IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 June 2024, accepted 28 June 2024, date of publication 2 July 2024, date of current version 12 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3422224

METHODS

An Adaptive Sampling Method Based on Expected Improvement Function and Residual Gradient in PINNs

YANBING LIU, LIPING CHEN, JIANWAN DING, AND YU CHEN
School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430070, China

Corresponding author: Jianwan Ding (dingjw@hust.edu.cn)

This work was supported by the Key Research and Development Program of Hubei Province under Grant 2021AAB001.

ABSTRACT In recent years, physics-informed neural networks (PINNs) have developed significantly as a deep learning technology. In analogy to the selection of grid cells in traditional numerical methods, the distribution of sample points used for training PINN can have a greater impact on the solution accuracy. Based on the Residual-based Adaptive Refinement (RAR) algorithm, many improved adaptive sampling algorithms have been proposed. However, these sampling algorithms rely on residuals as the error indicator and focus only on the sample points within the solution domain. Therefore, we introduce a novel adaptive sampling algorithm EI-RAR. This algorithm incorporates a new expected improvement (EI) function, which increases focus on the sample points at the boundaries of the solution domain. Additionally, EI-RAR integrates attention mechanism with a sample point generation algorithm, aimed at reinforcing the connection between newly-added and existing sample points. To increase the accuracy of solving problems with sharp solutions, we build upon the EI-RAR algorithm by incorporating gradient information of the residual values as a criterion for sample point selection, leading to the development of a second adaptive sampling algorithm EI-Grad. We select residual neural network and combine it with adaptive sampling algorithms for a series of numerical experiments, aiming to reduce the phenomenon of gradient vanishing during the training process. These experiments select the Diffusion equation, Burgers' equation, Allen-Cahn equation, and Navier-Stokes equation, respectively. Numerical results indicate that, with the same number of residual points, the EI-RAR algorithm is more precise compared to other sampling methods, and the EI-Grad algorithm can also effectively solve partial differential equations with sharp solutions.

INDEX TERMS Physics-informed neural networks, adaptive sampling, residual gradient, error indicator function.

I. INTRODUCTION

With the development of computer technology in recent decades, neural networks have developed rapidly and are widely used in multiple fields, such as computer vision [1] and natural speech processing [2]. In recent years, the solution-fitting ability of neural networks has attracted a lot of attention. Partial differential equations (PDEs) are derived from the abstraction or mathematical modeling of many physical phenomena. Traditional numerical methods such as the finite difference method [3], finite element method [4], and finite volume method [5], which are commonly employed in various industrial applications, often encounter difficulties, and consume considerable time due to the need for mesh refinement.

In 2019, Raissi et al. [6] applied deep neural networks to cases of solving PDEs and proposed the concept of physical-informed neural networks (PINNs). To adherence to the physical laws of the model, PDEs, along with their initial or boundary conditions, are constructed using automatic differentiation (AD) supported by TensorFlow/PyTorch. Then, these physical laws are embedded into the loss function of the neural network as a residual term to expand the space of admissible solutions. After undergoing this process, the

The associate editor coordinating the review of this manuscript and approving it for publication was Wanqing Zhao.

problem transforms into one of optimizing the loss function. Owing to the mesh-free methodology [7], PINNs provide a more streamlined and flexible approach compared to traditional numerical methods for solving partial differential equations.

As described in summaries by Mojgani et al. [8], PINNs have been employed to tackle a range of PDEs in physical problems in recent years, however, how to use it to attain greater precision in predictions along with higher computational efficiency requires deeper research. Overviewing the published papers, there are four mainstreams enhancing PINN currently: **(1) Change of neural network structure**. PINNs generally use fully connected neural networks [9], Cheng and Zhang [10] proposed Res-PINN using Resnet block, Sun and Feng [11] constructed a second-order neural network structure analogous to Res-PINN for solving parabolic PDEs, Stevens and Colonius [12] used the LSTM structure and incorporated the spatiotemporal dynamics of PDEs to reduce the error in solving time-varying PDEs, Rodriguez-Torrado et al. [13] combined the attention mechanism with RNN to form the physics-informed attention neural network (PIANN); **(2) Domain decomposition**. The domain of PDEs is decomposed into discrete subdomains by Jagtap et al. [14], then the PDEs are tackled in each subdomain using PINN, after that, they [15] proposed a generalized time-domain decomposition framework, which compensates for the defect that the time domain cannot be decomposed; **(3) Loss balancing**. Wang et al. [16], Bischof and Kraus [17] and Xiang et al. [18] balanced the effect of the different terms in the loss function and raised adaptive weights algorithms, Thakur et al. [19] used backward Euler discretization of time derivatives instead of AD and utilized the statistical properties of the data to obtain the weights of the loss terms; **(4) Adaptive sampling**. Since Mao et al. [20] manually selected more sampling points in specified areas using a priori knowledge, adaptive sampling algorithms has been studied greatly. A mainstream method of adaptive sampling algorithms is that continuously include sample points exhibiting large residuals in the training set during the PINN training process, like [21], [22], [23], and [24]. Mao and Meng [25] combined the idea of decomposing the domain by dividing the computational domain into multiple subdomains, added new residuals to the subdomains which contains larger mean value of residuals and larger mean gradient value of latent solution than other subdomains, finally calculated new errors by retraining the PINN until the accuracy requirements are met.

In this paper, we study the effectiveness of the improvement of the adaptive sampling algorithm on the performance of PINN, whose loss function has an additional residual term compared with the general neural network. To significantly improve performance, we need to select a large number of discrete spatiotemporal points within the domain, referred to as residuals, to optimize the loss function in PINN. The option of residuals is similar to grid cells in finite element, so that we can expect that the distribution of residuals plays an important role in PINN. However, many scholars, when conducting research on PINN, mostly use simple uniform sampling methods to generate training set, e.g., uniform random sampling, Latin hypercube sampling (LHS) [26], and the Sobol sequence [27] etc., which ignore the characteristic of the solutions of PDEs, thus leading to poor accuracy when solving certain PDEs. Adaptive sampling algorithms typically use residuals as error indicator function, leading to a sampling process that lacks attention to sample points on boundaries. And these algorithms struggle with solving equations with sharp solutions. To address these sampling issues, we have designed two new adaptive sampling algorithms based on the concept of probabilistic sampling. Our main contributions include: (1) construction of a new expected improvement function to replace the residual value as a new error indicator function, (2) integrating residual gradient with adaptive sampling algorithms, (3) combination of point generation algorithm and attention mechanism for generating sampling point admissible set.

The rest of this paper is organized as follows. Section II provides a summarize of PINN and the Res-PINN framework used in this paper. In Section III, we introduce the adaptive sampling algorithms EI-RAR and EI-Grad based on expected improvement function values and gradient information. In Section IV, examples of the sampling algorithm are given for a variety of PDEs in forward problems, including Diffusion, Burgers', Allen-Cahn, and Navier-Stokes equation. In Section V, we summarize our work.

II. STRUCTURE OF PINNS

PINNs enhance the learning capabilities of the model by integrating constraints based on physical laws. These constraints are derived from the partial differential equations of the relevant physical models, along with their boundary and initial conditions, aiding the model in effective learning and prediction. Consequently, PINNs can overcome the challenges faced by traditional data-driven models due to insufficient data, thereby improving the model's robustness and prediction accuracy.

A. SUMMARIZE OF PINNS

The main idea of PINNs is to employ neural network structures such as deep neural network (DNN), which are trained to be approximators of the solutions of PDEs, to infer the latent solutions in the PDEs. The general form of nonlinear PDEs is written as (1).

$$\begin{aligned} \boldsymbol{u}_t + \mathcal{N}_x[\boldsymbol{u}] &= 0, \boldsymbol{x} \in \Omega, t \in [0, T] \\ u(\boldsymbol{x}, 0) &= h(\boldsymbol{x}), \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}, t) &= g(\boldsymbol{x}, t), x \in \partial\Omega, t \in [0, T] \end{aligned} \tag{1}$$

where $(\boldsymbol{x}, \boldsymbol{t}) \in \Omega \times [0,T]$ represent the spatial and temporal coordinates respectively, and subscripts represent the partial differentiation of different variables, $\mathcal{N}_x$ is a nonlinear spatiotemporal differentiation operator. The latent solution $\boldsymbol{u}(\boldsymbol{x}, t)$ is solved under the constraints of initial conditions $h(\boldsymbol{x})$ and boundary conditions $g(\boldsymbol{x}, t)$.

