PDE tasks such as Nonlinear Schrödinger equation and Heat Conduction equation.

This paper is organized as follows. Section 2 mainly introduces the adaptive sampling method and our method, as well as the network architecture used. The third section is the experimental part, where we compared the effectiveness of our innovative method with the traditional PINN method, and demonstrated the effectiveness of the spatiotemporal resampling method through several classic numerical experiments. Section 4 is a summary of this article and discusses future prospects.

2. Methods

In this section, we introduce the adaptive sampling strategy, the innovative spatiotemporal resampling method proposed of this paper, and its corresponding initial condition processing scheme. It also briefly reviews the core model PINN, laying a theoretical foundation for the subsequent research content.

2.1 PINN

PINN have opened up a new avenue in solving PDE. As proposed by Lu et al.[27], the core of the PINN method is to integrate the physical constraint equations within the PDE into the neural network loss function.

Consider the general form of spatiotemporal PDE:

$$\begin{aligned} \mathcal{N}_{x,t}[u(x,t)] &= f(x,t), x \in \Omega, t \in (0,T], \\ B_{x,t}[u(x,t)] &= g(x,t), x \in \partial\Omega, t \in (0,T], \\ u(x,0) &= h(x), x \in \bar{\Omega}. \end{aligned} \tag{1}$$

Where $u(x,t)$ is the solution of the equation, $\mathcal{N}[\cdot]$ is a nonlinear differential operator, x is a space variable, and t is a time variable. $B[\cdot]$ represents the boundary operator, acting on $u(x,t)$, and $x \in \partial\Omega$ indicates that x takes the boundary points of region Ω. The initial conditions specify the value of the unknown function u in region Ω at time $t=0$, where $h(x)$ is a given function with respect to spatial variables. $f(x,t)$ is the given source term that determines the internal behavior of the equation, while $g(x,t)$ characterizes the physical conditions or constraints on the boundary.

PDE residuals: The neural network prediction value is substituted into the PDE, and the residual is defined as the difference from 0:

$$\mathcal{L}_r(w) = \frac{1}{N_r}\sum_{i=1}^{N_r} | \mathcal{N}_{x,t}[u(x_r^i, t_r^i; w)] - f(x_r^i, t_r^i) |^2 \tag{2}$$

Boundary condition loss: The error between the model's predicted solution and the actual boundary condition is calculated as:

$$\mathcal{L}_b(w) = \frac{1}{N_b}\sum_{i=1}^{N_b} | \mathcal{B}_{x,t}[u(x_b^i, t_b^i; w)] - g(x_b^i, t_b^i) |^2 \tag{3}$$

Initial condition loss: The error between the model's predicted solution and the actual initial condition is calculated as:

$$\mathcal{L}_0(w) = \frac{1}{N_0}\sum_{i=1}^{N_0} | u(x_0^i, 0; w) - h(x_0^i) |^2 \tag{4}$$

Total loss:

$$L(\omega) = \lambda_r L_r(\omega) + \lambda_b L_b(\omega) + \lambda_0 L_0(\omega) \tag{5}$$

In the relevant settings for solving PDE with PINN, λ_r, λ_b, and λ_0 are weight hyperparameters of the loss function. They are used to control the proportions of the residual loss $L_r(\omega)$, the boundary loss $L_b(\omega)$, and the initial condition loss $L_0(\omega)$ in the total loss, respectively. The larger the value, the more attention is paid to the corresponding constraint fitting. N_r, N_b and N_0 are parameters for the number of training data points. The principle of the PINN method is to construct a neural network to approximate the solution of the partial differential equation. It then employs automatic differentiation to calculate the derivative of the solution. Next, it optimizes the errors of the PDE, initial conditions, and boundary conditions as a loss function. In this manner, it can approximate the numerical solution of the PDE without the need for grid discretization.

The selection of sampling points is crucial not only in PINN but also in many other fields [28][29]. In the context of solving pdes with PINN, Lu et al.[30] proposed RAR, an adaptive sampling method to boost training efficiency. It selects new training points via PDE residuals in PINN training, improving accuracy and convergence. Practically, it samples randomly across the domain and picks the largest residual point, possibly overfocusing on local high-residual regions, ignoring others, and limiting global optimal search.To address this, researchers proposed RAD. It samples via residual probability density function, with each point's probability proportional to its PDE residual. This prioritizes high-error regions for encrypted sampling, reducing low-residual invalid computations and improving model accuracy and efficiency.

2.2 Spatial-temporal co-sampling method

To better capture spatio-temporal features and avoid non-convergence when solving PDE in specific spatio-temporal scenarios, we have made an innovative extension to the PINN: after the model iterates a fixed number of times, the time interval is divided into specified sub-intervals, and each sub-interval sequentially undergoes a two-stage optimized sampling process of "fixing time to sample space → fixing space to sample time".

