

RESEARCH ARTICLE | MARCH 25 2024

The improved backward compatible physics-informed neural networks for reducing error accumulation and applications in data-driven higher-order rogue waves FREE

Special Collection: [Rogue waves: Theory, Methods and Applications](#)Shuning Lin  ; Yong Chen  *Chaos* 34, 033139 (2024)<https://doi.org/10.1063/5.0191283>

Chaos

Special Topic: Nonautonomous Dynamics in the Climate Sciences

[Submit Today](#)

The improved backward compatible physics-informed neural networks for reducing error accumulation and applications in data-driven higher-order rogue waves

Cite as: Chaos 34, 033139 (2024); doi: 10.1063/5.0191283

Submitted: 12 December 2023 · Accepted: 26 February 2024 ·

Published Online: 25 March 2024



View Online



Export Citation



CrossMark

Shuning Lin¹ and Yong Chen^{1,2,a)}

AFFILIATIONS

¹School of Mathematical Sciences, Key Laboratory of Mathematics and Engineering Applications (Ministry of Education) and Shanghai Key Laboratory of PMMP, East China Normal University, Shanghai 200241, China

²College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China

Note: This paper is part of the Focus Issue, Rogue waves: Theory, Methods and Applications.

a) Author to whom correspondence should be addressed: yichen@sei.ecnu.edu.cn

ABSTRACT

Due to the dynamic characteristics of instantaneity and steepness, employing domain decomposition techniques for simulating rogue wave solutions is highly appropriate. Wherein, the backward compatible physics-informed neural network (bc-PINN) is a temporally sequential scheme to solve PDEs over successive time segments while satisfying all previously obtained solutions. In this work, we propose improvements to the original bc-PINN algorithm in two aspects based on the characteristics of error propagation. One is to modify the loss term for ensuring backward compatibility by selecting the earliest learned solution for each sub-domain as pseudo-reference solution. The other is to adopt the concatenation of solutions obtained from individual subnetworks as the final form of the predicted solution. The improved backward compatible PINN (Ibc-PINN) is applied to study data-driven higher-order rogue waves for the nonlinear Schrödinger (NLS) equation and the AB system to demonstrate the effectiveness and advantages. Transfer learning and initial condition guided learning (ICGL) techniques are also utilized to accelerate the training. Moreover, the error analysis is conducted on each sub-domain, and it turns out that the slowdown of Ibc-PINN in error accumulation speed can yield greater advantages in accuracy. In short, numerical results fully indicate that Ibc-PINN significantly outperforms bc-PINN in terms of accuracy and stability without sacrificing efficiency.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0191283>

The geometric structure of higher-order rogue waves can be classified into fundamental, triangular, pentagonal shapes, and even more complex patterns. Rogue wave solutions have been widely analytically studied but the research studies by using deep learning methods are still relatively scarce. It is suitable to simulate high-order rogue waves by using domain decomposition techniques, considering their complicated dynamic characteristics. In this paper, we put forward the improved backward compatible physics-informed neural network (Ibc-PINN) method with the aim of reducing error accumulation based on the traditional bc-PINN, a temporally sequential algorithm to solve PDEs. According to the characteristics of error propagation, we propose improvements in two aspects, namely, the loss function and the final form of the predicted solution. Moreover, Ibc-PINN is

applied to obtain data-driven higher-order rogue wave solutions of the nonlinear Schrödinger equation and the AB system. The numerical results demonstrate that the Ibc-PINN algorithm contributes to the mitigation of error accumulation speed, exhibiting significant advantages in terms of accuracy and stability compared to the bc-PINN.

I. INTRODUCTION

Rogue waves (also known as monster waves, killer waves, giant waves, and abnormal waves) have always been an important research direction in the field of nonlinear science. Due to the complexity of the rogue wave phenomenon, it is currently difficult to

provide a precise definition. Usually, in oceanography, people refer to a type of abnormally steep wave that suddenly appears on the sea level as the rogue wave. This type of wave has the characteristic of “coming without a shadow, going without a trace.” The concept of rogue waves was first proposed by British scientist Draper in 1964, who described a huge, sudden appearance of monster waves that overturned ships in Ref. 1. The rogue wave is a very tall and strong solitary wave, and its wave height is usually more than twice that of the highest wave around it. Due to the sudden onset and immense power, it can sometimes cause serious damage to ships and offshore structures. Naturally, rogue wave phenomena are not confined to the field of oceanography. In 2007, Solli *et al.*² utilized a new real-time detection technique to observe abnormally steep, massive, and rare optical rogue waves for the first time in nonlinear optical systems. Afterward, rogue wave phenomena have emerged in numerous research fields, such as Bose–Einstein condensation,³ plasma physics,⁴ fluid mechanics,⁵ meteorology,⁶ and finance.⁷

Scientists aspire to elucidate the ancient and enigmatic phenomenon of rogue waves through the application of nonlinear models. Research has revealed that the rational solution for the Nonlinear Schrödinger (NLS) equation can aptly describe rogue waves. This solution, first discovered by the British physicist Peregrine in 1983,⁸ is consequently referred to as the Peregrine soliton. In terms of mathematical expression, the first-order rational solution for the NLS equation is composed of a second-order rational polynomial and an exponential function. Physically, it describes a localized, steep wave in both spatial and temporal dimensions, with an amplitude three times higher than the background field. The evolution of this solution precisely aligns with the two fundamental characteristics of rogue waves: suddenness and steepness. Up to this point, utilizing existing soliton theories to investigate various rogue wave solutions has become one of the focal issues capturing the attention of scientists across diverse fields worldwide. Commonly employed research methods include the Hirota bilinear method,⁹ the Kadomtsev–Petviashvili (KP) hierarchy reduction method,¹⁰ the generalized Darboux transformation,¹¹ and the inverse scattering transformation,¹² among others.

With the rapid development of computer technology and the explosive growth of available data, methods for solving partial differential equations have expanded from traditional numerical approaches to data-driven methods. The physics-informed neural network (PINN) method¹³ has demonstrated extraordinary prospects among various data-driven techniques for solving PDEs. The main idea of PINN is to incorporate information from physical equations into the training of neural networks, rather than relying solely on given data. This enables it to provide accurate predictions even in situations where data are scarce or corrupted by noise. In response to various application scenarios and precision requirements, numerous variations and extensions of PINNs have emerged, such as fPINN¹⁴ for solving fractional differential equations, hPINN¹⁵ for inverse design, B-PINN¹⁶ to solve both forward and inverse nonlinear problems involving PDEs and noisy data, NN-aPC¹⁷ for solving stochastic PDEs, and so on. In the field of integrable systems, algorithms based on excellent properties including conserved quantities,¹⁸ Miura transformation,¹⁹ and symmetries.²⁰ Moving forward, there is anticipation for the synergistic integration of deep learning methods with other integrable theories, such

as Darboux transformation, Riemann–Hilbert method, and Lax pair.

Despite promising prospects, PINN has encountered difficulties in accurately and efficiently tackling challenges associated with large domains and multiscale problems. This motivates us to explore additional techniques for enhancing accuracy. Multiple reasons drive the integration of region decomposition technology into PINNs. First, it can decompose the scale of the original problem, dividing a complex problem into equivalent smaller parts according to a certain decomposition method, and then solving them one by one. Second, locality can be highlighted. The experience shows that neural networks tend to approximate smooth and continuous functions. This may lead to poor local prediction performance in complex situations, such as approaching steep or discontinuous solutions. Furthermore, in terms of the algorithm design, we can combine traditional numerical algorithms with neural network methods. Finally, parallel computing can also be used to improve efficiency. The cPINN method²¹ performs domain decomposition in space and the conserved quantities like fluxes are preserved by enforcing their continuity in the strong form at the common interfaces of neighboring sub-domains. The extended PINN (XPINN)²² is a space–time domain decomposition based deep learning framework and is an extension of the cPINN method. XPINN replaces the continuity of flux on the interface of adjacent regions with the more general condition of residual continuity, which can be extended to solve any type of PDEs. On this basis, a parallel algorithm for cPINNs and XPINNs is also proposed.²³ With regard to innovation in the algorithm design, FBPINN²⁴ is an overlapping domain decomposition framework inspired by classical finite element methods, and hp-VPINNs²⁵ are designed by combining variational principles. In addition to manual decomposition of the computational domain, Stiller *et al.* put forward an adaptive domain decomposition method called Gated PINN.²⁶ This study suggests that the decomposition of the computational domain can be learned by utilizing the mixture of expert approach.²⁷ Specifically, the decomposition method of time domain is also a hot research topic due to the particularity of time variables compared to spatial ones. Two time-adaptive approaches, adaptive sampling in time, and adaptive time marching strategy are introduced in Ref. 28. To ensure backward compatibility of the solution, the bc-PINN scheme²⁹ re-trains the neural network over successive time segments while satisfying the solution for all previous time segments. Given that the existing PINN formula cannot respect the inherent spatiotemporal causal structure of the evolution of physical systems, Wang *et al.*³⁰ put forth a general casual training algorithm for explicitly respecting physical causality during model training. In contrast to previously mentioned methods using hard constraints, causal PINN adjusts the initiation of training in subsequent time domains based on the magnitude of weights, employing a soft constraint approach. Moreover, the idea of ensemble learning is adopted in Ref. 31, and the computational region can be automatically expanded without artificially dividing the time domain.

In recent years, the use of deep learning methods to simulate rogue wave solutions has attracted widespread attention and sparked a new research trend. Since Chen proposed the concept of integrable deep learning,^{32,33} his team has conducted extensive research in the field of data-driven rogue waves with the aid of

the PINN algorithm and its variants. The dynamic behaviors of the rogue wave solution³⁴ for the nonlinear Schrödinger equation and the rogue periodic wave solution³⁵ for the Chen–Lee–Liu equation have been reproduced for the first time. Miao and Chen³⁶ also studied the data-driven resonance rogue wave solution for the (2+1)-dimensional KP equation, which is the first practice of deep learning methods in high-dimensional integrable systems. Later, Pu *et al.* also successfully simulated vector rogue waves for the Manakov³⁷ and Yajima–Oikawa systems.³⁸ In addition, other scholars have also made rich achievements and progress in this area of research.^{39–43}

Considering the complexity of the dynamics of rogue wave solutions (instantaneity and steepness), it is a good application scenario for using domain decomposition techniques. In this paper, in order to capture the complex dynamic behavior of higher-order rogue waves, we treat the integrable equation considered here as a time evolution model and adopt a time-phased training approach. The bc-PINN method draws inspiration from the traditional numerical format and employs a training approach that advances layer by layer in the temporal domain, while satisfying all obtained solutions of the previous time periods to ensure backward compatibility of the solution. For algorithms that perform time piecewise training to solve PDEs, error propagation is an unavoidable problem. Therefore, based on the characteristics of error propagation, we propose improvements to the original bc-PINN algorithm in the design of the loss function and the final form of the predicted solution. Then, the improved algorithm is applied to study the data-driven high-order rogue waves for the NLS equation⁸ and the AB system,⁴⁴ while analyzing the effectiveness of these improvements in reducing error accumulation by calculating the errors in each sub-domain.

The structure of the article is as follows. Section II gives a brief review of the original PINN and bc-PINN algorithms and puts forward the improved bc-PINN (Ibc-PINN) for reducing error accumulation. In Sec. III, the Ibc-PINN method is utilized to learn abundant dynamic behaviors of the first-order and higher-order rogue waves for the NLS equation and the AB system. Then, we conduct an analysis of error accumulation to disclose the necessity of proposing an Ibc-PINN method and analyze the errors across the entire spatiotemporal region to explore the role of each aspect of improvement in Sec. IV. In addition, numerical experiments are carried out to discuss the impact of changes in the network structure on both algorithm stability and accuracy. Finally, the conclusion and expectation are given in Sec. V.

II. METHODOLOGY

A. A brief review of PINN and bc-PINN

Consider the general form of a $N + 1$ -dimensional partial differential equation (PDE)

$$f\left(\mathbf{x}, t; \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_N}, \frac{\partial u}{\partial t}; \frac{\partial^2 u}{\partial x_1^2}, \dots, \frac{\partial^2 u}{\partial x_1 \partial x_N}, \frac{\partial^2 u}{\partial x_1 \partial t}; \dots\right) = 0, \quad (1)$$

$$\mathbf{x} = (x_1, \dots, x_N) \in \Omega, \quad t \in [T_0, T],$$

where Ω is a subset of $\mathbb{R}^N$, and f is a nonlinear function of the solution $u(\mathbf{x}, t)$ and partial derivatives of space coordinate $\mathbf{x}$ and time coordinate t . Then, the initial and boundary conditions are given as

follows:

$$\begin{aligned} u(\mathbf{x}, T_0) &= u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\ u(\mathbf{x}, t) &= \mathcal{B}(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \Gamma \times [T_0, T], \end{aligned} \quad (2)$$

and Γ denotes the boundary of spatial region Ω .

In this paper, our emphasis lies on the forward problems of partial differential equations where PINN encounters challenges in achieving precise solutions.

• PINN:

The physics-informed neural network (PINN) method is an effective tool in solving forward problems of PDEs, i.e., the initial-boundary value problem considered here.

Construct a feedforward neural network with a depth of L , which consists of one input layer, $L - 1$ hidden layers, and one output layer. The l th ($l = 0, 1, \dots, L$) layer has N_l neurons, which represents that it transmits N_l -dimensional output vector $\mathbf{x}^l$ to the $(l + 1)$ th layer as the input. The connection between layers is achieved by the following affine transformation $\mathcal{A}$ and activation function $\sigma(\cdot)$:

$$\mathbf{x}^l = \sigma(\mathcal{A}_l(\mathbf{x}^{l-1})) = \sigma(\mathbf{w}^l \mathbf{x}^{l-1} + \mathbf{b}^l), \quad (3)$$

where $\mathbf{w}^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\mathbf{b}^l \in \mathbb{R}^{N_l}$ denote the weight matrix and bias vector of the l th layer, respectively. Thus, the relation between input $\mathbf{x}^0$ and output $u(\mathbf{x}^0, \Theta)$ is given by

$$u(\mathbf{x}^0, \Theta) = (\mathcal{A}_L \circ \sigma \circ \mathcal{A}_{L-1} \circ \dots \circ \sigma \circ \mathcal{A}_1)(\mathbf{x}^0) \quad (4)$$

and here $\Theta = \{\mathbf{w}^l, \mathbf{b}^l\}_{l=1}^L$ represents the trainable parameters of PINN. For the $N + 1$ -dimensional PDE mentioned in (1), the input $\mathbf{x}^0$ is the combination of spatial and temporal coordinates $(\mathbf{x}, t)$.

Assume we can obtain the training data, including the initial dataset $\{\mathbf{x}_k^i, T_0, u_k^i\}_{i=1}^{N_i}$ and boundary dataset $\{\mathbf{x}_k^b, t_k^b, u_k^b\}_{i=1}^{N_b}$ via simple random sampling method, and the set of collocation points $\{\mathbf{x}_k^r, t_k^r\}$ of the PDE residual

$$R := f\left(\mathbf{x}, t; \frac{\partial \hat{u}}{\partial x_1}, \dots, \frac{\partial \hat{u}}{\partial x_N}, \frac{\partial \hat{u}}{\partial t}; \frac{\partial^2 \hat{u}}{\partial x_1^2}, \dots, \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_N}, \frac{\partial^2 \hat{u}}{\partial x_1 \partial t}; \dots\right) \quad (5)$$

by Latin hypercube sampling approach.⁴⁵ Then, the loss function that reflects the initial-boundary conditions and the PDE residual is defined as follows:

$$\text{MSE} = w_i \text{MSE}_I(\mathbf{x}_k^i, T_0) + w_b \text{MSE}_B(\mathbf{x}_k^b, t_k^b) + w_r \text{MSE}_R(\mathbf{x}_k^r, t_k^r), \quad (6)$$

$$\mathbf{x}_k^i \in \Omega, \quad (\mathbf{x}_k^b, t_k^b) \in \Gamma \times [T_0, T], \quad (\mathbf{x}_k^r, t_k^r) \in \Omega \times [T_0, T],$$

where

$$\begin{aligned} \text{MSE}_I(\mathbf{x}_k^i, T_0) &= \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}(\mathbf{x}_k^i, T_0, \Theta) - u_k^i|^2, \\ \text{MSE}_B(\mathbf{x}_k^b, t_k^b) &= \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}(\mathbf{x}_k^b, t_k^b, \Theta) - u_k^b|^2, \\ \text{MSE}_R(\mathbf{x}_k^r, t_k^r) &= \frac{1}{N_r} \sum_{k=1}^{N_r} |R(\mathbf{x}_k^r, t_k^r, \Theta)|^2. \end{aligned} \quad (7)$$

It measures the difference between the predicted values and the true values of each iteration. Here, we use $\hat{u}$ to represent the predicted solution and its partial derivatives of each order with respect to spatial and the temporal variables can be derived by automatic differentiation.⁴⁶ The neural network is trained to update weights and biases, and they need to be properly initialized before training. The bias vector is usually initialized to 0 and weight matrices can be initialized by some effective methods, like Xavier initialization,⁴⁷ He initialization,⁴⁸ and so on. Then, the trainable parameters of the neural network are iteratively updated to minimize the value of the loss function. There are many commonly used optimization algorithms, such as stochastic gradient descent (SGD), Adam, and L-BFGS.⁴⁹

• bc-PINN:

The backward compatible PINN (bc-PINN) is proposed for application scenarios where the accuracy of the PINN method significantly decreases, such as strong nonlinearity and high-order time-varying partial differential equations. This scheme uses a single neural network to sequentially solve PDE over successive time segments by retraining the same neural network, while satisfying all previously obtained solutions. It is, henceforth, referred as backward compatible PINN since this method ensures backward compatibility of the solution.

The sequential scheme of bc-PINN is shown in Fig. 1 and the specific steps are as follows.