According to the original definition of PINNs, a fully connected neural network consisting of multiple hidden layers utilized spatiotemporal coordinates $(\boldsymbol{x}, t)$ as the input of the network, which fits the solution $\boldsymbol{u}(\boldsymbol{x}, t)$ of the PDEs and obtains the approximate solution $\hat{\boldsymbol{u}}(\boldsymbol{x}, t; \theta)$, where θ represents a series of network parameters, such as weights and biases.

Before training the parameters θ, the loss function of PINNs needs to be constructed. The trained neural network can fit the latent solution $u(\boldsymbol{x}, t; \theta)$ to solve the PDEs, with the residual is defined as (2).

$$f\left(\boldsymbol{x}, t; \theta\right) := \frac{\partial}{\partial t}\hat{u}\left(\boldsymbol{x}, t; \theta\right) + \mathcal{N}_x\left[\hat{u}\left(\boldsymbol{x}, t; \theta\right)\right] \tag{2}$$

The mathematical laws in the physical model are embedded in the loss function of the neural network as a priori knowledge. Accordingly, the loss function formed by the combination of physical model and neural network is in the following form (3).

$$Loss = \lambda_f \mathcal{L}_{\mathrm{PDE}}\left(\theta; N_f\right) + \lambda_b \mathcal{L}_{\mathrm{BC}}\left(\theta; N_b\right) + \lambda_I \mathcal{L}_{\mathrm{IC}}\left(\theta; N_i\right) \tag{3}$$

where $\mathcal{L}_{\mathrm{PDE}}$, $\mathcal{L}_{\mathrm{BC}}$, $\mathcal{L}_{\mathrm{IC}}$ correspond to the residuals, boundary conditions, and initial conditions calculated by randomly selected points in domain, respectively, and N_f, N_b, N_i denote the quantity of points for residual term, boundary terms, and initial terms.

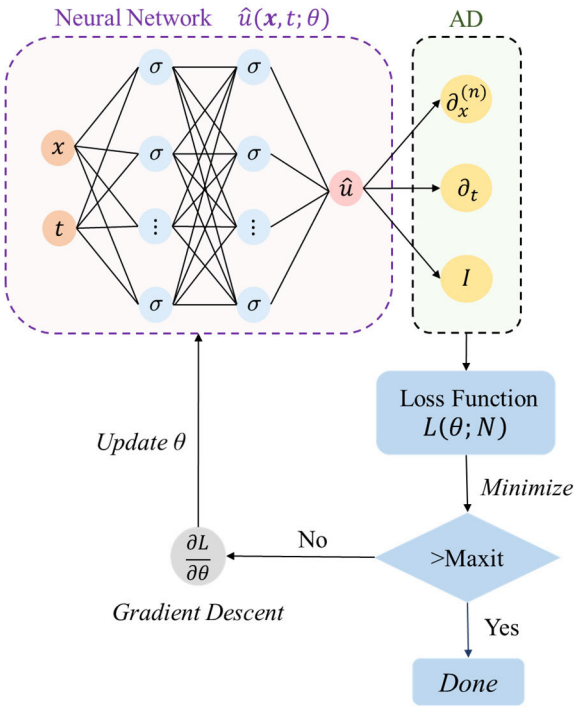

FIGURE 1. Flowchart of the physics-informed neural networks (PINNs).

The above process is the whole procedure of constructing a PINN for the physical model. First, convert an equation solving problem into a loss function minimization problem. Then, adjust the parameters θ through gradient descent algorithm including Adam [28] and LBFGS [29], to minimize the loss function as much as possible. Finally, PINNs fit the physical model with high precision. The framework of PINNs is shown in FIGURE 1.

B. IMPROVEMENT OF NETWORK STRUCTURE

The network structure of PINNs is forward neural network, while this structure is not suitable for all PDEs. He et al. [30] introduce residual physics-informed neural network (Res-PINN), and the residual neural network can retain the information of the previous hidden layer during the training process, preventing from vanishing gradient pathologies to some extent.

In this paper, we will use ResNet to fit PDEs. ResNet architectures simplify depth by embedding identity mappings into the network structure, reducing the complexity associated with direct function fitting. As shown in FIGURE 2, each residual block in ResNet uses shortcut connections to add the output from a previous layer to subsequent layers, not only alleviating the vanishing gradient problem but also preserving information integrity throughout the training process.

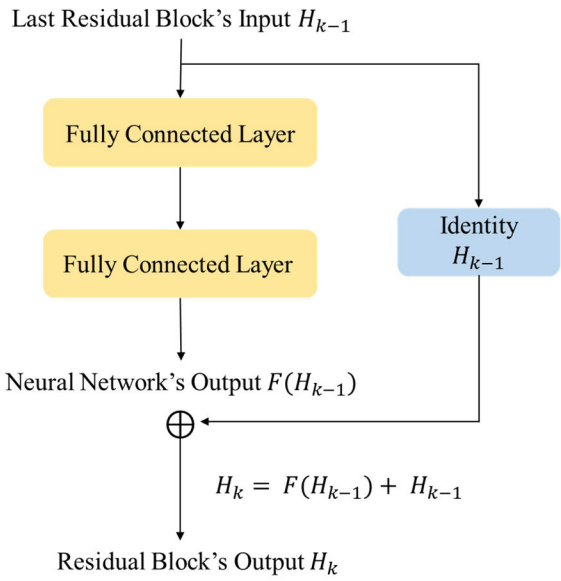

FIGURE 2. Illustration of the residual block.

III. ADAPTIVE SAMPLING ALGORITHM FOR PINN

Solving PDEs with PINN requires sampling from the domain of the physical model. In many PINN studies, a fixed number of residuals at predetermined locations are selected from the domain prior to training, and no additional sample points are added during the training process. The sampling methods frequently used in PINN are equispaced uniform grid, LHS, etc.. These sampling algorithms can provide good solutions in most cases except some specific PDEs. In addition, the training of neural networks is a non-convex problem, and suitable samples can lead to better training performance and

higher accuracy. Therefore, we enhance the performance of PINN by improving the sampling algorithm.

A. THE REVIEW OF ADAPTIVE PROBABILISTIC SAMPLING METHOD

The RAR algorithm aims to improve the model's accuracy by adding new sample points in regions with high residuals. However, this approach may overlook regions that are already well-trained, leading to instability during the training process. Based on the RAR algorithm, many improved adaptive sampling methods have subsequently been proposed.

Wu et al. [24] proposed the probabilistic sampling algorithm RAR-D and built a probability density function based on the residuals of sample points in the domain for resampling, which ensures that regions with large residuals are resampled with higher probability, while sample points with small residuals are also taken into consideration. This algorithm selects a set of sufficiently dense points $\mathbb{S}$ from the solution domain after the initial training of PINN, calculates the residual values $\varepsilon(\boldsymbol{x}, t) = \left|f(\boldsymbol{x}, t; \hat{\theta})\right|$ for all sample points in $\mathbb{S}$, then calculates the probability of each point in $\mathbb{S}$ according to the density function (4).

$$P(\boldsymbol{x}, t) \propto \frac{\varepsilon^k(\boldsymbol{x}, t)}{\mathbb{E}\left[\varepsilon^k(\boldsymbol{x}, t)\right]} + c \tag{4}$$

where $k \geq 0$ and $c \geq 0$ are two hyperparameters. Then calculate the sampling probability of all points according to the probability mass function (5).

$$\tilde{P}(\boldsymbol{x}, t) = \frac{P(\boldsymbol{x}, t)}{\sum_{\boldsymbol{x} \in \mathbb{S}} P(\boldsymbol{x}, t)} \tag{5}$$

Finally, the sample points are randomly selected from the set of sufficiently dense points $\mathbb{S}$ according to previous sampling probability and are merged into the initial set of sample points to retrain PINN.

