

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

Jun-Cai Pu(蒲俊才), Jun Li(李军), and Yong Chen(陈勇)
Citation: Chin. Phys. B, 2021, 30 (**6**): 060202. DOI: 10.1088/1674-1056/abd7e3
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

$\mathcal{H}_{\infty }$ state estimation for Markov jump neural networks with transition probabilities subject to the persistentdwell-time switching rule

Hao Shen(沈浩), Jia-Cheng Wu(吴佳成), Jian-Wei Xia(夏建伟), and Zhen Wang(王震)
Chin. Phys. B, 2021, 30 (**6**): 060203. DOI: 10.1088/1674-1056/abd7da

Constructing reduced model for complex physical systems via interpolation and neural networks

Xuefang Lai(赖学方), Xiaolong Wang(王晓龙, and Yufeng Nie(聂玉峰)
Chin. Phys. B, 2021, 30 (**3**): 030204. DOI: 10.1088/1674-1056/abd92e

Discontinuous event-trigger scheme for global stabilization of state-dependent switching neural networks with communication delay

Yingjie Fan(樊英杰), Zhen Wang(王震), Jianwei Xia(夏建伟), and Hao Shen(沈浩)
Chin. Phys. B, 2021, 30 (**3**): 030202. DOI: 10.1088/1674-1056/abc541

Design of passive filters for time-delay neural networks with quantized output

Jing Han(韩静), Zhi Zhang(章枝), Xuefeng Zhang(张学锋), and Jianping Zhou(周建平)
Chin. Phys. B, 2020, 29 (**11**): 110201. DOI: 10.1088/1674-1056/aba602

Finite-time Mittag-Leffler synchronization of fractional-order delayed memristive neural networks with parameters uncertainty and discontinuous activation functions

Chong Chen(陈冲), Zhixia Ding(丁芝侠), Sai Li(李赛), Liheng Wang(王利恒)
Chin. Phys. B, 2020, 29 (**4**): 040202. DOI: 10.1088/1674-1056/ab7803

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)[1], Jun Li(李军)[2], and Yong Chen(陈勇)[1,3,4,†]

[1] School of Mathematical Sciences, Shanghai Key Laboratory of Pure Mathematics and Mathematical Practice, and Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200241, China
[2] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
[3] College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China
[4] Department of Physics, Zhejiang Normal University, Jinhua 321004, China

(Received 17 December 2020; revised manuscript received 29 December 2020; accepted manuscript online 4 January 2021)

The nonlinear Schrödinger equation is a classical integrable equation which contains plenty of significant properties and occurs in many physical areas. However, due to the difficulty of solving this equation, in particular in high dimensions, lots of methods are proposed to effectively obtain different kinds of solutions, such as neural networks among others. Recently, a method where some underlying physical laws are embeded into a conventional neural network is proposed to uncover the equation's dynamical behaviors from spatiotemporal data directly. Compared with traditional neural networks, this method can obtain remarkably accurate solution with extraordinarily less data. Meanwhile, this method also provides a better physical explanation and generalization. In this paper, based on the above method, we present an improved deep learning method to recover the soliton solutions, breather solution, and rogue wave solutions of the nonlinear Schrödinger equation. In particular, the dynamical behaviors and error analysis about the one-order and two-order rogue waves of nonlinear integrable equations are revealed by the deep neural network with physical constraints for the first time. Moreover, the effects of different numbers of initial points sampled, collocation points sampled, network layers, neurons per hidden layer on the one-order rogue wave dynamics of this equation have been considered with the help of the control variable way under the same initial and boundary conditions. Numerical experiments show that the dynamical behaviors of soliton solutions, breather solution, and rogue wave solutions of the integrable nonlinear Schrödinger equation can be well reconstructed by utilizing this physically-constrained deep learning method.

Keywords: deep learning method, neural network, soliton solutions, breather solution, rogue wave solutions

PACS: 02.30.IK, 05.45.Yv, 07.05.Mh **DOI:** 10.1088/1674-1056/abd7e3

1. Introduction

In recent decades, more and more attention has been paid to the nonlinear problems in fluid mechanics, condensed matter physics, optical fiber communication, plasma physics, and even biology.[1–4] After establishing nonlinear partial differential equations to describe these nonlinear phenomena and then analyzing the analytical and numerical solutions of these nonlinear models, the essence of these nonlinear phenomena can be understood.[5] Therefore, the research of these nonlinear problems is essentially transformed into the study of nonlinear partial differential equations which describe these physical phenomena. Due to many basic properties of linear differential equations are not applicable to nonlinear differential equations, these nonlinear differential equations which the famous nonlinear Schrödinger equation belongs to are more difficult to solve compared with the linear differential equations. It is well known that the Schrödinger equation can be used to describe the quantum behavior of microscopic particles in quantum mechanics.[6] Furthermore, various solutions of this equation can describe the nonlinear phenomena in other physical fields, such as optical fiber, plasma, Bose–Einstein condensates, fluid mechanics, and Heisenberg ferromagnet.[7–16]

With the explosive growth of available data and computing resources, deep neural networks, *i.e.*, deep learning,[17] are applied in many areas including image recognition, video surveillance, natural language processing, medical diagnostics, bioinformatics, financial data analysis, and so on.[18–23] In scientific computing, especially, the neural network method[24–26] provides an ideal representation for the solution of differential equations[27] due to its universal approximation properties.[28] Recently, a physically constrained deep learning method called physics-informed neural network (PINN)[29] and its improvement[30] has been proposed which is particularly suitable for solving differential equations and corresponding inverse problems. It is found that the PINN architecture can obtain remarkably accurate solution with ex-

*Project supported by the National Natural Science Foundation of China (Grant No. 11675054), the Fund from Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (Grant No. ZF1213), and the Project of Science and Technology Commission of Shanghai Municipality (Grant No. 18dz2271000).

†Corresponding author. E-mail: ychen@sei.ecnu.edu.cn

© 2021 Chinese Physical Society and IOP Publishing Ltd

traordinarily less data. Meanwhile, this method also provides a better physical explanation for predicted solutions because of the underlying physical constraints which is usually described explicitly by the differential equations. In this paper, the computationally efficient physics-informed data-driven algorithm for inferring solutions to more general nonlinear partial differential equations, such as the integrable nonlinear Schrödinger equation, is studied.

As is known to all, the study of exact solutions for integrable equations which are used to describe complex physical phenomena in the real world have been paid more and more attention in plasma physics, optical fiber, fluid dynamics, and others fields.[13,31–35] The Hirota bilinear method, the symmetry reduction method, the Darboux transformation, the Bäcklund transformations, the inverse scattering method, and the function expansion method are powerful means to solve nonlinear integrable equations, and many other methods are based on them.[5,36–42] Although the computational cost of some direct numerical solutions of integrable equations is very high, with the revival of neural networks, the development of more effective deep learning algorithms to obtain data-driven solutions of nonlinear integrable equations has aroused great interest.[43–47] Li and Chen constructed abundant numerical solutions of second-order and third-order nonlinear integrable equations with different initial and boundary conditions by deep learning method based on the PINN model.[43,46,47] Previous works mainly focused on some simple solutions (*e.g.*, *N*-soliton solutions, kink solutions) of given system or integrable equation. Relatively, the research results of machine learning for constructing rogue waves are rare. In Ref. [48], the bias function including two backward shock waves and soliton generation and the generation of rogue waves are studied by using a single wave-layer feed forward neural network. As far as we know, the soliton solutions, breather solution, and rogue wave solutions[8,9] of the integrable nonlinear Schrödinger equation have not been given out by the deep learning method based on PINN. Therefore, we introduce the deep learning method with underlying physical constraints to construct the soliton solutions, breathing solution, and rogue wave solutions of integrable nonlinear Schrödinger equation in this work.

This paper is organized as follows. In Section 2, we introduce the physically constrained deep learning method and briefly present some problem setups. In Section 3, the one-soliton solution and two-soliton solution of the nonlinear Schrödinger equation are obtained by this approach, and the breather solution is derived in comparison with the two-soliton solution. Section 4 provides rogue wave solutions which contain one-order rogue wave and two-order rogue wave for the nonlinear Schrödinger equation, and the relative $\mathbb{L}_2$ errors of simulating the one-order rogue wave of nonlinear Schrödinger equation with different numbers of initial points sampled, collocation points sampled, network layers, and neurons per hidden layer are also given out in detail. Conclusion is given in the last section.