First, the time domain $[T_0, T]$ is divided into $n_{\max}$ segments

$$[T_0, T_1], [T_1, T_2], \dots, [T_{n-1}, T_n], \dots, [T_{n_{\max}-1}, T_{n_{\max}} = T], \quad (8)$$

and then we represent the n th time interval as $\Delta T_n = [T_{n-1}, T_n]$ and obtain $n_{\max}$ sub-domains $\Omega \times \Delta T_n$. Therefore, the neural network needs to be repeatedly trained for $n_{\max}$ times, and we record the solution obtained each time as $\hat{u}^{(n)} \triangleq \hat{u}^{(n)}(\Theta_n^*)$, ($n = 1, \dots, n_{\max}$).

For the first sub-domain $\Omega \times [T_0, T_1]$, the PINN method is applied to minimize the following loss function:

$$\text{MSE}_{\Delta T_1} = w_i \text{MSE}_I(\mathbf{x}_k^i, T_0) + w_b \text{MSE}_B(\mathbf{x}_k^b, t_k^b) + w_r \text{MSE}_R(\mathbf{x}_k^r, t_k^r), \quad (9)$$

$$\mathbf{x}_k^i \in \Omega, \quad (\mathbf{x}_k^b, t_k^b) \in \Gamma \times [T_0, T_1], \quad (\mathbf{x}_k^r, t_k^r) \in \Omega \times [T_0, T_1]$$

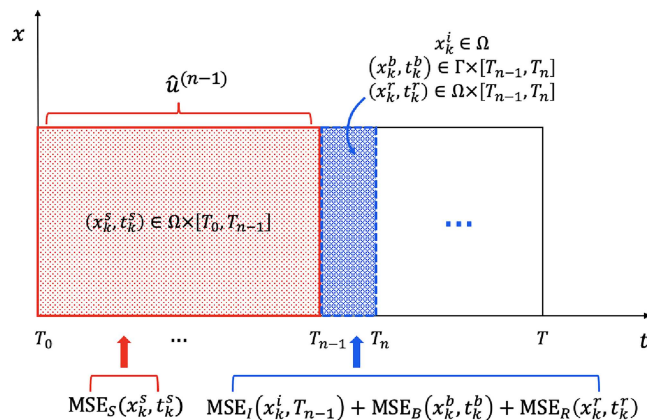


FIG. 1. Illustration of the bc-PINN for the one-dimensional scenario.

to obtain the solution $\hat{u}^{(1)} \triangleq \hat{u}^{(1)}(\Theta_1^*)$ of PDE, where $\Theta_1^* = \arg \min \text{MSE}_{\Delta T_1}(\Theta_1)$. The trained neural network is named subnet-1. $\text{MSE}_I(\mathbf{x}_k^i, T_0)$, $\text{MSE}_B(\mathbf{x}_k^b, t_k^b)$, and $\text{MSE}_R(\mathbf{x}_k^r, t_k^r)$ represent the errors of initial condition, boundary condition, and PDE constraints, respectively. The calculation formulas of which are similar to (7) except for replacing $\hat{u}$ with $\hat{u}^{(1)}$ and Θ with Θ_1 .

For all subsequent sub-domains $\Omega \times \Delta T_n$, ($n = 2, \dots, n_{\max}$), a new loss function is proposed, which differs in that it is enforced to satisfy the solution obtained from previous training

$$\begin{aligned} \text{MSE}_{\Delta T_n} = & w_i \text{MSE}_I(\mathbf{x}_k^i, T_{n-1}) + w_b \text{MSE}_B(\mathbf{x}_k^b, t_k^b) \\ & + w_r \text{MSE}_R(\mathbf{x}_k^r, t_k^r) + w_s \text{MSE}_S(\mathbf{x}_k^s, t_k^s), \\ & \mathbf{x}_k^i \in \Omega, \quad (\mathbf{x}_k^b, t_k^b) \in \Gamma \times [T_{n-1}, T_n], \\ & (\mathbf{x}_k^r, t_k^r) \in \Omega \times [T_{n-1}, T_n], \quad (\mathbf{x}_k^s, t_k^s) \in \Omega \times [T_0, T_{n-1}], \end{aligned} \quad (10)$$

where

$$\begin{aligned} \text{MSE}_S(\mathbf{x}_k^s, t_k^s) = & \frac{1}{N_s} \sum_{k=1}^{N_s} |\hat{u}^{(n)}(\mathbf{x}_k^s, t_k^s, \Theta_n) - \hat{u}^{(n-1)}(\mathbf{x}_k^s, t_k^s)|^2, \\ & \hat{u}^{(n-1)} \triangleq \hat{u}^{(n-1)}(\Theta_{n-1}^*), \\ & \Theta_{n-1}^* = \arg \min \text{MSE}_{\Delta T_{n-1}}(\Theta_{n-1}). \end{aligned} \quad (11)$$

Specifically, the solution $\hat{u}^{(n)}$ obtained by the n th trained network (subnet- n) is constrained to satisfy the predicted values of $\hat{u}^{(n-1)}$ obtained from the previous training at all previous time segments $[T_0, T_{n-1}]$. $\text{MSE}_S(\mathbf{x}_k^s, t_k^s)$ measures departure between the current predicted solution and the one from the previous training, and the introduction of this term ensures backward compatibility.

Finally, the result of bc-PINN is the predicted solution of the neural network trained for the last time in the entire spatiotemporal region, i.e.,

$$\hat{u}(\mathbf{x}, t) = \hat{u}^{(n_{\max})}(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \Omega \times [T_0, T] \quad (12)$$

where $\hat{u}^{(n_{\max})}$ is obtained by training subnet- $n_{\max}$.

B. The improved bc-PINN (lbc-PINN) for reducing error accumulation

For schemes like bc-PINN that perform time piecewise training to sequentially solve PDEs, error accumulation is inevitable. Here, we propose an improved bc-PINN method to alleviate this phenomenon.

In the sequential scheme, the neural network training of the posterior sub-domain is based on the training results of the anterior ones, which is reflected in the initial constraint MSE_I and the constraint MSE_S that ensures backward compatibility. Regarding MSE_I , except for the initial values of the first sub-domain being accurate, those of all other regions are the predicted values of the previous trained neural network at the end of the time domain, known as the “pseudo-initial values.” However, this part of the error cannot be eliminated and can only be reduced by training the previous sub-domains with sufficient accuracy. However, for MSE_S , we can modify its form to reduce error accumulation. This term enforces the current predicted solution to approximate the solution obtained

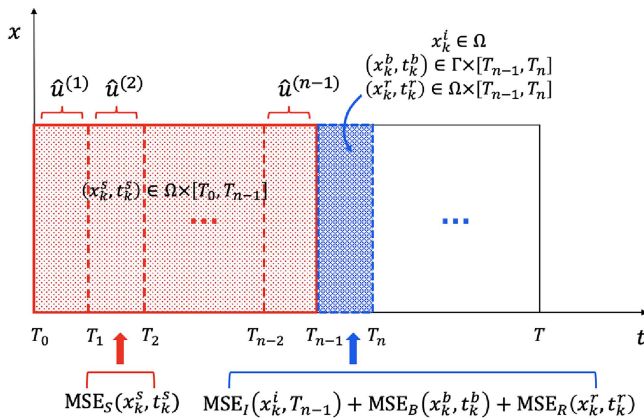


FIG. 2. Illustration of the improved bc-PINN for the one-dimensional scenario.

from the previous training at all previous time segments. Take $n = 3$ as an example and the corresponding MSE_S is

$$MSE_S(\mathbf{x}_k^s, t_k^s) = \frac{1}{N_s} \sum_{k=1}^{N_s} |\hat{u}^{(3)}(\mathbf{x}_k^s, t_k^s, \Theta_3) - \hat{u}^{(2)}(\mathbf{x}_k^s, t_k^s)|^2, \quad (13)$$

$$(\mathbf{x}_k^s, t_k^s) \in \Omega \times [T_0, T_2],$$

where the acquisition of $\hat{u}^{(2)}$ is based on the predicted values of $\hat{u}^{(1)}$ in $[T_0, T_1]$, rather than the exact solution of PDE. Therefore, there is unavoidable error propagation in the process from $\hat{u}^{(1)}$ to $\hat{u}^{(2)}$, and it is reasonable to believe that the accuracy of $\hat{u}^{(2)}$ in $[T_0, T_1]$ is lower than that of $\hat{u}^{(1)}$ in the high probability. Thus, it is advisable to make some modifications to the form of MSE_S so that the currently trained solution $\hat{u}^{(3)}$ approaches the predicted values of $\hat{u}^{(1)}$ in $[T_0, T_1]$ and $\hat{u}^{(2)}$ in $(T_1, T_2]$. Similarly, for the subsequent sub-domains, the prediction accuracy of $\hat{u}^{(n)}$ is probably higher than that of $\hat{u}^{(j)}, j = n + 1, \dots, n_{\max}$ in $\Omega \times (T_{n-1}, T_n]$. Accordingly, the improved bc-PINN makes the following modifications to MSE_S , and its sequential scheme is shown in Fig. 2.

Define the following set of subscripts:

$$\tau_1 = \{k | (\mathbf{x}_k^s, t_k^s) \in \Omega \times [T_0, T_1]\}, \quad (14)$$

$$\tau_j = \{k | (\mathbf{x}_k^s, t_k^s) \in \Omega \times (T_{j-1}, T_j]\}, j = 2, 3, \dots, n_{\max} - 1,$$

and then when $n = 2, \dots, n_{\max}$, MSE_S is modified into

$$MSE_S(\mathbf{x}_k^s, t_k^s) = \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{u}^{(n)}(\mathbf{x}_k^s, t_k^s, \Theta_n) - \hat{u}^{(j)}(\mathbf{x}_k^s, t_k^s)|^2, \quad (15)$$

where $\hat{u}^{(j)} \triangleq \hat{u}^{(j)}(\Theta_j^*)$, $\Theta_j^* = \arg \min MSE_{\Delta T_j}(\Theta_j)$.

To summarize, the loss function for the first sub-domain $\Omega \times [T_0, T_1]$ is given

$$MSE_{\Delta T_1} = w_i MSE_I(\mathbf{x}_k^i, T_0) + w_b MSE_B(\mathbf{x}_k^b, t_k^b) + w_r MSE_R(\mathbf{x}_k^r, t_k^r), \quad (16)$$

$$\mathbf{x}_k^i \in \Omega, \quad (\mathbf{x}_k^b, t_k^b) \in \Gamma \times [T_0, T_1], \quad (\mathbf{x}_k^r, t_k^r) \in \Omega \times [T_0, T_1],$$

where

$$MSE_I(\mathbf{x}_k^i, T_0) = \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}^{(1)}(\mathbf{x}_k^i, T_0, \Theta_1) - u_k^i|^2,$$

$$MSE_B(\mathbf{x}_k^b, t_k^b) = \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}^{(1)}(\mathbf{x}_k^b, t_k^b, \Theta_1) - u_k^b|^2, \quad (17)$$

$$MSE_R(\mathbf{x}_k^r, t_k^r) = \frac{1}{N_r} \sum_{k=1}^{N_r} |R(\mathbf{x}_k^r, t_k^r, \Theta_1)|^2,$$

and that for subsequent sub-domain $\Omega \times \Delta T_n (n = 2, \dots, n_{\max})$ is changed into

$$MSE_{\Delta T_n} = w_i MSE_I(\mathbf{x}_k^i, T_{n-1}) + w_b MSE_B(\mathbf{x}_k^b, t_k^b) + w_r MSE_R(\mathbf{x}_k^r, t_k^r) + w_s MSE_S(\mathbf{x}_k^s, t_k^s), \quad (18)$$

$$\mathbf{x}_k^i \in \Omega, \quad (\mathbf{x}_k^b, t_k^b) \in \Gamma \times [T_{n-1}, T_n],$$

$$(\mathbf{x}_k^r, t_k^r) \in \Omega \times [T_{n-1}, T_n], \quad (\mathbf{x}_k^s, t_k^s) \in \Omega \times [T_0, T_{n-1}],$$

where

$$MSE_I(\mathbf{x}_k^i, T_{n-1}) = \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}^{(n)}(\mathbf{x}_k^i, T_{n-1}, \Theta_n) - \hat{u}^{(n-1)}(\mathbf{x}_k^i, T_{n-1})|^2,$$

$$MSE_B(\mathbf{x}_k^b, t_k^b) = \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}^{(n)}(\mathbf{x}_k^b, t_k^b, \Theta_n) - u_k^b|^2,$$

$$MSE_R(\mathbf{x}_k^r, t_k^r) = \frac{1}{N_r} \sum_{k=1}^{N_r} |R(\mathbf{x}_k^r, t_k^r, \Theta_n)|^2, \quad (19)$$

$$MSE_S(\mathbf{x}_k^s, t_k^s) = \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{u}^{(n)}(\mathbf{x}_k^s, t_k^s, \Theta_n) - \hat{u}^{(j)}(\mathbf{x}_k^s, t_k^s)|^2,$$

$$\tau_1 = \{k | (\mathbf{x}_k^s, t_k^s) \in \Omega \times [T_0, T_1]\}, \quad \tau_j = \{k | (\mathbf{x}_k^s, t_k^s) \in \Omega \times (T_{j-1}, T_j]\},$$

$$j = 2, 3, \dots, n_{\max} - 1.$$

For the loss term $MSE_I(\mathbf{x}_k^i, T_{n-1})$, $\hat{u}^{(n-1)}$ serves the role of pseudo-initial values at $t = T_{n-1}$, and for $MSE_S(\mathbf{x}_k^s, t_k^s)$, $\hat{u}^{(1)}, \hat{u}^{(2)}, \dots, \hat{u}^{(n-1)}$ act as pseudo-reference solutions of the previous region $\Omega \times [T_0, T_{n-1}]$. The aforementioned weights w_i, w_b, w_r, w_s are used to scale the differences in the amplitude of each loss term. Note that the loss functions of bc-PINN and lbc-PINN in the first two sub-domains are the same, leading to identical training outcomes.

What is more, the final form of the predicted solution has also been improved

$$\hat{u}(\mathbf{x}, t) = \sum_{n=1}^{n_{\max}} \hat{u}^{(n)}(\mathbf{x}, t) \cdot \mathbb{1}_{\mathcal{D}_n}(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \Omega \times [T_0, T], \quad (20)$$

where

$$\mathcal{D}_n = \begin{cases} \Omega \times [T_0, T_1] & \text{if } n = 1, \\ \Omega \times (T_{n-1}, T_n] & \text{if } n = 2, 3, \dots, n_{\max}. \end{cases} \quad (21)$$

That is, we adopt the concatenation of solutions obtained from individual subnetworks as the final form of the predicted solution, rather

than relying solely on the solution learned by the last subnetwork like bc-PINN.

Finally, the workflow of Ibc-PINN can be briefly summarized in Fig. 3, where the evolution equation is taken as an example, and $\mathcal{N}[\cdot]$ represents the nonlinear operator involving various order derivatives with respect to spatial variables.

C. Comparison between bc-PINN and Ibc-PINN

We compare the methods theoretically before and after improvement from the following perspectives and the specific effects of two methods in practical applications are shown based on the numerical experimental results in the following two sections.

- **Accuracy** Ibc-PINN is an improved methodology building upon the foundation of bc-PINN with the aim of enhancing accuracy. In bc-PINN, the loss term MSE_S is introduced to ensure backward compatibility of the solution. Its reference baseline is the predicted solution obtained in the previous time segment, which

can be considered as a pseudo-reference solution across all prior regions with some margin of error. In addition, error propagation is unavoidable for algorithms performing time piecewise training to solve PDEs. Therefore, based on the characteristics of error propagation, we modify the loss term by selecting the earliest learned solution for each sub-domain as pseudo-reference solution. It not only reduces the error of the pseudo-reference solution used by Ibc-PINN in calculating the loss term MSE_S compared to bc-PINN but also helps to decelerate the speed of error accumulation, which contributes to improving the accuracy of the solution in subsequent training.

- **Efficiency** The main differences between Ibc-PINN and bc-PINN are reflected in two aspects: one is the data of pseudo-reference solution used in computing MSE_S , and the other is the final form of the predicted solution. All the data used originate from the training outcomes of intermediate subnetworks. Therefore, the improved method does not introduce almost any additional computational burden, and the impact on the efficiency of algorithm can be considered negligible.