B. IMPROVEMENT OF ADAPTIVE SAMPLING ALGORITHM

RAR-D shows a significant improvement in accuracy compared to traditional PINNs. However, in the computation process, it employs a hard constraint approach to embed the boundary and initial conditions into the neural network. If the hard constraint approach is not adopted, there would be a decrease in the solution accuracy. This is because the sampling algorithm overly concentrates on the points within the solution domain and neglects the points on the boundary.

To address the aforementioned issue, it is necessary to reselect an appropriate error indicator function. This function plays a crucial role in selecting suitable sample points from the admissible sample set $\mathbb{S}$ to effectively supplement the PINN training dataset. As an error indicator function, it should satisfy two characteristics [31], one is that it should be mathematically equivalent to the error value $e = \left|u - \hat{\boldsymbol{u}}(\boldsymbol{x}, t; \theta)\right|$, and the other is that it should be easy to calculate for each point $(\boldsymbol{x}, t)$. The error indicator function of all current adaptive sampling algorithms utilizes the residual of PDEs, but for some partial differential equation problems, this overreliance on the residual information will cause the internal region in the domain to be over-refined, thus neglecting solutions on the boundary. Therefore, we turn to choose the modified expected improvement (EI) function based on [32] referring to (6) as error indicator function in this paper. This function is a modification of the EI function originally proposed by Jones et al. [33]. for surrogate-based optimization problems. The EI function evaluates the feasibility of new sample points by considering the predicted value and its associated uncertainty.

$$EI(\boldsymbol{x}, t) = s\phi\left(-\frac{y(\boldsymbol{x}, t)}{s(\boldsymbol{x}, t)}\right) = s \cdot \exp\left(-0.5\left(\frac{y^2(\boldsymbol{x}, t)}{s^2(\boldsymbol{x}, t)}\right)\right) \tag{6}$$

where $y(\boldsymbol{x}, t)$ represents the predicted value of PINN at this point $(\boldsymbol{x}, t)$, and $s(\boldsymbol{x}, t)$ represents the corresponding residual of PINN at this point $(\boldsymbol{x}, t)$.

The partial derivative of $EI(\boldsymbol{x}, t)$ is obtained as (7).

$$\begin{aligned} \frac{\partial EI}{\partial s} &= (1 + \frac{y^2}{s^2}) \cdot \exp(-\frac{y^2}{s^2}) \\ \frac{\partial EI}{\partial y} &= -\frac{y}{s} \cdot \exp\left(-\frac{y^2}{2s^2}\right) \end{aligned} \tag{7}$$

From the above process of the partial derivative, it is obvious that the partial derivative of the EI function with respect to s is constantly greater than 0, which means that when the residual corresponding to the sample point $(\boldsymbol{x}, t)$ increases monotonically, the EI function value also increases. This indicates that as an error indicator function, the EI function can successfully identify sample points with higher residuals. Besides, since s as the residual is constantly greater than 0, the sign of the partial derivative of the EI function with respect to y is merely influenced by the sign of y. When $y < 0$, the partial derivative of the EI function with respect to y is greater than 0, when $y > 0$, it is less than 0, which means that the EI function value gradually increases when approaching $y = 0$. Hence, maximizing the value of the EI function can find the region where the predicted value of the PINN is close to 0.

However, the Dirichlet conditions of the partial differential equation are typically non-zero. In this case, assuming the Dirichlet conditions at the boundary points $(\boldsymbol{x}, t)$ are $\mathcal{D}(\boldsymbol{x}, t)$, the Expected Improvement function is modified as (8). Thus, maximizing the EI function value can find sample points that are closer to the Dirichlet conditions $\mathcal{D}(\boldsymbol{x}, t)$.

$$EI(\boldsymbol{x}, t) = s \cdot \exp\left(-\frac{(y - \mathcal{D}(\boldsymbol{x}, t))^2}{2s^2}\right) \tag{8}$$

In this way, the maximum value of the EI function can identify sample points with larger residuals in the PINN as well as sample points near the boundary of the domain. This ensures effective selection of sample points within the internal region, improving the solution accuracy within this area while maintaining the sampling algorithm's focus on the boundary points. Additionally, this method balances the distribution of sample points across different regions, preventing the increase in errors and instability in training caused by over-focusing on a particular region, thereby enhancing the overall accuracy and stability of the model.

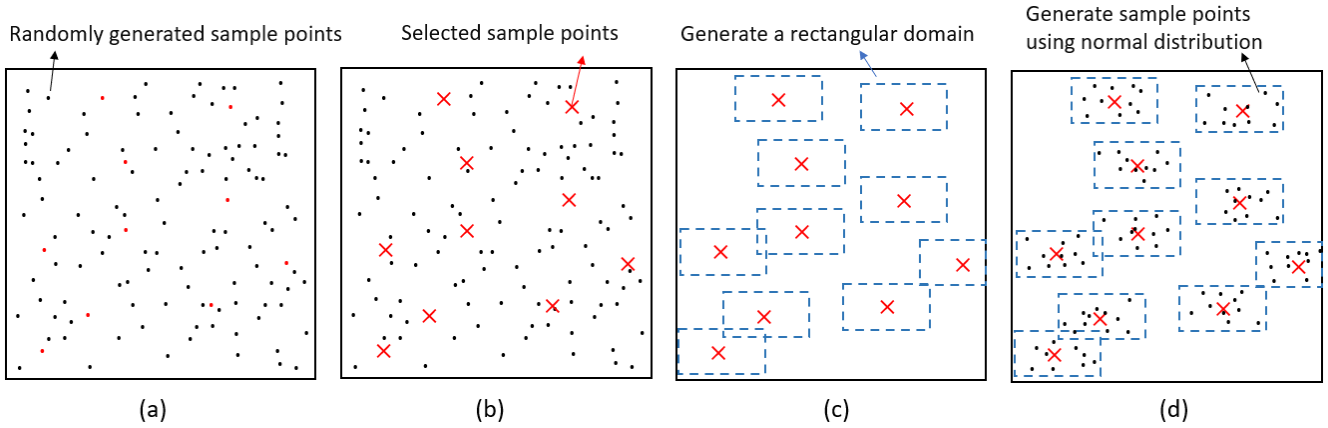

FIGURE 3. New admissible set generation process.

In the RAR-D algorithm, when resampling based on the value of residuals, the sample point set of each sampling is randomly generated in this iteration and is not associated with the admissible set in the last iteration, which greatly increases the randomness of each sampling and is not conducive to the convergence of PINN training and the improvement of accuracy. To enhance the connection between the existing sample points and the newly added ones, we generate new sample points based on the existing ones and use these points as the new admissible set. The generation process is as (9).

$$\boldsymbol{x}_{i,j} = \boldsymbol{x}_i + kN\left(0, \mathcal{L}_d\right), \quad i = 1, \ldots, I, j = 1, \ldots, J \tag{9}$$

where k is the hyperparameter, $J > 0$ is a well-specified integer representing the quantity of newly generated sample points, $\mathcal{N}\left(0, \mathcal{L}_d\right)$ stands for the d-dimensional normal distribution, and I is the quantity of newly added sample points in the last period. The process of generating admissible set is shown in FIGURE 3. FIGURE 3 illustrates the process of the sample point generation algorithm. Sample points are randomly distributed within the computational domain and a subset is selected based on the probability density function. Rectangular subdomains are then constructed around the chosen sample points, within which new points are generated using a normal distribution method. Ultimately, these newly generated points form the new admissible set.