2. Method

In this paper, we consider (1+1)-dimensional nonlinear Schrödinger equation as follows:

$$\mathrm{i}q_t + \alpha q_{xx} + \beta|q|^2 q = 0, \tag{1}$$

where α, β are arbitrary parameters, $\mathrm{i} = \sqrt{-1}$ and q are complex-valued solutions with respect to x and t. Based on the theory of integrable systems and PINN, we establish a physically-constrained deep learning method to approximate the potential solution $|q(x,t)|$ of this integrable equation. Here, the underlying laws of physics are described explicitly by this equation and embedded into the architecture with the help of automatic differentiation.[49] The physical constraints regarded as a kind of regularization are introduced into the network via this mechanism which enables us to understand the network architecture and the predicted solutions better. In addition, due to the physical constraints, the network is trained just with few data.

Specifically, the complex value solution $q(x,t)$ is formulated as $q = u + \mathrm{i}v$, where $u(x,t)$ and $v(x,t)$ are real-valued functions of x,t, and real part and imaginary part of $q(x,t)$, respectively. Then, equation (1) can be converted into

$$u_t + \alpha v_{xx} + \beta(u^2+v^2)v = 0, \tag{2}$$

$$v_t - \alpha u_{xx} - \beta(u^2+v^2)u = 0. \tag{3}$$

Accordingly, we define the physics-informed neural networks $f_u(x,t)$ and $f_v(x,t)$ respectively

$$f_u := u_t + \alpha v_{xx} + \beta(u^2+v^2)v, \tag{4}$$

$$f_v := v_t - \alpha u_{xx} - \beta(u^2+v^2)u, \tag{5}$$

and the solution $q(x,t)$ is trained to satisfy the networks (4) and (5) which are embedded into the mean-squared objective function (also called loss function)

$$\mathrm{Loss} = \mathrm{Loss}_u + \mathrm{Loss}_v + \mathrm{Loss}_{f_u} + \mathrm{Loss}_{f_v}, \tag{6}$$

where

$$\mathrm{Loss}_u = \frac{1}{N_q}\sum_{i=1}^{N_q}|u(x_u^i, t_u^i) - u^i|^2, \tag{7}$$

$$\mathrm{Loss}_v = \frac{1}{N_q}\sum_{i=1}^{N_q}|v(x_v^i, t_v^i) - v^i|^2, \tag{8}$$

$$\mathrm{Loss}_{f_u} = \frac{1}{N_\mathrm{f}}\sum_{j=1}^{N_\mathrm{f}}|f_u(x_{f_u}^j, t_{f_u}^j)|^2, \tag{9}$$

$$\mathrm{Loss}_{f_v} = \frac{1}{N_\mathrm{f}}\sum_{j=1}^{N_\mathrm{f}}|f_v(x_{f_v}^j, t_{f_v}^j)|^2. \tag{10}$$

Here the initial and boundary value data about $q(x,t)$ are denoted by $\{x_u^i,t_u^i,u^i\}_{i=1}^{N_q}$ and $\{x_v^i,t_v^i,v^i\}_{i=1}^{N_q}$. Similarly, the collocation points for $f_u(x,t)$ and $f_v(x,t)$ are specified as $\{x_{f_u}^j,t_{f_u}^j\}_{j=1}^{N_f}$ and $\{x_{f_v}^j,t_{f_v}^j\}_{j=1}^{N_f}$ which are sampled using the classical latin hypercube sampling (LHS) technique.[50] The Loss function (6) corresponds to the initial-boundary data and the networks (4) and (5) penalize Eqs. (2) and (3) not being satisfied on the collocation points. Specifically, the first and second terms on the right hand side of Eq. (6) attempt to fit the solution data, and the third and fourth terms learn to satisfy the networks f_u and f_v. The convergence of the loss function has been analyzed in previous works.[51]

In this paper, we optimize all loss functions simply using the L-BFGS algorithm which is a full-batch gradient-based optimization algorithm based on a quasi-Newton method.[52] In addition, we use relatively simple multilayer perceptrons (MLPs) with the Xavier initialization and the hyperbolic tangent (tanh) activation function.[43] All codes in this article are based on Python 3.7 and Tensorflow 1.15, and all numerical examples reported here are run on a DELL Precision 7920 Tower computer with 2.10 GHz 8-core Xeon Silver 4110 processor and 64-GB memory.

3. Soliton solutions and breather solution of the nonlinear Schrödinger equation

The (1+1)-dimensional focusing nonlinear Schrödinger equation is a classical integrable field equation for describing quantum mechanical systems, nonlinear wave propagation in optical fibers or waveguides, Bose–Einstein condensates, and plasma waves. In optics, the nonlinear term is generated by the intensity dependent index of a given material. Similarly, the nonlinear term for Bose–Einstein condensates is the result of the mean-field interactions about the interacting N-body system. We consider the focusing nonlinear Schrödinger equation along with Dirichlet boundary conditions given by

$$\begin{cases} \mathrm{i}q_t + q_{xx} + 2|q|^2 q = 0, x \in [x_0,x_1], t \in [t_0,t_1], \\ q(x,t_0) = q_0(x), \\ q(x_0,t) = q(x_1,t), \end{cases} \tag{11}$$

where $q_0(x)$ is an arbitrary complex-valued function of space variable x, x_0, and x_1 represent the lower and upper boundaries of x respectively, t_0 and t_1 represent the initial and terminal time instants of t respectively. In addition, this equation corresponds to Eq. (1) with $\alpha = 1$ and $\beta = 2$. Equation (11) is often used to describe the evolution of weakly nonlinear dispersive wave modulation. In view of the characteristic of its solution, it is called "self focusing" nonlinear Schrödinger equation. For water wave modulation, there is usually coupling between modulation and wave induced current, so in some cases, water wave modulation can also be described by the nonlinear Schrödinger equation.[2] The N-soliton solutions and breather solution of the above equation have been obtained by many different methods.[36,38,53] Here, we simulate the soliton solutions and breather solution using the physically constrained deep learning method, and compare them with the known exact solutions, so as to prove the effectiveness of solving the numerical solutions $q(x,t)$ by neural networks. Specifically, the N-soliton solution of nonlinear Schrödinger equation have been derived by the Riemann–Hilbert method,[53] and the N-soliton solution is formed as

$$q(x,t) = -2\mathrm{i}\frac{\det R}{\det M}, \tag{12}$$

where M is a matrix of $N \times N$,

$$M = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1N} \\ M_{21} & M_{22} & \cdots & M_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ M_{N1} & M_{N2} & \cdots & M_{NN} \end{bmatrix}, \tag{13}$$

with

$$M_{jk} = \frac{\mathrm{e}^{-(\theta_k+\theta_j^*)} + c_j^* c_k \mathrm{e}^{\theta_k+\theta_j^*}}{\zeta_j^* - \zeta_k}, \quad j,k = 1,\ldots,N, \tag{14}$$

and R is a matrix of $(N+1)\times(N+1)$,

$$R = \begin{bmatrix} 0 & c_1\mathrm{e}^{\theta_1} & \cdots & c_N\mathrm{e}^{\theta_N} \\ \mathrm{e}^{-\theta_1^*} & M_{11} & \cdots & M_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{e}^{-\theta_N^*} & M_{N1} & \cdots & M_{NN} \end{bmatrix}, \tag{15}$$

with $\theta_k = -\mathrm{i}\zeta_k x - 2\mathrm{i}\zeta_k^2 t$ $(k = 1,\ldots,N)$, ζ_k and c_i $(i = 1,\ldots,N)$ are complex value constants. After taking the positive integer N, one can obtain the corresponding N-soliton solutions and breather solution of the nonlinear Schrödinger Eq. (11).

