



Solution of the Hirota equation using a physics-informed neural network method with embedded conservation laws

Ruibo Zhang · Jin Su · Jinqian Feng

Received: 22 March 2023 / Accepted: 23 April 2023 / Published online: 14 May 2023
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

Abstract The solution of the integrable Hirota equation has attracted considerable attention in the applications of nonlinear optics, electromagnetics, and many other natural sciences. In this paper, we propose an improved physics-informed neural network (IPINN) method to study numerical solutions of the Hirota equation, which embeds energy conservation laws into a traditional neural network through the Lax pair formulation. Our simulation results show that the proposed method can predict the solutions and parameters of the Hirota equation more accurately than the traditional physics-informed neural network method. In addition, the influence on the rogue wave solution for the Hirota equation of the three factors of the IPINN method that are, number of network layers and hidden layer neurons, sampling points, and noises, is also analyzed in detail. In our study, it is worth noting that the presented method can achieve good prediction with fewer training data and iterations.

Keywords Hirota equation · Lax pair · Neural network · Conservation law · Data-driven solution · Data-driven parameter discovery

R. Zhang · J. Su (✉) · J. Feng
School of Science, Xi'an Polytechnic University,
Xi'an 710048, Shaanxi, China
e-mail: sujin@xpu.edu.cn

R. Zhang
e-mail: zhangruibo202106@163.com

1 Introduction

The nonlinear Schrödinger equation (NLSE) is

$$iQ_t + Q_{xx} + 2|Q|^2Q = 0, \quad (x, t) \in \mathbb{R}^2, \quad (1)$$

where $Q(x, t)$ are complex-valued solutions. Equation (1) describes the optical pulse propagation when the pulse width is greater than 100 femtosecond, such as in nonlinear optics [1], plasmas [2], fluid mechanics [3], and ocean [4]. The NLSE was first introduced by Hull et al. [5] as a model for superfluid helium behavior, but its importance was not fully realized until the development of optical fiber communication technology in the 1970 s. With some further research undergoing, the NLSE can only describe the nonlinear behavior in continuous media and cannot handle the discrete nonlinear phenomena in discrete systems. Therefore, Mio et al. [6] introduced the discrete time derivative term in NLSE and then put forward the derivative nonlinear Schrödinger equation (DNLS). Kaup and Newell [7] solved the appropriate inverse scattering problem and obtained the single soliton solution of DNLS. Xiang et al. and Jia et al. [8, 9] obtained the breather, soliton, and rogue wave solutions of coupled DNLS using the Darboux transformation. However, with the deep study of physical phenomena, researchers began to realize that the low-order nonlinear Schrödinger equation could not completely describe all phenomena. For ultrashort pulse systems (e.g., electron dynamics of graphene nanoring

[10], ranging systems [11]), high-order dispersion and nonlinear effects are the key influential factors in the propagation of sub-picosecond or femtosecond pulses [12, 13]. The dynamics of such systems are described by the high-order nonlinear Schrödinger equation (HNLSE). The third-order NLS equation (alias the Hirota equation [14]) is a basic HNLSE physical model, and it is completely integrable. In recent decades, many scholars have spent a great deal of time and done a lot of work studying the Hirota equation. Zhang and Yuan [15] studied the defocusing Hirota equation, constructed the N th-iterated binary Darboux transformation by the limit technique, derived multi-dark soliton solutions from the nonzero background, and discussed the properties of dark solitons. By using the Hirota bilinear method, Zuo et al. [16] obtained the higher-order solutions of the generalized Hirota equation and the exact solutions of the S -symmetric, T -symmetric, and ST -symmetric nonlocal Hirota equations. Hao and Zhang [17] studied the interaction of different types of solitons and gave the corresponding approximate eigenvalue and interaction period formula. Recently, richer solutions and new physical phenomena of the Hirota equation and its variants have been revealed by various methods [18–22].