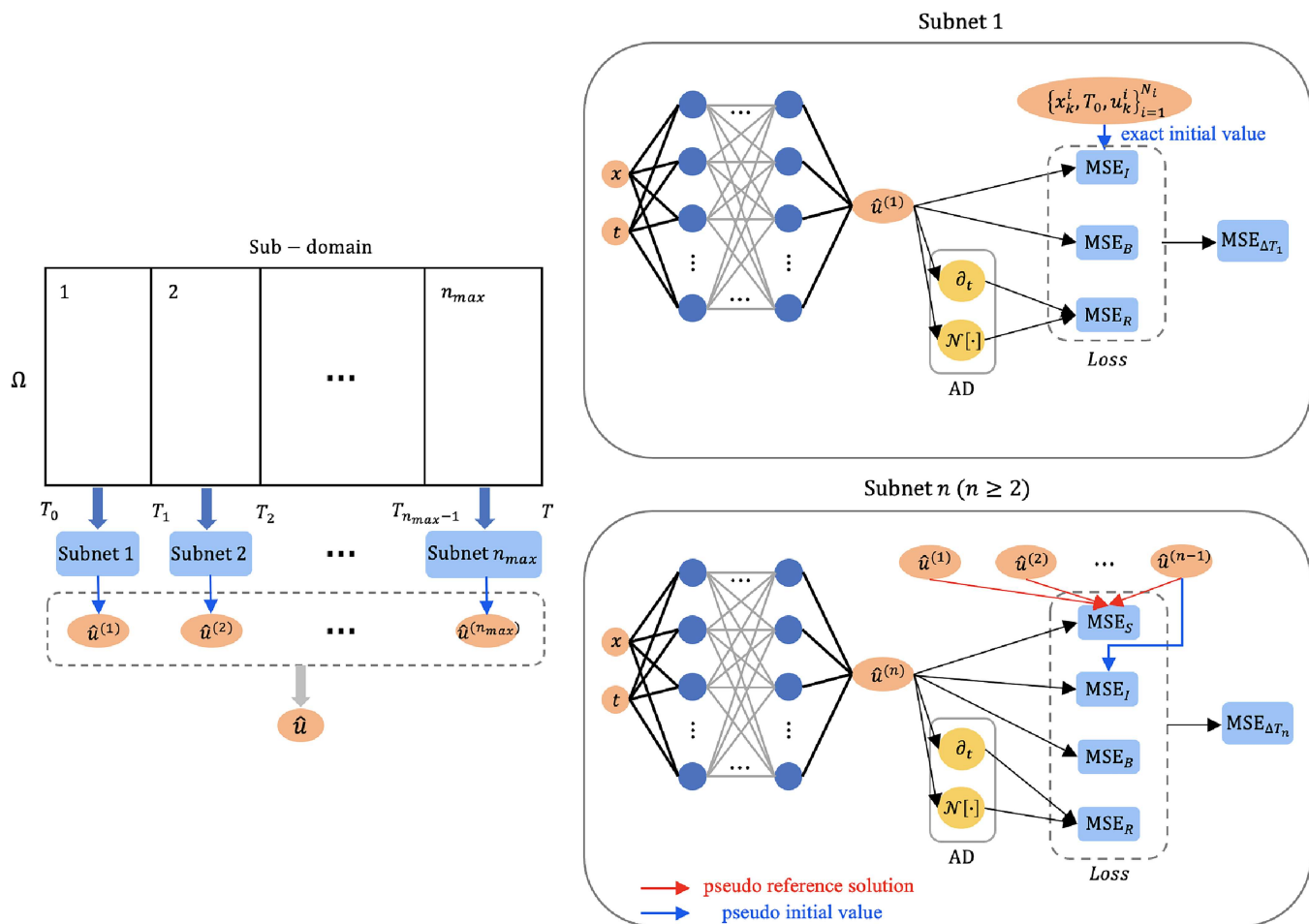


FIG. 3. Schematic diagram of Ibc-PINN.

- **Stability** By adopting the concatenation of solutions obtained from individual subnetworks as the final form of the predicted solution, the lbc-PINN is more stable since insufficient training in a specific subdomain only affects the accuracy of the predicted solutions from that subregion onward. The successful prediction results in previous sub-domains remains unaffected and will not be rendered futile. However, the results of bc-PINN rely entirely on the success or failure of training in the last sub-domain. This will result in an immediate severe deterioration of accuracy across all trained regions if the network training performs poorly in a specific subdomain.
- **Storage requirement of data** Unless there are additional analysis requirements, bc-PINN only needs to store the final weight matrices and bias vectors of the subnetwork from the last training. In contrast, lbc-PINN needs to store the training results of all subnetworks, leading to larger data storage requirements.

III. NUMERICAL EXPERIMENTS

Rogue wave is a steep wave that is localized in both spatial and temporal directions, with an amplitude of three times the height of the background field. Due to the complexity of the shape and structure, many rogue wave solutions, such as high-order rogue waves, are always difficult to obtain by using standard PINN. Generally speaking, the geometric structure of high-order rogue wave solutions can be categorized into fundamental, triangular, pentagonal, heptagonal shapes, and even more complex patterns. Since high-order rogue waves may exhibit multiple peaks in a certain spatiotemporal region, it is appropriate to use domain decomposition methods to capture their intricate dynamic behaviors.

In this section, we applied the lbc-PINN method to numerically simulate the first-order and higher-order rogue waves of the nonlinear Schrödinger equation (NLS) equation and the AB system and compared the results with bc-PINN to demonstrate the effectiveness of the improved method.

A. The nonlinear Schrödinger equation

The nonlinear Schrödinger equation is one of the most classical and well-known integrable equations, which contains many excellent properties. It can be employed to characterize the quantum behavior exhibited by microscopic particles in quantum mechanics⁵⁰ as well as nonlinear phenomena in other physical domains.^{51,52} Numerous commonly employed techniques,^{8,53,54} including the Darboux transformation, the Riemann Hilbert method, and so on have been utilized to acquire localized wave solutions for the NLS equation.

1. Data-driven first-order rogue waves

In this subsection, we consider the nonlinear Schrödinger equation

$$iq_t + q_{xx} + 2|q|^2q = 0, \quad (22)$$

where $q(x, t)$ is a complex-valued solution regarding spatial coordinate x and temporal coordinate t . In optics, the nonlinear term arises from the material's intensity-dependent index. This equation can be reduced from the extended nonlinear Schrödinger equation

by setting β to 0,

$$iU_t + U_{xx} + 2|U|^2U + 2i\beta(|U|^2U_x - U^2U_x^*) - 2\beta U \int (|U|^2)_t dx = 0. \quad (23)$$

Through the generalized Darboux transformation, the N th-order rogue wave $U[N]$ for the extended nonlinear Schrödinger equation is derived in Ref. 55, and thus, we can get the corresponding arbitrary-order rogue wave solution $q[N]$ of the standard NLS equation by taking $\beta = 0$.

- $\beta = 0$

By constructing the first-step generalized Darboux transformation, the first-order rogue wave solution for the extended NLS equation is presented as

$$U[1] = \exp[i\theta] \frac{F_1 + iG_1}{D_1}, \quad (24)$$

where

$$\begin{aligned} \theta &= -2\beta x + (4\beta^2 + 2)t, \\ F_1 &= -4x^2 - 32\beta tx - 16(4\beta^2 + 1)t^2 + 3, \quad G_1 = 16t, \\ D_1 &= 4x^2 + 32\beta tx + 16(4\beta^2 + 1)t^2 + 1. \end{aligned} \quad (25)$$

After choosing $\beta = 0$, we arrive at the explicit expression of the corresponding first-order rogue wave $q(x, t)$ for the NLS equation.

The computational domain is taken as $\Omega = [x_0, x_1] = [-2, 2]$ and $[T_0, T] = [-1, 1]$. Then, the initial and boundary conditions are obtained based on the above exact solution. For all the examples showcased here, we choose a straightforward approach by uniformly dividing the time domain. Recognizing that an excessively fine partition might escalate computational costs, while an overly coarse partition could compromise accuracy, we adopt a moderate division based on the size of the selected time domain and the complexity of the solution. The time domain is divided into $n_{\max} = 4$ segments,

$$\begin{aligned} [T_0, T_1] &= [-1, -0.5], \quad [T_1, T_2] = [-0.5, 0], \\ [T_2, T_3] &= [0, 0.5], \quad [T_3, T] = [0, 0.5]. \end{aligned} \quad (26)$$

We divide the whole spatial and time region into 512 and 201 discrete equidistance points, respectively. Then, the solution is discretized into 512×201 data points in the grid points to generate the discretized dataset. In each sub-domain, $N_i = 128$ and $N_b = 50$ training points are randomly selected from the initial-boundary dataset. To reiterate, only the initial values of the first sub-domain are taken from the exact solution, while those of the other sub-domains are the predicted values of the solution obtained from the previous training. In addition, $N_r = 20\,000$ collocation points are generated by means of the Latin hypercube sampling method. It should be noted that as the trained sub-domain gradually move backwards, N_s will increase since the range of previously trained regions expands. Here, we set $N_s = 128 \times 50 \times (n - 1)$ for the n th sub-domain ($n = 2, 3, 4$).

In view of the complexity of the structure of complex-valued solution $q(x, t)$, it is decomposed into real and imaginary parts, i.e.,

$q(x, t) = u(x, t) + iv(x, t)$, which correspond to the two outputs of the neural network. After substituting it into (22), the PDE residual R can be divided into two corresponding parts,

$$\begin{aligned} R_u &:= -v_t + u_{xx} + 2(u^2 + v^2)u, \\ R_v &:= u_t + v_{xx} + 2(u^2 + v^2)v. \end{aligned} \quad (27)$$

We choose the weights $w_i = w_b = w_r = w_s = 1$ for all the examples presented in this paper to facilitate the analysis of the effect of

the improvements. For the first sub-domain, the PINN method is applied to minimize the following loss function:

$$\begin{aligned} \text{MSE}_{\Delta T_1} &= \text{MSE}_I(x_k^i, T_0) + \text{MSE}_B(x_k^b, t_k^b) + \text{MSE}_R(x_k^r, t_k^r), \\ T_0 &= -1, \quad x_k^i \in [-2, 2], \quad (x_k^b, t_k^b) \in \{-2, 2\} \times [-1, -0.5], \\ (x_k^r, t_k^r) &\in [-2, 2] \times [-1, -0.5], \end{aligned} \quad (28)$$

where

$$\begin{aligned} \text{MSE}_I(x_k^i, T_0) &= \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}^{(1)}(x_k^i, T_0, \Theta_1) - u_k^i|^2 + \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{v}^{(1)}(x_k^i, T_0, \Theta_1) - v_k^i|^2, \\ \text{MSE}_B(x_k^b, t_k^b) &= \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}^{(1)}(x_k^b, t_k^b, \Theta_1) - u_k^b|^2 + \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{v}^{(1)}(x_k^b, t_k^b, \Theta_1) - v_k^b|^2, \\ \text{MSE}_R(x_k^r, t_k^r) &= \frac{1}{N_r} \sum_{k=1}^{N_r} |R_u(x_k^r, t_k^r, \Theta_1)|^2 + \frac{1}{N_r} \sum_{k=1}^{N_r} |R_v(x_k^r, t_k^r, \Theta_1)|^2, \end{aligned} \quad (29)$$

and $\hat{u}^{(1)}$ and $\hat{v}^{(1)}$ denote the predicted solutions by training subnet-1. $\{x_k^i, T_0, u_k^i, v_k^i\}_{k=1}^{128}$ and $\{x_k^b, t_k^b, u_k^b, v_k^b\}_{k=1}^{50}$ denote the initial and boundary data, respectively. For subsequent three sub-domains ($n = 2, 3, 4$), MSE_S is added into the loss function

$$\begin{aligned} \text{MSE}_{\Delta T_n} &= \text{MSE}_I(x_k^i, T_{n-1}) + \text{MSE}_B(x_k^b, t_k^b) + \text{MSE}_R(x_k^r, t_k^r) + \text{MSE}_S(x_k^s, t_k^s), \\ x_k^i &\in [-2, 2], \quad (x_k^b, t_k^b) \in \{-2, 2\} \times [T_{n-1}, T_n], \quad (x_k^r, t_k^r) \in [-2, 2] \times [T_{n-1}, T_n], \quad (x_k^s, t_k^s) \in [-2, 2] \times [-1, T_{n-1}], \\ \tau_1 &= \{k | (x_k^s, t_k^s) \in [-2, 2] \times [-1, T_1]\}, \quad \tau_j = \{k | (x_k^s, t_k^s) \in [-2, 2] \times (T_{j-1}, T_j]\} (j = 2, \dots, n-1), \end{aligned} \quad (30)$$

where

$$\begin{aligned} \text{MSE}_I(x_k^i, T_{n-1}) &= \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}^{(n)}(x_k^i, T_{n-1}, \Theta_n) - \hat{u}^{(n-1)}(x_k^i, T_{n-1})|^2 + \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{v}^{(n)}(x_k^i, T_{n-1}, \Theta_n) - \hat{v}^{(n-1)}(x_k^i, T_{n-1})|^2, \\ \text{MSE}_B(x_k^b, t_k^b) &= \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}^{(n)}(x_k^b, t_k^b, \Theta_n) - u_k^b|^2 + \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{v}^{(n)}(x_k^b, t_k^b, \Theta_n) - v_k^b|^2, \\ \text{MSE}_R(x_k^r, t_k^r) &= \frac{1}{N_r} \sum_{k=1}^{N_r} |R_u(x_k^r, t_k^r, \Theta_n)|^2 + \frac{1}{N_r} \sum_{k=1}^{N_r} |R_v(x_k^r, t_k^r, \Theta_n)|^2, \\ \text{MSE}_S(x_k^s, t_k^s) &= \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{u}^{(n)}(x_k^s, t_k^s, \Theta_n) - \hat{u}^{(j)}(x_k^s, t_k^s)|^2 + \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{v}^{(n)}(x_k^s, t_k^s, \Theta_n) - \hat{v}^{(j)}(x_k^s, t_k^s)|^2, \end{aligned} \quad (31)$$

$\hat{u}^{(n)}$ and $\hat{v}^{(n)}$ are the predicted solutions by training subnet- n in the n th sub-domain.

Here, we construct a seven-layer feedforward neural network with 128 neurons per hidden layer and use the tanh activation function to learn the first-order rogue wave solution for the NLS equation. The loss functions of subnet-1 and subnet- n ($n = 2, 3, 4$) have been given in (28) and (30), respectively. For subnet-1, the weights are initialized with Xavier initialization method and the

biases is initialized to 0. For the subsequent subnetworks, we adopt transfer learning⁵⁶ and weight freezing techniques. To be specific, the latter subnet inherits the saved weight matrixes and bias vectors of the previous subnet at the end of the iteration process as the initialization parameters, and thus the subsequent training is based on the previous results by leveraging the transfer learning technique instead of training from scratch. In addition, we use weight freezing to freeze the parameters of the first two layers to prevent them from

updating in subsequent training, while only training the weights and biases of the subsequent layers. The advantage of this technique is that it can retain the features learned by the pre-trained model, reduce the training time, and prevent the model from overfitting.

In this study, the first-order optimization algorithm (Adam) is first adopted to train the neural network with a certain number of iterations and then proceed by using the second-order algorithm (L-BFGS) to reach the ideal local optimal solution. This training strategy that combines first-order and second-order optimization algorithms can balance efficiency and accuracy. The number of iterations for the Adam optimizer is taken as 10 000 here. Meanwhile, the initial condition guided learning (ICGL) technique²⁹ is used here in order to better learn the initial conditions. The core idea is to employ a fraction of the overall iterations to train the neural network, focusing solely on matching the initial conditions of that specific time interval. Specifically, the loss function for ICGL in the first sub-domain is as follows:

$$\begin{aligned} \text{MSE}_{SI} = & \frac{1}{N_{SI}} \sum_{k=1}^{N_{SI}} |\hat{u}^{(1)}(x_k^{si}, t_k^{si}, \Theta_1) - u_k^{si}|^2 \\ & + \frac{1}{N_{SI}} \sum_{k=1}^{N_{SI}} |\hat{v}^{(1)}(x_k^{si}, t_k^{si}, \Theta_1) - v_k^{si}|^2, \end{aligned} \quad (32)$$

$$(x_k^{si}, t_k^{si}) \in \Omega \times [T_0, T_1],$$

while that in n th sub-domain ($n \geq 2$) is defined as

$$\begin{aligned} \text{MSE}_{SI} = & \frac{1}{N_{SI}} \sum_{k=1}^{N_{SI}} |\hat{u}^{(n)}(x_k^{si}, t_k^{si}, \Theta_n) - \hat{u}^{(n-1)}(x_k^{si}, T_{n-1})|^2 \\ & + \frac{1}{N_{SI}} \sum_{k=1}^{N_{SI}} |\hat{v}^{(n)}(x_k^{si}, t_k^{si}, \Theta_n) - \hat{v}^{(n-1)}(x_k^{si}, T_{n-1})|^2, \end{aligned} \quad (33)$$

$$(x_k^{si}, t_k^{si}) \in \Omega \times [T_{n-1}, T_n],$$

where $\{u_k^{si}, v_k^{si}\}$ is the initial data at the point (x_k^{si}, T_0) and $\hat{u}^{(n-1)}(x_k^{si}, T_{n-1})$ ($\hat{v}^{(n-1)}(x_k^{si}, T_{n-1})$) denotes the value of the predicted solution $\hat{u}^{(n-1)}$ ($\hat{v}^{(n-1)}$) at $t = T_{n-1}$. The neural network is trained using 10% of the total number of iterations of the Adam

optimizer, and we choose $N_{SI} = 128 * 51$ to match only the initial condition of that time segment. The dynamic behavior of first-order rogue wave has been successfully learned by Ibc-PINN and the relative $\mathbb{L}_2$ error of $|q(x, t)|$ is 4.843×10^{-4} . The density plot, the absolute error and the 3d plot of the data-driven rogue wave solution are displayed in Fig. 4. It can be seen that there is one peak and two valleys, with the highest value of the peak roughly appearing near the point $(0, 0)$.

$$\bullet \beta = \frac{1}{3}$$

The NLS equation is a special case when $\beta = 0$ of the Kundu–Eckhaus (KE) equation^{57,58}

$$iq_t + q_{xx} + 2|q|^2 q + 4\beta^2 |q|^4 q - 4i\beta (|q|^2)_x q = 0, \quad (34)$$

which contains quintic nonlinearity and the Raman effect in nonlinear optics. The Kundu–Eckhaus equation can be converted to the extended NLS equation above by means of a gauge transformation. Therefore, the following formula:

$$q[N] = U[N] \exp \left[2i\beta \int |U[N]|^2 dx \right] \quad (35)$$

gives rise to the corresponding N th-order rogue waves for the Kundu–Eckhaus equation after deriving the rogue wave solutions for the extended NLS equation through the generalized Darboux transformation.⁵⁵

Here is the explicit first-order rogue wave solution of the Kundu–Eckhaus equation

$$q[1] = U[1] \exp \left[2i\beta \int |U[1]|^2 dx \right], \quad (36)$$

where $U[1] = \exp[i\theta] \frac{F_1 + iG_1}{D_1}$, $\int |U[1]|^2 dx = \frac{H_1}{D_1}$ with θ, F_1, G_1, D_1 shown in (25) and

$$H_1 = 4x^3 + 16(4\beta^2 + 1)t^2x + 9x + 32\beta(x^2 + 1)t. \quad (37)$$

We select the spatiotemporal region as $\Omega \times [T_0, T] = [-1.5, 1.5] \times [-0.5, 0.5]$, $\beta = \frac{1}{3}$ and divide the whole spatiotemporal region

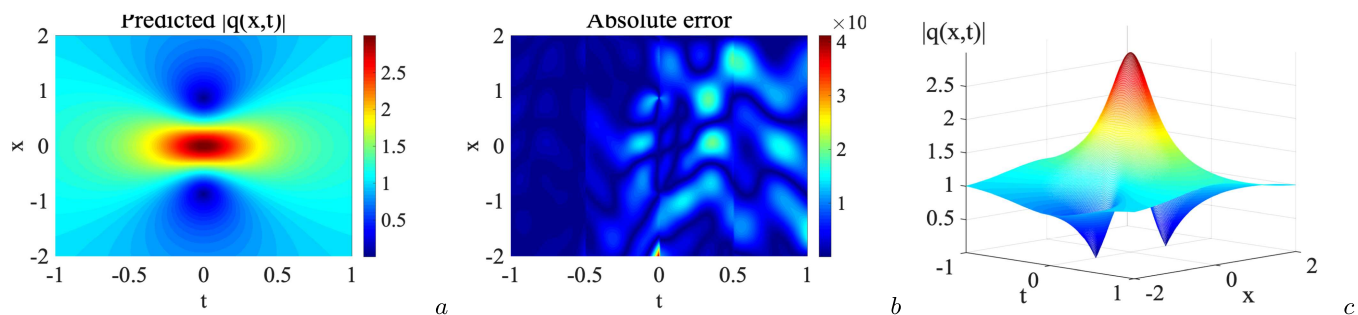


FIG. 4. Data-driven first-order rogue wave ($\beta = 0$) by Ibc-PINN: (a) the density diagram of the predicted solution $|q(x, t)|$; (b) the density diagram of absolute error; (c) the three-dimensional plot of the data-driven wave solution $|q(x, t)|$.