3.1. One-soliton solution

In this subsection, we numerically construct one-soliton solution of Eq. (11) based on the neural network structure with 9 hidden layers and 40 neurons per hidden layer. When $N = 1$, we have

$$R = \begin{pmatrix} 0 & c_1\mathrm{e}^{\theta_1} \\ \mathrm{e}^{-\theta_1^*} & M_{11} \end{pmatrix}, \quad M_{11} = \frac{\mathrm{e}^{-(\theta_1+\theta_1^*)} + c_1^* c_1 \mathrm{e}^{\theta_1+\theta_1^*}}{\zeta_1^* - \zeta_1},$$

$\det R = -c_1\mathrm{e}^{\theta_1-\theta_1^*}, \det M = M_{11}$, so the solution (12) is

$$\begin{aligned} q(x,t) &= -2\mathrm{i}\frac{\det R}{\det M} \\ &= 2\mathrm{i}(\zeta_1^* - \zeta_1)\frac{c_1\mathrm{e}^{\theta_1-\theta_1^*}}{\mathrm{e}^{-(\theta_1+\theta_1^*)} + |c_1|^2\mathrm{e}^{\theta_1+\theta_1^*}}. \end{aligned} \tag{16}$$

Letting

$$\zeta_1 = \xi + \mathrm{i}\eta, \quad c_1 = \mathrm{e}^{-2\eta x_0 + \mathrm{i}\sigma_0},$$

where ξ, η are the real and imaginary parts of ζ_1 respectively, and x_0, σ_0 are real parameters. Then the above one-soliton solution (16) can be reduced to

$$q(x,t) = 2\eta\,\mathrm{sech}[2\eta(x + 4\xi t - x_0)]$$

$$\times \mathrm{e}^{[-2\mathrm{i}\xi x-4\mathrm{i}(\xi^2-\eta^2)t+\mathrm{i}\sigma_0]}. \tag{17}$$

One can obtain the exact one-soliton solution of the nonlinear Schrödinger Eq. (11) after taking $\eta = 1, \xi = 1, x_0 = 0, \sigma_0 = 1$ into Eq. (17) as follows:

$$q(x,t) = 2\mathrm{sech}(8t+2x)\,\mathrm{e}^{(-2\mathrm{i}x+\mathrm{i})}. \tag{18}$$

Then we take $[x_0, x_1]$ and $[t_0, t_1]$ in Eq. (11) as $[-5.0, 5.0]$ and $[-0.5, 0.5]$, respectively. The corresponding initial condition is obtained by substituting a specific initial value into Eq. (18)

$$q_0(x) = 2\mathrm{sech}(2x-4)\,\mathrm{e}^{(-2\mathrm{i}x+\mathrm{i})}. \tag{19}$$

We employ the traditional finite difference shcemes on even grids in MATLAB to simulate Eq. (11) with the initial data (19) to acquire the training data. Specifically, dividing space $[-5.0, 5.0]$ into 513 points and time $[-0.5, 0.5]$ into 401 points, one-soliton solution q is discretized into 401 snapshots accordingly. We sub-sample a smaller training dataset that contain initial-boundary subsets by randomly extracting $N_q = 100$ from original initial-boundary data and $N_f = 10000$ collocation points which are generated by LHS.[50] After giving a dataset of initial and boundary points, the latent one-soliton solution $q(x,t)$ is successfully learned by tuning all learnable parameters of the neural network and regulating the loss function (6). The model achieves a relative $\mathbb{L}_2$ error of 2.566069×10^{-2} in about 726 seconds, and the number of iterations is 8324.

In Fig. 1, the density diagrams, the figures at different instants of the latent one-soliton solution $q(x,t)$, the error diagram about the difference between exact one-soliton solution and hidden one-soliton solution, and the loss curve figure are plotted respectively. The panel (a) of Fig. 1 clearly compares the exact solution with the predicted spatiotemporal solution. Obviously, combining with the panel (b), we can see that the error between the numerical solution and the exact solution is very small. We particularly present a comparison between the exact solution and the predicted solution at different time instants $t = -0.25, 0, 0.25$ in the bottom panel of panel (a). It is obvious that as time t increases, the one-soliton solution propagates along the negative direction of the x axis. The three-dimensional motion of the predicted solution and the loss curve at different iterations are given out in detail in panels (c) and (d) of Fig. 1. The results show that the loss curve is very smooth which proves the effectiveness and stability of the integrable deep learning method.

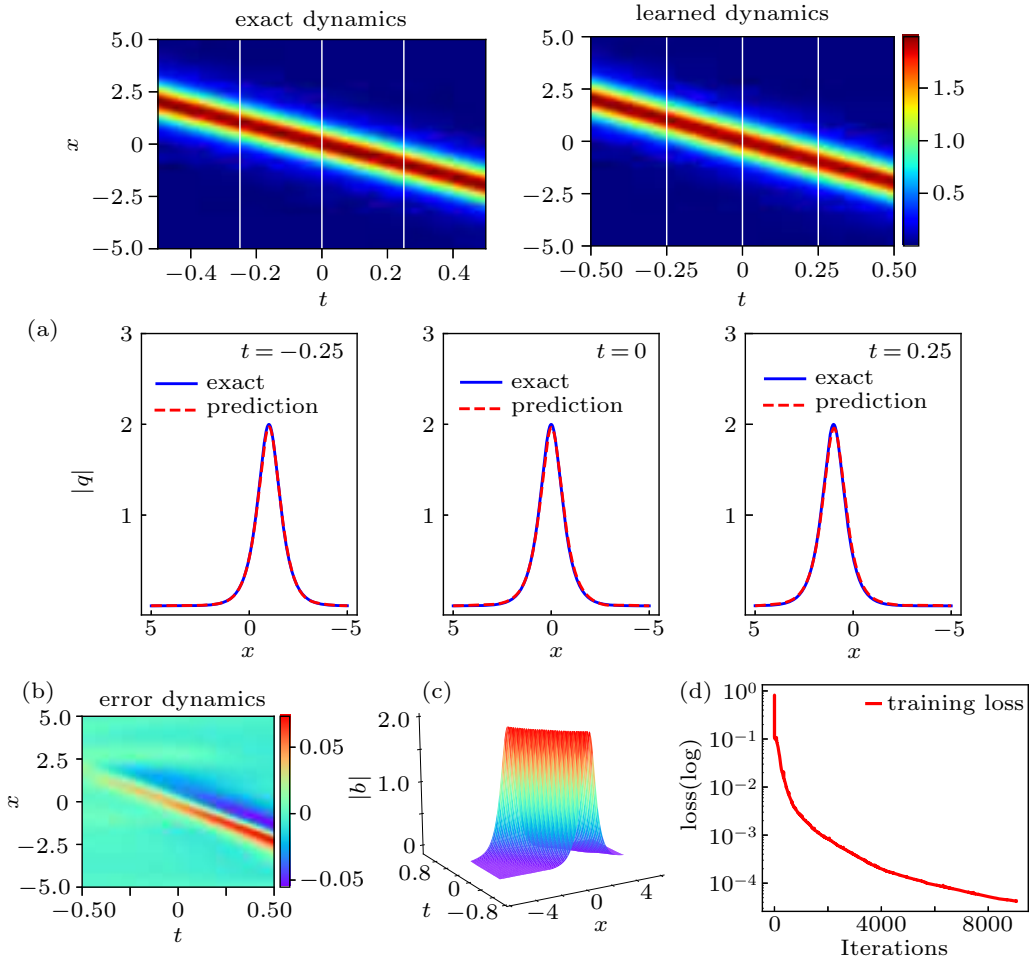


Fig. 1. The one-soliton solution $q(x,t)$: (a) the density diagrams and figures at three different instants, respectively; (b) the error density diagram; (c) the three-dimensional motion; (d) the loss curve figure.

