

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

Li-Jun Chang(常莉君), Yi-Fan Mo(莫一凡), Li-Ming Ling(凌黎明), and De-Lu Zeng(曾德炉)
Citation: Chin. Phys. B, 2022, 31 (**6**): 060201. DOI: 10.1088/1674-1056/ac4cc5
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

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

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

Memory-augmented adaptive flocking control for multi-agent systems subject to uncertain external disturbances

Ximing Wang(王希铭), Jinsheng Sun(孙金生), Zhitao Li(李志韬), and Zixing Wu(吴梓杏)
Chin. Phys. B, 2022, 31 (**2**): 020203. DOI: 10.1088/1674-1056/ac21c1

Soliton, breather, and rogue wave solutions for solving the nonlinear Schrödinger equation using a deep learning method with physical constraints

Jun-Cai Pu(蒲俊才), Jun Li(李军), and Yong Chen(陈勇)
Chin. Phys. B, 2021, 30 (**6**): 060202. DOI: 10.1088/1674-1056/abd7e3

Exact solution of the Gaudin model with Dzyaloshinsky-Moriya and Kaplan-Shekhtman-Entin-Wohlman-Aharony interactions

Fa-Kai Wen(温发楷) and Xin Zhang(张鑫)
Chin. Phys. B, 2021, 30 (**5**): 050201. DOI: 10.1088/1674-1056/abcf43

Restricted Boltzmann machine: Recent advances and mean-field theory

Aurélien Decelle, Cyril Furtlehner
Chin. Phys. B, 2021, 30 (**4**): 040202. DOI: 10.1088/1674-1056/abd160

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

Li-Jun Chang(常莉君), Yi-Fan Mo(莫一凡), Li-Ming Ling(凌黎明), and De-Lu Zeng(曾德炉)†

School of Mathematics, South China University of Technology, Guangzhou 510640, *China*

(Received 22 November 2021; revised manuscript received 5 January 2022; accepted manuscript online 24 January 2022)

Rogue waves are a class of nonlinear waves with extreme amplitudes, which usually appear suddenly and disappear without any trace. Recently, the parity-time ($\mathcal{PT}$)-symmetric vector rogue waves (RWs) of multi-component nonlinear Schrödinger equation (n-NLSE) are usually derived by the methods of integrable systems. In this paper, we utilize the multi-stage physics-informed neural networks (MS-PINNs) algorithm to derive the data-driven $\mathcal{PT}$ symmetric vector RWs solution of coupled NLS system in elliptic and X-shapes domains with nonzero boundary condition. The results of the experiment show that the multi-stage physics-informed neural networks are quite feasible and effective for multi-component nonlinear physical systems in the above domains and boundary conditions.

Keywords: nonlinear Schrödinger equation, vector rogue waves, deep learning, numerical simulations

PACS: 02.30.Ik, 02.60.Cb, 07.05.Mh **DOI:** 10.1088/1674-1056/ac4cc5

1. Introduction

Nonlinear Schrödinger (NLS) equations generally can be used to describe the nonlinear wave phenomena in diverse physics fields and have attracted more and more attention especially in hydrodynamics, nonlinear optical fibers, planar wave guides, and Bose–Einstein condensates theory.[1–5] For the integrable NLS systems, there exist lots of exact solutions including solitons, breathers, and rogue waves (RWs). They are all nonlinear waves with localized structure both in time and space. Note that, on the one hand, some solitons will produce breathers when they are disturbed periodically. And the RWs are the limit form of the breathers. On the other hand, solitons are stable waves and invariant of shape in the evolution under vanishing background, while RWs are unstable waves and the variation of shape in the evolution under non-vanishing background due to modulation instability. Especially, the amplitudes of RWs are usually two times or even higher than those of its surrounding waves, hence they frequently cause sea accidents and pose a great threat to people's navigation safety for years. RWs were firstly constructed analytically by Peregrine,[4] expressed by the first-order functions. But these solutions did not get much attention until Solli *et al.* observed optical RWs in an optical fiber experiment.[6] Zakharov *et al.*[7] found the physical mechanism of usual RW generation was frequently related with modulation instability. Afterwards, RWs of different physical systems were one after another derived by related Darboux transformation.[8–16] Furthermore, recent studies have shown that the vector RWs of n-component NLS equations are the $\mathcal{PT}$ symmetry under the constraints of some parameters and classified into n types in terms of the degree of polynomials.[17–19] In this paper, our research concentrates on data-driven $\mathcal{PT}$-symmetric vector RW solutions for the focusing multi-component NLS equation with nonzero boundary conditions:

$$\mathrm{i}\boldsymbol{q}_t + \frac{1}{2}\boldsymbol{q}_{xx} + ||\boldsymbol{q}||_2^2\boldsymbol{q} = \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{q}(x,t) = (q_1(x,t),\ldots,q_n(x,t))^{\mathrm{T}}$ is the complex vector field and the symbol "T" represents the transpose. The subscripts t and x denote the partial derivative with respect to t and x, respectively. $||\boldsymbol{q}||_2 = \sqrt{|q_1|^2+\cdots+|q_n|^2}$ stands for the Euclidean norm of the vector $\boldsymbol{q}$.

In the last decades, deep learning has achieved great success in a wide variety of areas because of strong data representation ability, such as image processing, speech recognition, natural language processing, and many more.[20–24] Since neural networks (NNs) are universal approximation of functions,[25] as we know, it is natural to solve differential equations using NNs. Lagaris *et al.*[26] presented a method to solve ordinary differential equations (ODEs) and partial differential equations (PDEs) by artificial neural networks (ANNs) for a given form of trainable solution. Sirignsno *et al.*[27] introduced deep Galerkin method (DGM) to solve high dimensional PDE by Monte Carlo method for fast computation of second derivatives and an integral along its respective domain under proper measurement. Raissi *et al.*[28] put forward physics-informed neural networks (PINNs) method to solve PDEs and inverse problems through automatic differentiation technique (AD)[29] by random sampling points in the space domain. After that the variations of PINNs[30–46] have widely applied to solve distinct kinds of PDEs over years. Lin *et al.*[35] utilized the two-stage PINN to simulate abundant localized wave solutions of integrable equations by introducing the measurement of conserved quantities into mean squared error loss method. Pu *et al.*[36–38] proposed