Algorithm 1: Spatio-Temporal Sampling-Optimized Adaptive PINN Algorithm
Require:Time interval: $\mathcal{T} = [t_{\text{start}}, t_{\text{end}}]$ split into N sub-intervals T_i of length $\Delta t = \frac{t_{\text{end}} - t_{\text{start}}}{N}$
Ensure: Spatio-temporal sampling →Update training set →Optimize Adaptive PINN
repeat
1: Temporally-fixed, spatially-adaptive sampling:
2: Randomly select a set of time points $\mathcal{T}_{i,s} \subset \mathcal{T}_i$ from the sub – interval T_i
3: $\theta \leftarrow \text{Train}\left(\text{Adapt PINN}(\theta), \mathcal{D} \cup \text{Resample}\left(\mathcal{S}_t, \frac{\text{Loss}_{PDE}(\text{PINN}(\theta), \mathcal{S}_t)}{\sum_{(t,x)\in\mathcal{S}_t} \text{Loss}_{PDE}(\text{PINN}(\theta),(t,x))}\right)\right)$
4: Spatially-fixed, temporally-adaptive sampling:
5: Randomly select a set of spatial points X_s from the sub-interval T_i
6: $\theta \leftarrow \text{Train}\left(\text{PINN}(\theta), \mathcal{D} \cup \text{Resample}\left(\mathcal{S}_x, \frac{\text{Loss}_{PDE}(\text{PINN}(\theta), \mathcal{S}_x)}{\sum_{(t,x)\in\mathcal{S}_x} \text{Loss}_{PDE}(\text{PINN}(\theta),(t,x))}\right)\right)$
Until: The training residual on the updated training set is less than the predefined value.

Although the RAD method can improve solution efficiency, it still has some limitations in terms of the time dimension. In addition, the application of the RAD method is mainly focused on one-dimensional problems, and two-dimensional PDEs have not been solved, which limits its applicability in complex scenarios.In order to overcome these limitations, we propose an adaptive physics-informed neural network with alternate sampling in time and space, called ATS-PINN. The core of our method for solving spatio-temporal PDEs is spatio-temporal collaborative sampling, which aims to optimize the layout of omni-directional sampling points (Algorithm 1). The algorithm first partitions the temporal domain $\mathcal{T} = [t_{\text{start}}, t_{\text{end}}]$ into N sub-intervals T_i with $\Delta t = \frac{t_{\text{end}} - t_{\text{start}}}{N}$. For each sub-interval T_i, a set of temporal points $\mathcal{T}_{i,s}$ is randomly selected for time-fixed, space-adaptive sampling. The model parameters θ are updated through the Train($\cdot$) function using a combination of the base dataset D and resampled points. Similarly, spatial points X_s are sampled from spatial sub-intervals, and temporal points are adaptively resampled using S_{tx} to form the final spatio-temporal

•Time domain fixed, spatial adaptive sampling

In each sub-interval, several time points t_k are randomly selected, and the spatial discrete point $\{x_i\}_{i=1}^{N}$ is sampled at these time points to construct the sample point set $\{(x_i, t^*)\}_{i=1}^{N}$.Then, the current PINN model is used to predict $\hat{u}(x,t;\theta)$, and the corresponding residual $R(x_i, t^*)$ is calculated.

$$R(x,t) = \mathcal{N}_{x,t}[\hat{u}(x,t)] - f(x,t) \tag{6}$$

Based on the residual distribution, a normalized probability density function is constructed, and a new set of spatial points $\{x_i^{'}\}_{i=1}^{M}$ is resampled to generate a new training sample set. Finally, the training data set $\mathcal{S}_{\text{updated}} = \mathcal{S}_{\text{old}} \cup \mathcal{S}_{\text{new}}$ is updated and the training is continued to optimize the network parameter θ.

The formula of probability density function is:

$$P(x_i \mid t^*) = \frac{\mid R(x_i, t^*) \mid}{\sum_{j=1}^{N} \mid R(x_j, t^*) \mid} \tag{7}$$

•Space fixed, time domain adaptive sampling

After the time domain of the sub-interval is fixed and the spatial adaptive sampling is completed, several fixed spatial points x^* are randomly selected, and time sampling is performed on these spatial positions to generate a time point set $\{t_i\}_{i=1}^{N}$, thereby constructing a sample point set $\{(x^*, t_i)\}_{i=1}^{N}$. Then, the current PINN model is used to predict $\hat{u}(x,t;\theta)$ and calculate the corresponding residuals $L_r(\theta)$.

Based on the residual distribution, the normalized probability density function is constructed, and the time point is resampled accordingly to generate a new training sample $\{(x^*, t_i)\}_{i=1}^{N}$, update the data set $S_{\text{new}} = \{(x^*, t_i^{'})\}_{i=1}^{M}$ and continue training to optimize the network parameter θ.(Figure1)

PINN training proceed son new sampling points, iteratively updating network parameters θ using a loss function:

$$L(\theta):\ L(\theta)=\lambda_r L_r(\theta)+\lambda_b L_b(\theta)+\lambda_0 L_0(\theta) \tag{8}$$

(In the $L(\theta)$ expression, $L_r(\theta)$, $L_b(\theta)$ and $L_0(\theta)$ are defined as in Section 2.1). The PINN processes the boundary and initial conditions of PDE by explicitly adding their corresponding loss terms to the total loss function.

If termination conditions (max iterations or residual convergence) are met, training stops. Otherwise, alternate spatial-time point optimization continues for better model accuracy and generalization.