3.2. Two-soliton solution and breather solution

Now, we numerically construct the two-soliton solution and breather solution of Eq. (11) based on the neural network architecture with 9 hidden layers and 80 neurons per hidden layer. When $N=2$, the solution (12) can also be written out explicitly. We have

$$F=\begin{pmatrix} 0 & c_1\mathrm{e}^{\theta_1} & c_2\mathrm{e}^{\theta_2} \\ \mathrm{e}^{-\theta_1^*} & M_{11} & M_{12} \\ \mathrm{e}^{-\theta_2^*} & M_{21} & M_{22} \end{pmatrix}, \quad M=\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix},$$

where

$$M_{11}=\frac{\mathrm{e}^{-(\theta_1+\theta_1^*)}+c_1^*c_1\mathrm{e}^{\theta_1+\theta_1^*}}{\zeta_1^*-\zeta_1},$$

$$M_{12}=\frac{\mathrm{e}^{-(\theta_2+\theta_1^*)}+c_1^*c_2\mathrm{e}^{\theta_2+\theta_1^*}}{\zeta_1^*-\zeta_2},$$

$$M_{21}=\frac{\mathrm{e}^{-(\theta_1+\theta_2^*)}+c_2^*c_1\mathrm{e}^{\theta_1+\theta_2^*}}{\zeta_2^*-\zeta_1},$$

$$M_{22}=\frac{\mathrm{e}^{-(\theta_2+\theta_2^*)}+c_2^*c_2\mathrm{e}^{\theta_2+\theta_2^*}}{\zeta_2^*-\zeta_2},$$

with $\theta_1=-\mathrm{i}\zeta_1 x-2\mathrm{i}\zeta_1^2 t, \theta_1^*=\mathrm{i}\zeta_1^* x+2\mathrm{i}\zeta_1^{*2}t, \theta_2=-\mathrm{i}\zeta_2 x-2\mathrm{i}\zeta_2^2 t, \theta_2^*=-\mathrm{i}\zeta_2^* x-2\mathrm{i}\zeta_2^{*2}t$, ζ_j $(j=1,2)$ are complex value constants, so one can derive the general form of two-soliton solution as follows:

$$q(x,t)=2\mathrm{i}\frac{c_2M_{11}\mathrm{e}^{\theta_2-\theta_2^*}-c_2M_{12}\mathrm{e}^{\theta_2-\theta_1^*}-c_1M_{21}\mathrm{e}^{\theta_1-\theta_2^*}+c_1M_{22}\mathrm{e}^{\theta_1-\theta_1^*}}{M_{11}M_{22}-M_{12}M_{21}}. \tag{20}$$

According to the relationship between the two-soliton solution and the breather solution, we can know that when $\mathrm{Re}(\zeta_1)\neq\mathrm{Re}(\zeta_2)$, the solution $q(x,t)$ is a two-soliton solution, and when $\mathrm{Re}(\zeta_1)=\mathrm{Re}(\zeta_2)$, the solution $q(x,t)$ degenerates into a bound state which is also called the breather solution. Given appropriate parameters

$$\zeta_1=0.1+0.7\mathrm{i}, \quad \zeta_2=-0.1+0.4\mathrm{i}, \quad c_1=c_2=1, \tag{21}$$

we can obtain the exact two-soliton solution from the formulae (20)

$$q(x,t)=\frac{-2\mathrm{i}A}{B}, \tag{22}$$

where

$$\begin{aligned} A =\ & (0.224+0.158\mathrm{i})\mathrm{e}^{\mathrm{i}(1.92t+0.32\mathrm{i}t-0.2x-0.8\mathrm{i}x)} \\ & -(0.224+0.232\mathrm{i})\mathrm{e}^{-\mathrm{i}(-0.6t+0.56\mathrm{i}t-0.2x+1.4\mathrm{i}x)} \\ & +(0.224-0.232\mathrm{i})\mathrm{e}^{\mathrm{i}(0.6t+0.56\mathrm{i}t+0.2x+1.4\mathrm{i}x)} \\ & +(0.518\mathrm{i}-0.224)\mathrm{e}^{-\mathrm{i}(-1.92t+0.32\mathrm{i}t+0.2x-0.8\mathrm{i}x)}, \\ B =\ & 1.25\mathrm{e}^{0.88t+0.6x}+0.13\mathrm{e}^{-0.24t-2.2x} \\ & +1.25\mathrm{e}^{-0.88t-0.6x}-1.12\mathrm{e}^{\mathrm{i}(1.32t-0.4x)} \\ & -1.12\mathrm{e}^{-\mathrm{i}(1.32t-0.4x)}+0.13\mathrm{e}^{0.24t+2.2x}. \end{aligned}$$

On the other hand, given other appropriate parameters

$$\zeta_1=0.7\mathrm{i}, \quad \zeta_2=0.4\mathrm{i}, \quad c_1=c_2=1, \tag{23}$$

one can obtain the exact breather solution

$$q(x,t)=\frac{-2\mathrm{i}C}{D}, \tag{24}$$

where

$$\begin{aligned} C =\ & -0.246\mathrm{i}\,\mathrm{e}^{-\mathrm{i}(-0.64t+1.4\mathrm{i}x)}+0.462\mathrm{i}\,\mathrm{e}^{-\mathrm{i}(-1.96t-0.8\mathrm{i}x)} \\ & -0.264\mathrm{i}\,\mathrm{e}^{\mathrm{i}(0.64t+1.4\mathrm{i}x)}+0.462\mathrm{i}\,\mathrm{e}^{-\mathrm{i}(-1.96t+0.8\mathrm{i}x)}, \\ D =\ & 1.21\mathrm{e}^{-0.6x}+1.21\mathrm{e}^{0.6x}-1.12\mathrm{e}^{1.32\mathrm{i}t} \\ & +0.09\mathrm{e}^{2.2x}-1.12\mathrm{e}^{-1.32\mathrm{i}t}+0.09\mathrm{e}^{-2.2x}. \end{aligned}$$

Now we take $[x_0,x_1]$ and $[t_0,t_1]$ in Eq. (11) as $[-5.0,5.0]$ and $[-3.0,3.0]$, respectively. For instance, we consider the initial condition of the two-soliton solution based on Eq. (22)

$$q_0(x)=\frac{-2\mathrm{i}A'}{B'}, \tag{25}$$

where

$$\begin{aligned} A' =\ & (0.224+0.158\mathrm{i})\mathrm{e}^{\mathrm{i}(-5.76-0.96\mathrm{i}-0.2x-0.8\mathrm{i}x)} \\ & -(0.224+0.232\mathrm{i})\mathrm{e}^{-\mathrm{i}(1.8-1.68\mathrm{i}-0.2x+1.4\mathrm{i}x)} \\ & +(0.224-0.232\mathrm{i})\mathrm{e}^{\mathrm{i}(-1.8-1.68\mathrm{i}+0.2x+1.4\mathrm{i}x)} \\ & +(0.518\mathrm{i}-0.224)\mathrm{e}^{-\mathrm{i}(5.76-0.96\mathrm{i}+0.2x-0.8\mathrm{i}x)}, \\ B' =\ & 1.25\mathrm{e}^{-2.64+0.6x}+0.13\mathrm{e}^{0.72-2.2x} \\ & +1.25\mathrm{e}^{2.64-0.6x}-1.12\mathrm{e}^{\mathrm{i}(-3.96-0.4x)} \\ & -1.12\mathrm{e}^{-\mathrm{i}(-3.96-0.4x)}+0.13\mathrm{e}^{-0.72+2.2x}. \end{aligned}$$

Similarly, the initial condition of the breather solution is given

$$q_0(x)=\frac{-2\mathrm{i}C'}{D'}, \tag{26}$$

where

$$\begin{aligned} C' =\ & -0.246\mathrm{i}\,\mathrm{e}^{-\mathrm{i}(1.92+1.4\mathrm{i}x)}+0.462\mathrm{i}\,\mathrm{e}^{-\mathrm{i}(5.88-0.8\mathrm{i}x)} \\ & -0.264\mathrm{i}\,\mathrm{e}^{\mathrm{i}(-1.92+1.4\mathrm{i}x)}+0.462\mathrm{i}\,\mathrm{e}^{-\mathrm{i}(5.88+0.8\mathrm{i}x)}, \\ D' =\ & 1.21\mathrm{e}^{-0.6x}+1.21\mathrm{e}^{0.6x}-1.12\mathrm{e}^{-3.96\mathrm{i}} \\ & +0.09\mathrm{e}^{2.2x}-1.12\mathrm{e}^{3.96\mathrm{i}}+0.09\mathrm{e}^{-2.2x}. \end{aligned}$$

With the same data generation and sampling method in Subsection 3.1, we numerically simulate the two-soliton solution and the breather solution of the nonlinear Schrödinger

equation (11) using the physically-constrained deep learning method mentioned above. After training the two-soliton solution, the neural network achieves a relative $\mathbb{L}_2$ error of 5.500792×10^{-2} in about 2565 seconds, and the number of iterations is 17789. However, the network model for learning breather solution achieves a relative $\mathbb{L}_2$ error of 9.689267×10^{-3} in about 1934 seconds, and the number of iterations is 13488. Apparently, since the breather solution is a special form of the two-soliton solution and accordingly the solution structure is simpler, the training of the breather solution takes remarkably less time, the relative error is obviously smaller, and moreover the result is better than that of the two-soliton solution from Figs. 2 and 3.