†Corresponding author. E-mail: dlzeng@scut.edu.cn

© 2022 Chinese Physical Society and IOP Publishing Ltd

the improved PINN with neuron-wise locally adaptive activation function to simulate vector localized waves of Manakov system such as one-rational soliton solutions, RWs solution, breathers, and their interaction solutions. Wang *et al.*[39] combined the classical spectral method with PINN algorithm to correct the error perturbation caused by modulation instability on boundary problem and further simulate more accurate RW or breather solutions of NLS equation with a smaller numerical error for long time. However, mainly due to the difficulties of the computational complexity and the algorithmic instability in solving n-component coupled nonlinear equations, previous studies of data-driven RWs mostly focused on one-component systems. The dynamical behaviors and relevant patterns of data-driven RWs of multi-component systems such as two- and three-component are less studied with deep learning method. Recently, in this respect, Mo *et al.*[42] proposed the MS-PINN algorithm which picks the pre-fixed points for training the network again to simulate the data-driven degenerate/non-degenerate vector solitons for coupled NLS equation with the zero boundary theory. The MS-PINN algorithm combines the multiple thoughts of time-adaptive, adaptive sampling of collocation points, error measurement, and adaptive-weight[34,43–45] and obtains good numerical results on 2-NLS systems in the rectangle shape domain. Peng *et al.*[46] used the Riemann–Hilbert method and PINN algorithm to solve N-double poles solutions for the non-local Hirota equation with nonzero boundary conditions. Inspired by above some papers, one of our goals of this work is to extend the zero boundary theory to the non-zero ones for the vector NLS equation with the MS-PINN algorithm.[42] In addition, we use the elliptic and X-shape spatio-temporal boundary conditions to simulate the $\mathcal{PT}$ symmetric RWs of 2-NLS and 3-NLS systems. To the best of our knowledge, these results have never been reported before.

The structure of this paper is assigned as follows. In Section 2, we introduce briefly the PINN algorithm and give detailed descriptions about MS-PINN training method. In Section 3, we use the MS-PINN to simulate data-driven vector RW solutions of the 2-NLS system (1) in elliptic and X-shapes domains. Meanwhile, data-driven $\mathcal{PT}$-symmetric two-vector RW solutions of the 3-NLS system are also studied. In Section 4, some conclusions and discussions are given.

2. Multi-stage physics informed neural networks

In general, if we want to solve Eq. (1) by using deep learning, the most common neural networks are PINNs. The PINNs set up a deep neural network with the input x, t, and the output $\boldsymbol{q}(x,t)$, such that the output values can well approximate the solution of Eq. (1). PINNs take physical information $\boldsymbol{f}(x,t)$ defined by Eq. (2) into consideration and use it as a part of the total loss functions on the spatio-temporal domain beside satisfying the initial and boundary conditions. And then we minimize the loss function through the optimization algorithm such as Adam[47] or L-BFGS[48] in order to obtain the optimal parameters. Define the residual function $\boldsymbol{f}(x,t)$ of n-component NLS system Eq. (1) below

$$\boldsymbol{f}(x,t) := \mathrm{i}\boldsymbol{q}_t + \frac{1}{2}\boldsymbol{q}_{xx} + ||\boldsymbol{q}||_2^2\boldsymbol{q}, \tag{2}$$

where $\boldsymbol{q}(x,t)$ and $\boldsymbol{f}(x,t)$ are both complex-valued functions and can be reformulated as $\boldsymbol{q}(x,t) = \boldsymbol{a}(x,t) + \mathrm{i}\boldsymbol{b}(x,t)$ and $\boldsymbol{f}(x,t) = \boldsymbol{f_a}(x,t) + \mathrm{i}\boldsymbol{f_b}(x,t)$. $\boldsymbol{a}(x,t) = (a_1(x,t),\ldots,a_n(x,t))$ and $\boldsymbol{b}(x,t) = (b_1(x,t),\ldots,b_n(x,t))$. $\boldsymbol{f_a}(x,t)$ and $f_b(x,t)$ are real value functions and by the above residual formula, they satisfy

$$\begin{aligned}\boldsymbol{f_a} &= \boldsymbol{b}_t - \frac{1}{2}\boldsymbol{a}_{xx} - \left(|\boldsymbol{a}|^2 + |\boldsymbol{b}|^2\right)\boldsymbol{a},\\ \boldsymbol{f_b} &= \boldsymbol{a}_t + \frac{1}{2}\boldsymbol{b}_{xx} + \left(|\boldsymbol{a}|^2 + |\boldsymbol{b}|^2\right)\boldsymbol{b}.\end{aligned} \tag{3}$$

We can formulate the total loss function by the most common mean square error $\widetilde{\mathrm{Loss}} = \mathrm{Loss_f} + \mathrm{Loss_i}$, where

$$\mathrm{Loss_f} = \frac{1}{N_\mathrm{f}}\sum_{j=1}^{N_\mathrm{f}}\left(\left|\boldsymbol{f_a}\left(x_\mathrm{f}^j, t_\mathrm{f}^j\right)\right|^2 + \left|\boldsymbol{f_b}\left(x_\mathrm{f}^j, t_\mathrm{f}^j\right)\right|^2\right), \tag{4}$$

$$\mathrm{Loss_i} = \frac{1}{N_\mathrm{u}}\sum_{j=1}^{N_\mathrm{u}}\left(\left|\boldsymbol{a}\left(x_\mathrm{u}^j, t_\mathrm{u}^j\right) - \boldsymbol{a}_\mathrm{u}^j\right|^2 + \left(\left|\boldsymbol{b}\left(x_\mathrm{u}^j, t_\mathrm{u}^j\right) - \boldsymbol{b}_\mathrm{u}^j\right|^2\right). \tag{5}$$

Here $\{(x_\mathrm{f}^j, t_\mathrm{f}^j)\}_{j=1}^{N_\mathrm{f}}$ represents the training collocation points which can be sampled using the classical latin hypercube sampling (LHS) technique[49] in the space–time domain. $\{(x_\mathrm{u}^j, t_\mathrm{u}^j)\}_{j=1}^{N_\mathrm{u}}$ denotes the sampled points on the initial and boundary domains. N_f, N_u are the numbers of corresponding sample points respectively. $\mathrm{Loss_f}$ makes the output value produced by deep neural network meet nonlinear coupled system. $\mathrm{Loss_i}$ forces the predicted solution to satisfy the initial and boundary conditions of coupled NLS equation. With the help of reverse mode differentiation, we can get the partial derivative of the target function with respect to time variable and spatial variable.[29] To obtain the optimal parameters of the training model, we minimize the total loss function to update the parameters with the optimizer.[48] For more information of PINNs, please refer to the work.[28]

In the multi-stage physics informed neural network (MS-PINN) algorithm, as shown in Fig. 1, we do not sample data points in the boundary region and the initial region respectively according to the traditional classical sampling method, but choose the adaptive spatio-temporal boundary points in manner of surrounding sampling on time and space together. Take the elliptic domain as an example, as displayed in Fig. 1(a), we first treat the boundary points of the elliptic domain in the direction of time and space as the spatio-temporal boundary points (yellow point) and then decompose the whole domain into N parts which are kept apart from the spatio-temporal boundary points, forming one-to-one correspondence stages. Each stage we collect some collocation

points (blue points) in advance. Suppose the total collocation points are $X_{\rm f} = \left\{(x_{\rm f}^j, t_{\rm f}^j)\right\}_{j=1}^{N_{\rm f}}$, the spatio-temporal boundary points are $X_i = \left\{(x_0^j, t_0^j)\right\}_{j=1}^{N_0}$. So it matters that how to select collocation points of each stage. We first designate the distance of the j-th collocation point from the spatio-temporal boundary points, as defined below:

$$\mathrm{dis}_j = \min_{i=1,\cdots,N_0} ||(x_{\rm f}^j, t_{\rm f}^j) - (x_0^i, t_0^i)||_2^2, \quad j = 1,\ldots,N_{\rm f}. \tag{6}$$