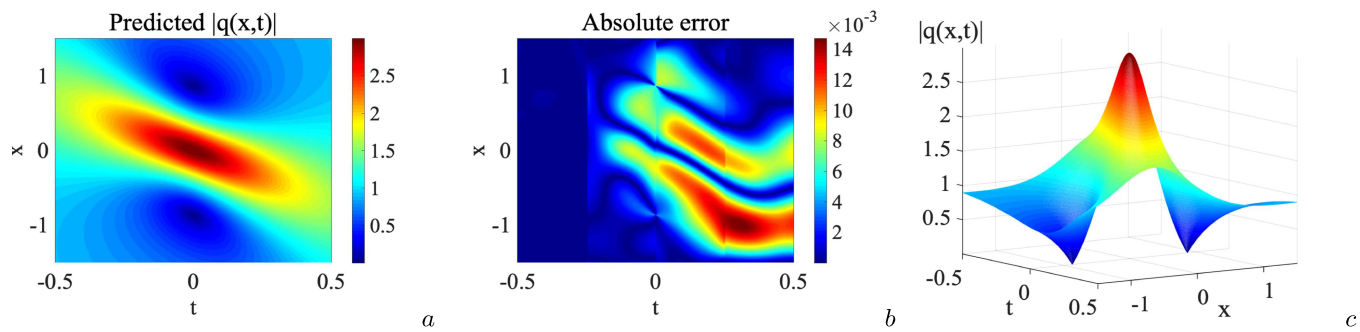


FIG. 5. Data-driven first-order rogue wave ($\beta = \frac{1}{3}$) by lbc-PINN: (a) the density diagram of the predicted solution $|q(x, t)|$; (b) the density diagram of absolute error; (c) the three-dimensional plot of the data-driven rogue wave solution $|q(x, t)|$.

into $n_{\max} = 4$ sub-domains,

$$\begin{aligned} &[-1.5, 1.5] \times [-0.5, -0.25], \quad [-1.5, 1.5] \times [-0.25, 0], \\ &[-1.5, 1.5] \times [0, 0.25], \quad [-1.5, 1.5] \times [0.25, 0.5]. \end{aligned} \quad (38)$$

Then, the PDE residuals are changed into

$$\begin{aligned} R_u &:= -v_t + u_{xx} + 2(u^2 + v^2)u + 4\beta^2(u^2 + v^2)^2u \\ &\quad + 4\beta(2uu_x + 2vv_x)v, \\ R_v &:= u_t + v_{xx} + 2(u^2 + v^2)v + 4\beta^2(u^2 + v^2)^2v \\ &\quad - 4\beta(2uu_x + 2vv_x)u. \end{aligned} \quad (39)$$

Other details are similar to that of $\beta = 0$ except for changing the number of nodes in each hidden layer to 64, so we omit them here.

The relative $\mathbb{L}_2$ error of the data-driven solution $|q(x, t)|$ for the KE equation obtained by lbc-PINN is 3.789×10^{-3} . Although the network structure and other parameter selection of above two first-order rogue wave solutions are consistent on the whole, the data-driven solution with $\beta = \frac{1}{3}$ has lower precision even with smaller selected training domain. The reason behind this is that the form of the governing function is more complex, thereby increasing the difficulty of training when $\beta \neq 0$, which involves quintic and Raman-effect nonlinear terms. According to Fig. 5, the shape of the rogue wave does not change drastically but it rotates a certain angle in a clockwise direction as β changes from zero to non zero.

Finally, relative $\mathbb{L}_2$ errors generated by bc-PINN and lbc-PINN are summarized in Table I for intuitive comparison. The lbc-PINN method has improved accuracy, especially for the case when $\beta = 0$.

2. Data-driven second-order rogue waves

Based on the above derivation, the second-order rogue wave solution of the extended NLS equation naturally takes the form

$$U[2] = \exp[i\theta] \frac{F_2 + iG_2}{D_2}, \quad (40)$$

where the concrete expressions of F_2, G_2, D_2 are not detailed here due to space limitations, and they contain free parameters (m_1 and n_1) and were given in Ref. 55. After taking $\beta = 0$, we obtain the

corresponding second-order rogue wave solution $q(x, t)$ for the NLS equation.

Once the spatiotemporal region is selected, the corresponding initial and boundary conditions can be derived from the exact solution. After acquiring the initial-boundary data obtained by taking two different sets of free parameters (m_1 and n_1), the lbc-PINN method is utilized to learn second-order rogue wave solutions.

- $m_1 = 0, n_1 = 0$

Here, the spatiotemporal region $\Omega \times [T_0, T] = [-2, 2] \times [-0.8, 0.8]$ are also equally divided into four sub-domains. The loss function (28) is employed for optimizing subnet-1 in the first time segment $[-0.8, -0.4]$, while the loss (30) for subnet- n ($n = 2, 3, 4$) in subsequent ones ($[-0.4, 0]$, $[0, 0.4]$, $[0.4, 0.8]$).

We establish a neural network with the depth of 7 and width of 128. Other parameters are selected as $N_i = 128$, $N_b = 50$, $N_r = 20000$, and incremental $N_s = 128 \times 50 \times (n - 1)$ for the n th sub-domain. The effective techniques mentioned above have also been adopted, including the initial condition guided learning (ICGL), transfer learning and weight freezing, and specific details are consistent with Subsection III A 1.

When $m_1 = 0, n_1 = 0$, we successfully simulate the fundamental second-order rogue wave solution by lbc-PINN shown in Fig. 6. The maximum value of $|q(x, t)|$ is approximately 5, reaching at point (0, 0).

- $m_1 = 10, n_1 = 0$

Under this set of parameter values, the morphology of the rogue wave solution becomes more complex. Thus, we will divide

TABLE I. Relative $\mathbb{L}_2$ errors of the data-driven first-order rogue waves for the NLS and KE equations by bc-PINN and lbc-PINN.

	Ω	$[T_0, T]$	Relative $\mathbb{L}_2$ error of $ q(x, t) $	
			bc-PINN	lbc-PINN
$\beta = 0$	$[-2, 2]$	$[-1, 1]$	1.062×10^{-3}	4.843×10^{-4}
$\beta = \frac{1}{3}$ (KE)	$[-1.5, 1.5]$	$[-0.5, 0.5]$	4.471×10^{-3}	3.789×10^{-3}

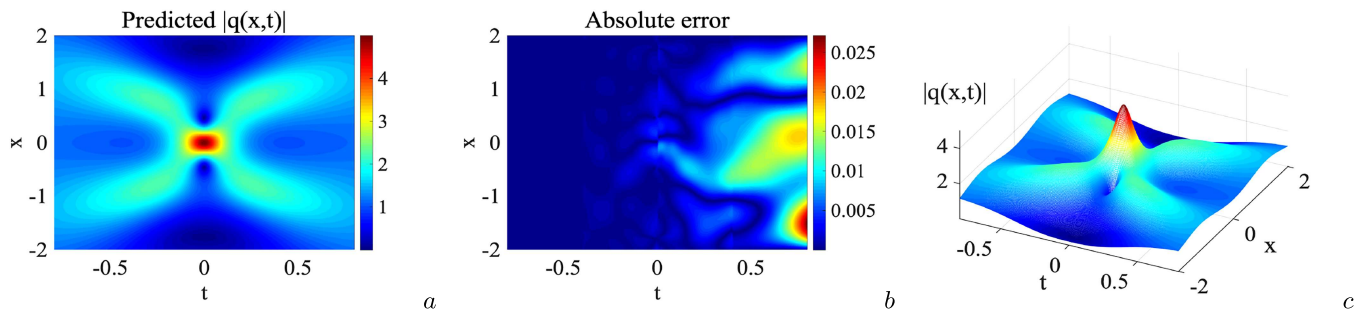


FIG. 6. Data-driven second-order rogue wave ($m_1 = 0, n_1 = 0$) by lbc-PINN: (a) The density diagram of the predicted solution $|q(x, t)|$; (b) The density diagram of absolute error; (c) The three-dimensional plot of the data-driven rogue wave solution $|q(x, t)|$.

the selected time domain $[T_0, T] = [-1.5, 1.5]$ into $n_{\max} = 6$ equal parts to apply the lbc-PINNs. Take the space range as $\Omega = [-4, 4]$ and choose $N_i = 256$ due to the expansion of the spatial region. Accordingly, N_s is changed to $256 \times 50 \times (n - 1)$ and we select $N_{sl} = 256 \times 51$ for the n th sub-domain. Other parameter settings, including network structure, are consistent with $m_1 = n_1 = 0$.

By means of lbc-PINN, data-driven second-order rogue wave is displayed in Fig. 7. It vividly illustrates that the fundamental second-order rogue wave is separated into three first-order rogue waves: a single and a double spatial hump. We also observe that the single hump appears at $t \approx -1$ and rapidly decays while two spatial humps rise up simultaneously at $t \approx 0.6$. The spatial coordinates corresponding to the three peaks are $x = 0, x \approx \pm 2$ separately.

Comparing the performance of the two methods shown in Table II, the error of bc-PINN is reduced to approximately half of its original value after improvement.

3. Data-driven third-order rogue waves

Similarly, after denoting

$$\begin{aligned} \Psi_1[l] = & \Psi_1^{[0]} + \sum_{j=1}^l T_1[j] \Psi_1^{[1]} + \sum_{j=1}^l \sum_{k=1}^{j-1} T_1[j] T_1[k] \Psi_1^{[2]} \\ & + \dots + T_1[l] T_1[l-1] \dots T_1[1] \Psi_1^{[l]}, \end{aligned} \quad (41)$$

the N th-step generalized Darboux transformation results in

$$\Psi[N] = T[N] T[N-1] \dots T[1] \Psi, \quad (42)$$

$$T[l] = \zeta I - H[l-1] \Lambda_l H[l-1]^{-1}, \quad (43)$$

$$U[N] = U[0] - 2i(\zeta_1 - \zeta_1^*) \sum_{l=0}^{N-1} \frac{\psi_1[l] \phi_1[l]^*}{(|\psi_1[l]|^2 + |\phi_1[l]|^2)}, \quad (44)$$

where

$$(\psi_1[l], \phi_1[l])^T = \Psi_1[l], \quad (45)$$

$$H[l-1] = \begin{pmatrix} \psi_1[l-1] & -\phi_1[l-1]^* \\ \phi_1[l-1] & \psi_1[l-1]^* \end{pmatrix}, \quad (46)$$

$$\Lambda_l = \begin{pmatrix} \zeta_1 & 0 \\ 0 & \zeta_1^* \end{pmatrix}, 1 \leq l \leq N \quad (47)$$

and can derive the N -order rogue wave solution $U[N]$ for the extended NLS equation. The specific details of the parameters involved have been presented in Ref. 55 and will not be elaborated here. Then, we choose $\beta = 0, N = 3$ and simulate the third-order rogue wave solution $q(x, t)$ for the NLS equation by lbc-PINN.

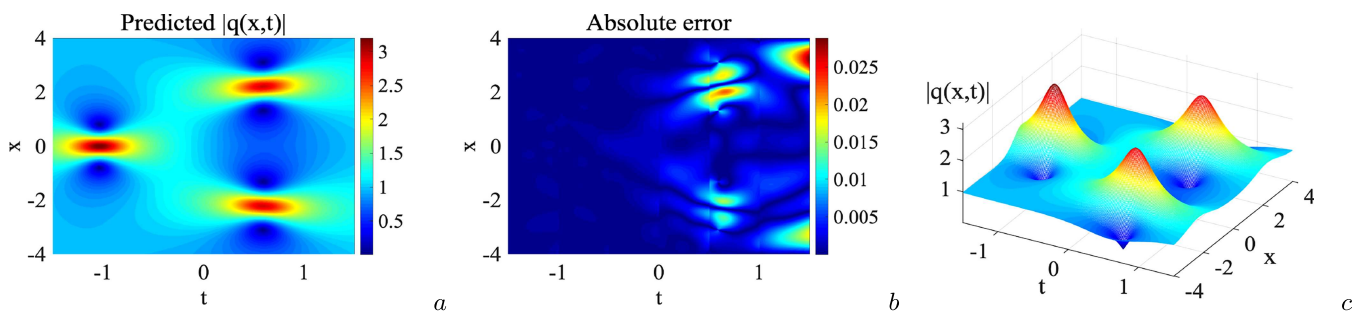


FIG. 7. Data-driven second-order rogue wave ($m_1 = 10, n_1 = 0$) by lbc-PINN: (a) the density diagram of the predicted solution $|q(x, t)|$; (b) the density diagram of absolute error; (c) the three-dimensional plot of the data-driven rogue wave solution $|q(x, t)|$.

TABLE II. Relative $\mathbb{L}_2$ errors of the data-driven second-order rogue waves for the NLS equation by bc-PINN and lbc-PINN.

	Ω	$[T_0, T]$	Relative $\mathbb{L}_2$ error of $ q(x, t) $	
			bc-PINN	lbc-PINN
$m_1 = 0$	$[-2, 2]$	$[-0.8, 0.8]$	6.368×10^{-3}	3.690×10^{-3}
$m_1 = 10$	$[-4, 4]$	$[-1.5, 1.5]$	6.580×10^{-3}	3.543×10^{-3}

- $m_1 = 10, n_1 = 0, m_2 = 0, n_2 = 0$

We divide the spatiotemporal region $\Omega \times [T_0, T] = [-5, 5] \times [-2, 1.5]$ into $n_{\max} = 7$ sub-domains with a time step of 0.5. The parameters of each dataset in the loss function are taken as $N_i = 128, N_b = 50, N_r = 20\,000$ and $N_s = 128 \times 50 \times (n - 1)$ for the n th sub-domain. Afterward, a neural network with a depth of 7 and a width of 64 is established to learn dynamic behaviors of the solution in each subregion. We opt for a blend of the Adam and L-BFGS optimizers, commencing with 10 000 iterations using the Adam optimization algorithm, followed by subsequent iterations employing the L-BFGS algorithm until convergence. Additionally, during the initial training phase in each subregion, we leverage ICGL technology to match the initial conditions. This step accounts for 10% of the Adam iteration count and N_{SI} is taken as 128×51 . Similarly, techniques such as transfer learning and weight freezing are employed to accelerate the training speed of the network, and the number of frozen weight layers is set to 2.

The density diagrams of the predicted $|q(x, t)|$ obtained by bc-PINN and lbc-PINN as well as the curve plots to show

comparison between the predicted and exact solutions at the three temporal snapshots are depicted in Fig. 8. Under this parameter selection of $m_1 = 10, n_1 = 0, m_2 = 0, n_2 = 0$, the fundamental third-order rogue wave splits into a triangular arrangement of six first-order rogue waves. A single hump forms at $t \approx -1.75$, followed by the symmetric appearance of two rogue waves around $t \approx -0.5$, and a triple spatial hump rapidly rises at $t \approx 1.1$ and $t \approx 1.2$. We notice that the contour lines of the density plot for the third-order rogue wave obtained by bc-PINN are irregular and lack smoothness. Moreover, as shown in Fig. 8(b), there is a certain gap between the predicted solution and the exact solution curve. In contrast, the predictive outcomes of lbc-PINN demonstrate a notable concordance with the reference solution. The absolute error plots in Fig. 9 also reveals the difference in accuracy between these two methods.

- $m_1 = 0, n_1 = 0, m_2 = 50, n_2 = 0$

Here, we take the number of sub regions $n_{\max}$ as 6 and each subnet contains six hidden layers, with 128 neurons per hidden layer. Other than this, all other configurations remain consistent with the previous example. Ultimately, Fig. 10 illustrates third-order rogue wave of pentagonal pattern reconstructed by using lbc-PINN. Table III summarizes the relative errors of data-driven third-order rogue waves for these two patterns, with lbc-PINN maintaining higher accuracy compared to the unimproved method.