Figures 2 and 3 show the density diagrams, the profiles at different instants and error density diagrams of the two-soliton solution and the breather solution, respectively. From the bottom panel of panels (a) in Fig. 2, we can clearly see that the intersection of two solitary waves with different wave widths and amplitudes produces a peak of a higher amplitude different from the former two solitary waves, which satisfies the law of conservation of energy. We reveal the profiles of the three moments at $t=-1.50$, 0, 1.50, respectively, and find that the amplitude is the largest when $t=0$. From soliton theory, we know that the two solitary waves have elastic collision. Similarly, one can look at the breather solution shown in panel (a) of Fig. 3 it is a special bound state two-soliton solution formed by two solitary waves with the same wave velocity, wave width and amplitude, and has a periodic motion with respect to time t. The panel (b) of Figs. 2 and 3 shows the error dynamics of the difference between the exact solution and the predicted solution for the two-soliton solution and the breather solution, respectively. In Fig. 4, the corresponding three-dimensional motion of the two-soliton solution and the breather solution are shown, respectively. It is evident that the breather solution is more symmetric than the general two-soliton solution.

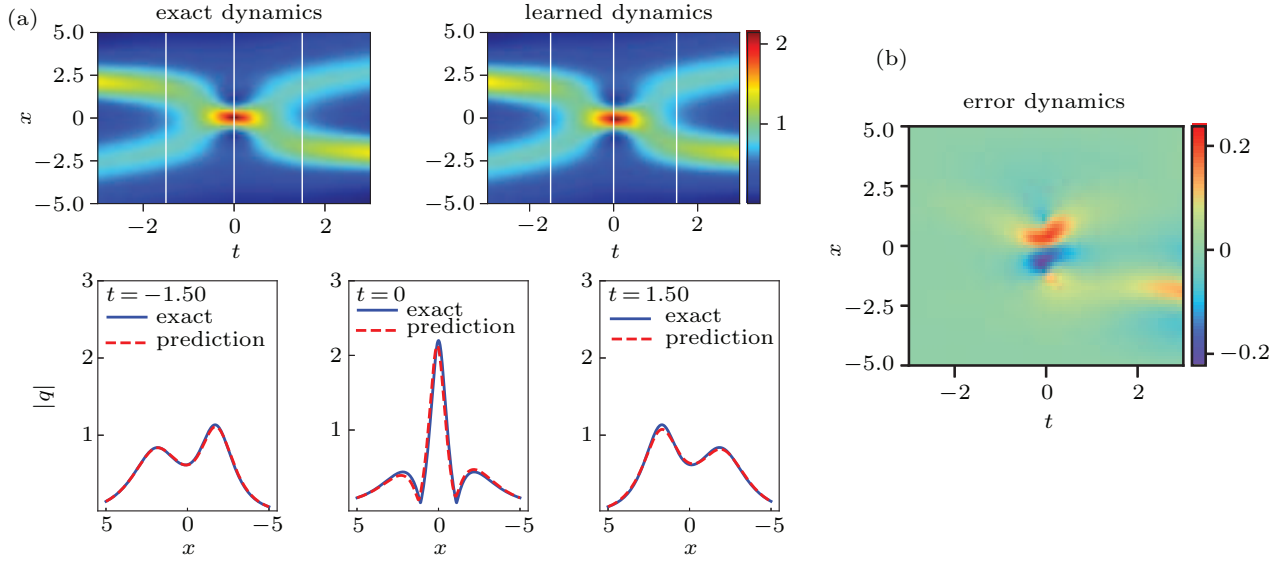


Fig. 2. The two-soliton solution $q(x,t)$: (a) the density diagrams and the profiles at different moments; (b) the error density diagram.

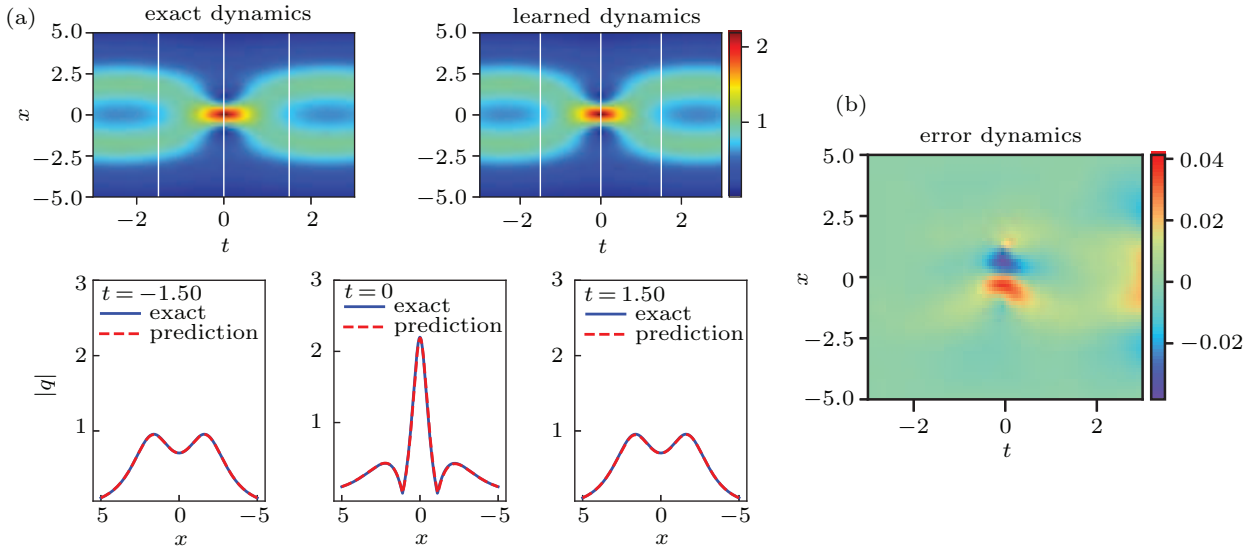


Fig. 3. The breather solution $q(x,t)$: (a) the density diagrams and the profiles at different moments; (b) the error density diagram.

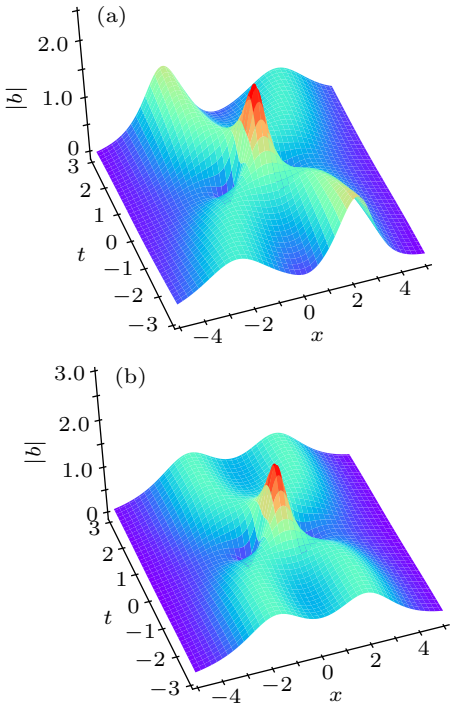


Fig. 4. The three-dimensional motion of $q(x,t)$: (a) the two-soliton solution; (b) the breather solution.

For the numerical simulation of the three-soliton solution, we only need to take $N=3$ in Eqs. (12)–(15) to get the exact solution of the three-soliton solution, and then discretize the initial and boundary value data of the exact solution as our original dataset and train our network to simulate the corresponding three-soliton solution numerically. Similarly, N-soliton solutions can be learned by the same approach. Of course, the higher the order of soliton solution, the more complex the form of the solution, then the longer the resulting network training time takes.