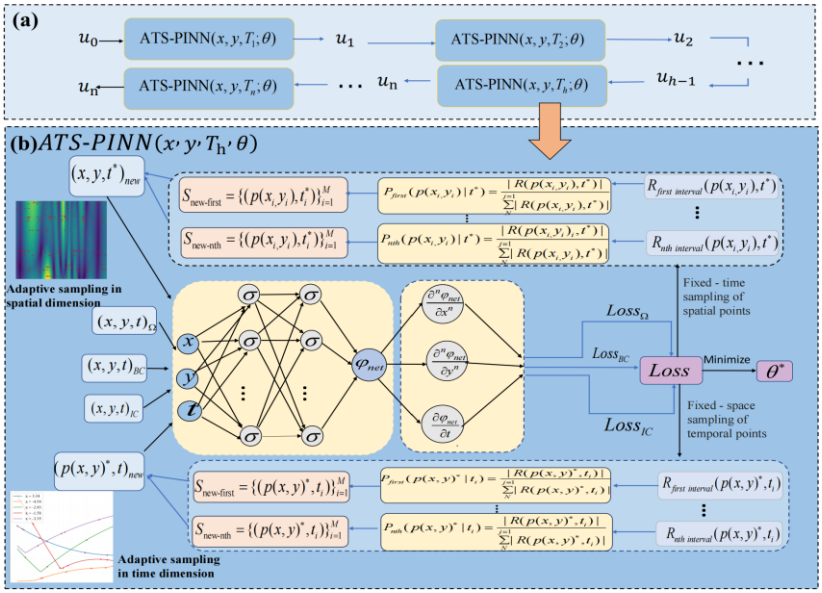

Figure 1: The network architecture of ATS-PINN.

(a) ATS-PINN integration architecture diagram. (b) ATS-PINN internal detailed mechanism diagram. This figure reflects the spatiotemporal adaptive sampling optimization process, and includes a two-stage progressive sampling strategy and corresponding visual sampling graph.

Figure 1 depicts our iterative process based on the Adaptive PINN, which takes spatiotemporal coordinates (x,y,T) and model parameters θ as inputs. During training, spatial and temporal adaptive sampling are executed alternately (the interaction logic is shown in Figure 1b). Moreover, the sampling of each subinterval inherits the state from the previous one (for example, u_0 is processed by the network to obtain u_1, and subsequent iterations continue with the parameters θ and output states), enabling the transfer of spatiotemporal information across different scales. In the core process, the residual formula $R(x,t)=N_{x,t}[\hat{u}(x,t)]-f(x,t)$ is first used to quantify the deviation. Here, $\hat{u}(x,t;\theta)$ represents the prediction of the PINN at the spatiotemporal point (x,t), $N_{x,t}$ is the equation operator acting on $\hat{u}$ (such as the differential operator in PDE), and $f(x,t)$ is the true physical source term. The difference among them, $R(x,t)$, reflects the extent to which the prediction deviates from the real physical laws.

2.3 The initial conditions of sub- intervals

Our innovation focuses on the strategy of partition-based initial condition transfer. The core idea is to divide the time interval into multiple sub-intervals after a fixed number of model iterations. For each sub-interval, we perform a two-stage optimized sampling process: first, fixing time to sample

space, and then fixing space to sample time. This iterative sampling process is designed to better adapt to the spatio-temporal characteristics of the problem. Mathematically, consider the following equation system:

$$\begin{cases} u(x,0) = h(x) \\ u^{(k)}(t_{k-1}) = u(t_{k-1}) \end{cases} \tag{9}$$

The first equation $u(x,0) = h(x)$ provides the initial condition for the PDE defined within the first sub-interval. In the second equation $u^{(k)}(t_{k-1}) = u(t_{k-1})$, the solution state $u(t_{k-1})$ from the k-1th sub-interval is used as the initial condition for the k-th sub-interval. In this way, the model can capture spatio-temporal features sequentially and hierarchically.

3. Results

In this section, four experiments were conducted to compare with the PINN method. For the treatment of initial and boundary conditions, sampling points were selected from these conditions and incorporated into the loss function. The fundamental model employed in this paper is PINN, and all related code implementations are based on the PyTorch framework. The process and results of the test are as follows.

3.1 Heat conduction equation

This section will elaborate on the numerical experiments carried out on the task of solving heat conduction equations with different orders in spatio-temporal co-sampling, which is of great significance for understanding the performance of spatio-temporal co-sampling in dealing with such problems.

3.1.1 One-dimensional heat conduction equation

We first consider the following one-dimensional heat conduction equation problem :

$$\begin{gathered} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + e^{-t}(\sin(\pi x) - \pi^2 \sin(\pi x)) = 0, x \in (-1,1), t \in (0,1), \\ u(x,0) = \sin(\pi x), x \in [-1,1], \\ u(-1,t) = 0, u(1,t) = 0, t \in [0,1]. \end{gathered} \tag{10}$$

First, training and test data are generated based on the equation and initial conditions, with initial random sampling to ensure diversity and representativeness. A fully connected neural network is built as the solution model: the input layer has 2 neurons (for spatial and temporal variables); the hidden layer consists of 3 layers with 32 neurons each; the output layer has 1 neuron (outputting the equation solution). The hidden layers use tanh as the activation function and Glorot uniform for weight initialization. The network output is transformed to better meet boundary conditions and problem characteristics.

In the initial training, the Adam optimizer is used to train the model, and the learning rate is set to 10^{-3}.After obtaining the initial solution, the interval training is performed: the time interval is divided into multiple sub-intervals, and two sampling strategies are used in each sub-interval. Firstly, the time space is fixed, and the time points are randomly selected in the sub-interval. Then, the space points are sampled at a fixed set of time points $\{t_1, t_2, ..., t_m\}$, and the residual value of each sample point is calculated:

$$\text{Residual}(x_i, t_j) = \left| \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + e^{-t_j} (\sin(\pi x_i) - \pi^2 \sin(\pi x_i)) \right| \tag{11}$$

The probability density function is generated by the normalized residual: $P(x,t) = \frac{R(x,t)}{\sum R(x,t)}$, and then the spatial points are resampled according to the probability density function, and the regions with larger residuals are preferentially selected to update the training set. Local optimization is performed on the sampling points at a fixed time point to improve the spatial resolution.