That is to say, we compare every collocation points with all spatio-temporal boundary points $X_{\rm i}$, then select the minimum value as the distance of the collocation point from $X_{\rm i}$. Followed by putting the distance in order from the smallest value to the largest value, thus the sets of these collocation points corresponding to N stages are selected

$$\left\{ X_{\rm f}^k = \left\{(x_{\rm f}^j, t_{\rm f}^j)\right\}_{j=1}^{(k/N)N_{\rm f}} \middle| k = 1,\ldots,N \right\}, \ X_{\rm f}^k \subset X_{\rm f}^{k+1}.$$

To obtain more accurate prediction, we choose N_k points which correspond to the minimum values of $\mathrm{Loss}_{\rm f}$ computed by Eq. (4) in sequence as pre-fixed points (good points) $X_{\rm p}^k = \left\{(x_{\rm p}^j, t_{\rm p}^j)\right\}_{j=1}^{N_k}$ which actually could be considered as a kind of history information produced by each stage to further get the smaller loss function. Then we transmit the pre-fixed points (red points) produced by last stage to train in the next stage.

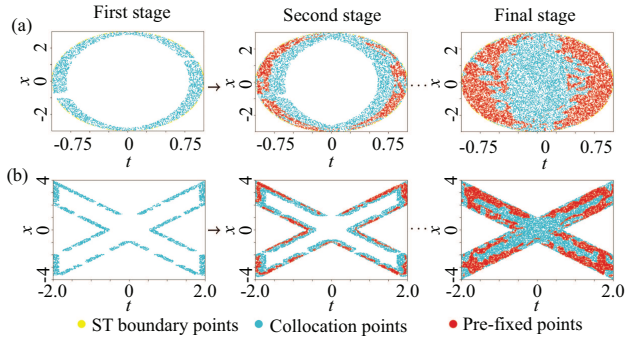


Fig. 1. Adaptive sampling collocation points in space-time. (a) and (b) The pre-fixed points from the one stage next stage the in elliptic domain and X shape domain, respectively.

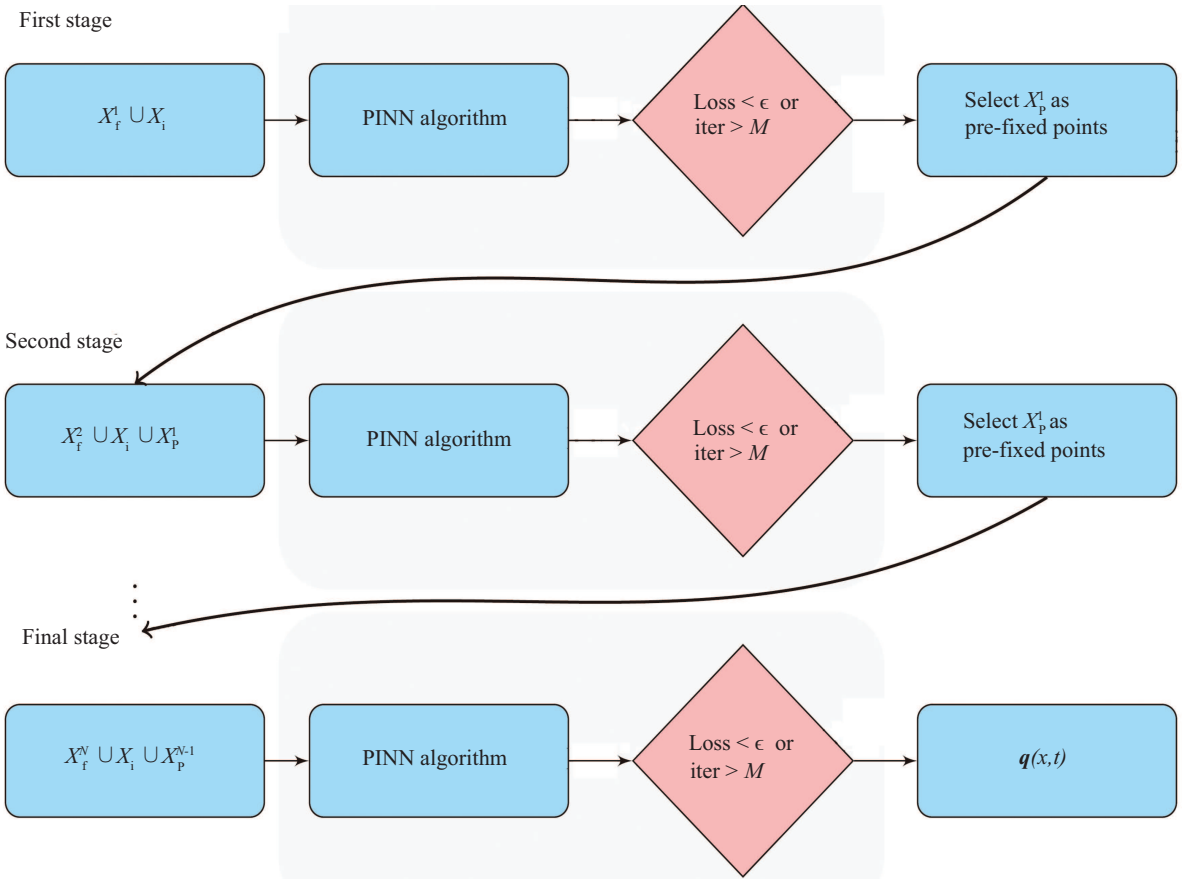


Fig. 2. The MS-PINNs architecture. The first stage: the input $X_{\rm f}^1 \cup X_{\rm i}$. By PINN algorithm the pre-fixed points $X_{\rm p}^1$ are obtained. The second stage: the input are $X_{\rm f}^2 \cup X_{\rm i} \cup X_{\rm p}^1$. By PINN algorithm, we obtain pre-fixed points $X_{\rm p}^2$. The last stage: the input is $X_{\rm f}^N \cup X_{\rm i} \cup X_{\rm p}^{N-1}$. By PINN algorithm, the output is $\boldsymbol{q}(x,t)$.