To further strengthen the connection between the two consecutive samplings, we introduce the attention mechanism which means that we calculate residuals of newly generated sample point using PINN in last sampling and this sampling respectively. Then, these two residuals are coupled with each other under attention mechanism [34]. The calculation process of attention mechanism is (10).

$$\begin{aligned} w_0 &= a\left(y_0, z_0\right) \\ w_1 &= a\left(y_1, z_1\right) \\ &\vdots \\ w_n &= a\left(y_n, z_n\right) \end{aligned} \tag{10}$$

where $y_0, y_1, \ldots, y_n$ represent EI function values of points calculated by the current PINN, $z_0, z_1, \ldots, z_n$ represent EI function values of the same points calculated by the last PINN, the function $a(\cdot)$ represents the coupling relationship between these two residuals, which involves a form of scalar operation, $w_0, w_1, \ldots, w_n$ are coefficients of sampling probability. These coefficients multiplied by the sampling probability of current points generates an updated sampling probability.

The probability density function formula for the adaptive sampling algorithm is updated to (11).

$$P_{EI}\left(\boldsymbol{x}, t\right) \propto a\left(EI_n, EI_{n-1}\right) \frac{EI^k\left(\boldsymbol{x}, t\right)}{\mathbb{E}\left[EI^k\left(\boldsymbol{x}, t\right)\right]} + c \tag{11}$$

where EI_n denotes the EI function value of points calculated by the current PINN, and EI_{n-1} denotes the EI function value of the same points calculated by the last PINN. In the numerical experiments of the RAR-D algorithm, the results are generally better under the parameter set of $k = 2, c = 0$. We choose $k = 2, c = 0$ as parameters in the sampling algorithm and calculate the probability mass function as (12).

$$\tilde{P}_{El}\left(\boldsymbol{x}, t\right) = \frac{P_{EI}\left(\boldsymbol{x}, t\right)}{\sum_{x \in S} P_{EI}\left(\boldsymbol{x}, t\right)} \tag{12}$$

At this point, we can select the newly added sample points based on the probability mass function. The algorithm is described in detail as Algorithm 1.

When dealing with partial differential equations that have sharp solutions, the aforementioned algorithm evidently does not take into account the characteristics of such equations. Therefore, it is necessary to incorporate gradient information into the adaptive sampling algorithm as well. One of the training objectives of PINN is to make the residual $f(\boldsymbol{x}; \theta)$ of each point in domain be zero. Yu et al. [35] further proposed that the derivative of the residual of any point $(\boldsymbol{x}, t)$ should also be zero.

Inspired by this idea, we enhance the adaptive sampling algorithm based on the RAR-D algorithm and the gradient information of the residual. If there are discontinuous solution

Algorithm 1 EI-RAR Algorithm

Select the initial training point set $\mathcal{S}$ using uniformly random sampling;
Train PINN in the training set $\mathcal{S}$;
Select an admissible set $\mathcal{L}$ using uniformly random sampling;
Repeat
  $\mathcal{S}_0 \longleftarrow$ According to (12) randomly sampled a set of sufficiently dense points in the admissible set $\mathcal{L}$;
  $\mathcal{S} \longleftarrow \mathcal{S} \cup \mathcal{S}_0$;
  Retrain PINN in the training set $\mathcal{S}$;
  In the points of $\mathcal{S}_0$, a new admissible set $\mathcal{L}$ is generated based on (9);
Until the number of residuals or the average residual value reaches the limit;

Algorithm 2 EI-Grad Algorithm

Select the initial training point set $\mathcal{S}$ using uniformly random sampling;
Train PINN in the training set $\mathcal{S}$;
Select an admissible set $\mathcal{L}$ using uniformly random sampling;
Repeat
  $\mathcal{S}_0 \longleftarrow$ According to (12) randomly sampled a set of sufficiently dense points in the admissible set $\mathcal{L}$;
  $\mathcal{S}_1 \longleftarrow$ Sample points corresponding to the maximum and minimum ratios;
  $\mathcal{S} \longleftarrow \mathcal{S} \cup \mathcal{S}_0 \cup \mathcal{S}_1$;
  Retrain PINN in the training set $\mathcal{S}$;
  In the points of $\mathcal{S}_0$ and $\mathcal{S}_1$, a new admissible set $\mathcal{L}$ is generated based on (9);
Until the number of residuals or the average residual value reaches the limit;

regions in the domain, underfitting will arise at the edges. Consequently, larger residuals will occur in these edge areas, and the gradient of the residuals will change significantly. Therefore, identifying residual points with larger gradients can enhance the PINN's focus on discontinuous regions. Derivation of the residuals comes to (13).

$$\nabla f(\boldsymbol{x};\theta) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right) \tag{13}$$

Calculate the ratio of the residual value of each sample point to the residual values of other sample points, and obtain the maximum and minimum absolute values of these ratios as (14).

$$\begin{aligned} R_{max} &= max\left\{\left|\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial f}{\partial x_2}}\right|, \left|\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial f}{\partial x_3}}\right|, \cdots, \left|\frac{\frac{\partial f}{\partial x_{n-1}}}{\frac{\partial f}{\partial x_n}}\right|\right\} \\ R_{min} &= min\left\{\left|\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial f}{\partial x_2}}\right|, \left|\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial f}{\partial x_3}}\right|, \cdots, \left|\frac{\frac{\partial f}{\partial x_{n-1}}}{\frac{\partial f}{\partial x_n}}\right|\right\} \end{aligned} \tag{14}$$

During each resampling, calculate R_{max} and R_{min} for all sample points, and identify the sample points with the largest R_{max} and the smallest R_{min}. Include these points in the initial training sample set and retrain PINN along with the sample points selected by the probability mass function. After incorporating gradient information, the details of the algorithm are as shown in Algorithm 2.

IV. EXPERIMENTS

We evaluated the performance of Res-PINN with various sampling algorithms including RAR-D sampling algorithm, Grad-RAR sampling algorithm and the proposed sampling algorithm in this paper. These algorithms are used for solving various PDEs and we compare the solution errors with each other. To measure the effect of our proposed algorithm, the relative L^2 error between the exact value $u(x_i, t_i)$ and the approximate value $\hat{u}(x_i, t_i)$ at the sample point $\{x_i, t_i\}_{i=1}^N$ obtained by Res-PINN is used and is calculated as (15).

$$L^2 error : e = \frac{\sqrt{\sum_{i-1}^{N} \left|\hat{u}(x_i, t_i) - u(x_i, t_i)\right|^2}}{\sqrt{\sum_{i-1}^{N} \left|u(x_i, t_i)\right|^2}} \tag{15}$$

A. DIFFUSION EQUATION

Here, we consider a 1D diffusion equation, whose computational domain is $[-1,1] \times [0, 1]$.

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + e^{-t}\left(-\sin(\pi x) + \pi^2 \sin(\pi x)\right), \\ u(x, 0) &= \sin(\pi x), \\ u(-1, t) &= u(1, t) = 0 \end{aligned} \tag{16}$$

where u represents the concentration of diffusion material. The diffusion equation has an exact solution $u(x, t) = \sin(\pi x)e^{-t}$. To maintain the accuracy of the contrast experiments, we unify the structure of PINN used for multiple sampling algorithms to Res-PINN. This network uses a hyperbolic activation function (tanh) and it consists of three residual blocks, each of which contains a fully connected layer, and each layer contains 60 neurons.

Before the training commences, we randomly place 40 residual points within the computational domain and randomly position 10 initial points along with 20 boundary points. During the initial training phase of the PINN, we use Adam optimizer with a learning rate of 5×10^{-4} to minimize the loss function in PINN and iterate 10^4 times. Since the diffusion equation does not have sharp solutions, we opt for the EI-RAR algorithm proposed in this paper for solving it. During the adaptive sampling process, for each added sample point, we perform 1000 iterations using the Adam optimization, until the number of added sample points reaches 4000.