Similarly, the fixed space points are optimized for the time points, and the time point sample $\{t_1, t_2, ..., t_m\}$ is generated on a fixed set of space points $\{x_1, x_2, ..., x_n\}$. Repeat the above steps until the entire time interval is traversed. At the same time, the initial conditions of the subsequent sub-intervals strictly follow the end state of the previous sub-interval to ensure the continuity of the sampling process and the inheritance of the data. In Figure 2, we show the comparison between the exact solution and the predicted solution obtained by using PINN, RAR and ATS-PINN methods at different times t = 0.5, 0.75. The results indicate that the predictions made by the ATS-PINN method are in good agreement with the evolution depicted by the exact solution.
The fluctuation trend of the surface and the numerical variation range of different regions are highly consistent, indicating that the model successfully captures the main modes and key features of the PDE solution when simulating the solution of the partial differential equation.

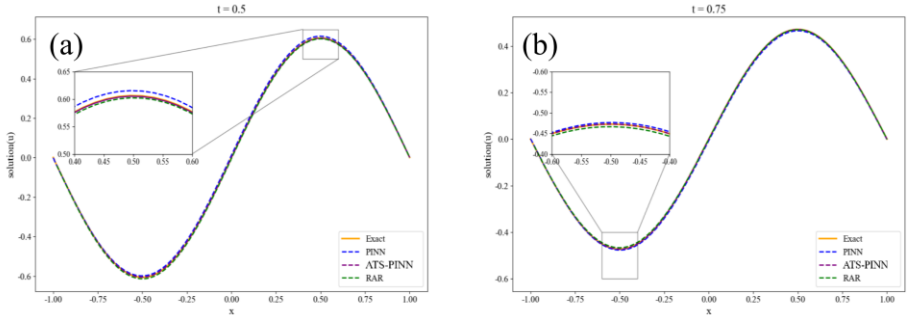

Figure 2: Comparison of exact, PINN,RAR and ATS-PINN methods of the One-dimensional heat conduction equationat different times.

Figure 3 compares 3D surface graphs of partial differential equation results. The left shows the exact solution distribution over space (x) and time (t); the right shows the spatio-temporal sampling model's predicted solution distribution in the same dimension. Both use color coding to represent solution values, facilitating intuitive evaluation of how close the model's prediction is to the exact solution.

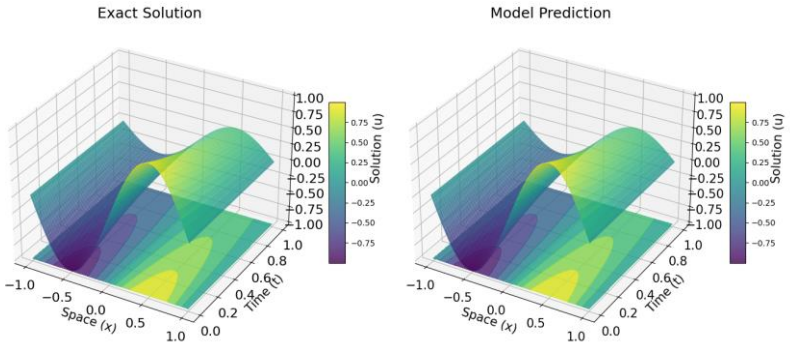

Figure 3: Exact solution (left) and predictive solution(right) of one-dimensional heat conduction PDE.

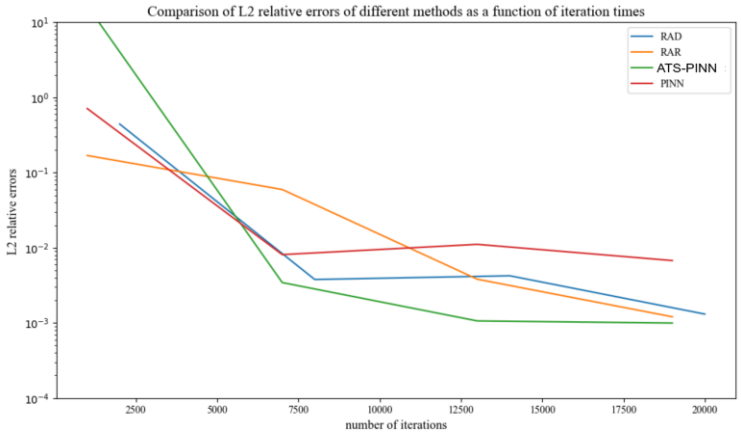

Figure 4: The loss curve function curve of PINN, RAD, RAR, and ATS-PINN methods for optimizing iterations.

Figure 4 shows the number of iterations and the loss functions of the four methods (PINN, RAD, RAR, and ATS-PINN). It can be seen that the optimization effect of the ATS-PINN method is superior to other methods.

Next, we visually analyze the sampling points. The residual heatmap shows residual distribution across spatial points when solving the heat conduction equation: warm colors (yellow) indicate high residuals (PINN prediction errors), while cool tones (blue) mean low residuals . Red dots in Figure 5 represent points reselected via residual-driven sampling. From the visualization results of sampling points, it can be observed that there are more samples in areas with large residuals, and samples are also taken in areas with small residuals.