B. The AB system

The AB system, initially proposed by Pedlosky⁴⁴ using singular perturbation theory, serves as model equations to depict marginally

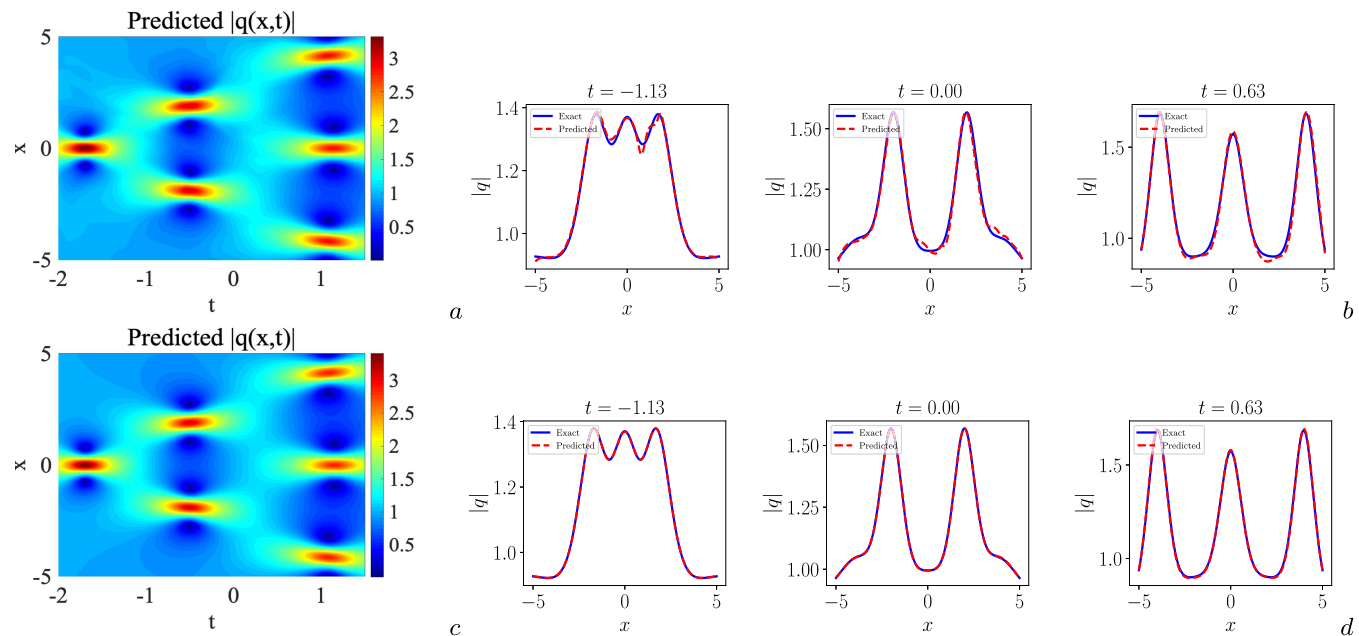


FIG. 8. Data-driven third-order rogue wave ($m_1 = 10, n_1 = m_2 = n_2 = 0$): the density diagrams of the predicted solution $|q(x, t)|$: (a) by bc-PINN and (c) by lbc-PINN; comparison between the predicted and exact solutions at the three temporal snapshots of $|q(x, t)|$: (b) by bc-PINN and (d) by lbc-PINN.

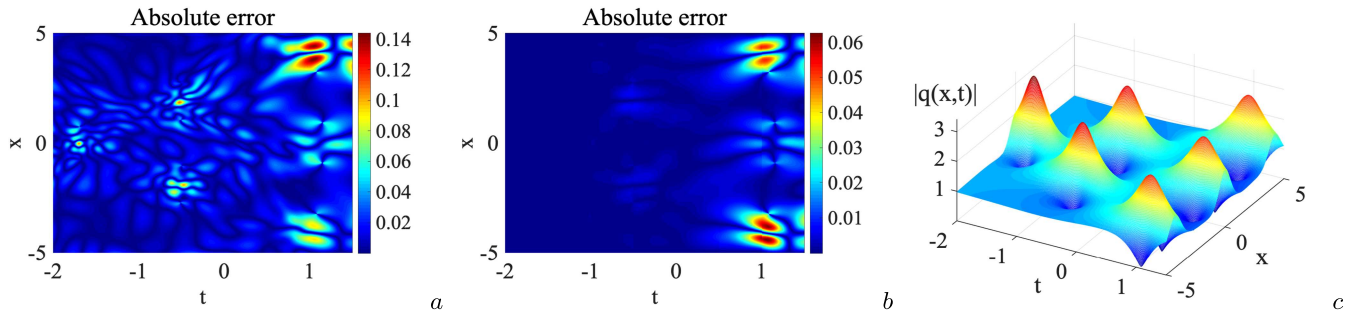


FIG. 9. Data-driven third-order rogue wave ($m_1 = 10, n_1 = m_2 = n_2 = 0$): the density diagrams of absolute error: (a) by bc-PINN and (b) by lbc-PINN; (c) the three-dimensional plot of the data-driven rogue wave solution $|q(x, t)|$ by lbc-PINN.

unstable baroclinic wave packets^{59,60} and ultra-short pulses in nonlinear optics.⁶¹ The study of the dynamic properties of the AB system has yielded rich results, including the envelope solitary waves and periodic waves,⁶² the Lax pair,⁶³ the conservation laws, modulational instability and breather solutions via the Darboux transformation,⁶⁴ the N -soliton solutions by using the dressing method,⁶⁵ and rogue wave solutions via the generalized Darboux transformation.⁶⁶

1. Data-driven first-order rogue waves

We investigate the AB system in the canonical form⁶³

$$A_{xt} = AB, \quad (48)$$

$$B_x = -\frac{1}{2}(|A|^2)_t. \quad (49)$$

Here, A and B are wave amplitudes satisfying the normalization condition

$$|A_t|^2 + B^2 = 1, \quad (50)$$

and x and t represent semi-characteristic normalized coordinates. By using the generalized Darboux transformation, a unified formula

of N th-order rogue wave solution for the AB system is given⁶⁶

$$A[N] = A[0] - 4i(\lambda_1 - \lambda_1^*) \sum_{j=0}^{N-1} \frac{\psi_1[j] \phi_1[j]^*}{(|\psi_1[j]|^2 + |\phi_1[j]|^2)}, \quad (51)$$

$$B[N] = B[0] - 4i(\lambda_1 - \lambda_1^*) \times \sum_{j=0}^{N-1} \frac{[|\psi_1[j]|^2 (|\phi_1[j]|^2)_t - |\phi_1[j]|^2 (|\psi_1[j]|^2)_t]}{(|\psi_1[j]|^2 + |\phi_1[j]|^2)^2}. \quad (52)$$

Specifically, the following plane wave solutions are taken as seed solutions:

$$A[0] = e^{i\theta}, \quad B[0] = -\frac{1}{\sqrt{1+a^2}}, \quad (53)$$

with the real constant a and $\theta = \frac{a\sqrt{1+a^2}x+t}{\sqrt{1+a^2}}$, and after substituting them into the Lax pair of the AB system, the first-order rogue wave is derived based on formulas (51) and (52) with $N = 1$ and the fixed spectral parameter $\lambda_1 = -\frac{a}{2} + \frac{i}{2}$,

$$A[1] = e^{i\theta} \left(1 + \frac{F_1 + iH_1}{D_1} \right), \quad B[1] = \frac{1}{\sqrt{1+a^2}} \frac{G_1}{D_1}, \quad (54)$$

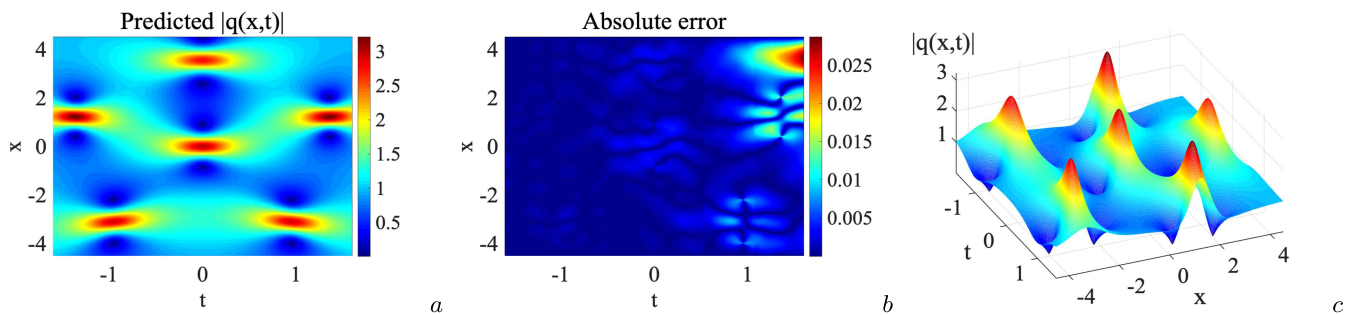


FIG. 10. Data-driven third-order rogue wave ($m_1 = n_1 = n_2 = 0, m_2 = 50$) by lbc-PINN: (a) The density diagram of the predicted solution $|q(x, t)|$; (b) The density diagram of absolute error; (c) The three-dimensional plot of the data-driven rogue wave solution $|q(x, t)|$.

TABLE III. Relative $\mathbb{L}_2$ errors of the data-driven third-order rogue waves for the NLS equation by bc-PINN and lbc-PINN.

	Ω	$[T_0, T]$	Relative $\mathbb{L}_2$ error of $ q(x, t) $	
			bc-PINN	lbc-PINN
$m_1 = 10, m_2 = 0$	$[-5, 5]$	$[-2, 1.5]$	1.932×10^{-2}	6.953×10^{-3}
$m_1 = 0, m_2 = 50$	$[-4.5, 4.5]$	$[-1.6, 1.6]$	6.419×10^{-3}	2.985×10^{-3}

where

$$\begin{aligned}
 F_1 &= (2a^4 + 4a^2 + 2)x^2 - 4\sqrt{1+a^2}axt + 2t^2 - 2a^4 - 4a^2 - 2, \\
 H_1 &= 4\sqrt{1+a^2}t, \\
 D_1 &= -(a^4 + 2a^2 + 1)x^2 + 2a\sqrt{1+a^2}xt - t^2 - a^4 - 2a^2 - 1, \\
 G_1 &= -a(a^2 + 1)^4 x^4 + 4a^2(a^2 + 1)^{5/2} tx^3 - 2a(a^2 + 1)((3a^2 + 1)t^2 + a^6 + 5a^4 + 7a^2 + 3)x^2 \\
 &\quad + 4\sqrt{1+a^2}(a^2 t^2 + a^6 + 4a^4 + 5a^2 + 2)xt - at^4 - (2a^5 + 8a^3 + 6a)t^2 - a^9 + 6a^5 + 8a^3 + 3a.
 \end{aligned}$$

After selecting $a = \frac{1}{10}$ and the spatiotemporal region $\Omega \times [T_0, T] = [-4, 4] \times [-4, 4]$, we can obtain the corresponding initial and boundary data. Then, we divide the time domain into $n_{\max} = 4$ segments and utilize the lbc-PINN method to simulate the dynamic behavior of first-order rogue wave. The entire region is divided into 512×201 equidistant grid points, and the number of initial data points, boundary data points and residual configuration points is taken as $N_i = 128$, $N_b = 50$ and $N_r = 20\,000$ separately. The increasing N_s is selected as $N_s = 128 \times 50 \times (n - 1)$ for the n th sub-domain.

Decomposing the complex-valued solution $A(x, t)$ into real part $u(x, t)$ and imaginary part $v(x, t)$ and substituting them into the AB system can derive three PDE residuals

$$\begin{aligned}
 R_1 &:= u_{xt} - uB, \\
 R_2 &:= v_{xt} - vB, \\
 R_3 &:= B_x + uu_t + vv_t.
 \end{aligned} \tag{55}$$

Then, the loss function of the first sub-domain is given

$$\text{MSE}_{\Delta T_1} = \text{MSE}_I(x_k^i, T_0) + \text{MSE}_B(x_k^b, t_k^b) + \text{MSE}_R(x_k^r, t_k^r), \tag{56}$$

$$T_0 = -4, \quad x_k^i \in [-4, 4], \quad (x_k^b, t_k^b) \in \{-4, 4\} \times [-4, -2], \quad (x_k^r, t_k^r) \in [-4, -4] \times [-4, -2],$$

where

$$\begin{aligned}
 \text{MSE}_I(x_k^i, T_0) &= \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}^{(1)}(x_k^i, T_0, \Theta_1) - u_k^i|^2 + \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{v}^{(1)}(x_k^i, T_0, \Theta_1) - v_k^i|^2 + \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{B}^{(1)}(x_k^i, T_0, \Theta_1) - B_k^i|^2, \\
 \text{MSE}_B(x_k^b, t_k^b) &= \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}^{(1)}(x_k^b, t_k^b, \Theta_1) - u_k^b|^2 + \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{v}^{(1)}(x_k^b, t_k^b, \Theta_1) - v_k^b|^2 + \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{B}^{(1)}(x_k^b, t_k^b, \Theta_1) - B_k^b|^2, \\
 \text{MSE}_R(x_k^r, t_k^r) &= \frac{1}{N_r} \sum_{k=1}^{N_r} |R_1(x_k^r, t_k^r, \Theta_1)|^2 + \frac{1}{N_r} \sum_{k=1}^{N_r} |R_2(x_k^r, t_k^r, \Theta_1)|^2 + \frac{1}{N_r} \sum_{k=1}^{N_r} |R_3(x_k^r, t_k^r, \Theta_1)|^2,
 \end{aligned} \tag{57}$$

and the PINN is applied to learn the real and imaginary parts ($\hat{u}^{(1)}$ and $\hat{v}^{(1)}$) of $\hat{A}^{(1)}$ and $\hat{B}^{(1)}$. In order to ensure the backward compatibility of the solution and simultaneously reduce the accumulation of errors, the modified loss term MSE_S is introduced into the loss function for subsequent three sub-domains ($n = 2, 3, 4$)

$$\begin{aligned}
 \text{MSE}_{\Delta T_n} &= \text{MSE}_I(x_k^i, T_{n-1}) + \text{MSE}_B(x_k^b, t_k^b) + \text{MSE}_R(x_k^r, t_k^r) + \text{MSE}_S(x_k^s, t_k^s), \\
 x_k^i &\in [-4, 4], \quad (x_k^b, t_k^b) \in \{-4, 4\} \times [T_{n-1}, T_n], \quad (x_k^r, t_k^r) \in [-4, 4] \times [T_{n-1}, T_n], \quad (x_k^s, t_k^s) \in [-4, 4] \times [-4, T_{n-1}], \\
 \tau_1 &= \{k | (x_k^s, t_k^s) \in [-4, 4] \times [-4, T_1]\}, \quad \tau_j = \{k | (x_k^s, t_k^s) \in [-4, 4] \times (T_{j-1}, T_j]\} \quad (j = 2, \dots, n-1),
 \end{aligned} \tag{58}$$

where

$$\begin{aligned}
 \text{MSE}_I(x_k^i, T_{n-1}) &= \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{u}^{(n)}(x_k^i, T_{n-1}, \Theta_n) - \hat{u}^{(n-1)}(x_k^i, T_{n-1})|^2 + \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{v}^{(n)}(x_k^i, T_{n-1}, \Theta_n) - \hat{v}^{(n-1)}(x_k^i, T_{n-1})|^2 \\
 &\quad + \frac{1}{N_i} \sum_{k=1}^{N_i} |\hat{B}^{(n)}(x_k^i, T_{n-1}, \Theta_n) - \hat{B}^{(n-1)}(x_k^i, T_{n-1})|^2, \\
 \text{MSE}_B(x_k^b, t_k^b) &= \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{u}^{(n)}(x_k^b, t_k^b, \Theta_n) - u_k^b|^2 + \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{v}^{(n)}(x_k^b, t_k^b, \Theta_n) - v_k^b|^2 + \frac{1}{N_b} \sum_{k=1}^{N_b} |\hat{B}^{(n)}(x_k^b, t_k^b, \Theta_n) - B_k^b|^2, \\
 \text{MSE}_R(x_k^r, t_k^r) &= \frac{1}{N_r} \sum_{k=1}^{N_r} |R_1(x_k^r, t_k^r, \Theta_n)|^2 + \frac{1}{N_r} \sum_{k=1}^{N_r} |R_2(x_k^r, t_k^r, \Theta_n)|^2 + \frac{1}{N_r} \sum_{k=1}^{N_r} |R_3(x_k^r, t_k^r, \Theta_n)|^2, \\
 \text{MSE}_S(x_k^s, t_k^s) &= \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{u}^{(n)}(x_k^s, t_k^s, \Theta_n) - \hat{u}^{(j)}(x_k^s, t_k^s)|^2 + \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{v}^{(n)}(x_k^s, t_k^s, \Theta_n) - \hat{v}^{(j)}(x_k^s, t_k^s)|^2 \\
 &\quad + \frac{1}{N_s} \sum_{j=1}^{n-1} \sum_{k \in \tau_j} |\hat{B}^{(n)}(x_k^s, t_k^s, \Theta_n) - \hat{B}^{(j)}(x_k^s, t_k^s)|^2.
 \end{aligned} \tag{59}$$

Here, $\hat{A}^{(n)} = \hat{u}^{(n)} + i\hat{v}^{(n)}$ and $\hat{B}^{(n)}$ are the predicted solutions by training subnet- n in n th sub-domain.

For each subnetwork, the width and depth are taken as 7 and 128, respectively. The selection of optimizers and the application of ICGL and transfer learning techniques are similar to those used in the nonlinear Schrödinger equation, and are not further elaborated upon here. The dynamic behaviors of the first-order rogue wave for

the AB system are successfully reproduced by using Ibc-PINN, as displayed in Fig. 11. The waveform of A component is the standard eye-shaped Peregrine soliton whereas the B component exhibits the characteristic shape of four peaky-shaped rogue wave. Moreover, the results in Table IV indicate that under this network structure configuration, the network training of bc-PINN fails and there is still a considerable error compared to the exact solution.