Besides, we train the same experiment at least five times for all PINNs with different sampling methods, such as RAR-D sampling algorithm and EI-RAR sampling algorithm proposed in this paper and calculate the mean and standard deviation of the L^2 errors in TABLE 1.

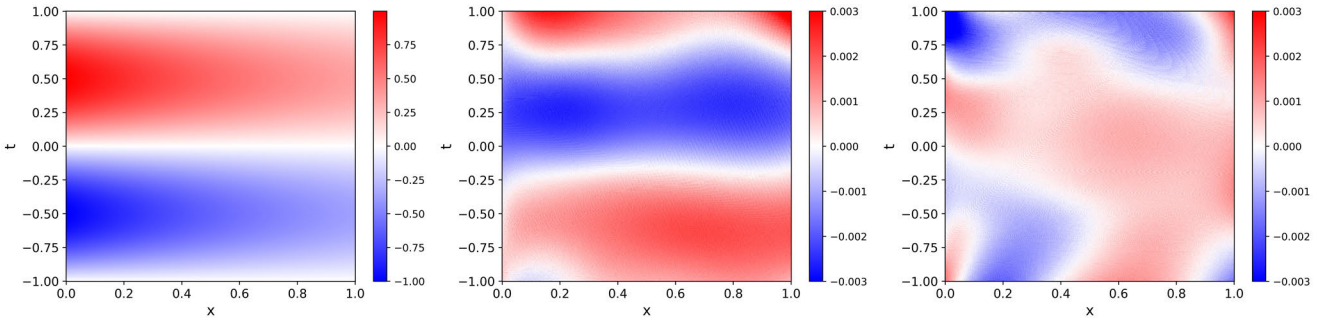

FIGURE 4. New admissible set generation process. Results for the diffusion equations. (a) Predicted solution by EI-RAR sampling algorithm, (b)pointwise error of diffusion equation using RAR-D sampling algorithm, (c) pointwise error of diffusion equation using EI-RAR sampling algorithm.

TABLE 1. Relative L^2 error of diffusion equation with various sampling algorithms.

Sampling Method	Random	RAR-D	EI-RAR
Error	0.67±0.21%	0.27±0.08%	0.16±0.02%

From the TABLE 1, it is apparent that EI-RAR sampling algorithm, employing Res-PINN, can solve the diffusion equation more stably and accurately when boundary and initial conditions are not working as hard constrains in the neural network. FIGURE 4. shows the pointwise error between the concentration value and its corresponding exact value using RAR-D sampling algorithm and EI-RAR sampling algorithm in Diffusion equation. And it can be seen that the error has a more uniform distribution in the figure produced by EI-RAR with a better fit intuitively.

B. BURGERS' EQUATION

The 1D Burgers' equation is described as (17), whose computational domain is $[-1, 1] \times [0, 1]$.

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} &= \nu\frac{\partial^2 u}{\partial x^2}, \\ u(-1, t) &= u(1, t) = 0, \\ u(x, 0) &= -\sin(\pi x) \end{aligned} \tag{17}$$

We select the parameter $\nu = 0.01/\pi$, consistent with the parameter used by Raissi et al. [6] in their pioneering PINN paper. When the diffusion coefficient is small, the nonlinear solution of the convection term is steep, which means that the Burgers' equation has a steep solution. Choosing parameters used in previous studies allows our algorithm to be more easily compared with these studies. Res-PINN chooses tanh as the activation function, and to verify the validity of selecting the EI function as the error indicator function, boundary conditions and initial conditions are not constrained to PINN as hard constrains during the calculation of Burgers' equation.

Due to the sharp solutions presented by the Burgers' equation, the EI-Grad algorithm proposed in this paper is selected as the adaptive sampling algorithm during the training process. In the training phase, the initial training points for the PINN are determined using a random sampling algorithm, where 2000 interior points, 100 boundary sample points, and 200 initial sample points are selected at random from within the computational domain. We used Res-PINN with four residual blocks, each containing a fully connected neural layer with 50 neurons. During the initial training phase of the PINN, we use Adam optimizer with a learning rate of 10^{-3} to minimize the loss function in PINN and iterate 2×10^4 times. In the subsequent training phase of the PINN, parameters are first updated by Adam optimizer with 10^3 steps, then, they are trained by L-BFGS with 10^3 steps. Increase the number of sampling points by 20 in each iteration until a total of 2000 additional sampling points are reached. For each kind of sampling algorithm of the PINN, we performed the experiments multiple times.

Different sampling algorithms are combined with Res-PINN to solve the one-dimensional Burgers' equation, and the mean values and standard deviation of errors are calculated and shown in the TABLE 2. The table shows that using the EI-Grad algorithm to solve the one-dimensional Burgers' equation without hard constraints is more accurate and will have lower standard deviation, indicating that EI-Grad algorithm makes the Res-PINN converge better.

TABLE 2. Relative L^2 error of Burgers' equation with different sampling algorithms.

Sampling Method	Random	RAR-D	EI-Grad
Error	1.06±0.49%	0.33±0.11%	0.09±0.06%

FIGURE 5. exhibits the plots of the pointwise error using Random, and EI-Grad combined with the Res-PINN and the reference plot of velocity values in the one-dimensional Burgers' equation. After adding the gradient information, the error in the discontinuous region of the domain can be effectively reduced, and EI-Grad sampling algorithm can reduce the error level even further.