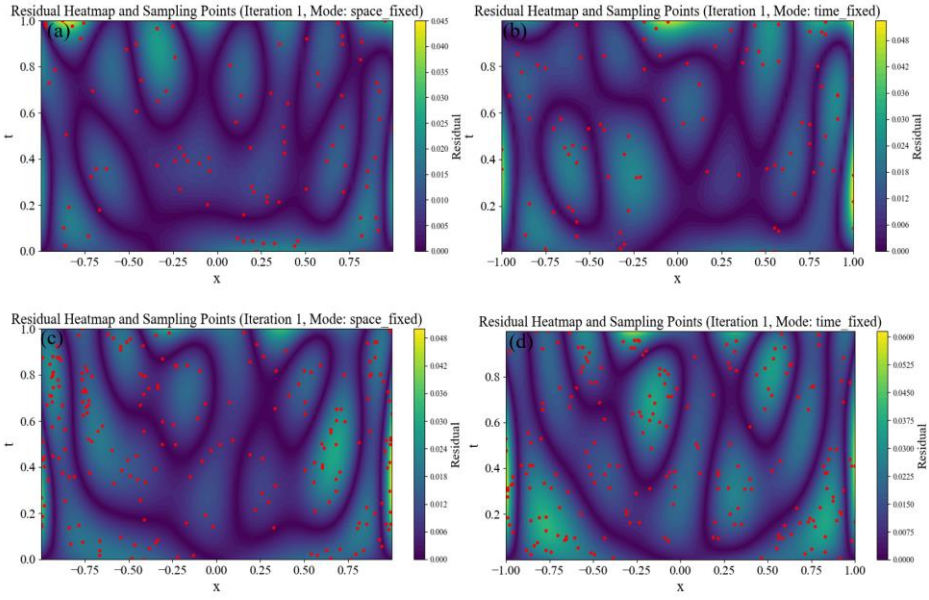

Figure 5: Sampling diagram of the heat conduction equation.

(a) Sampling points (marked by red dots): Fix x , sample t . (b) Sampling points (small number): Fix t , sample x . (c) Sampling points (large number): Fix x , sample t . (d) Sampling points (large number): Fix t , sample x .Red indicates sampling points.

3.1.2 Two-dimensional heat conduction equation

The second case study is the two-dimensional heat conduction equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta u + 2\pi^2 e^{-t}\sin(\pi x)\sin(\pi y), \\ u(x,y,0) &= \sin(\pi x)\sin(\pi y), \\ u(x,y,t) &= 0. \end{aligned} \tag{12}$$

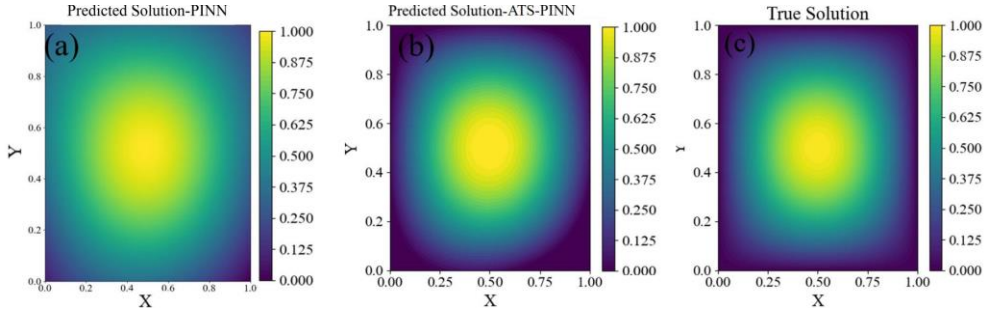

Figure 6: PINN model predicted solution (a), and ATS-PINN model predicted solution (b) and accurate solution (c) of two-dimensional heat conduction PDE.

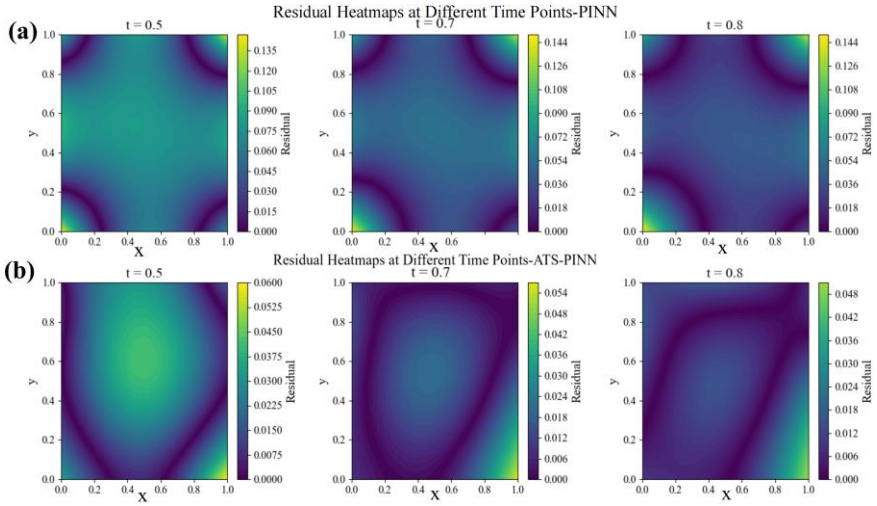

Figure 7: Example of Two-dimensional heat conduction equation.

(a) The predicted absolute error of PINN. (b) The predicted absolute error of ATS-PINN.

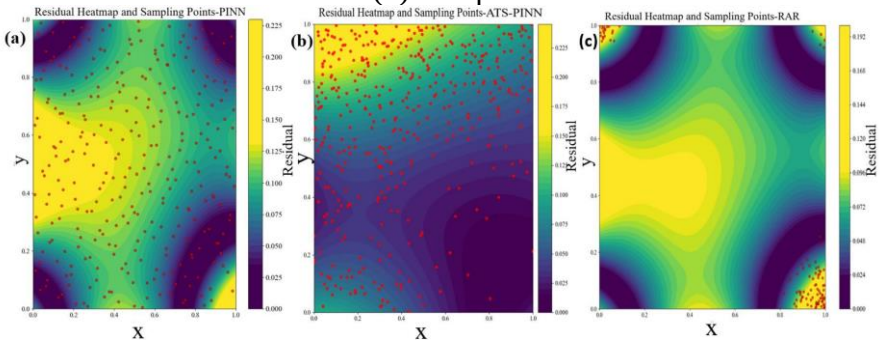

Figure 8: Sampling point distribution map.