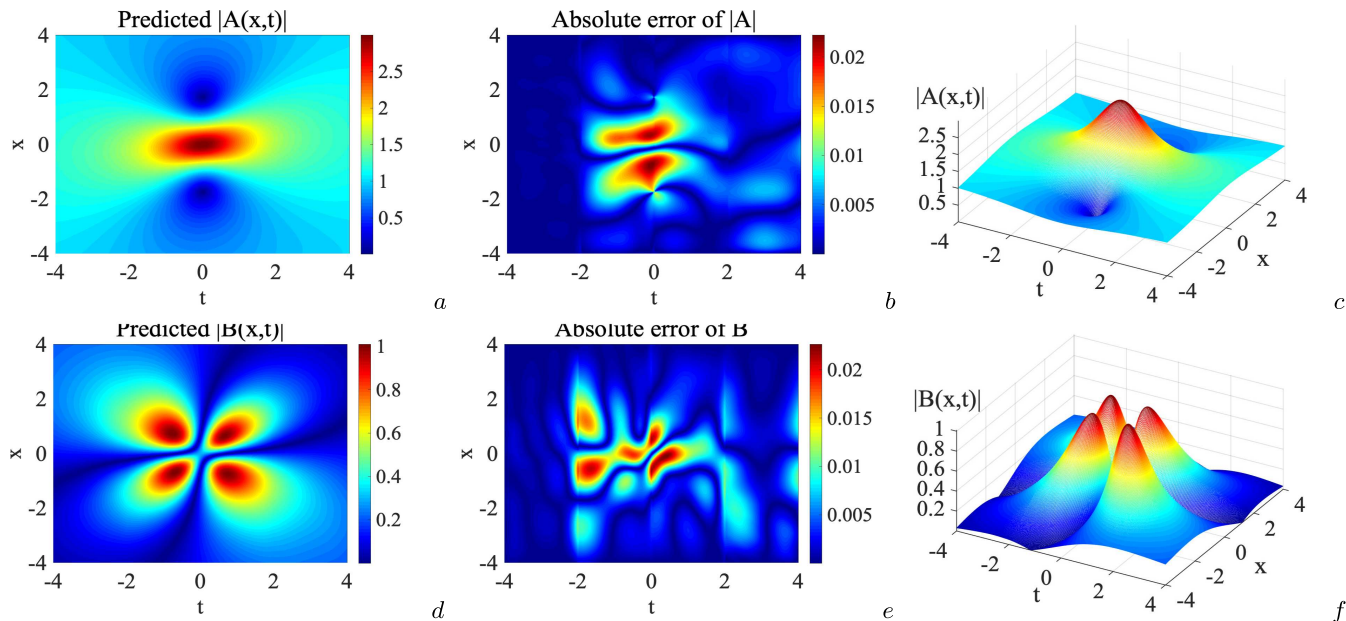


FIG. 11. Data-driven first-order rogue wave by Ibc-PINN: the density diagrams of the predicted solutions: (a) $|A(x, t)|$ and (d) $|B(x, t)|$; the density diagrams of absolute error: (b) $|A(x, t)|$ and (e) $|B(x, t)|$; the three-dimensional plots of the data-driven wave solutions: (c) $|A(x, t)|$ and (f) $|B(x, t)|$.

TABLE IV. Relative $\mathbb{L}_2$ errors of the data-driven rogue waves for the AB system by bc-PINN and lbc-PINN.

		bc-PINN	lbc-PINN
First-order	$ A(x, t) $	2.083×10^{-1}	4.238×10^{-3}
	$B(x, t)$	5.070×10^{-1}	1.442×10^{-2}
Second-order	$ A(x, t) $	3.309×10^{-3}	2.671×10^{-3}
	$B(x, t)$	1.212×10^{-2}	9.891×10^{-3}

2. Data-driven second-order rogue waves

Naturally, the second-order rogue wave solution for the AB system with free parameters a_1, m_1 , and n_1 can be derived⁶⁶ by performing the second-step generalized Darboux transformation (DT). Here, we refrain from providing the explicit expressions in this context due to the intricate nature of their representation. The value of a_1 is fixed at $\frac{1}{10}$, and once the spatiotemporal domain is chosen and the remaining two free parameters are determined, we can acquire the corresponding initial and boundary conditions.

The time domain $[-4, 4]$ is divided into $n_{\max} = 6$ equidistant segments. For the n th sub-domain, we choose $N_t = 128$, $N_b = 50$, $N_r = 2000$, and $N_s = 128 \times 50 \times (n - 1)$ and establish a sub network with six hidden layers, each consisting of 64 neurons.

The selection of free parameters $m_1 = 0, n_1 = 0$ corresponds to the fundamental second-order rogue wave, which is simulated by lbc-PINN shown in Fig. 12. Obviously, the maximum amplitude of predicted $|A(x, t)|$ occurs at the center $(0, 0)$, and it is roughly five times the amplitude of the background wave. For B component,

there are twelve peaks with a peak value of 1 around the center. Finally, Table IV provides the relative error of solutions obtained by using two methods, and the accuracy of lbc-PINN still remains higher.

IV. ANALYSIS AND DISCUSSION

A. Analysis of error accumulation

In this part, we take the second and third-order rogue waves for the nonlinear Schrödinger equation as examples to analyze the cumulative effect of errors by calculating the relative $\mathbb{L}_2$ errors of bc-PINN and lbc-PINN in each sub-domain.

For the second-order ($m_1 = 0, n_1 = 0$) and third-order rogue waves ($m_1 = n_1 = n_2 = 0, m_2 = 50$), the entire regions were divided into four and six sub-domains, respectively, and a total of four and six subnets were trained. We calculate the relative $\mathbb{L}_2$ errors on $N_x \times N_t = 512 \times 51$ grid points for each sub-domain to evaluate the accuracy of models and the specific results are listed in Tables V–VIII.

According to Tables V and VII, we first analyze the error accumulation phenomenon of bc-PINN. Obviously, the relative $\mathbb{L}_2$ error of subnet-2 on sub-domain 1 has been almost an order of magnitude larger than that of subnet-1, and as the sequential number of subnets increases, the error on sub-domain 1 roughly shows an upward trend. There is also a similar phenomenon of error accumulation in other sub-domains. The values in the last column are the relative $\mathbb{L}_2$ errors of the solution by training the last sub network [i.e., the final form of the predicted solution obtained by bc-PINN shown in (12)] in each sub-domain. Those on the diagonal are the errors generated by the predicted solution of the n th subnet on the n th sub-domain,

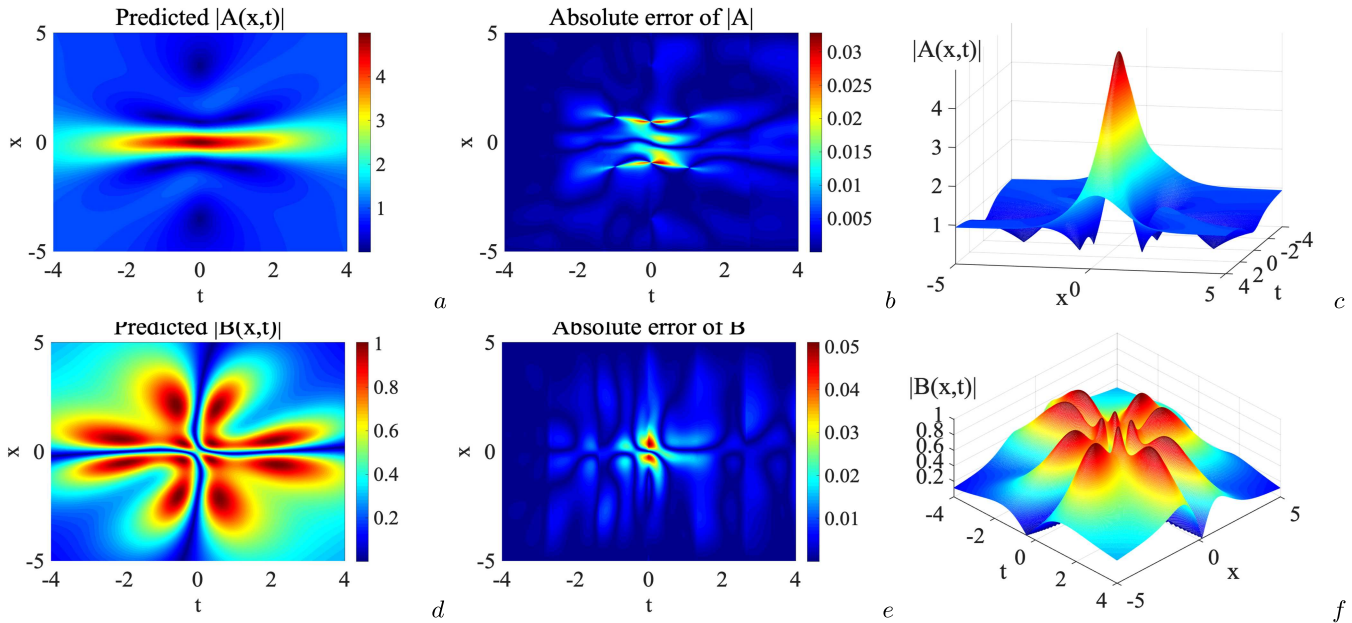
**FIG. 12.** Data-driven second-order rogue wave by lbc-PINN: the density diagrams of the predicted solutions: (a) $|A(x, t)|$ and (d) $|B(x, t)|$; the density diagrams of absolute error: (b) $|A(x, t)|$ and (e) $|B(x, t)|$; the three-dimensional plots of the data-driven wave solutions: (c) $|A(x, t)|$ and (f) $|B(x, t)|$.

TABLE V. Relative $\mathbb{L}_2$ errors of the data-driven second-order rogue wave ($m_1 = 0$, $n_1 = 0$) for the NLS equation in each sub-domain by bc-PINN.^a

		subnet-1	subnet-2	subnet-3	subnet-4
Relative $\mathbb{L}_2$ error of $u(x, t)$	sub-domain 1	1.015×10^{-4}	2.543×10^{-3}	1.026×10^{-2}	8.885×10^{-3}
	sub-domain 2	$\diagdown$	8.519×10^{-4}	6.831×10^{-3}	5.008×10^{-3}
	sub-domain 3	$\diagdown$	$\diagdown$	7.742×10^{-3}	7.764×10^{-3}
	sub-domain 4	$\diagdown$	$\diagdown$	$\diagdown$	4.013×10^{-2}
Relative $\mathbb{L}_2$ error of $v(x, t)$	sub-domain 1	6.315×10^{-5}	1.571×10^{-3}	5.985×10^{-3}	5.162×10^{-3}
	sub-domain 2	$\diagdown$	1.260×10^{-3}	1.018×10^{-2}	7.282×10^{-3}
	sub-domain 3	$\diagdown$	$\diagdown$	1.366×10^{-2}	1.371×10^{-2}
	sub-domain 4	$\diagdown$	$\diagdown$	$\diagdown$	1.382×10^{-2}
Relative $\mathbb{L}_2$ error of $ q(x, t) $	sub-domain 1	5.543×10^{-5}	1.307×10^{-3}	5.293×10^{-3}	4.656×10^{-3}
	sub-domain 2	$\diagdown$	6.178×10^{-4}	5.443×10^{-3}	3.975×10^{-3}
	sub-domain 3	$\diagdown$	$\diagdown$	3.868×10^{-3}	3.990×10^{-3}
	sub-domain 4	$\diagdown$	$\diagdown$	$\diagdown$	1.071×10^{-2}

^aThe boldface values on the diagonal indicate the error of the earliest learned predictive solution in each subregion, while the boldface values in the last column denote the error of the solution obtained from training the final subnetwork.

while noting that the accuracy of this solution is the highest for this sub-domain. In addition, by comparing the values in the last column of the table with those on the diagonal, it can be intuitively seen that subsequent training of the subnets will inevitably be affected by the accumulation of errors, leading to a decrease in accuracy.

The performance of lbc-PINN in various sub regions is shown in Tables VI and VIII. When calculating MSE_S , we modified the original form in bc-PINN to reduce error accumulation. To reiterate, with regard to subnet-1 and subnet-2, the training results of bc-PINN and lbc-PINN are the same, and there are differences in the results between the two methods starting from subnet-3. By comparing the values on the diagonal and the last column, we observe that the error accumulation speed of lbc-PINN is significantly slower than that of bc-PINN. After comparing the

errors on the diagonal in Tables V–VIII, the highest accuracy that lbc-PINN can achieve in each sub-domain is also higher than that of bc-PINN attributed to the reduction of error accumulation. Moreover, not to mention that bc-PINN is associated with errors in the last column, which are even greater than those on the diagonal.

In order to compare the performance of the two methods more intuitively, we draw a line graph as shown in Fig. 13. The black and red lines represent the relative $\mathbb{L}_2$ error of $|q(x, t)|$ achieved by bc-PINN and lbc-PINN, respectively, while the blue dashed line represents the minimum error of bc-PINN in each sub-domain, namely, the diagonal values in Tables V and VII, annotated on the graph as “best bc-PINN.” Note that the red line is always below the blue dashed line and furthermore, the difference between the blue

TABLE VI. Relative $\mathbb{L}_2$ errors of the data-driven second-order rogue wave ($m_1 = 0$, $n_1 = 0$) for the NLS equation in each sub-domain by lbc-PINN.^a

		subnet-1	subnet-2	subnet-3	subnet-4
Relative $\mathbb{L}_2$ error of $u(x, t)$	sub-domain 1	1.015×10^{-4}	2.543×10^{-3}	7.961×10^{-3}	2.167×10^{-3}
	sub-domain 2	$\diagdown$	8.519×10^{-4}	5.842×10^{-3}	1.991×10^{-3}
	sub-domain 3	$\diagdown$	$\diagdown$	6.108×10^{-3}	6.465×10^{-3}
	sub-domain 4	$\diagdown$	$\diagdown$	$\diagdown$	3.268×10^{-2}
Relative $\mathbb{L}_2$ error of $v(x, t)$	sub-domain 1	6.315×10^{-5}	1.571×10^{-3}	5.609×10^{-3}	1.058×10^{-3}
	sub-domain 2	$\diagdown$	1.260×10^{-3}	9.123×10^{-3}	2.894×10^{-3}
	sub-domain 3	$\diagdown$	$\diagdown$	1.103×10^{-2}	1.132×10^{-2}
	sub-domain 4	$\diagdown$	$\diagdown$	$\diagdown$	1.020×10^{-2}
Relative $\mathbb{L}_2$ error of $ q(x, t) $	sub-domain 1	5.543×10^{-5}	1.307×10^{-3}	4.602×10^{-3}	9.862×10^{-4}
	sub-domain 2	$\diagdown$	6.178×10^{-4}	5.347×10^{-3}	1.610×10^{-3}
	sub-domain 3	$\diagdown$	$\diagdown$	2.684×10^{-3}	3.189×10^{-3}
	sub-domain 4	$\diagdown$	$\diagdown$	$\diagdown$	7.022×10^{-3}

^aThe boldface values on the diagonal indicate the error of the earliest learned predictive solution in each subregion, while the boldface values in the last column denote the error of the solution obtained from training the final subnetwork.

TABLE VII. Relative $\mathbb{L}_2$ errors of the data-driven third-order rogue wave ($m_1 = n_1 = n_2 = 0$, $m_2 = 50$) for the NLS equation in each sub-domain by bc-PINN.^a

		subnet-1	subnet-2	subnet-3	subnet-4	subnet-5	subnet-6
Relative $\mathbb{L}_2$ error of $u(x, t)$	sub-domain 1	3.980×10^{-4}	1.327×10^{-3}	1.933×10^{-3}	3.123×10^{-3}	4.536×10^{-3}	6.979×10^{-3}
	sub-domain 2	↖	5.857×10^{-4}	1.670×10^{-3}	3.097×10^{-3}	3.270×10^{-3}	5.652×10^{-3}
	sub-domain 3	↖	↖	2.065×10^{-3}	3.365×10^{-3}	4.531×10^{-3}	6.939×10^{-3}
	sub-domain 4	↖	↖	↖	4.830×10^{-3}	5.716×10^{-3}	8.763×10^{-3}
	sub-domain 5	↖	↖	↖	↖	7.516×10^{-3}	1.019×10^{-2}
	sub-domain 6	↖	↖	↖	↖	↖	1.990×10^{-2}
Relative $\mathbb{L}_2$ error of $v(x, t)$	sub-domain 1	3.917×10^{-4}	1.085×10^{-3}	2.179×10^{-3}	3.199×10^{-3}	4.393×10^{-3}	6.568×10^{-3}
	sub-domain 2	↖	5.577×10^{-4}	1.713×10^{-3}	3.522×10^{-3}	3.851×10^{-3}	6.530×10^{-3}
	sub-domain 3	↖	↖	2.333×10^{-3}	3.738×10^{-3}	4.244×10^{-3}	7.770×10^{-3}
	sub-domain 4	↖	↖	↖	3.909×10^{-3}	4.776×10^{-3}	8.020×10^{-3}
	sub-domain 5	↖	↖	↖	↖	1.284×10^{-2}	1.521×10^{-2}
	sub-domain 6	↖	↖	↖	↖	↖	2.120×10^{-2}
Relative $\mathbb{L}_2$ error of $ q(x, t) $	sub-domain 1	3.033×10^{-4}	8.866×10^{-4}	1.476×10^{-3}	2.226×10^{-3}	3.306×10^{-3}	4.996×10^{-3}
	sub-domain 2	↖	4.405×10^{-4}	1.183×10^{-3}	2.204×10^{-3}	2.521×10^{-3}	3.904×10^{-3}
	sub-domain 3	↖	↖	1.014×10^{-3}	2.181×10^{-3}	2.884×10^{-3}	5.141×10^{-3}
	sub-domain 4	↖	↖	↖	1.243×10^{-3}	2.510×10^{-3}	5.460×10^{-3}
	sub-domain 5	↖	↖	↖	↖	4.065×10^{-3}	6.620×10^{-3}
	sub-domain 6	↖	↖	↖	↖	↖	1.045×10^{-2}

^aThe boldface values on the diagonal indicate the error of the earliest learned predictive solution in each subregion, while the boldface values in the last column denote the error of the solution obtained from training the final subnetwork.

dashed line and the red one shows an increasing trend, which indicates that our improvement helps to slow down error accumulation and improve the accuracy of each sub region compared to bc-PINN. In other word, due to the refinement made in the loss term MSE_S , even though bc-PINN adopts a concatenated form of solution, the

error propagation speed is also faster than lbc-PINN. Considering that the examples presented in Fig. 13 are divided into at most six sub regions, if the number of sub-domains divided increases in future research, the slowdown of lbc-PINN in error accumulation speed may yield greater advantages in accuracy.