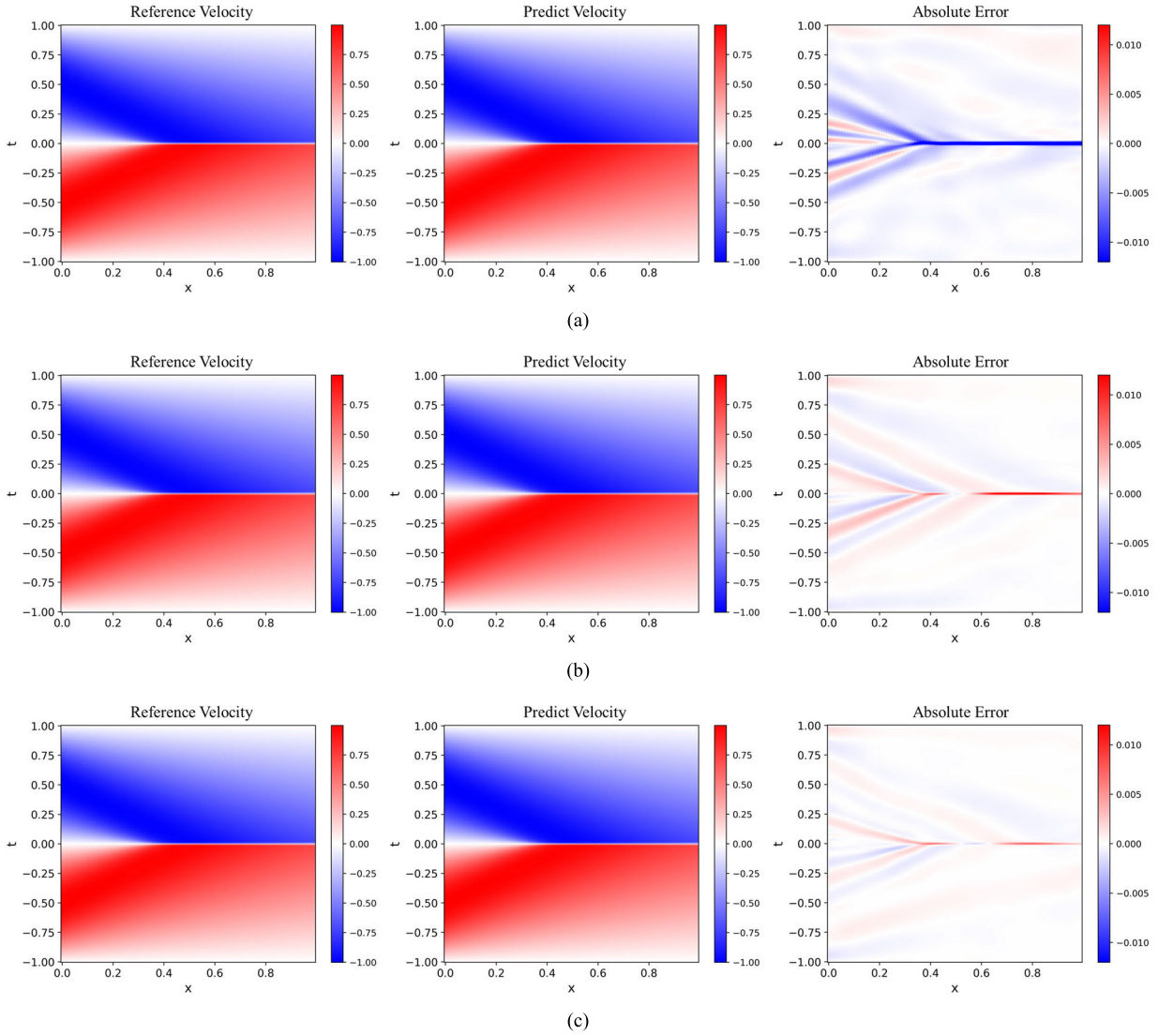

FIGURE 5. Results for Burgers' equation using different sampling algorithm. (a) Reference velocity, predict velocity and absolute error obtained by random sampling algorithm, (b) reference velocity, predict velocity and absolute error obtained by RAR-D sampling algorithm, (c) reference velocity, predict velocity and absolute error obtained by EI-Grad sampling algorithm.

C. ALLEN-CAHN EQUATION

We consider the Allen-Cahn equation as follows:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D\frac{\partial^2 u}{\partial x^2} + 5(u - u^3), x \in [-1, 1], t \in [0, 1], \\ u(x, 0) &= x^2 \cos(\pi x), \\ u(-1, t) &= u(1, t) = -1. \end{aligned} \tag{18}$$

where the value of the diffusion coefficient D is chosen to be 0.001.

Using the neural network and training procedure similar to previous experiments, the Allen-Cahn equation is calculated by random sampling algorithm, RAR-D sampling algorithm and EI-RAR sampling algorithm. For the initial training of PINN, we randomly select 2000 points within the interior, 100 points along the boundary, and 240 initial points within the computational domain as initial training dataset. The architecture of the PINN utilized is Res-PINN that consists of four residual blocks, with each block containing a fully connected layer of 64 neurons. In the subsequent training phase of the PINN, we first employ the Adam optimizer for 10^3 steps to update the parameters. Then, the training continues with the L-BFGS optimizer for another 10^3 steps. The training process includes progressively increasing the number of sampling points by 20 per iteration until an additional 2000 sampling points have been incorporated.

The L^2 errors corresponding to different sampling algorithms are shown in the TABLE 3. It is evident that the EI-RAR algorithm demonstrates a significant improvement in the accuracy of solving Allen-Cahn equation.

To further demonstrate the generalization and applicability of our proposed adaptive sampling algorithm, we use not only a diffusion coefficient $D = 0.001$ but also test the algorithm's performance under different parameter settings,

TABLE 3. Relative L^2 error of Allen-Cahn equation with different sampling algorithms for $D = 0.001$.

Sampling Method	Random	RAR-D	EI-RAR
Error	6.47±1.25%	2.87±0.93%	0.97±0.26%

such as $D = 0.0001$ for Allen–Cahn equation with periodic boundary conditions. In the initial training, we randomly select 5000 sample points within the domain, along with 100 boundary points and 300 initial points. The network architecture chosen is a residual neural network with four residual blocks, each containing a fully connected neural layer with 128 neurons. The adaptive sampling algorithm iterates for 100 rounds, selecting 50 sample points in each iteration. Finally, we present the L^2 errors for different sampling algorithms under the diffusion coefficient $D = 0.0001$ for the Allen-Cahn equation in the TABLE 4. The results show that even with different parameter choices for the partial differential equation, our algorithm maintains stability and accuracy.

TABLE 4. Relative L^2 error of Allen-Cahn equation with different sampling algorithms for $D = 0.0001$.

Sampling Method	Random	RAR-D	EI-RAR
Error	89.95±4.57%	4.02±1.07%	1.96±0.61%

D. FLOW IN A LID-DRIVEN CAVITY

In this section, we focus on the classical benchmark problem in fluid dynamics, i.e., the two-dimensional lid-driven cavity steady-state flow. The model is comprised of a two-dimensional cavity which has a tangential velocity on the upper boundary and zero velocity on other three boundaries, just as shown in the FIGURE 6. This motion gives rise to a flow phenomenon whose characteristics show a large vortex in the center of the cavity and some small vortices in the corners. And the value of Reynolds number affects the size and number of vortices in the flow.

The motion is governed by the incompressible Navier-Stokes equation, which can be written in dimensionless form as (19).

$$\begin{aligned} (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} &= \frac{1}{Re}\cdot\nabla^2\boldsymbol{u} - \nabla\mathbf{p}, \\ \nabla\cdot\boldsymbol{u} &= 0 \end{aligned} \tag{19}$$

where $\boldsymbol{u}(\boldsymbol{x}) = (u(\boldsymbol{x}), v(\boldsymbol{x}))$ is a velocity vector field, p is a scalar pressure field, and the size of the computational domain is $\boldsymbol{x} = (x, y) \in [0, 1] \times [0, 1]$. In addition, the upper boundary of the cavity has a tangential velocity $u = 1$, and the remaining three boundaries of the cavity have zero velocity, and Re denotes the Reynolds number of the fluid, and we choose $Re = 100$ in this case.

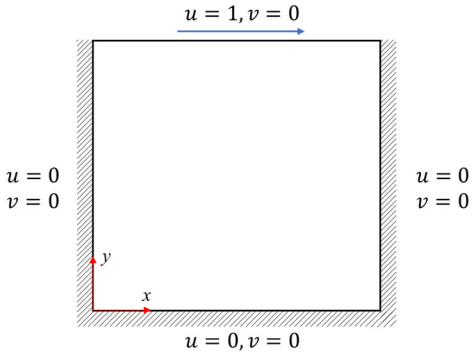

FIGURE 6. Schematic diagram of the problem description of flow in a lid-driven cavity.

For the above Navier-Stokes equation, the residual form is constructed according to the process of PINN as (20).

$$\begin{aligned} r_\theta^u(x,y) &:= u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \\ r_\theta^v(x,y) &:= u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \\ r_\theta^c(x,y) &:= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \end{aligned} \tag{20}$$

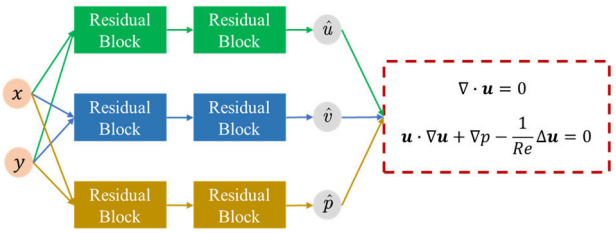

FIGURE 7. Parallel-PINN structure in velocity-representation of Navier-Stokes equation.

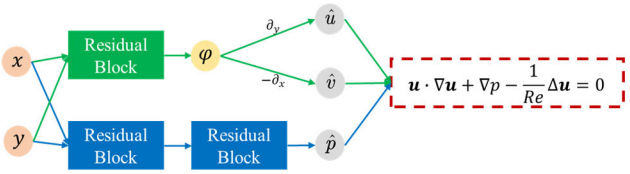

FIGURE 8. Parallel-PINN structure in streamfunction-pressure representation of Navier-Stokes equation.