4. Rogue wave solutions of the nonlinear Schrödinger equation

Recently, the research of rogue wave has been one of the hot topics in many areas such as optics, ocean dynamics, plasma, Bose–Einstein condensate, and even finance.[8,9,54–56] In addition to the peak amplitude more than twice of the background wave, rogue waves also have the characteristics of instability and unpredictability. Therefore, the study and application of rogue waves play a momentous role in real life, especially in avoiding the damage to ships caused by ocean rogue waves. As a one-dimensional integrable scalar equation, the nonlinear Schrödinger equation plays a key role in describing rogue waves. In 1983, Peregrine[2] first gave a rational rogue waves to the nonlinear Schrödinger equation, whose generation principle is identified as the evolution of the breather waves when the period tends to infinity. At present, the researches on rogue wave of this equation through data-driven methods, such as machine learning, are relatively less. Marcucci *et al.*[48] have studied the computational machine in which nonlinear waves replace the internal layers of neural networks, discussed learning conditions, and demonstrated functional interpolation, datasets, and Boolean operations. When considering the solitons, rogue waves, and shock waves of the nonlinear Schrödinger equation, highly nonlinear and even discontinuous regions play a leading role in the network training and solution calculation. In this section, we construct the rogue wave solutions of the nonlinear Schrödinger equation by the neural network with underlying physical constraints. Here, we consider the another form of focusing nonlinear Schrödinger equation along with Dirichlet boundary conditions given by

$$\begin{cases} \mathrm{i}q_t+\frac{1}{2}q_{xx}+|q|^2q=0, x\in[x_0,x_1], t\in[t_0,t_1], \\ q(x,t_0)=q_0(x), \\ q(x_0,t)=q(x_1,t), \end{cases} \tag{27}$$

where $q_0(x)$ is an arbitrary complex-valued function of space variable x, here x_0, x_1 represent the lower and upper boundaries of x respectively, and t_0, t_1 represent the initial and terminal time instants of t respectively. In addition, this equation corresponds to Eq. (1) with $\alpha=1/2$ and $\beta=1$. The rogue wave solutions of Eq. (27) can be obtained by lots of different tools.[11] Therefore, we can get respectively the one-order rogue wave and the two-order rogue wave of Eq. (27) as follows:

$$q(x,t)=\left[1-\frac{4(1+2\mathrm{i}t)}{4t^2+4x^2+1}\right]\mathrm{e}^{\mathrm{i}t}, \tag{28}$$

$$q(x,t)=\left(1+\frac{G+\mathrm{i}tH}{D}\right)\mathrm{e}^{\mathrm{i}t}, \tag{29}$$

where

$$\begin{aligned} G &= \frac{3}{8}-3x^2-2x^4-9t^2-10t^4-12t^2x^2, \\ H &= \frac{15}{4}+6x^2-4x^4-2t^2-4t^4-8t^2x^2, \\ D &= \frac{3}{32}+\frac{9}{8}x^2+\frac{1}{2}x^4+\frac{2}{3}x^6+\frac{33}{8}t^2+\frac{9}{2}t^4 \\ &\quad +\frac{2}{3}t^6-3t^2x^2+2t^2x^4+2t^4x^2. \end{aligned}$$

In the following two parts, we will construct the training dataset to reconstruct our predicted solutions based on the above two rogue wave solutions by constructing a neural network with 9 hidden layers and 40 neurons per hidden layer.

4.1. One-order rogue wave

In this subsection, we will numerically uncover the one-order rogue wave of the nonlinear Schrödinger equation using the neural network method above. Now, we take $[x_0,x_1]$ and $[t_0,t_1]$ in Eq. (27) as $[-2.0,2.0]$ and $[-1.5,1.5]$, respectively. The corresponding initial condition is obtained from Eq. (28), we have

$$q_0(x)=\left(1+\frac{-4+12\mathrm{i}}{4x^2+10}\right)\mathrm{e}^{-1.5\mathrm{i}}. \tag{30}$$

Next, we obtain the initial and boundary value dataset by the same data discretization method in Subsection 3.1, and then we can simulate precisely the one-order rogue wave solution by feeding the data into the network. By randomly subsampling $N_q = 100$ from the original dataset and selecting $N_f = 10000$ configuration points which are generated by LHS, a training dataset composed of initial-boundary data and collocation points is generated. After training, the neural network model achieves a relative $\mathbb{L}_2$ error of 7.845201×10^{-3} in about 871 seconds, and the number of iterations is 9584.

Our experiment results are summarized in Fig. 5, and we simulate the solution $q(x,t)$ and then obtain the density diagrams, profiles at different instants, error dynamics diagrams, three dimensional motion and loss curve figure of the one-order rogue wave. Specifically, the magnitude of the predicted spatio-temporal solution $|q(x,t)|$ is shown in the top panel of panel (a) of Fig. 5. It can be simply seen that the amplitude of the rogue wave solution changes greatly in a very short time from the bottom panel of Fig. 5(a). Meanwhile, we present a comparison between the exact and the predicted solution at different time instants $t = -0.75, 0, 0.75$. Figure 5(b) reveals the relative $\mathbb{L}_2$ error becomes larger as the time increases. From Fig. 5(d), we can observe that when the number of iterations is more than 2000, there are some obvious fluctuations which we could call "burr" in the training, it does not exist during the training process about the one-soliton solution of the nonlinear Schrödinger equation. With only a handful of initial-boundary data, one can accurately capture the intricate nonlinear dynamical behavior of the integrable Schrödinger equation by this method.

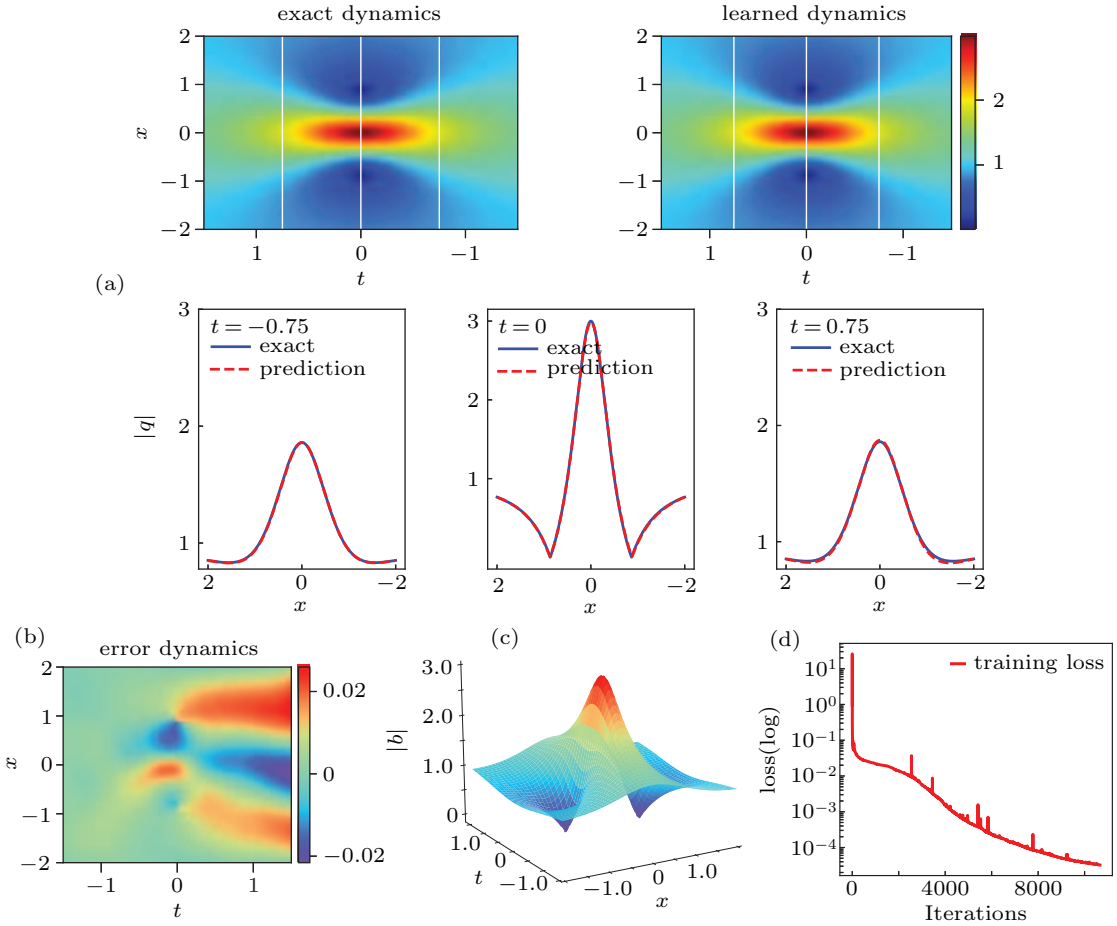


Fig. 5. The one-order rogue wave solution $q(x,t)$: (a) the density diagram and profiles at three different instants; (b) the error density diagram; (c) the three-dimensional motion; (d) the loss curve.