In recent years, with the explosion of data volume and the rapid development of computer hardware, deep learning algorithms represented by neural networks have achieved brilliant results in computer vision [23], natural language processing [24], reinforcement learning [25], and so on. Thus, some scholars use neural networks to solve partial differential equations (PDEs) [26–28]. Compared with traditional numerical methods, neural networks do not need to grid generation the solution region, and more importantly, they can handle high-dimensional PDEs [29, 30]. However, pure data-driven algorithms often have poor robustness, cannot guarantee the convergence of the algorithm, and lack physical interpretability. For this reason, some researchers utilize the Gaussian process regression method to solve PDE [31]. But the Gaussian process itself has some limitations [32]. Therefore, Raissi et al. [33–35] proposed a physics-informed neural network method (PINN), which embeds known physical prior knowledge into the network for learning. Specifically, the control equation is used as part of the loss function, and then, the network is trained using the general approximation ability of the neural network [36] and

automatic differentiation (AD) techniques [37]. PINN has been successfully applied to complex problems in computational science and engineering, such as cardiovascular systems [38], finance [39], and turbulence [40]. Particularly, for integrable equations, Fang et al. [41] designed a conservation law constrained PINN method. They used the law of conservation of energy and the law of conservation of momentum to constrain the loss function of PINN and studied the soliton solutions and rogue wave solutions of the NLSE, Korteweg–de Vries, and modified Korteweg–de Vries equations. However, for HNLSE with higher-order dispersion and nonlinear effects, there are not given the corresponding conservation laws combined with neural networks for solving. Zhou and Yan [42] used the original PINN to solve the forward and inverse problems of the Hirota equation, but it requires a large number of collocation points. Bai [43] utilized the PINN with a local adaptive activation function to obtain the bright soliton solution and rogue wave solution of the Hirota equation, yet it needs more iterations. Inspired by these works, to predict data-driven of the Hirota equation with less data, we propose an improved PINN (IPINN) method embedded with energy conservation laws, which is constructed by utilizing the Lax pair formulation. We will mainly focus on the bright soliton solution and the rogue wave solution, and parameter inversion of the Hirota equation simulated by using the IPINN method. In short, the main contributions of this paper are as follows:

- (1) The energy conservation law of the Hirota equation is derived using Lax pairs and embedding into the traditional neural network to impose stricter constraints on the loss function of the network, which improves the performance of the traditional PINN with less training data;
- (2) Compared with the traditional PINN method, the new IPINN method can more accurately predict the bright soliton solution, the rogue wave solution, the second-order term, the third-order term coefficients, and the parameters contained in the third-order term;
- (3) For the rogue wave solution, the influences of noise and network structure on the IPINN method are also investigated, and the results show that IPINN can also predict the behavioral

changes of the rogue wave solution under noisy data well.

This paper is organized as follows. In Sect. 2, the energy conservation law for the Hirota equation is constructed through Lax pair and embed into the IPINN method. In Sect. 3, the bright soliton solution and the rogue wave solution of the Hirota equation are obtained by the IPINN method and compared with the traditional PINN method's, and the robustness of the IPINN method is discussed under different number of network layers, each hidden layer neurons, sampling points, and noises. In Sect. 4, the unknown parameters in Hirota equation are identified by the proposed method. Finally, the conclusion is given in Sect. 5.

2 Methodology

In this section, we consider the Hirota equation along with Dirichlet boundary conditions given by

$$\begin{cases} iQ_t + \alpha(Q_{xx} + 2|Q|^2Q) + i\beta(Q_{xxx} + 6|Q|^2Q_x) = 0, \\ x \in (-L, L), t \in (t_0, t_1), \\ Q(x, t_0) = Q_0(x), x \in [-L, L], \\ Q(-L, t) = Q(L, t), t \in [t_0, t_1], \end{cases} \tag{2}$$

where $i = \sqrt{-1}$. $-L$ and L represent the upper and lower boundaries of the spatial domain x , respectively. t_0 and t_1 denote the start time and end time of the time domain t , respectively. $Q(x, t)$ is a complex envelope field. Assume $Q(x, t) = u(x, t) + iv(x, t)$, where $u(x, t)$ and $v(x, t)$ are the real and imaginary parts of $Q(x, t)$, respectively. α represents the group velocity dispersion coefficient, and β denotes the third-order dispersion coefficient. We solve Eq. (2) in terms of the real and imaginary parts and define