TABLE VIII. Relative $\mathbb{L}_2$ errors of the data-driven third-order rogue wave ($m_1 = n_1 = n_2 = 0$, $m_2 = 50$) for the NLS equation in each sub-domain by lbc-PINN.^a

		subnet-1	subnet-2	subnet-3	subnet-4	subnet-5	subnet-6
Relative $\mathbb{L}_2$ error of $u(x, t)$	sub-domain 1	3.980×10^{-4}	1.327×10^{-3}	2.021×10^{-3}	2.663×10^{-3}	4.578×10^{-3}	4.301×10^{-3}
	sub-domain 2	↖	5.857×10^{-4}	1.720×10^{-3}	2.473×10^{-3}	3.848×10^{-3}	3.552×10^{-3}
	sub-domain 3	↖	↖	1.906×10^{-3}	3.883×10^{-3}	3.836×10^{-3}	5.183×10^{-3}
	sub-domain 4	↖	↖	↖	3.546×10^{-3}	4.918×10^{-3}	5.269×10^{-3}
	sub-domain 5	↖	↖	↖	↖	6.795×10^{-3}	7.909×10^{-3}
	sub-domain 6	↖	↖	↖	↖	↖	1.292×10^{-2}
Relative $\mathbb{L}_2$ error of $v(x, t)$	sub-domain 1	3.917×10^{-4}	1.085×10^{-3}	2.328×10^{-3}	2.604×10^{-3}	4.039×10^{-3}	4.657×10^{-3}
	sub-domain 2	↖	5.577×10^{-4}	2.059×10^{-3}	2.821×10^{-3}	4.325×10^{-3}	4.296×10^{-3}
	sub-domain 3	↖	↖	2.151×10^{-3}	4.180×10^{-3}	4.304×10^{-3}	4.726×10^{-3}
	sub-domain 4	↖	↖	↖	3.113×10^{-3}	4.728×10^{-3}	4.669×10^{-3}
	sub-domain 5	↖	↖	↖	↖	9.264×10^{-3}	1.024×10^{-2}
	sub-domain 6	↖	↖	↖	↖	↖	1.340×10^{-2}
Relative $\mathbb{L}_2$ error of $ q(x, t) $	sub-domain 1	3.033×10^{-4}	8.866×10^{-4}	1.578×10^{-3}	1.829×10^{-3}	3.040×10^{-3}	3.062×10^{-3}
	sub-domain 2	↖	4.405×10^{-4}	1.408×10^{-3}	1.768×10^{-3}	2.736×10^{-3}	2.566×10^{-3}
	sub-domain 3	↖	↖	9.628×10^{-4}	2.656×10^{-3}	2.590×10^{-3}	3.453×10^{-3}
	sub-domain 4	↖	↖	↖	1.059×10^{-3}	2.877×10^{-3}	2.909×10^{-3}
	sub-domain 5	↖	↖	↖	↖	2.793×10^{-3}	4.114×10^{-3}
	sub-domain 6	↖	↖	↖	↖	↖	6.690×10^{-3}

^aThe boldface values on the diagonal indicate the error of the earliest learned predictive solution in each subregion, while the boldface values in the last column denote the error of the solution obtained from training the final subnetwork.

B. Error analysis

In the last subsection, we demonstrate the necessity of proposing an lbc-PINN method that can slow down the accumulation of errors by analyzing the errors of two methods in each sub-domain. The previous analysis is conducted from the perspective of sub regions. Here, we analyze the relative $\mathbb{L}_2$ errors and absolute error plots across the entire spatiotemporal region.

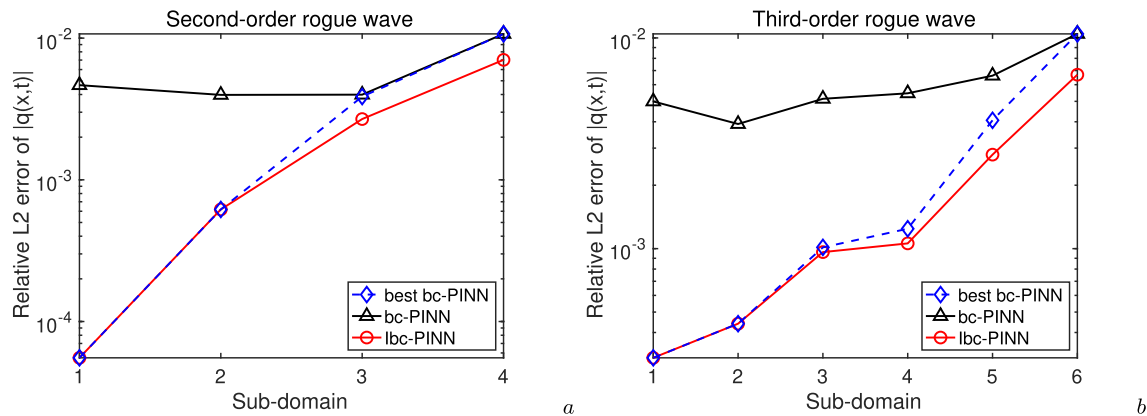


FIG. 13. Relative $\mathbb{L}_2$ errors in each sub-domain: (a) data-driven second-order rogue wave ($m_1 = 0, n_1 = 0$); (b) data-driven third-order rogue wave ($m_1 = n_1 = n_2 = 0, m_2 = 50$).

TABLE IX. Relative $\mathbb{L}_2$ errors of the data-driven rogue wave solutions $|q(x, t)|$ for the NLS and KE equations by bc-PINN, lbc-PINN(unjoined), and lbc-PINN.

		bc-PINN	lbc-PINN(unjoined)	lbc-PINN
First-order rogue waves	$\beta = 0$	1.062×10^{-3}	6.776×10^{-4}	4.843×10^{-4}
	$\beta = \frac{1}{3}$ (KE)	4.471×10^{-3}	3.909×10^{-3}	3.789×10^{-3}
Second-order rogue waves	$m_1 = 0$	6.368×10^{-3}	3.902×10^{-3}	3.690×10^{-3}
	$m_1 = 10$	6.580×10^{-3}	4.183×10^{-3}	3.543×10^{-3}
Third-order rogue waves	$m_1 = 10, m_2 = 0$	1.932×10^{-2}	9.368×10^{-3}	6.953×10^{-3}
	$m_1 = 0, m_2 = 50$	6.419×10^{-3}	4.020×10^{-3}	2.985×10^{-3}

TABLE X. Relative $\mathbb{L}_2$ errors of the data-driven rogue wave solutions for the AB system by bc-PINN, lbc-PINN(unjoined), and lbc-PINN.

		bc-PINN	lbc-PINN(unjoined)	lbc-PINN
First-order rogue wave	$ A(x, t) $	2.083×10^{-1}	4.612×10^{-3}	4.238×10^{-3}
	$B(x, t)$	5.070×10^{-1}	1.595×10^{-2}	1.442×10^{-2}
Second-order rogue wave	$ A(x, t) $	3.309×10^{-3}	2.719×10^{-3}	2.671×10^{-3}
	$B(x, t)$	1.212×10^{-2}	9.881×10^{-3}	9.891×10^{-3}

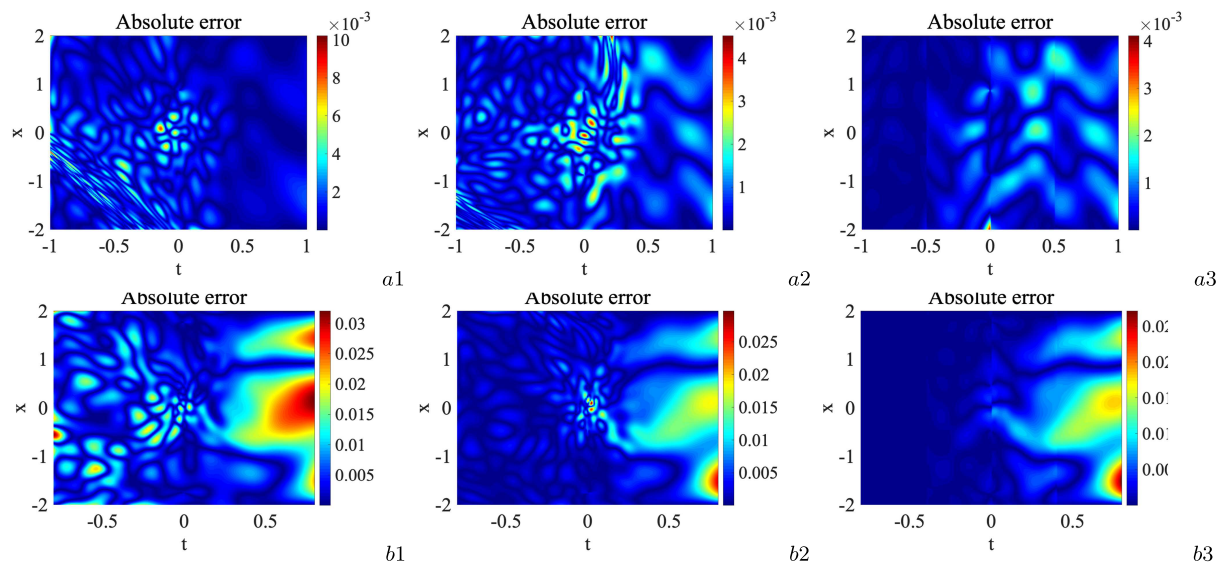


FIG. 14. The absolute error density diagrams of data-driven first-order rogue wave ($\beta = 0$): (a1) by bc-PINN, (a2) lbc-PINN(unjoined), and (a3) lbc-PINN; the absolute error density diagrams of data-driven second-order rogue wave ($m_1 = 0, n_1 = 0$): (b1) by bc-PINN, (b2) lbc-PINN(unjoined), and (b3) lbc-PINN.

TABLE XI. Relative $\mathbb{L}_2$ errors of the data-driven first-order rogue wave solution $|q(x, t)|$ for the NLS equation by using different number of hidden layers and neurons per layer.

Layers	Neurons							
	32		64		96		128	
	bc-PINN	lbc-PINN	bc-PINN	lbc-PINN	bc-PINN	lbc-PINN	bc-PINN	lbc-PINN
4	8.875×10^{-4}	5.564×10^{-4}	5.889×10^{-4}	3.224×10^{-4}	5.799×10^{-4}	2.327×10^{-4}	1.327×10^{-3}	1.630×10^{-4}
6	8.402×10^{-4}	3.389×10^{-4}	5.292×10^{-4}	1.558×10^{-4}	5.854×10^{-4}	1.874×10^{-4}	1.062×10^{-3}	4.843×10^{-4}
8	6.323×10^{-4}	2.505×10^{-4}	6.050×10^{-4}	3.505×10^{-4}	$1.398e+00$	6.662×10^{-4}	5.955×10^{-3}	4.871×10^{-3}

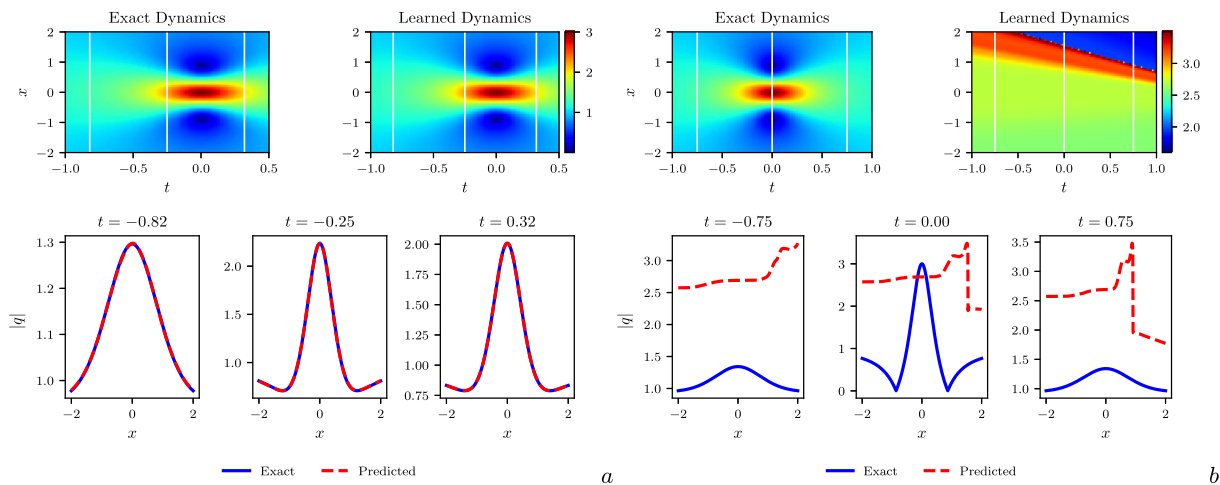


FIG. 15. Instability of data-driven first-order rogue wave for the NLS equation by bc-PINN: (a) predicted results of the penultimate stage; (b) predicted results of the last stage.

The Ibc-PINN method proposed in this study has been improved in two aspects for bc-PINN: one is to modify the loss term MSE_S and the other is to take the joined form as the final form of the predicted solution, as shown in (15) and (20). To illustrate that both of these improvements contribute to accuracy, the errors generated by the three methods [bc-PINN, Ibc-PINN(unjoined), and Ibc-PINN] over the entire region are shown in Tables IX and X. Among them, Ibc-PINN(unjoined) refers to the modification of bc-PINN only in the form of MSE_S , without using a concatenated version in the final form of the predicted solution.

According to the results in the tables, it can be seen that Ibc-PINN is optimal in terms of accuracy, while Ibc-PINN(unjoined)

is suboptimal, which implies that the improvements in these two aspects can both improve accuracy to a certain extent.

In addition, the absolute error plots of the first-order ($\beta = 0$) and second-order rogue waves ($m_1 = 0, n_1 = 0$) for the nonlinear Schrödinger equation are displayed in Fig. 14. The three columns in the figure correspond to the results of the bc-PINN, Ibc-PINN(unjoined), and Ibc-PINN methods, while each row corresponds to a different data-driven rogue wave solution.

We can observe the characteristics of the error distribution of the three methods. Due to the joined form adopted in the final representation of the predicted solutions in Ibc-PINN, there are distinct traces of concatenation in the absolute error graphs of the third

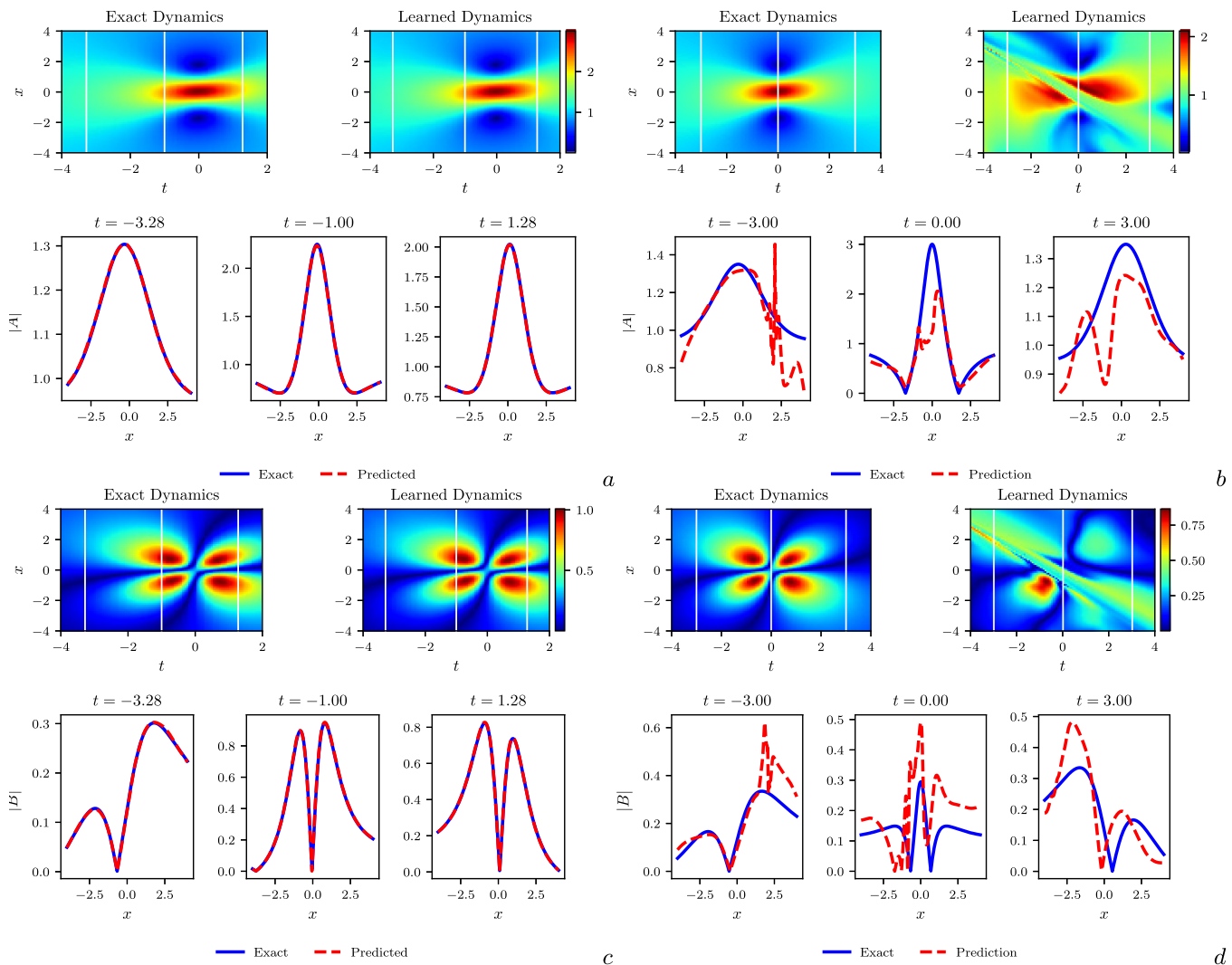


FIG. 16. Instability of data-driven first-order rogue wave for the AB system by bc-PINN: predicted results of the penultimate stage: (a) $|A(x, t)|$ and (c) $|B(x, t)|$; predicted results of the last stage: (b) $|A(x, t)|$ and (d) $|B(x, t)|$.

column. For the bc-PINN, larger absolute errors primarily occur in the following three typical regions: (1) near the initial time; (2) where the wave amplitude is large or the gradient is significant; (3) near the final time. The phenomenon of significant errors in bc-PINN near the initial time is contrary to the typical error propagation patterns in time piecewise training methods. For the first sub-domain, accurate initial conditions are given, while the training for subsequent sub-domains is based on pseudo-initial values. Theoretically, the accuracy of the first sub region should have been the highest, but the red part indicating a large error value can be observed near the initial time in the first column of absolute error graphs. For Ibc-PINN(unjoined) and Ibc-PINN, the absolute error near the initial time is significantly reduced, especially for Ibc-PINN, which performs well in the first sub-domain with an absolute error of almost zero. It reflects that the violation of error propagation law mentioned above can be corrected by the improvement proposed in this study.