In addition, based on the same initial and boundary values of the one-order rogue waves in the case of $N_q = 100$ and $N_f = 10000$, we employ the control variable method often used in applied sciences to study the effects of different numbers of network layers and neurons per hidden layer on the one-order rogue wave dynamics of nonlinear Schrödinger equation. The relative $\mathbb{L}_2$ errors of different network layers and different neurons per hidden layer are given in Table 1. From the data in Table 1, we can see that when the number of network layers is fixed, the more the number of single-layer neurons, the smaller the relative error becomes. Due to the influence of randomness caused by some factors, there are some cases that do not conform with the above conclusion. However, when the number of single-layer neurons is fixed, the influence of the number of network layers on the relative error is not obvious. To sum up, we can draw the conclusion that the network layers and the single-layer neurons jointly determine the relative $\mathbb{L}_2$ error to some extent. In the case of the same training dataset. Table 2 shows the relative

$\mathbb{L}_2$ error with 9 network layers and 40 neurons per hidden layer when taking different numbers of subsampling points N_q in the initial-boundary data and collocation points N_f. From Table 2, we can see that the influence of N_q on the relative $\mathbb{L}_2$ error of the network is not obvious, which also indicates the network model with physical constraints can uncover accurate predicted solutions with smaller initial-boundary data and relatively many sampled collocation points.

Table 1. One-order rogue wave of the nonlinear Schrödinger equation: Relative final prediction error estimations in the $\mathbb{L}_2$ norm for different numbers of network layers and neurons per hidden layer.

Neurons \ Layers	20	30	40	50	60
5	2.765905×10^{-3}	2.903368×10^{-4}	4.961406×10^{-4}	5.232502×10^{-4}	9.323978×10^{-4}
7	2.699082×10^{-3}	1.328768×10^{-3}	4.030658×10^{-4}	3.633812×10^{-4}	1.448091×10^{-3}
9	2.732954×10^{-3}	2.618465×10^{-3}	7.845201×10^{-3}	6.880915×10^{-4}	6.797486×10^{-4}
11	4.641999×10^{-3}	1.779715×10^{-3}	1.440061×10^{-3}	9.148106×10^{-4}	1.581767×10^{-3}

Table 2. One-order rogue wave of the nonlinear Schrödinger equation: Relative final prediction error measurements in the $\mathbb{L}_2$ norm for different numbers of N_q and N_f.

N_q \ N_f	6000	8000	10000	12000	14000
80	1.473695×10^{-2}	9.200569×10^{-3}	7.967294×10^{-3}	6.034213×10^{-3}	3.575678×10^{-3}
100	1.176106×10^{-2}	3.082057×10^{-3}	7.845201×10^{-3}	5.495886×10^{-3}	1.332274×10^{-2}
120	1.525780×10^{-2}	1.265775×10^{-2}	4.175621×10^{-2}	2.402183×10^{-3}	6.568740×10^{-3}

4.2. Two-order rogue wave

In the next example, we consider the two-order rogue wave of the nonlinear Schrödinger equation, and properly take $[x_0,x_1]$ and $[t_0,t_1]$ in Eq. (27) as $[-2.0,2.0]$ and $[-0.5,0.5]$. Here we consider the corresponding initial condition from Eq. (29) as follows:

$$q_0(x)=\left[1+\frac{-70.5-30x^2-2x^4-1.5\mathrm{i}(-21-12x^2-4x^4)}{39.75+4.5x^2+5x^4+(2/3)x^6}\right]\mathrm{e}^{-1.5\mathrm{i}}. \tag{31}$$

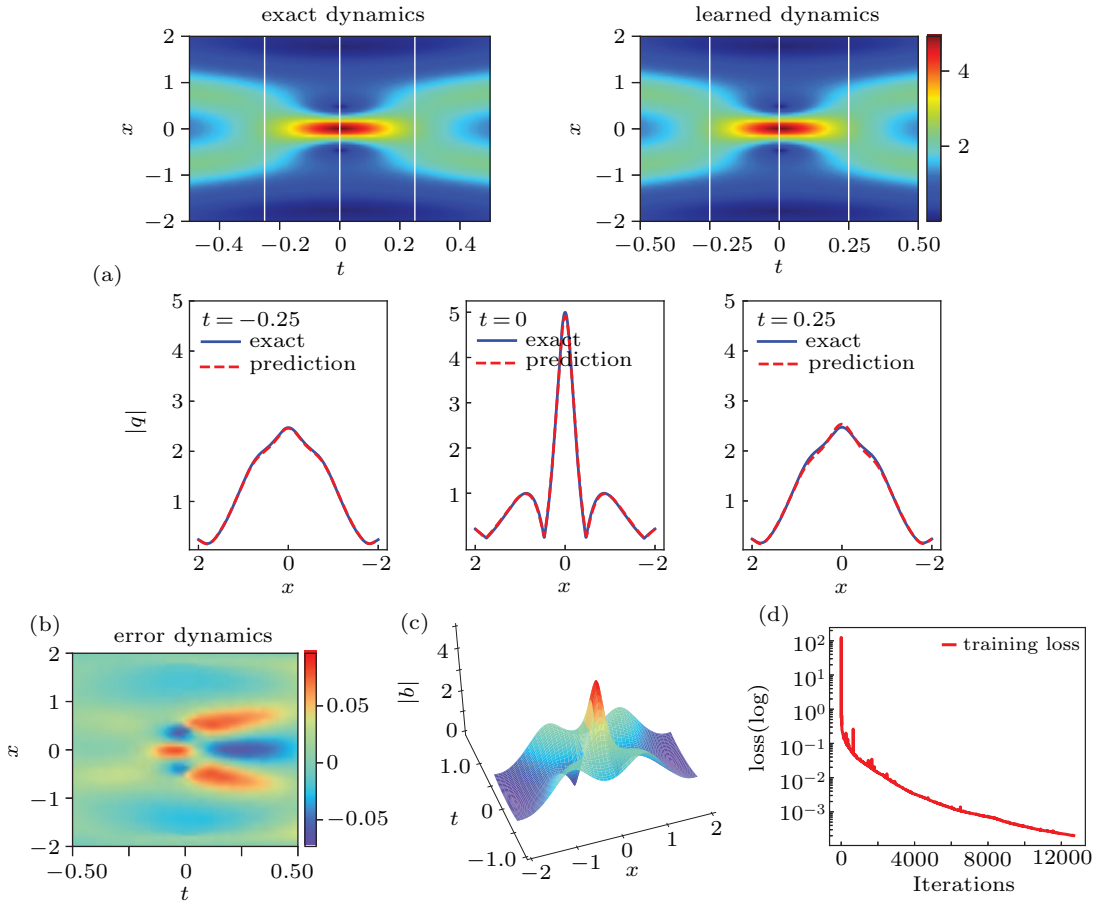


Fig. 6. The two-order rogue wave solution $q(x,t)$: (a) the density diagrams and the snapshots at three different instants; (b) the error density diagram; (c) the three-dimensional motion; (d) the loss curve figure.

We use the same data discretization method in Subsection 3.1 to collect the initial and boundary data. In the network architecture, initial and boundary training dataset of $N_q = 100$ are randomly subsampled from the original initial-boundary data. In addition, configuration points of $N_f = 10000$ are sampled by LHS. Finally, the hidden two-order rogue wave solution of nonlinear Schrödinger equation is approximated fairly accurately by constraining the loss function with underlying physical laws. The neural network model achieves a relative $\mathbb{L}_2$ error of 1.665401×10^{-2} in about 1090 seconds, and the number of iterations is 11450.