(a) Distribution of sampling points on PINN method (b) Distribution of sampling points on ATS-PINN method (c) Distribution of sampling points on RAR method.

The model The experiment's model still uses FNN as its main structure. Visualization results (Figure 6) clearly show high consistency between the model's predictions and real solutions across the entire spatial domain, with accurate capture of key distribution trends and features—verifying

the model's effectiveness in solving time-varying PDE problems. Figure 7 shows ATS-PINN has smaller absolute errors and more accurate predictions. To better observe data distribution, sampling points (red scatter points across the area) of PINN, ATS-PINN, and RAR were visualized with residual heatmaps (colors indicating residual magnitudes). Results (Figure 8) show ATS-PINN's sampling better captures key regional info and avoids invalid sampling compared to other methods.

3.2 Nonlinear Schrödinger Equation

In this study, we consider the following Nonlinear Schrödinger (NLS) equation:

$$i\frac{\partial h}{\partial t}+\frac{1}{2}\frac{\partial^2 h}{\partial x^2}+|h|^2 h=0,\ x\in[-5,5],\quad t\in\left[0,\frac{\pi}{2}\right] \tag{13}$$

Where $h=h_{real}+ih_{imag}$ is a complex-valued function of the spatial variable x and the temporal variable t.

The NLS equation can be split into its real and imaginary parts:

$$\frac{\partial h_{imag}}{\partial t}+\frac{1}{2}\frac{\partial^2 h_{real}}{\partial x^2}+(h_{real}^2+h_{imag}^2)h_{real}=0 \tag{14}$$

$$-\frac{\partial h_{real}}{\partial t}+\frac{1}{2}\frac{\partial^2 h_{imag}}{\partial x^2}+(h_{real}^2+h_{imag}^2)h_{imag}=0 \tag{15}$$

Periodic boundary conditions are imposed to ensure the periodicity of $h(x,t)$ in the x-direction. The initial conditions are set as:

$$h_{real}(x,0)=\frac{1}{\cosh(x)},\ h_{imag}(x,0)=0 \tag{16}$$

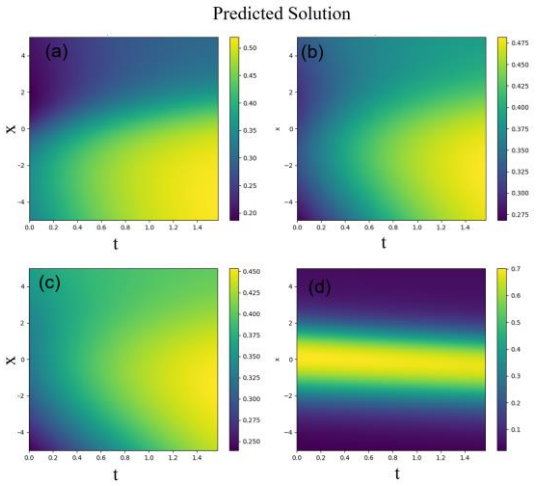

Figure 9: Save the prediction results at a specific number of iterations.

(a) 1000 prediction results; (b) 2000 prediction results; (c) 3000 prediction results; (d)10000 prediction results.

To solve the NLS equation, a feedforward neural network is constructed. The network takes (x,t) as input and outputs (h_{real},h_{imag}) .The architecture of the network consists of two neurons in the input layer, two neurons in each of the middle four layers, and two neurons in the output layer. The activation function used is tanh. During the training process, the learning rate is set to 10^{-4} .The

prediction results(Figure 9) are saved at specific iteration numbers (1000, 2000, 3000, 10000). The L-BFGS optimizer is employed for final training. Finally, by drawing the heat map of $|h(x,t)|$ （$|h(x,t)|=\sqrt{h_{real}^2+h_{imag}^2}$）under specific iterations, the prediction results of the model at different training stages are visually displayed, and the performance of the model is improved through resampling strategy. Our model's prediction results are visualized with PINN's prediction results (Figure 10).

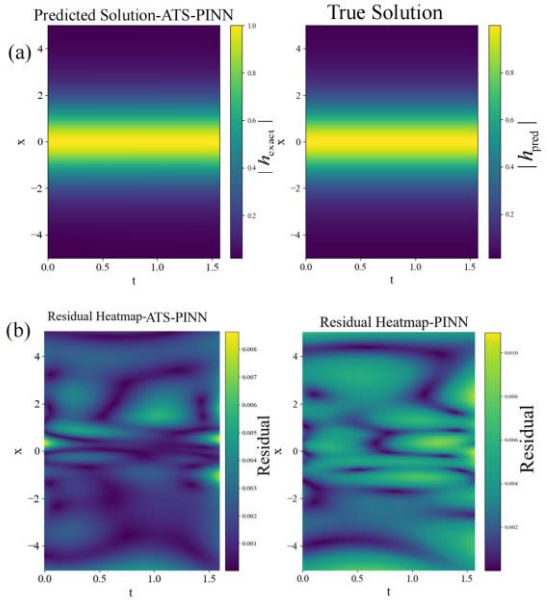

Figure 10: Example of Nonlinear Schrödinger equation.