$$\begin{aligned} -v_t + \alpha(u_{xx} + 2(u^2 + v^2)u) \\ - \beta(v_{xxx} + 6(u^2 + v^2)v_x) = 0, \end{aligned} \tag{3}$$

$$\begin{aligned} u_t + \alpha(v_{xx} + 2(u^2 + v^2)v) \\ + \beta(u_{xxx} + 6(u^2 + v^2)u_x) = 0. \end{aligned} \tag{4}$$

Next, the deep neural network is used to approximate the solutions of Eqs. (3) and (4), and then, the governing equation is embedded into the network. The derivative of the solution to time and space is obtained by using AD

technique. Specifically, we define the residual networks $f_u(x, t)$ and $f_v(x, t)$, which are given by the left-hand side of Eqs. (3) and (4), respectively

$$\begin{aligned} f_u := & -\hat{v}_t + \alpha(\hat{u}_{xx} + 2(\hat{u}^2 + \hat{v}^2)\hat{u}) \\ & - \beta(\hat{v}_{xxx} + 6(\hat{u}^2 + \hat{v}^2)\hat{v}_x), \end{aligned} \tag{5}$$

$$\begin{aligned} f_v := & \hat{u}_t + \alpha(\hat{v}_{xx} + 2(\hat{u}^2 + \hat{v}^2)\hat{v}) \\ & + \beta(\hat{u}_{xxx} + 6(\hat{u}^2 + \hat{v}^2)\hat{u}_x). \end{aligned} \tag{6}$$

where N_u and N_v are nonlinear operators consisting of functions of real-valued solutions and functions of spatial derivatives, respectively.

Next, we will adopt the Lax pair [44] to construct the infinitely many conservation laws for Eq. (2) as the following forms:

$$\Phi_x = U\Phi, \Phi_t = V\Phi, \tag{7}$$

where $\Phi = (\Phi_1, \Phi_2)^T$ is the eigenfunction (the symbol ‘‘T’’ represents the transpose of the vector), Φ_1 and Φ_2 are the functions of x and t , and U and V are specifically in the form of

$$U = \begin{pmatrix} \lambda & Q^* \\ Q & -\lambda \end{pmatrix}, \tag{8}$$

$$V = \begin{pmatrix} A & B^* \\ B & -A \end{pmatrix}, \tag{9}$$

with

$$\begin{cases} A = -i|Q|^2 + \beta Q^*Q_x - \beta QQ_x^* - 2i\lambda\beta|Q|^2 + 2i\lambda^2 + 4i\lambda^3\beta, \\ B = -Q_x - i\beta Q_{xx} - 2i\beta|Q|^2Q + \lambda[-2\beta Q_x + 2iQ] + 4i\lambda^2\beta Q, \end{cases} \tag{10}$$

where λ is a complex eigenvalue parameter, and the symbol ‘‘*’’ denotes to the complex conjugation. Further, based on Eq. (7), introduce the fractional function $\Gamma = \Phi_2/\Phi_1$. We can obtain the following Riccati equation:

$$\Gamma_x = iQ - 2i\lambda\Gamma - iQ^*\Gamma^2. \tag{11}$$

Then, setting [45]

$$\Gamma = \sum_{m=1}^{\infty} \gamma_m \lambda^{-m}, \tag{12}$$

substituting Eq. (12) into Eq. (11), and letting the coefficients of the same power of λ equal zero, we can get

$$\left\{ \begin{array}{l} \gamma_1 = \frac{1}{2}Q \\ \gamma_2 = \frac{1}{4}iQ_t, \\ \gamma_3 = \frac{1}{8}(-|Q|^2Q - Q_{xx}) \\ \vdots \\ \gamma_{n+1} = \frac{i}{2}(\gamma_{n,t} + iQ \sum_{k=0}^{n-1} \gamma_k \gamma_{n-k}), (n = 2, 3, \dots), \end{array} \right. \tag{13}$$