The detailed illustration is shown in Fig. 6. The top panel of Fig. 6(a) gives the density map of hidden solution $q(x,t)$, and when combing Fig. 6(b) with the bottom panel in Fig. 6(a), we can see that the relative error is relatively large at $t = 0.25$. From Fig. 6(d), in contrast with the one-order rogue wave solution, the fluctuation (burr phenomenon) of the loss function is obvious when the number of iterations is less than 3000.

5. Summary and discussion

In this paper, we introduced a physically-constrained deep learning method based on PINN to solve the classical integrable nonlinear Schrödinger equation. Compared with traditional numerical methods, it has no mesh size limits. Moreover, due to the physical constraints, the network is trained with just few data and has a better physical interpretability. This method showcases a series of results of various problems in the interdisciplinary field of applied mathematics and computational science which opens a new path for using deep learning to simulate unknown solutions and correspondingly discover the parametric equations in scientific computing.

Specifically, we apply the data-driven algorithm to deduce the soliton solutions, breather solution, and rogue wave solutions to the nonlinear Schrödinger equation. We outline how different types of solutions (such as general soliton solutions, breather solution, and rogue wave solutions) are generated due to different choices of initial and boundary value data. Remarkably, these results show that the deep learning method with physical constraints can exactly recover different dynamical behaviors of this integrable equation. Furthermore, the sizes of space-time variable x and interval t are selected by the dynamical behaviors of these solutions. For the breathers, in particular, the wider the interval of time variable t, the better we can see the dynamical behavior in this case. However, with a wider range of time interval t, the training effect is not very good. So more complex boundary conditions, such as Neumann boundary conditions, Robin boundary conditions or other mixed boundary conditions, may be considered. Similarly, for the integrable complex modified Korteweg–de Vries (mKdV) equation, the Dirichlet boundary conditions cannot recover the ideal rogue wave solutions.

The influence of noise on our neural network model is not introduced in this paper. This kind of physical factors in real life should be considered to show the network's robustness. Compared with static LHS sampling with even mesh sizes, more adaptive sampling techniques should be considered in some special problems, for example, discontinuous fluid flows such as shock wave. In addition, more general nonlinear Schrödinger equation, such as the derivative Schrödinger equation, is not investigated in this work. These new problems and improvements will be considered in the future research.

References

[1] Draper L 1966 *Weather* **21** 2
[2] Peregrine D H 1983 *J. Aust. Math. Soc. Ser. B* **25** 16
[3] Zabusky N J and Kruskal M D 1965 *Phys. Rev. Lett.* **15** 240
[4] Parkins A S and Walls D F 1998 *Phys. Rep.* **303** 1
[5] Ablowitz M J and Clarkson P A 1992 *Solitons, Nonlinear Evolution Equations and Inverse Scattering* (Cambridge: Cambridge University Press)
[6] Schrödinger E 1926 *Phys. Rev.* **28** 1049
[7] Guo B L, Ling L M and Liu Q P 2012 *Stud. Appl. Math.* **130** 317
[8] Solli D R, Ropers C, Koonath P and Jalali B 2007 *Nature* **450** 1054
[9] Chabchoub A, Hoffmann N P and Akhmediev N 2011 *Phys. Rev. Lett.* **106** 204502
[10] Qiao Z J 1994 *J. Math. Phys.* **35** 2971
[11] Akhmediev N, Ankiewicz A and Soto-Crespo J M 2009 *Phys. Rev. E* **80** 026601
[12] Ohta Y and Yang J K 2012 *Proc. R. Soc. A* **468** 1716
[13] Hasegawa A and Tappert F 1973 *Appl. Phys. Lett.* **23** 142
[14] Kavitha L and Daniel M 2003 *J. Phys. A: Math. Gen.* **36** 10471
[15] Qiao Z J 1993 *J. Math. Phys.* **34** 3110
[16] Wang B, Zhang Z and Li B 2020 *Chin. Phys. Lett.* **37** 030501
[17] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436
[18] Mitchell T M 1997 *Machine Learning* (McGraw-Hill Press series in computer science)
[19] Bishop C M 2006 *Pattern Recognition and Machine Learning* (Springer)
[20] Alipanahi B, Delong A, Weirauch M T and Frey B J 2015 *Nat. Biotechnol.* **33** 831
[21] Duda R O, Hart P E and Stork D G 2000 *Pattern Classification* (Wiley-Interscience Press)
[22] Lake B M, Salakhutdinov R and Tenenbaum J B 2015 *Science* **350** 1332
[23] Krizhevsky A, Sutskever I and Hinton G 2017 *Communications of the Acm* **60** 84
[24] Mcculloch W S and Pitts W 1943 *Bull. Math. Biophys.* **5** 115
[25] Rosenblatt F 1958 *Psychological Review* **65** 386
[26] Bryson A E and Ho Y C 1975 *Applied Optimal Control: Optimization, Estimation, and Control* (Taylor and Francis Press)
[27] Lagaris I E, Likas A and Fotiadis D I 1998 *IEEE Transactions on Neural Networks* **9** 987
[28] Hornik K, Stinchcombe M and White H 1989 *Neural Netw.* **2** 359
[29] Raissi M, Perdikaris P and Karniadakis G E 2019 *J. Comput. Phys.* **378** 686
[30] Jagtap A D, Kharazmi E and Karniadakis G E 2020 *Comput. Methods Appl. Mech. Engrg.* **365** 113028
[31] Lax P D 1968 *Comm. Pure. Appl. Math.* **21** 467
[32] Yu S J, Toda K and Fukuyama T 1998 *J. Phys. A: Math. Gen.* **31** 10181
[33] Iwao M and Hirota R 1997 *J. Phys. Soc. Jpn.* **66** 577
[34] Osman M S, Ghanbari B and Machado J A T 2019 *Eur. Phys. J. Plus* **134** 20
[35] Dong J J, Li B and Yuen M W 2020 *Commun. Theor. Phys.* **72** 025002
[36] Hirota R 2004 *Direct Methods in Soliton Theory* (Springer-verlag Press)
[37] Geng X G and Tam H W 1999 *J. Phys. Soc. Jpn.* **68** 1508

[38] Matveev V B and Salle M A 1991 *Darboux Transformation and Solitons* (Springer Press)
[39] Olver P J 1993 *Applications of Lie Groups to Differential Equations* (Springer Press)
[40] Zakharov V E, Manakov S V, Novikov S P and Pitaevskii L P 1984 *The Theory of Solitons: The Inverse Scattering Method* (Consultants Bureau Press)
[41] Pu J C and Chen Y 2020 *Mod. Phys. Lett. B* **34** 2050288
[42] Zhang Z, Yang X Y, Li W T and Li B 2019 *Chin. Phys. B* **28** 110201
[43] Li J and Chen Y 2020 *Commun. Theor. Phys.* **72** 105005
[44] Bongard J and Lipson H 2007 *Proc. Natl. Acad. Sci. USA* **104** 9943
[45] Raissi M, Perdikaris P and Karniadakis G E 2017 *J. Comput. Phys.* **348** 683
[46] Li J and Chen Y 2020 *Commun. Theor. Phys.* **72** 115003
[47] Li J and Chen Y 2020 *Commun. Theor. Phys.* **73** 015001
[48] Marcucci G, Pierangeli D and Conti C 2020 *Phys. Rev. Lett.* **125** 093901
[49] Baydin A G, Pearlmutter B A, Radul A A and Siskind J M 2018 *J. Mach. Learn. Res.* **18** 1
[50] Stein M L 1987 *Technometrics* **29** 143
[51] Choromanska A, Henaff M, Mathieu M, Arous G B and LeCun Y 2015 *Proc. 18 Int. Conf. on Artificial Intelligence and Statistics, PMLR* **38** 192
[52] Liu D C and Nocedal J 1989 *Math. Program.* **45** 503
[53] Yang J K 2010 *Nonlinear Waves in Integrable and Nonintegrable Systems* (SIAM, Philadelphia Press)
[54] Bludov Y V, Konotop V V and Akhmediev N 2009 *Phys. Rev. A* **80** 033610
[55] Moslem W M 2011 *Phys. Plasmas* **18** 032301
[56] Yan Z Y 2011 *Phys. Lett. A* **375** 4274