(a) The exact solution (right) and the predicted solution (left) of the Schrödinger PDE. (b) The predicted residual of PINN method (right) and ATS-PINN method(left) in predicting Nonlinear Schrödinger Equation.

3.3 Klein-Gordon equation

In this study, we consider the following Klein-Gordon (K-G) equation :

$$\frac{\partial^2 u}{\partial t^2}+\alpha\frac{\partial^2 u}{\partial x^2}+\beta u+\gamma u^k=-x\cos(t)+x^2\cos^2(t),\quad x\in[-1,1],\quad t\in[0,10] \tag{17}$$

Where $u(x,t)$ denotes the unknown function with respect to the spatial variable x and the temporal variable t.

The initial conditions are set as:

$$u(x,0)=x,\quad \frac{\partial u}{\partial t}(x,0)=0 \tag{18}$$

Dirichlet boundary conditions are imposed to constrain the solution at the boundaries of the spatial domain:

$$u(-1,t)=-\cos(t),\quad u(1,t)=\cos(t) \tag{19}$$

We also specify the following parameters for the equation:

$$\alpha=-1,\beta=0,\gamma=1,k=2. \tag{20}$$

The reference solution is given by: $u(x,t) = x\cos(t)$

To solve the K-G equation, a feedforward neural network is constructed. For verification and visualization, heatmaps comparing the true solution, our predicted solution, and the PINN model's output are generated to evaluate the overall spatiotemporal fitting performance, while error heatmaps (for both our model and PINN) are plotted to visualize residual distributions at specific time points and resampling locations, thus validating the effectiveness of the adaptive sampling strategy(Figure 11).

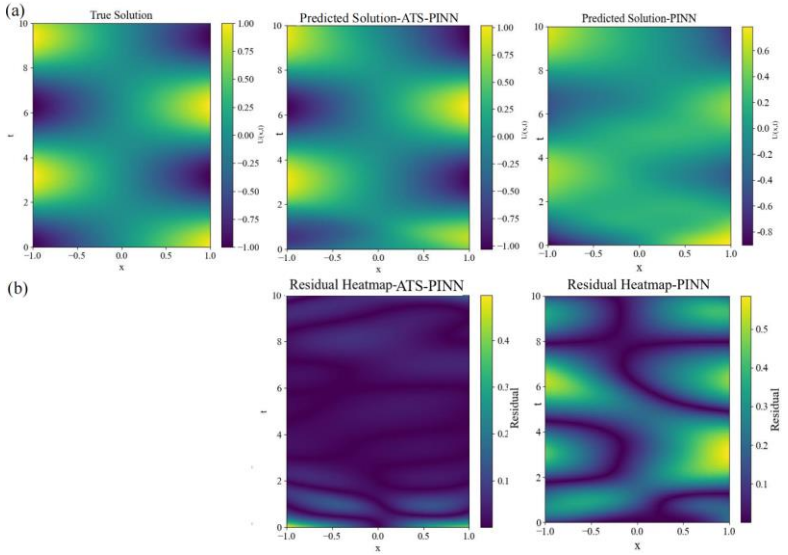

Figure 11: Example of Klein-Gordon equation.

(a) The exact solution of the Klein-Gordon partial differential equation (left), the predicted solution by the ATS-PINN method (middle), and the predicted solution by the PINN method (right). (b) The predicted residuals of the PINN method (right) and the ATS-PINN method (left) in predicting the Klein-Gordon equation.

4. Results

We propose a spatio-temporal collaborative sampling strategy to address the issues of low sampling efficiency and insufficient spatiotemporal continuity in solving PDE using PINN. By dynamically connecting time subintervals, residual-driven adaptive resampling, and incremental training, we ensure spatiotemporal continuity, balance sampling in high and low residual regions, to improve solution accuracy and efficiency. To systematically evaluate the effectiveness of the spatio-temporal collaborative sampling strategy in solving PDE, we conduct a comprehensive study on multiple representative one-dimensional and two-dimensional PDE cases. The research results indicate that the spatio-temporal collaborative sampling strategy can improve the solution accuracy of the PINN. In the comparative experiments of RAR, RAD, and PINN, our method has a smaller absolute error and improved solution performance. In one-dimensional and two-dimensional heat conduction equation experiments, by comparing the model training process under traditional methods and the new strategies, we find that this mechanism can effectively shorten the convergence time of the model, allowing the model to reach the ideal solution state faster. The experimental data shows that as the number of time interval segments gradually increases from 2 to 8, the final loss value of the model shows a continuous downward trend. This phenomenon shows that by reasonably increasing the number of segments in the time interval, the loss function value can be effectively reduced, thereby significantly improving the accuracy of solving the two-

dimensional heat equation. In the Schrödinger equation experiment, our proposed "knowledge inheritance" mechanism and optimized sampling strategy can significantly improve the accuracy of solving equations compared to traditional PINN. Our method not only improves sampling efficiency but also significantly enhances the accuracy of the model solution, resulting in a solution that is closer to the actual value.

We further validate the effectiveness of this strategy in improving solution efficiency through comparative experiments on typical PDE: one-dimensional heat conduction, two-dimensional heat conduction, Klein-Gordon equation, and Schrödinger equation. Whether facing cross-dimensional problems (from 1D to 2D) or different types of equations such as elliptic and parabolic equations, this strategy demonstrates good adaptability and universality.

References

[1] Raissi M, Yazdani A, Karniadakis G E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations [J]. Science, 2020, 367 (6481): 1026-1030.