According to the above residuals and the boundary conditions in Navier-Stokes equation, the loss function in PINN is defined as (21).

$$L(\theta) = L_{r_u}(\theta) + L_{r_v}(\theta) + L_{r_c}(\theta) + L_{u_b}(\theta) + L_{v_b}(\theta) \tag{21}$$

where the first three terms represent residual loss terms formed by the PDEs expressed by the law of conservation of momentum and the law of conservation of mass, and the

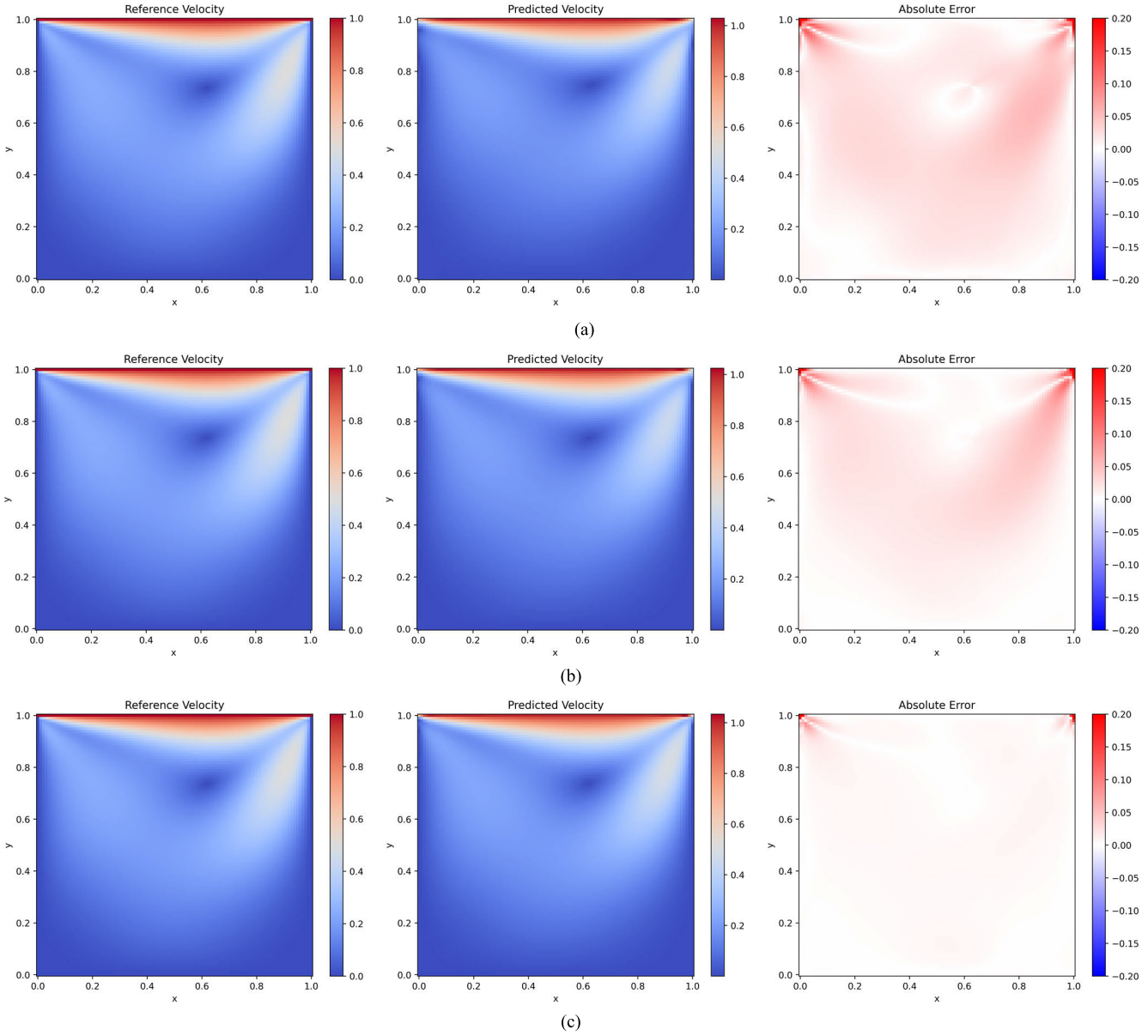

FIGURE 9. Results for lid-driven cavity steady-state flow using different sampling algorithm. (a) Reference velocity, predict velocity and absolute error obtained by random sampling algorithm, (b) reference velocity, predict velocity and absolute error obtained by RAR-D sampling algorithm, (c) reference velocity, predict velocity and absolute error obtained by EI-Grad sampling algorithm.

last two terms are the loss terms formed by the boundary conditions in both directions of the lid-driven cavity flow.

In the process of solving this fluid mechanics problem, the PDEs are expressed in the form of multiple latent solutions, therefore, using only one forward neural network for training to solve the cavity flow problem will cause a large error. Besides, when the neural network is retrained after resampling, the neural network becomes less robust due to this simple structure, and it is easy to cause large oscillations in the parameters. In this case, all parameters in the entire physical informed neural network are not optimized. Therefore, we connect multiple neural networks together in parallel, each of which uses the same spatiotemporal coordinates as input in Res-PINN, to solve one of the multiple latent solutions in the PDE. And we name this network structure parallel-PINN. Then, all the multiple latent solutions solved by the parallel-PINN utilize automatic differentiation to construct the loss function for the lid-driven cavity flow problem. The velocity-pressure representation using parallel-PINN is shown schematically in FIGURE 7.

For the initial training of PINN, we randomly select 2000 points within the interior and 200 points along the boundary within the computational domain as initial training dataset. Two Res-PINNs are utilized in a parallel configuration, where each Res-PINN is comprised of three residual blocks, and each block contains 48 neurons.

During the initial training phase of the PINN, we use Adam optimizer with a learning rate of 5×10^{-4} to minimize the loss function and iterate 2×10^4 times. Subsequently, we utilize the L-BFGS optimizer to iterate for 2×10^4 times.. In retraining the PINN, parameters are first updated by Adam optimizer with 2×10^3 steps, then, they are trained by L-BFGS with 3×10^3 steps. Increase the number of sampling points by 40 in each iteration until a total of 4000 additional sampling points are reached.

The relative L^2 errors corresponding to different sampling algorithms are obtained by training the parallel-PINN and are shown in TABLE 5.

TABLE 5. Relative L^2 error of Navier-Stokes equation in velocity-representation with different sampling algorithms.

Sampling Method	Random	RAR-D	EI-Grad
Error	0.4824	0.2377	0.1482

In the case of solving the Navier-Stokes equation, the stream function $\varphi(x, y)$ is used and the continuity equation in the system equation is automatically satisfied as a result, i.e. (22).

$$u = \frac{\partial \varphi}{\partial y}, v = -\frac{\partial \varphi}{\partial x} \tag{22}$$

The mapping relationship of the physical informed neural network is $(x, y) \rightarrow (\varphi, p)$, which means that streamfunction-pressure representation can effectively reduce the number of parameters to be trained in Res-PINN. The streamfunction-pressure representation using parallel-PINN is shown schematically in FIGURE 8.

We use parallel-PINN combined with RAR-D sampling algorithm, Grad-RAR sampling algorithm, and EI-Grad sampling algorithm to solve the cavity flow problem. Plots for reference velocity, predicted velocity and their pointwise error are shown in FIGURE 9.

Corresponding to different sampling algorithms, the relative L^2 errors, using the streamfunction-pressure representation of Navier-Stokes equation, are shown in TABLE 6.

TABLE 6. Relative L^2 Error of Navier-Stokes equation in streamfunction-pressure with different sampling algorithms.

Sampling Method	Random	RAR-D	EI-Grad
Error	0.1351	0.0624	0.0108

Comparing the accuracies of parallel-PINN with different sampling algorithms for the lid-driven cavity flow problem, it is evident that the EI-Grad algorithm can reduce the solution error of velocity and show better results solving the PDEs involving multiple latent solutions..

In addition, after comparing the two different forms of Navier-Stokes equation with sampling algorithm, it can be found that the solution error for the lid-driven cavity flow problem has a lower order of magnitude when streamfunction-pressure representation is chosen.