To more clearly represent the algorithm flowing-chart, we give the model architecture, as shown in Fig. 2. In the 1-st stage the inputs are the collocation points $X_{\rm f}^1$ and spatio-temporal boundary points $X_{\rm i}$, then we train the parameters by using PINN algorithm. When the loss function less than the given threshold value or the training iteration reaches our set M, we manually select the $X_{\rm p}^1$ pre-fixed points which contribute to the smallest loss value of $\mathrm{Loss}_{\rm f}$. Similarly, in the 2-nd stage the inputs are $X_{\rm f}^2$, $X_{\rm i}$, and $X_{\rm p}^1$, we still train the model using PINN algorithm and when the loss function less than

the given threshold value or the training reaches the maximum iteration, we manually select the $X_{\rm P}^2$ pre-fixed points which contribute to the smallest loss value and so on. At every stage of training, we can get the approximate value $\boldsymbol{q}(\boldsymbol{x},\boldsymbol{t})$ and use it to predict $\left\{\boldsymbol{a}_{\rm f}^j,\boldsymbol{b}_{\rm f}^j\right\}_{j=1}^{N_k}$ at pre-fixed points $\left\{(x_{\rm f}^j,t_{\rm f}^j)\right\}_{j=1}^{N_k}$ as another constraint term of the loss function.

$$\text{Loss}_{\rm s}=\frac{1}{N_k}\sum_{j=1}^{N_k}\left(\left|\boldsymbol{a}\left(x_{\rm f}^j,t_{\rm f}^j\right)-\boldsymbol{a}_{\rm f}^j\right|^2+\left|\boldsymbol{b}\left(x_{\rm f}^j,t_{\rm f}^j\right)-\boldsymbol{b}_{\rm f}^j\right|^2\right). \tag{7}$$

Thus the total loss function could be defined below:

$$\text{Loss}=\lambda_1\text{Loss}_1+\lambda_2\text{Loss}_2+\text{Loss}_{\rm f},$$

where $\text{Loss}_1=\text{Loss}_{\rm i},\text{Loss}_2=\text{Loss}_{\rm s}$ are the loss functions produced by the spatio-temporal boundary conditions and the pre-fixed points, respectively. $\text{Loss}_{\rm f}$ denotes n-NLS equation residual loss. λ_1 and λ_2 denote the adaptive penalty of loss functions in each stage, which are updated in an exponential moving average way where it could smooth the balancing and avoid drastic changes of the the loss space between optimization steps[45,47]

$$\begin{aligned}&\lambda_i=(1-\alpha)\lambda_i+\alpha\hat{\lambda}_i,\\&\hat{\lambda}_i=\frac{\max_\theta\{|\nabla_\theta\text{Loss}_{\rm f}|\}}{\overline{|\nabla_\theta\text{Loss}_{\rm i}|}},\quad i=1,2,\end{aligned} \tag{8}$$

where $\overline{|\nabla_\theta\text{Loss}|}$ denotes the mean of $|\nabla_\theta\text{Loss}|$ with respect to parameters θ. α is the hyper-parameter which is recommended values with 0.9.

3. Data-driven $\mathcal{PT}$ symmetric rogue waves of n-NLS equation

RWs, as a special type of nonlinear waves, have attracted more and more attentions in optical systems and other scientific disciplines. Recently, many studies on $\mathcal{PT}$-symmetric vector RW solutions in any n-component NLS models have also been proposed. However, due to modulation instability mechanism and the difficulty of observational conditions for RWs, we do not have a complete understanding of RW phenomena. Thus in this section, we would like to precisely simulate the $\mathcal{PT}$ symmetric vector RW solutions of 2-NLS and 3-NLS systems with the non-zero spatio-temporal boundary conditions by MS-PINN algorithm.

3.1. The $\mathcal{PT}$ symmetric fundamental vector RW solution of 2-NLS system

Specifically, the fundamental vector RW solutions

$$\boldsymbol{q}(x,t)=(q_1(x,t),\ldots,q_n(x,t))$$

of the n-NLS equations have been derived by Ref. [18], and the formula is

$$q_j=q_j^{\rm bg}[1+R_j(x+\alpha_0+\frac{1}{2\zeta},t)]\exp(-2{\rm i}\omega_j) \tag{9}$$

for $j=1,\ldots,n$, where

$$\begin{aligned}&q_j^{\rm bg}=a_j\exp({\rm i}\theta_j),\quad \theta_j=b_jx-\left(\frac{1}{2}b_j^2-||\boldsymbol{a}||_2^2\right)t,\\&R_j(x,t)=\frac{1}{a_j^2}\frac{2{\rm i}(b_jx-\zeta^2t)-1}{x^2+\zeta^2t^2+\zeta^2/4},\end{aligned}$$

$a_j=\zeta\csc\omega_j,b_j=\zeta\cot\omega_j$, a_j, b_j, α_0, ζ, $\omega_j\in\mathbb{R}$ are the parameters which nontrivially contribute to the profile of the vector RWs. The $\mathcal{PT}$-symmetric structure of a vector RW is $\boldsymbol{q}_j(x,t)=\mathcal{PT}\boldsymbol{q}_j(x,t)=\mathcal{P}\boldsymbol{q}_j^*(x,-t)$. $\mathcal{T}$ is the conventional bosonic the time-reversal operator: $t\to-t$, ${\rm i}\to-{\rm i}$, where $*$ denotes the complex conjugation.

For a 2-NLS system, we can obtain explicitly the vector RW solution $\boldsymbol{q}(x,t)=(q_1(x,t),q_2(x,t))$ with the parameters $\omega_1=\pi/3$, $\omega_2=2\pi/3$, $\zeta=1$, $\alpha_0=-1/2$, $t\in[-1,1]$, $x\in[-4,4]$ by Eq. (9). We simulate the data-driven vector RWs by deep neural network with 3 hidden layers and 50 neurons per hidden layer. The activation function is the hyperbolic tangent (tanh) function. By dividing the whole domain into 4 parts and following 4 stages, we set the numbers of total collocation points $N_{\rm f}=2\times10^4$, spatio-temporal boundary points $N_0=200$ and pre-fixed points $N_k=(0.6k/4)N_{\rm f}=3\times10^3k$, $(k=1,\ldots,4)$ which means 60% of collocation points are selected as pre-fixed points. In each stage the number of iterations of is 5000 and the learning rate ε is $1e-4$ for training with optimizer L-BFGS.

We know RWs usually admit one peak and two valleys, one valley and two peaks and two peaks and two valleys. For the RW of scalar NLS equation, RW usually has one peak and two valleys and the maximum amplitude is twice height of the background field. For the RWs of multi-component NLS equation, there exist the bright, dark and four-petal type RWs, which are the generalization of classic RWs. For more information about the classification of fundamental RW types, please refer to these articles.[18,50,51] Each component q_j of the 2-NLS system is a kind of the fundamental RW and the types of the structures are controlled by ω_j. As shown in Fig. 3, q_1 and q_2 are eye-shaped RWs with one hump and two valleys in top panel. Then we take relative $\mathbb{L}_2$ (RL_2) error as evaluation metrics in order to validate the method with simulated data. At the same time, considering the effect of noise in real-world observations, we add 5% white noise data and 10% white noise data to the spatio-temporal boundary condition. As shown in Fig. 4, the absolute error $|q_1|+|q_2|$ of the MS-PINN method is nearly 30 times less than that of the PINN method. The absolute errors of the MS-PINN method with 5% and 10% white noise data still nearly 20 times and 6 times less than PINN algorithm.