C. The impact of neural network architecture

Here, we explore the impact of network architecture on experimental results, specifically investigating whether the improved method still enhances accuracy when changing the number of hidden layers and neurons. Due to the time-consuming nature of time-segmented training approaches, we take the first-order rogue wave solution for the NLS equation as an example to illustrate.

The number of hidden layers changes from 4 to 8 with step size 2 and the number of neurons in each hidden layer changes from 32 to 128 with step size 32. The relative L_2 errors of the bc-PINN and Ibc-PINN methods are presented in Table XI. It is noteworthy that increasing the number of hidden layers or neurons does not necessarily lead to smaller errors. However, the improved method consistently demonstrate a certain level of accuracy enhancement.

Additionally, we observed that when the number of hidden layers is 8 and the number of neurons is 96, the bc-PINN method converges to an incorrect solution, resulting in a significant error. This phenomenon is not incidental and is also reflected in the data-driven first-order rogue wave of the AB system, as shown in Table IV. It indicates the instability in the predictive performance of bc-PINN under certain network structure settings, whereas the improved method can effectively alleviate or even avoid such occurrences. More specifically, we further analyzed the specific performance in each sub region based on the results of bc-PINN for these two examples. The predicted results of the penultimate stage and the final stage using the bc-PINN method are shown in the first and second columns of Figs. 15 and 16, respectively. It can be observed that the penultimate subnetwork effectively captures the dynamic behaviors of rogue wave solutions with minor errors, whereas the predictive accuracy of the final subnetwork is significantly compromised. It illustrates that for bc-PINN, once the network training performs poorly in a certain sub-domain, the success achieved in previous regions is nullified and the errors generated by this stage of training instantly pollute the entire region. This is due to the fact that the form of the solution adopted by bc-PINN is entirely dependent on whether the training of the last subnet in the last sub-domain is successful or not. In contrast, for Ibc-PINN, if the training in a specific sub-domain is inadequate, it only affects the accuracy of the

predicted solutions from that subregion onward. This also reveals the necessity for us to propose improvements to the ultimate form of the predicted solution shown in (20). The underlying reasons for the instability in accuracy of bc-PINN require further investigation, and targeted improvements can be proposed to address this issue in future research.

V. CONCLUSIONS

The bc-PINN is a sequential method to train physics informed neural networks over successive time segments while satisfying the solution for all previous time segments. Based on the characteristics of error propagation, we have made improvements in two aspects, namely, the loss function and the final form of the predicted solution. First, the loss term for ensuring backward compatibility is modified by selecting the earliest learned solution for each sub-domain as pseudo-reference solution. It can reduce the cumulative speed of errors to improve the accuracy of the solution in subsequent training. Second, we take the joined form of solutions obtained from individual subnetworks as the final form of the predicted solution, rather than relying solely on the solution learned by the last subnetwork. Its advantage lies in the fact that insufficient training in a specific subdomain only affects the accuracy of the predicted solutions from that subregion onward. This stands in contrast to bc-PINN, where inadequate training instantly compromises the accuracy of the entire region, rendering prior successful training in other subdomains futile. The improved bc-PINN (Ibc-PINN) is applied to successfully obtain data-driven higher-order rogue wave solutions for the nonlinear Schrödinger equation and the AB system. We also explore the impact of neural network architecture on performance and several cases were identified where the accuracy of bc-PINN is significantly compromised, while Ibc-PINN consistently maintains stability. In summary, both improvements contribute to the enhancement of the algorithm's accuracy and stability compared to the original bc-PINN method.

The domain decomposition technique possesses strong flexibility. In each sub-domain, a separate network can be employed, having its own set of parameters, including network width and depth, activation functions, and optimization methods. The discussion and research on this part can be carried out in subsequent work. Moreover, different subdomains can be partitioned into varying sizes and adaptive domain decomposition methods can be designed in future study based on the distinct characteristics of solutions in different sub regions. In this paper, we investigate the dynamic characteristics of rogue waves up to the third order, which exhibit geometric structures of triangle and pentagon. In fact, rogue wave patterns have been widely analytically studied but the research by using deep learning methods are still relatively scarce. Data-driven rogue waves of the fourth order and beyond can be simulated to showcase more intricate geometric structures including heptagon and nonagon in the future research.

ACKNOWLEDGMENTS

The authors thank Xin Wang for providing data support. The project is supported by National Natural Science Foundation

of China (Nos. 12175069 and 12235007), Science and Technology Commission of Shanghai Municipality (Nos. 21JC1402500 and 22DZ2229014), and Natural Science Foundation of Shanghai (No. 23ZR1418100).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Shuning Lin: Conceptualization (lead); Methodology (equal); Software (lead); Writing – original draft (equal); Writing – review & editing (equal). **Yong Chen:** Methodology (equal); Project administration (lead); Supervision (lead); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹L. Draper, “‘Freak’ ocean waves,” *Weather* **21**(1), 2–4 (1966).
- ²D. R. Solli, C. Ropers, P. Koonath, and B. Jalali, “Optical rogue waves,” *Nature* **450**(7172), 1054–1057 (2007).
- ³Y. V. Bludov, V. V. Konotop, and N. Akhmediev, “Matter rogue waves,” *Phys. Rev. A* **80**, 033610 (2009).
- ⁴W. M. Moslem, P. K. Shukla, and B. Eliasson, “Surface plasma rogue waves,” *Euro. Phys. Lett.* **96**, 25002 (2011).
- ⁵V. B. Efimov, A. N. Ganshin, G. V. Kolmakov, P. V. E. McClintock, and L. P. Mezhev-Deglin, “Rogue waves in superfluid helium,” *Eur. Phys. J. Spec. Top.* **185**, 181–193 (2010).
- ⁶L. Stenflo and M. Marklund, “Rogue waves in the atmosphere,” *J. Plasma Phys.* **76**(3–4), 293–295 (2010).
- ⁷Z. Y. Yan, “Financial rogue waves,” *Commun. Theor. Phys.* **54**(5), 947 (2010).
- ⁸D. H. Peregrine, “Water waves, nonlinear Schrödinger equations and their solutions,” *J. Australian Math. Soc. Ser. B Appl. Math.* **25**(1), 16–43 (1983).
- ⁹Y. Ohta and J. K. Yang, “Rogue waves in the Davey-Stewartson I equation,” *Phys. Rev. E* **86**, 036604 (2012).
- ¹⁰J. C. Chen, Y. Chen, B. F. Feng, K. Maruno, and Y. Ohta, “General high-order rogue waves of the (1+1)-dimensional Yajima-Oikawa system,” *J. Phys. Soc. Jpn.* **87**(9), 094007 (2018).
- ¹¹B. L. Guo, L. M. Ling, and Q. P. Liu, “Nonlinear Schrödinger equation: Generalized Darboux transformation and rogue wave solutions,” *Phys. Rev. E* **85**(2), 026607 (2012).
- ¹²X. E. Zhang and Y. Chen, “Inverse scattering transformation for generalized nonlinear Schrödinger equation,” *Appl. Math. Lett.* **98**, 306–313 (2019).
- ¹³M. 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.* **378**, 686–707 (2019).
- ¹⁴G. F. Pang, L. Lu, and G. E. Karniadakis, “fPINNs: Fractional physics-informed neural networks,” *SIAM J. Sci. Comput.* **41**(4), A2603–A2626 (2019).
- ¹⁵L. Lu, R. Pestourie, W. J. Yao, Z. C. Wang, F. Verdugo, and S. G. Johnson, “Physics-informed neural networks with hard constraints for inverse design,” *SIAM J. Sci. Comput.* **43**(6), B1105–B1132 (2021).
- ¹⁶L. Yang, X. H. Meng, and G. E. Karniadakis, “B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data,” *J. Comput. Phys.* **425**, 109913 (2021).

- ¹⁷D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis, “Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems,” *J. Comput. Phys.* **397**, 108850 (2019).
- ¹⁸S. N. Lin and Y. Chen, “A two-stage physics-informed neural network method based on con- served quantities and applications in localized wave solutions,” *J. Comput. Phys.* **457**, 111053 (2022).
- ¹⁹S. N. Lin and Y. Chen, “Physics-informed neural network methods based on Miura transformations and discovery of new localized wave solutions,” *Phys. D* **445**, 133629 (2023).
- ²⁰Z. Y. Zhang, H. Zhang, L. S. Zhang, and L. L. Guo, “Enforcing continuous symmetries in physics-informed neural network for solving forward and inverse problems of partial differential equations,” *J. Comput. Phys.* **492**, 112415 (2023).
- ²¹A. 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.* **365**, 113028 (2020).
- ²²A. 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,” *Commun. Comput. Phys.* **28**, 2002–2041 (2020).
- ²³K. Shukla, A. D. Jagtap, and G. E. Karniadakis, “Parallel physics-informed neural networks via domain decomposition,” *J. Comput. Phys.* **447**, 110683 (2021).
- ²⁴B. Moseley, A. Markham, and T. Nissen-Meyer, “Finite basis physics-informed neural networks (FBPINNs): A scalable domain decomposition approach for solving differential equations,” *Adv. Comput. Math.* **49**(4), 62 (2023).
- ²⁵E. Kharazmi, Z. Zhang, and G. E. Karniadakis, “hp-VPINNs: Variational physics-informed neural networks with domain decomposition,” *Comput. Methods Appl. Mech. Eng.* **374**, 113547 (2021).
- ²⁶P. Stiller, F. Bethke, M. Böhme, R. Pausch, S. Torge, A. Debus, J. Vorberger, M. Bussmann, and N. Hoffmann, “Large-scale neural solvers for partial differential equations, driving scientific and engineering discoveries through the convergence of HPC, big data and AI,” in *17th Smoky Mountains Computational Sciences and Engineering Conference, SMC 2020, Oak Ridge, TN, USA, August 26–28, 2020, Revised Selected Papers 17* (Springer International Publishing, 2020), pp. 20–34.
- ²⁷N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” *arXiv:1701.06538* (2017).
- ²⁸C. L. Wight and J. Zhao, “Solving Allen-Cahn and Cahn-Hilliard equations using the adaptive physics informed neural networks,” *arXiv:2007.04542* (2020).
- ²⁹R. Matthey and S. Ghosh, “A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations,” *Comput. Methods Appl. Mech. Eng.* **390**, 114474 (2022).
- ³⁰S. F. Wang, S. Sankaran, and P. Perdikaris, “Respecting causality is all you need for training physics-informed neural networks,” *arXiv:2203.07404* (2022).
- ³¹K. Haitsiukevich and A. Ilin, “Improved training of physics-informed neural networks with model ensembles,” in *2023 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2023), pp. 1–8.
- ³²J. Li and Y. Chen, “Solving second-order nonlinear evolution partial differential equations using deep learning,” *Commun. Theor. Phys.* **72**(10), 105005 (2020).
- ³³J. Li and Y. Chen, “A deep learning method for solving third-order nonlinear evolution equations,” *Commun. Theor. Phys.* **72**(11), 115003 (2020).
- ³⁴J. C. Pu, J. Li, and Y. Chen, “Soliton, breather and rogue wave solutions for solving the nonlinear Schrödinger equation using a deep learning method with physical constraints,” *Chin. Phys. B* **30**(6), 060202 (2021).
- ³⁵W. Q. Peng, J. C. Pu, and Y. Chen, “PINN deep learning for the Chen-Lee-Liu equation: Rogue wave on the periodic background,” *Commun. Nonlinear Sci. Numer. Simul.* **105**, 106067 (2022).
- ³⁶Z. W. Miao and Y. Chen, “Physics-informed neural network method in high-dimensional integrable systems,” *Mod. Phys. Lett. B* **36**(1), 2150531 (2022).
- ³⁷J. C. Pu and Y. Chen, “Data-driven vector localized waves and parameters discovery for Manakov system using deep learning approach,” *Chaos, Solitons Fractals* **160**, 112182 (2022).
- ³⁸J. C. Pu and Y. Chen, “Data-driven forward-inverse problems for Yajima-Oikawa system using deep learning with parameter regularization,” *Commun. Nonlinear Sci. Numer. Simul.* **118**, 107051 (2023).

- ³⁹L. Wang and Z. Y. Yan, "Data-driven rogue waves and parameter discovery in the defocusing nonlinear Schrödinger equation with a potential using the PINN deep learning," *Phys. Lett. A* **404**, 127408 (2021).
- ⁴⁰M. Zhong, S. Gong, S. F. Tian, and Z. Y. Yan, "Data-driven rogue waves and parameters discovery in nearly integrable PT-symmetric Gross-Pitaevskii equations via PINNs deep learning," *Phys. D* **439**, 133430 (2022).
- ⁴¹J. C. Chen, J. Song, Z. J. Zhou, and Z. Y. Yan, "Data-driven localized waves and parameter discovery in the massive Thirring model via extended physics-informed neural networks with interface zones," *Chaos, Solitons Fractals* **176**, 114090 (2023).
- ⁴²J. H. Li and B. Li, "Mix-training physics-informed neural networks for the rogue waves of nonlinear Schrödinger equation," *Chaos, Solitons Fractals* **164**, 112712 (2022).
- ⁴³Y. B. Zhang, H. Y. Liu, L. Wang, and W. R. Sun, "The line rogue wave solutions of the nonlocal Davey-Stewartson I equation with PT symmetry based on the improved physics-informed neural network," *Chaos* **33**(1), 013118 (2023).
- ⁴⁴J. Pedlosky, "Finite amplitude baroclinic wave packets," *J. Atmos. Sci.* **29**, 680–686 (1972).
- ⁴⁵M. L. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics* **29**(2), 143–151 (1987).
- ⁴⁶A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.* **18**, 1–43 (2018).
- ⁴⁷X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.* **9**, 249–256 (2010).
- ⁴⁸K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (ICCV, 2015), pp. 1026–1034.
- ⁴⁹D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.* **45**(1), 503–528 (1989).
- ⁵⁰E. Schrödinger, "An undulatory theory of the mechanics of atoms and molecules," *Phys. Rev.* **28**(6), 1049 (1926).
- ⁵¹A. Hasegawa and F. Tappert, "Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. I. Anomalous dispersion," *Appl. Phys. Lett.* **23**, 142–144 (1973).
- ⁵²L. Kavitha and M. Daniel, "Integrability and soliton in a classical one-dimensional site-dependent biquadratic Heisenberg spin chain and the effect of nonlinear inhomogeneity," *J. Phys. A* **36**(42), 10471–10492 (2003).
- ⁵³V. B. Matveev and M. A. Salle, *Darboux Transformations and Solitons* (Springer, Berlin, 1991).
- ⁵⁴J. K. Yang, *Nonlinear Waves in Integrable and Nonintegrable Systems* (Society for Industrial and Applied Mathematics, 2010).
- ⁵⁵X. Wang, B. Yang, Y. Chen, and Y. Q. Yang, "Higher-order rogue wave solutions of the Kundu-Eckhaus equation," *Phys. Scr.* **89**(9), 095210 (2014).
- ⁵⁶S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010).
- ⁵⁷A. Kundu, "Landau-Lifshitz and higher-order nonlinear systems gauge generated from nonlinear Schrödinger-type equations," *J. Math. Phys.* **25**, 3433–3438 (1984).
- ⁵⁸F. Calogero and W. Eckhaus, "Nonlinear evolution equations, rescalings, model PDEs and their integrability: I," *Inverse Probl.* **3**, 229–262 (1987).
- ⁵⁹I. Moroz, "Slowly modulated baroclinic waves in a three-layer model," *J. Atmos. Sci.* **38**, 600–608 (1981).
- ⁶⁰I. M. Moroz and J. Brindley, "Evolution of baroclinic wave packets in a flow with continuous shear and stratification," *Proc. Roy. Soc. London A* **377**, 397–404 (1981).
- ⁶¹R. Dodd, J. Eilbeck, J. Gibbon, and H. Morris, *Solitons and Nonlinear Wave Equations* (Academic Press, London, 1982), p. 630.
- ⁶²B. Tan and J. P. Boyd, "Envelope solitary waves and periodic waves in the AB equations," *Stud. Appl. Math.* **109**, 67–87 (2002).
- ⁶³A. M. Kamchatnov and M. V. Pavlov, "Periodic solutions and Whitham equations for the AB system," *J. Phys. A, Math. Gen.* **28**, 3279–3288 (1995).
- ⁶⁴R. Guo, H. Q. Hao, and L. L. Zhang, "Dynamic behaviors of the breather solutions for the AB system in fluid mechanics," *Nonlinear Dyn.* **74**, 701–709 (2013).
- ⁶⁵J. Y. Zhu and X. G. Geng, "The AB equations and the $\bar{\partial}$ -dressing method in semi-characteristic coordinates," *Math. Phys. Anal. Geom.* **17**, 49–65 (2014).
- ⁶⁶X. Wang, Y. Q. Li, F. Huang, and Y. Chen, "Rogue wave solutions of AB system," *Commun. Nonlinear Sci. Numer. Simulat.* **20**, 434–442 (2015).