V. CONCLUSION

In this paper, we notice that when solving PDEs, the adaptive sampling algorithm can effectively improve the performance of PINN, which is essentially similar to the selection of grid cells in traditional numerical methods. Based on the concept of probabilistic sampling using residual points, we modify the error indicator of general adaptive sampling algorithm and propose the EI-RAR algorithm. First, we choose an expected improvement function to replace the residual of the sample points as the new error indicator function, which can make us find the points closer to the boundary of the computational domain while focusing on the high residuals. Then, the sample points with special residual gradient information are added to improve the fitting accuracy of PINN in the discontinuous solution region, which makes us to further propose the EI-Grad adaptive sampling algorithm.. In addition, this paper introduces point generation algorithm and attention mechanism into our algorithm, so that each iteration sampling process can effectively use the last sampling information and the convergence speed of the neural network increases. Moreover, we choose residual neural network as the network structure of PINN to reduce the probability of vanishing gradient pathologies in training.

In order to verify that our sampling algorithm can effectively enhance PINN, we choose Diffusion equation, Burgers' equation, Allen-Cahn equation and flow in a lid-driven cavity as experiment tests. In Diffusion equation and Allen-Cahn equation, compared with Random and RAR-D sampling algorithms, EI-RAR has lower solution error and better robustness. In Burgers' equation, EI-Grad can significantly reduce the solution error of the discontinuous region in the solution domain. Regarding the problem of the flow in a lid-driven cavity, we first choose parallel-PINN to increase the complexity of PINN network structure. Then, velocity-representation and streamfunction-pressure representation are used to solve the Navier-Stokes equation. The results show that the EI-Grad sampling algorithm can raise the solution accuracy of the velocity field dramatically.

In the future, we will mainly study how to select error indicator function and probability sampling algorithm to improve the solution accuracy and training speed of PINN without hard constraints. Up to now, we have only studied one-dimensional spatiotemporal problems and two-dimensional spatial problems. Therefore, in the near future, we will explore how to select sampling points of PDEs in higher dimensions to increase their solution accuracy.

REFERENCES

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, Feb. 2018, Art. no. 7068349.
[2] K. R. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*. Jodhpur, India: Springer, 2020, pp. 603–649.
[3] B. Wendroff, "Difference methods for initial-value problems (Robert D. Richtmyer and K. W. Morton)," *SIAM Rev.*, vol. 10, no. 3, pp. 381–383, Jul. 1968.
[4] S. Chai, Y. Zou, C. Zhou, and W. Zhao, "Weak Galerkin finite element methods for a fourth order parabolic equation," *Numer. Methods Partial Differ. Equ.*, vol. 35, no. 5, pp. 1745–1755, Mar. 2019.

[5] C. Yang, R. Niu, and P. Zhang, "Numerical analyses of liquid slosh by finite volume and lattice Boltzmann methods," *Aerosp. Sci. Technol.*, vol. 113, Jun. 2021, Art. no. 106681.
[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
[7] A. B. Buhendwa, S. Adami, and N. A. Adams, "Inferring incompressible two-phase flow fields from the interface motion using physics-informed neural networks," *Mach. Learn. Appl.*, vol. 4, Jun. 2021, Art. no. 100029.
[8] R. Mojgani, M. Balajewicz, and P. Hassanzadeh, "Kolmogorov n-width and Lagrangian physics-informed neural networks: A causality-conforming manifold for convection-dominated PDEs," *Comput. Methods Appl. Mech. Eng.*, vol. 404, Feb. 2023, Art. no. 115810.
[9] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, Jan. 2022.
[10] C. Cheng and G.-T. Zhang, "Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems," *Water*, vol. 13, no. 4, p. 423, Feb. 2021.
[11] K. Sun and X. Feng, "A second-order network structure based on gradient-enhanced physics-informed neural networks for solving parabolic partial differential equations," *Entropy*, vol. 25, no. 4, p. 674, Apr. 2023.
[12] B. Stevens and T. Colonius, "FiniteNet: A fully convolutional LSTM network architecture for time-dependent partial differential equations," 2020, *arXiv:2002.03014*.
[13] R. Rodriguez-Torrado, P. Ruiz, L. Cueto-Felgueroso, M. C. Green, T. Friesen, S. Matringe, and J. Togelius, "Physics-informed attention-based neural network for solving non-linear partial differential equations," 2021, *arXiv:2105.07898*.
[14] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems," *Comput. Methods Appl. Mech. Eng.*, vol. 365, Jun. 2020, Art. no. 113028.
[15] A. D. Jagtap and G. E. Karniadakis, "Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations," in *Proc. AAAI Spring Symp.*, vol. 10, Nov. 2021, pp. 1–6.
[16] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. 3055–3081, Jan. 2021.
[17] R. Bischof and M. Kraus, "Multi-objective loss balancing for physics-informed deep learning," 2021, *arXiv:2110.09813*.
[18] Z. Xiang, W. Peng, X. Liu, and W. Yao, "Self-adaptive loss balanced physics-informed neural networks," *Neurocomputing*, vol. 496, pp. 11–34, Jul. 2022.
[19] S. Thakur, M. Raissi, H. Mitra, and A. Ardekani, "Temporal consistency loss for physics-informed neural networks," 2023, *arXiv:2301.13262*.
[20] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Comput. Methods Appl. Mech. Eng.*, vol. 360, Mar. 2020, Art. no. 112789.
[21] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," *SIAM Rev.*, vol. 63, no. 1, pp. 208–228, Jan. 2021.
[22] J. M. Hanna, J. V. Aguado, S. Comas-Cardona, R. Askri, and D. Borzacchiello, "Residual-based adaptivity for two-phase flow simulation in porous media using physics-informed neural networks," *Comput. Methods Appl. Mech. Eng.*, vol. 396, Jun. 2022, Art. no. 115100.
[23] W. Peng, W. Zhou, X. Zhang, W. Yao, and Z. Liu, "RANG: A residual-based adaptive node generation method for physics-informed neural networks," 2022, *arXiv:2205.01051*.
[24] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," *Comput. Methods Appl. Mech. Eng.*, vol. 403, Jan. 2023, Art. no. 115671.
[25] Z. Mao and X. Meng, "Physics-informed neural networks with residual/gradient-based adaptive sampling methods for solving partial differential equations with sharp solutions," *Appl. Math. Mech.*, vol. 44, no. 7, pp. 1069–1084, Jul. 2023.
[26] C. Rao, H. Sun, and Y. Liu, "Physics-informed deep learning for incompressible laminar flows," *Theor. Appl. Mech. Lett.*, vol. 10, no. 3, pp. 207–212, Mar. 2020.
[27] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 4, pp. 86–112, Jan. 1967.
[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[29] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, nos. 1–3, pp. 503–528, Aug. 1989.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[31] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *J. Comput. Phys.*, vol. 404, Mar. 2020, Art. no. 109136.
[32] N. Metta, R. Ramachandran, and M. Ierapetritou, "A novel adaptive sampling based methodology for feasible region identification of compute intensive models using artificial neural network," *AIChE J.*, vol. 67, no. 2, Feb. 2021, Art. no. e17095.
[33] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.
[34] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.
[35] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, "Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems," *Comput. Methods Appl. Mech. Eng.*, vol. 393, Apr. 2022, Art. no. 114823.

YANBING LIU received the B.S. degree from Chongqing University, Chongqing, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. His current research interests include adaptive sampling algorithms for PINN and two-phase flow using PINN.

LIPING CHEN received the Ph.D. degree from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, China. He is currently a Professor with the Huazhong University of Science and Technology. His research interests include model-based systems engineering and multi-domain physical unification modeling theory.

JIANWAN DING received the Ph.D. degree from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, China, in 2006. He is currently an Associate Professor with the Huazhong University of Science and Technology. His research interests include multi-domain system modeling and simulation and mechanical system dynamics analysis.

YU CHEN received the B.S. degree from Huazhong Agricultural University, Hubei, Wuhan. He is currently pursuing the Ph.D. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. His research interest includes the trajectory planning of robots and robot control.