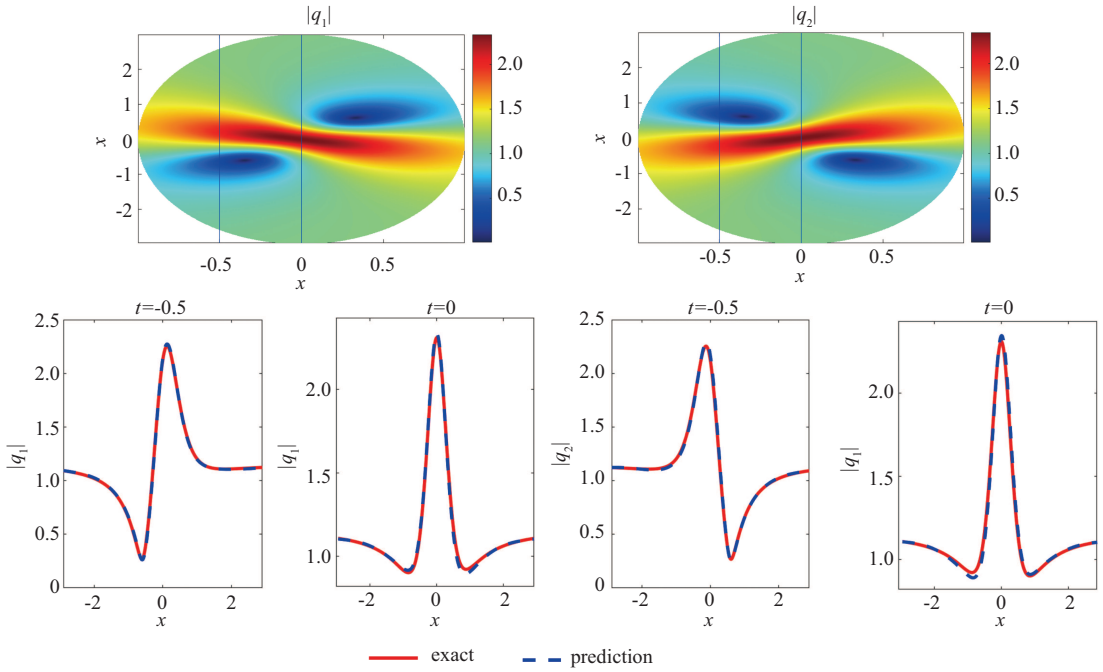


Fig. 3. The intensity of $\mathcal{PT}$ symmetric vector RW solutions of the 2-NLS system simulated by the MS-PINN algorithm in elliptic area and the comparison between the intensity of the predicted solutions and accurate solutions of two components system at times $t = -0.5$ and $t = 0$ corresponding to the two temporal snapshots depicted by the blue vertical lines in the top panel.

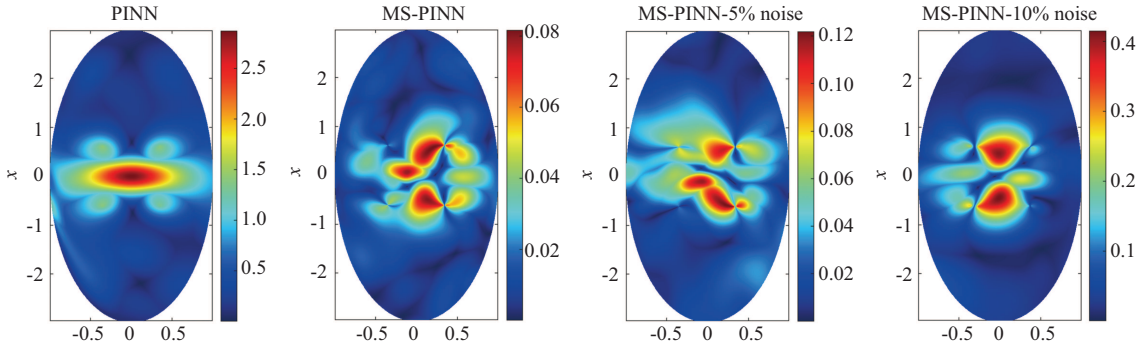


Fig. 4. The absolute error $|q_1| + |q_2|$ of PINN method, MS-PINN algorithm with clean data, MS-PINN algorithm with 5% white noise data and with 10% white noise data.

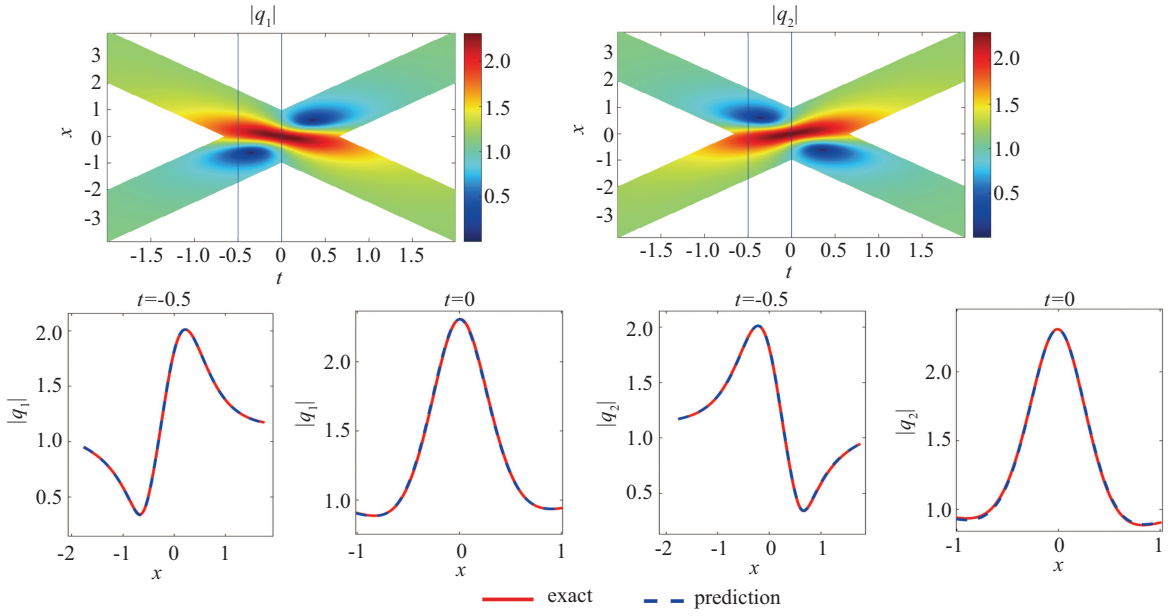


Fig. 5. The intensity of $\mathcal{PT}$ symmetric vector RWs of the 2-NLS equation simulated by the MS-PINN method in X-shape domain and the comparison between the intensity of the predicted solutions and accurate solutions of two components system at times $t = -0.5$ and $t = 0$ corresponding to the two temporal snapshots depicted by the blue vertical lines in the top panel.