[2] Raissi M, Perdikaris P, Karniadakis G E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations [J]. Journal of Computational Physics, 2019, 378: 686-707.

[3] Waheed U B. Kronecker neural networks overcome spectral bias for PINN-Based wavefield computation[J]. IEEE Geoscience and Remote Sensing Letters, 2022, 19: 1-5.

[4] Roy S, Sarkar D R, Annavarapu C, et al. Adaptive Interface-PINN (AdaI-PINN) for transient diffusion: Applications to forward and inverse problems in heterogeneous media[J]. Finite Elements in Analysis and Design, 2025, 244: 104305.

[5] McClenny L D, Braga-Neto U M. Self-adaptive physics-informed neural networks [J]. Journal of Computational Physics, 2023, 474: 111722.

[6] Yu J, Lu L, Meng X, et al. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems[J]. Computer Methods in Applied Mechanics and Engineering, 2022, 393: 114823.

[7] Bai Y, Chaolu T, Bilige S. The application of improved physics-informed neural network (IPINN) method in finance[J]. Nonlinear Dynamics, 2022, 107(4): 3655-3667.

[8] Xiang Z, Peng W, Liu X, et al. Self-adaptive loss balanced physics-informed neural networks[J]. Neurocomputing, 2022, 496: 11-34.

[9] Mattey R, Ghosh S. A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations[J]. Computer Methods in Applied Mechanics and Engineering, 2022, 390: 114474.

[10] Dong S, Li Z. Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations[J]. Computer Methods in Applied Mechanics and Engineering, 2021, 387: 114129.

[11] Yan J F, Pakalapati S N R, Nguyen T V, et al. Mathematical modeling of cathodic protection using the boundary element method with a nonlinear polarization curve[J]. Journal of the Electrochemical Society, 1992, 139(7): 1932.

[12] Raymond S J, Cecchi N J, Alizadeh H V, et al. Physics-informed machine learning improves detection of head impacts[J]. Annals of Biomedical Engineering, 2022, 50(11): 1534-1545.

[13] Zhang H, Wang M, Li H, et al. MeshKINN: A self-supervised mesh generation model based on Kolmogorov–Arnold-Informed neural network[J]. Expert Systems with Applications, 2025, 275: 126959.

[14] Cui K, Gao T, Shi D, et al. A Physics-Informed Event-Triggered Learning Approach to Long-Term Spacecraft Li-Ion Battery State-of-Charge Estimation[J]. IEEE Transactions on Industrial Informatics, 2024.

[15] Zhang R, Su J, Feng J. Solution of the Hirota equation using a physics-informed neural network method with embedded conservation laws[J]. Nonlinear Dynamics, 2023, 111(14): 13399-13414.

[16] Cao R, Su J, Feng J, et al. PhyICNet: Physics-informed interactive learning convolutional recurrent network for spatiotemporal dynamics[J]. Electronic Research Archive, 2024, 32(12).

[17] Krishnapriyan A, Gholami A, Zhe S, et al. Characterizing possible failure modes in physics-informed neural networks [J]. Advances in Neural Information Processing Systems, 2021, 34: 26548-26560.

[18] Wu C, Zhu M, Tan Q, et al. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks[J]. Computer Methods in Applied Mechanics and Engineering, 2023, 403: 115671.

[19] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks[J]. SIAM Journal on Scientific Computing, 2021, 43(5): A3055-A3081.

[20] Wang S, Yu X, Perdikaris P. When and why PINN fail to train: A neural tangent kernel perspective[J]. Journal of Computational Physics, 2022, 449: 110768.

[21] Pienaar J J, Boman A S, Malan K M. Hilbert curves for efficient exploratory landscape analysis neighbourhood

sampling[C]//International Conference on the Applications of Evolutionary Computation (Part of EvoStar). Cham: Springer Nature Switzerland, 2024: 293-309.
[22] Nabian M A, Gladstone R J, Meidani H. Efficient training of physics-informed neural networks via importance sampling[J]. Computer-Aided Civil and Infrastructure Engineering, 2021, 36(8): 962-977.
[23] Luo J, Yang Y, Yuan Y, et al. An Imbalanced Learning-based Sampling Method for Physics-informed Neural Networks[J]. arXiv preprint arXiv:2501.11222, 2025.
[24] Fang Q, Mou X, Li S. A physics-informed neural network based on mixed data sampling for solving modified diffusion equations[J]. Scientific Reports, 2023, 13(1): 2491.
[25] Anticev J, Aghdaei A, Cheng W, et al. SGM-PINN: Sampling Graphical Models for Faster Training of Physics-Informed Neural Networks[C]//Proceedings of the 61st ACM/IEEE Design Automation Conference. 2024: 1-6.
[26] Guo J, Wang H, Gu S, et al. TCAS-PINN: Physics-informed neural networks with a novel temporal causality-based adaptive sampling method[J]. Chinese Physics B, 2024, 33(5): 050701.
[27] Lu L, Pestourie R, Yao W, et al. Physics-informed neural networks with hard constraints for inverse design [J]. SIAM Journal on Scientific Computing, 2021, 43 (6): B1105-B1132.
[28] Ouyang X, Chang H, Liu Z, et al. Application of adaptive sampling method in hull form optimization[J]. J. Shanghai Jiaotong Univ, 2022, 56: 937-943.
[29] Zhang S, Chi C, Yao Y, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9759-9768.
[30] Lu L, Meng X, Mao Z, et al. DeepXDE: A deep learning library for solving differential equations[J]. SIAM Review, 2021, 63(1): 208-228.