In order to explain the applicability of the algorithm in complex geometry domain for 2-NLS equation, we replace the oval region with the X-shape domain, as shown in Fig. 5. By the comparison between the predicted solutions and exact solutions of 2-NLS equation, we find the MS-PINN algorithm also well approximate the exact solution. Figure 6 shows the absolute error of the PINN method is about much 100 times, 30 times, and 10 times larger than those of the MS-PINN method with clean training data, 5% white noise data and 10% white noise data, respectively. It can be seen that the data-driven solutions of nonlinear coupled system can be still kept relatively stable against the small perturbation.

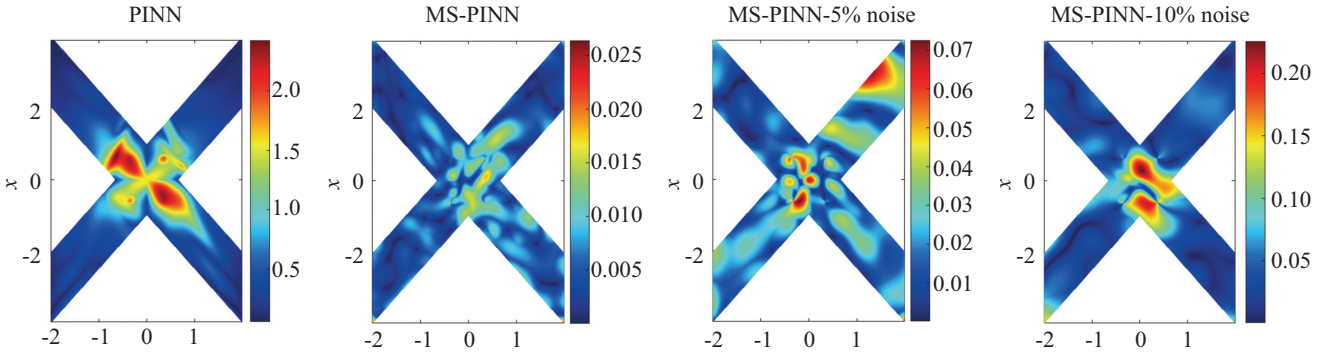


Fig. 6. The absolute error $|q_1|+|q_2|$ of PINN method, MS-PINN algorithm with clean data, MS-PINN algorithm with 5% white noise data and with 10% white noise data.

Table 1. The comparison of the PINN method and MS-PINN method base on 5 independent repeated experiments for 2-NLSE in elliptic-and X-shape regions.

Method	RL_2 error	Time
PINN-E	$28.86 \pm 0.35\%$	1820.64 ± 214.51
MS-PINN-E	$0.63 \pm 0.18\%$	896.64 ± 58.08
MS-PINN-E (5% noise data)	$1.24 \pm 0.18\%$	995.26 ± 52.07
MS-PINN-E (10% noise data)	$3.29 \pm 0.46\%$	1037.88 ± 42.45
PINN-X	$24.79 \pm 1.79\%$	1753.97 ± 82.26
MS-PINN-X	$0.23 \pm 0.01\%$	1289.24 ± 88.91
MS-PINN-X (5% noise data)	$1.04 \pm 0.08\%$	1484.01 ± 41.71
MS-PINN-X (10% noise data)	$2.60 \pm 0.09\%$	1508.53 ± 41.08

Because the neural network may get different solutions from different initial data, we train the neural network from random initialization for 5 times. More detailed information about RL_2 error and time cost with PINN method and MS-PINN algorithm in elliptic- and X-shape domains are displayed in Table 1. "-E, -X" denote the abbreviations of ellipse and X shape. PINN-E and MS-PINN-X represent the uses of the PINN and MS-PINN algorithm in ellipse and X-shape region, respectively.

3.2. The $\mathcal{PT}$ symmetric vector two-RW solutions of 3-NLS system

In the above subsection, we have simulated the $\mathcal{PT}$ symmetric fundamental vector RW solutions of the relative 2-NLS equations and analyzed the RL_2 error of MS-PINN algorithm with 5% white noise data and 10% white noise data. To our knowledge, simulating asymptotic behaviors of the $\mathcal{PT}$ symmetric vector RW solutions of 3-NLS system with deep learning has not been evaluated at present. For 3-NLS equation, we choose an exact vector two-RW solutions $\boldsymbol{q}(x,t)=(q_1(x,t),q_2(x,t),q_3(x,t))$ given by setting the special parameters[18]

$$q_1(x,t)=\frac{B_1+\mathrm{i}B_2}{A}\mathrm{e}^{(\mathrm{i}/2)(9t+2x+59)},\ q_2(x,t)=\frac{C}{A}\mathrm{e}^{5\mathrm{i}t},\ q_3(x,t)=q_1^*(x,-t), \tag{10}$$

where

$$\begin{aligned}
B_1&=2\sqrt{2}\left[2x^3+(1-2t)x^2+(2t^2+2t-120)x-2t^3-3t^2-120t-\frac{119}{2}\right],\\
B_2&=-2\sqrt{2}\left[x^4+(2t^2-121)x^2+(4t^2+2t)x+t^4+120t^2-2t+3600\right],\\
A&=2x^4+(4t^2-238)x^2+(-8t^2+2)x+2t^4+244t^2+7201,\\
C&=8\left[-\frac{t^4}{4}+\mathrm{i}t^3+\left(-\frac{1}{2}x^2+x-29\right)t^2+\mathrm{i}(x^2+59)t-\frac{x^4}{4}+\frac{121}{4}x^2+\frac{3x}{4}-\frac{6959}{8}\right].
\end{aligned}$$

We simulate the data-driven $\mathcal{PT}$-symmetric vector RW solutions by deep neural networks with 3 hidden layers neural network and 200 neurons per hidden layer. By dividing the whole domain into 4 parts, we have naturally 4 stages. The data-driven vector RW solutions of system resulted from the MS-PINN algorithm with the randomly chosen spatio-temporal boundary points $N_0=400$, total number of collocation points $N_\mathrm{f}=4\times10^4$ and the number of pre-fixed points $N_k=(0.6k/4)N_\mathrm{f}=6\times10^3k$,

($k = 1,2,3,4$). In other words, we consider 60% of collocation points as pre-fixed points in trained domain for each stage. In each stage the maximum iteration M is 2×10^4 and the learning rate threshold ε is 10^{-3} with optimizer L-BFGS.

Figure 7 shows the intensity plots of 3-NLS equation on $x\in[-20,20], t\in[-0.5,0.5]$. Top panel shows the density evolutions of 3-NLS system. Bottom panel shows the intensity of three components at $t=-0.25$ and $t=0.25$ corresponding to the two temporal snapshots depicted by the white horizontal lines in the top panel. Based on the classification of fundamental RW types,[18] the density plots of q_1 and q_3 for the 3-NLS system are both four-petaled RWs since $\omega_1,\omega_3\in(\pi/6,\pi/3)\cup(2\pi/3,5\pi/6)$, while the intensity plot of q_2 is the bright RW due to $\omega_2\in[\pi/3,2\pi/3]$. The simulation shows that the data-driven predicted solutions are very close to the exact solutions given before, which further demonstrates the observability of RWs in the physical experiments.

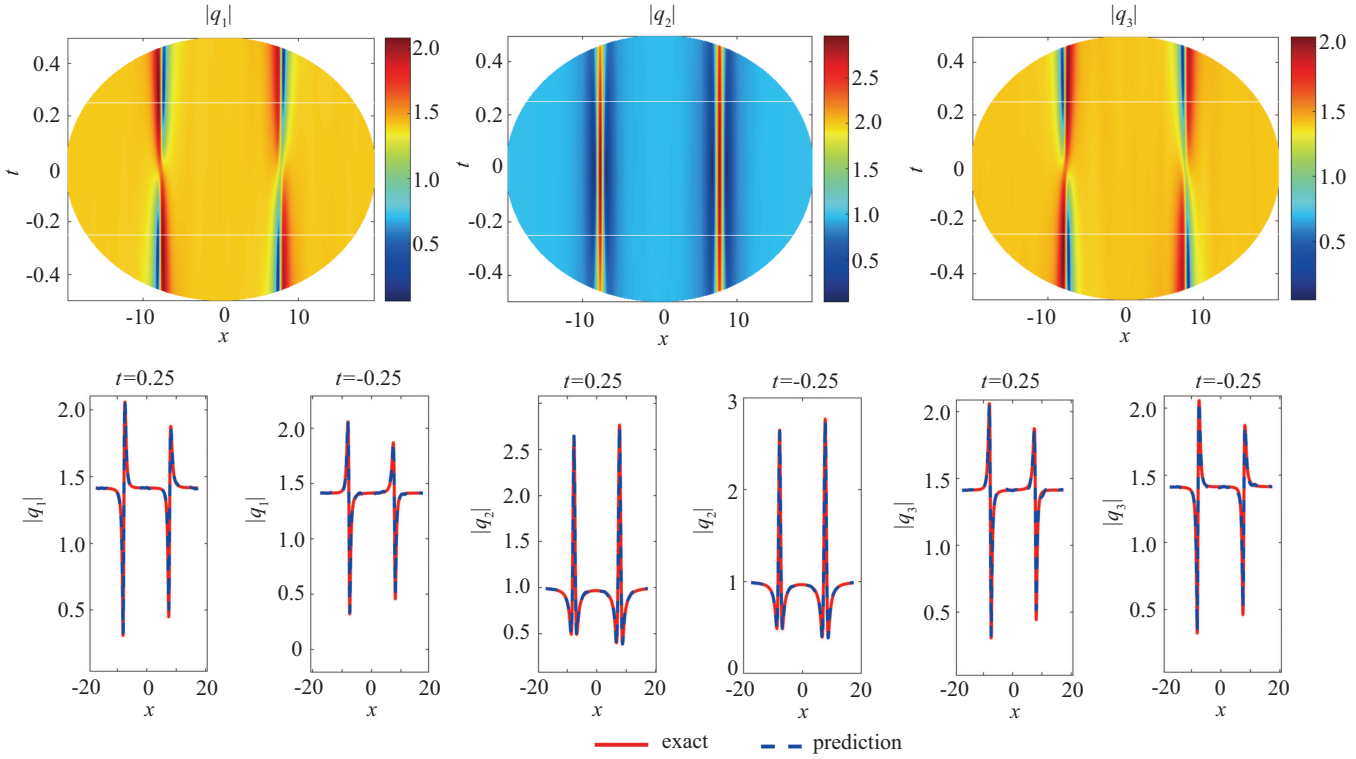


Fig. 7. The intensity plots of the $\mathcal{PT}$ symmetric vector RW solutions of 3-NLS system simulated by MS-PINN method and the six plots of the comparison between the of the predicted solutions and accurate solutions of three components system at times $t=-0.25$ and $t=0.25$.

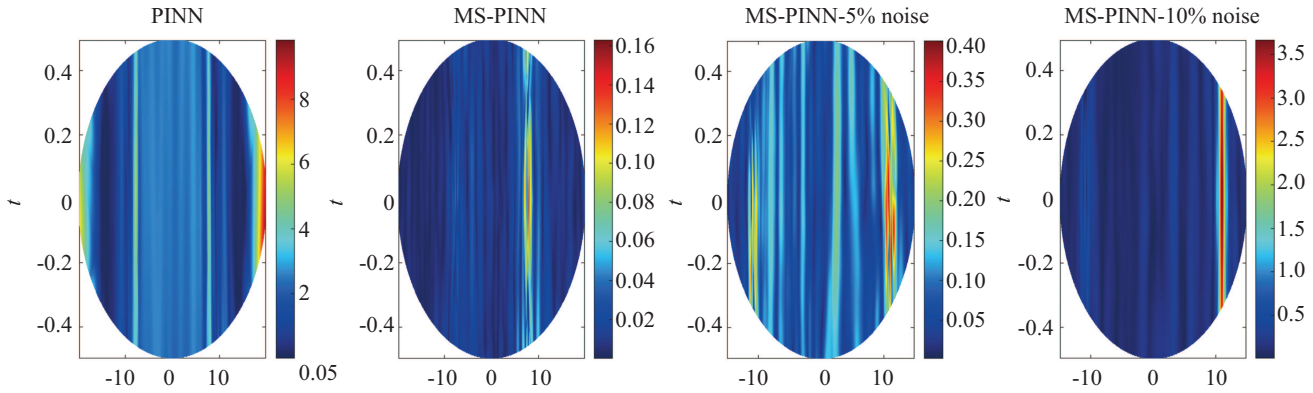


Fig. 8. The absolute error $|q_1|+|q_2|+|q_3|$ of PINN method, MS-PINN algorithm with clean data, MS-PINN algorithm with 5% white noise data and with 10% white noise data, respectively.

The absolute error $|q_1|+|q_2|+|q_3|$ of the PINN method and that of the MS-PINN method (with clean data, 5% white noise data and 10% white noise data) are displayed in Fig. 8. Obviously, the MS-PINN method obtains much 50 times better experimental effect than that of the PINN algorithm on 3-NLS equation. Though we add some white noise data to spatio-temporal boundary condition, the absolute errors are still much less than that of the PINN algorithm. For 3-NLS system, more specific RL_2 errors and time costs with PINN method and MS-PINN algorithm with white noise data are shown in Table 2.

Table 2. The comparison of the PINN method and MS-PINN method base on 5 independent repeated experiments for 3-NLSE.

Method	RL_2 error (%)	Time
PINN	49.70 ± 9.26	12835.22 ± 1256.40
MS-PINN (clean data)	1.78 ± 0.46	6911.70 ± 443.23
MS-PINN (5% noise data)	10.34 ± 3.69	9863.74 ± 354.53
MS-PINN (10% noise data)	30.68 ± 10.75	10370.80 ± 337.53

4. Conclusion

In this work, we utilize a multi-stage deep learning training algorithm to simulate data-driven $\mathcal{PT}$-symmetric vector RW solutions of 2-NLS systems and two-vector RWs of 3-NLS systems with spatio-temporal boundary condition (non-zero boundary condition). In fact, the spatio-temporal conditions do not conform to the traditional numerical methods of solving the equation in manner of time evolution, but comply with physics dynamics in the way of simultaneous evolution of time and space. Numerical simulations show that the MS-PINN algorithm can well recover different dynamical behaviors of RW solutions in the 2-NLS and 3-NLS equations. In the future, we would like to explore whether the MS-PINN algorithm is applied to the discovery problems of other integrable NLS and non-integrable nonlinear wave system by adding more prior knowledge about the geometry and physical properties of system.

Acknowledgments

Project supported by National Natural Science Foundation of China (Grant Nos. 11771151, 61571005, and 61901160), the Science and Technology Program of Guangzhou (Grant No. 201904010362), and the Fundamental Research Program of Guangdong Province, China (Grant No. 2020B1515310023).

References

[1] Draper L 1967 *Mar. Geol.* **5** 133
[2] Parkins A S and Walls D F 1998 *Phys. Rep.* **303** 1
[3] Agrawal G P 2001 *Nonlinear Fiber Optics*, 3rd edn. (San Diego: Academic Press) pp. 195–211
[4] Peregrine D H 1983 *ANZIAM J.* **25** 16
[5] Zabusky N J and Martin D K 1965 *Phys. Rev. Lett.* **15** 240
[6] Solli D R, Ropers C, Koonath P and Jalali B 2007 *Nature* **450** 1054
[7] Zakharov V E and Ostrovsky L A 2009 *Physica D* **238** p. 540
[8] Akhmediev N, Adrian A and Soto-Crespo J M 2009 *Phys. Rev. E* **80** 026601
[9] Guo B L and Ling L M 2011 *Chin. Phys. Lett.* **28** 110202
[10] Guo B L, Tian L X, Yan Z Y, Ling L M and Wang Y F 2017 *Rogue Waves* (Berlin: De Gruyter)
[11] Ling L M, Zhao L C and Guo B L 2016 *Commun. Nonlinear Sci. Numer. Simul.* **32** 285
[12] Shabat A and Zakharov V 1972 *JETP Lett.* **34** 62
[13] Calini A and Schober C M 2008 *Extreme Ocean Waves* p. 31
[14] Ohta Y and Yang J K 2012 *P. Roy. Soc. A-Math. P.* **468** 1716
[15] Zhang X E and Chen Y 2018 *Nonlinear Dyn.* **93** 2169
[16] Feng B F, Ling L M and Daisuke A T 2020 *Stud. Appl. Math.* **144** 46
[17] Zhang G Q, Ling L M, Yan Z Y and Konotop V V 2021 *Chaos* **31** 063120
[18] Zhang G Q, Ling L M and Yan Z Y 2021 *J. Nonlinear Sci.* **31** 1
[19] Yang B and Yang J K 2020 *J. Math. Anal. Appl.* **487** 124023
[20] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge: The MIT Press)
[21] AnzaiAnzai Y 2012 *Pattern recognition and machine learning* (Morgan Kaufmann Publishers)
[22] Deng L, Hinton G and Kingsbury B 2013 *Proc. IEEE Int. Conf. Acoust Speech Signal Process* p. 8599
[23] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu, K and Kuksa P 2011 *J. Mach. Learn. Res.* **12** 2493
[24] Krizhevsky A, Sutskever I and Hinton G E 2012 *Adv. Neural Inf. Process Syst.* **25** 1097
[25] Hornik K, Stinchcombe M and White H 1989 *Neural Networks* **2** 359
[26] Lagaris I E, Likas A and Fotiadis D I 1998 *IEEE Trans. Neural Netw.* **9** 987
[27] Sirignano J and Spiliopoulos K 2018 *J. Comput. Phys.* **375** 1339
[28] Raissi M, Perdikaris P and Karniadakis G E 2019 *J. Comput. Phys.* **378** 686
[29] Baydin A G, Pearlmutter B A, Radul A A and Siskind J M 2018 *J. Mach. Learn. Res.* **18** 1
[30] Pang G F, Lu L and Karniadakis G E 2019 *SIAM J. Sci. Comput.* **41** A2603
[31] Dwivedi V, Parashar N and Srinivasan B 2019 arXiv: 1907.08967
[32] Pang G, D'Elia M, Parks M and Karniadakis G E 2020 *J. Comput. Phys.* **422** 109760
[33] Meng X, Li Z, Zhang D and Karniadakis G E 2020 *Comput. Methods Appl. Mech. Eng.* **370** 113250
[34] Jagtap A D and Karniadakis G E 2020 *J. Comput. Phys.* **28** 2002
[35] Lin S N and Chen Y 2107 arXiv: 2107.01009
[36] Pu J C, Peng W Q and Chen Y 2021 *Wave Motion* **107** 102823
[37] Pu J C, Li J and Chen Y 2021 *Chin. Phys. B* **30** 060202
[38] Pu J C and Chen Y 2109 arXiv: 2109.09266
[39] Wang R Q, Ling L M, Zeng D L and Feng B F 2021 *Commun. Nonlinear Sci. Numer. Simul.* **101** 105896
[40] Wang L and Yan Z Y 2021 *Phys. Lett. A* **404** 127408
[41] Pu J C, Li J and Chen Y 2021 *Chin. Phys. B* **30** 060202
[42] Mo Y F, Ling L M and Zeng D L 2022 *Phys. Lett. A* **421** 127739
[43] Wight C L and Zhao J 2021 *Commun. Comput. Phys.* **29** 930
[44] Lu L, Meng X, Mao Z and Karniadakis G E 2021 *SIAM Review* **63** 208
[45] Wang S F, Teng Y J and Perdikaris P 2021 *SIAM J. Sci. Comput.* **43** A3055
[46] Peng W Q and Chen Y 2111 arXiv: 2111.12424
[47] Kingma D P and Ba J 2015 *Anon. International Conference on Learning Representations* (San Dego)
[48] Liu D C and Nocedal J 1989 *Math. Program.* **45** 503
[49] Stein M 1987 *Technometrics* **29** 143
[50] Zhao L C, Xin G G, Yang Z Y 2014 *Phys. Rev. E* **90** 022918
[51] Ling L M, Zhao L C, Yang Z Y and Guo B L 2017 *Phys. Rev. E* **96** 022211
[52] Cao Q H and Dai C Q 2021 *Chin. Phys. Lett.* **38** 090501
[53] Zhang X M, Qin Y H, Ling L M and Zhao L C 2021 *Chin. Phys. Lett.* **38** 090201
[54] Wang B, Zhang Z and Li B 2020 *Chin. Phys. Lett.* **37** 030501
[55] Ma X R, Tu Z C and Ran S J 2021 *Chin. Phys. Lett.* **38** 110301
[56] Zhang G Q and Yan Z Y 2018 *Commun. Nonlinear Sci. Numer. Simul.* **62** 117