where γ_m 's ($m = 1, 2, 3, \dots$) are the functions of x and t to be determined. Using compatibility conditions $(\ln \Phi_1)_{xt} = (\ln \Phi_1)_{tx}$, we can obtain

$$(i\lambda + iQ^*\Gamma)_t = (A + B\Gamma)_x. \tag{14}$$

Similarly, substituting Eqs. (10), (12), and (13) into Eq. (14) and collecting the coefficients of the same power of λ , we can yield the infinitely many conservation laws for Eq. (2) as follows:

$$\frac{\partial}{\partial t} \Gamma_m + \frac{\partial}{\partial x} \Theta_m = 0, \tag{15}$$

with

$$\left\{ \begin{array}{l} \Gamma_1 = \frac{1}{2}i|Q|^2, \\ \Theta_1 = \frac{1}{2}[\alpha(QQ_x - QQ_x^*) - i\beta(Q_xQ_x^* - Q^*Q_{xx} - QQ_{xx}^* - 3|Q|^4)], \\ \Gamma_2 = -\frac{1}{4}Q^*Q_x, \\ \Theta_2 = \frac{1}{4}[i\alpha(|Q|^4 - Q_xQ_x^* + Q^*Q_{xx}) - \beta(6Q^*Q_x|Q|^2 - Q_x^*Q_{xx} + Q_xQ_{xx}^* + Q^*Q_{xxx})] \\ \vdots \\ \Gamma_l = iQ^*\gamma_{x,l}, l = 3, 4, \dots, \\ \Theta_l = [i\beta Q_{xx}^* - \alpha Q_x^* + 2i\beta Q^*|Q|^2]\gamma_l - 2[i\alpha Q^* + \beta Q_x^*]\gamma_{l+1} - 4i\beta Q^*\gamma_{l+2}, \end{array} \right. \tag{16}$$

where Γ_m 's and Θ_m 's are the conserved densities and conserved fluxes, respectively. Substituting Γ_1 and Θ_1 into Eq. (15), we can obtain the energy conservation law of Eq. (2), which can be described as

$$i[uu_t + vv_t + \alpha(uv_{xx} - vu_{xx}) + \beta(uu_{xxx} + vv_{xxx} + 6(u^2 + v^2)(uu_x + vv_x))] = 0. \tag{17}$$

In the following, the real and imaginary parts of Eq. (17) are separated and defined as

$$f_{EC-\hat{u}} := 0, \tag{18}$$

$$f_{EC-\hat{v}} := \hat{u}\hat{u}_t + \hat{v}\hat{v}_t + \alpha(\hat{u}\hat{v}_{xx} - \hat{v}\hat{u}_{xx}) + \beta(\hat{u}\hat{u}_{xxx} + \hat{v}\hat{v}_{xxx} + 6(\hat{u}^2 + \hat{v}^2)(\hat{u}\hat{u}_x + \hat{v}\hat{v}_x)). \tag{19}$$

Now, we can give the loss function for solving Eq. (2) using IPINN specifically as follows:

$$Loss = Loss_{\hat{u}} + Loss_{\hat{v}} + Loss_f + Loss_{\hat{c}}, \tag{20}$$

where

$$\left\{ \begin{array}{l} Loss_{\hat{u}} = \frac{1}{N_0} \sum_{i=1}^{N_0} (|\hat{u}(x_u^i, t_u^i) - u^i|^2 + |\hat{v}(x_u^i, t_u^i) - v^i|^2), \\ Loss_{\hat{v}} = \frac{1}{N_b} \sum_{j=1}^{N_b} (|\hat{u}(-L, t_v^j) - \hat{u}(L, t_v^j)|^2 + |\hat{v}(-L, t_v^j) - \hat{v}(L, t_v^j)|^2), \\ Loss_f = \frac{1}{N_f} \sum_{k=1}^{N_f} (|f_u(x_{f_u}^k, t_{f_u}^k)|^2 + |f_v(x_{f_v}^k, t_{f_v}^k)|^2), \\ Loss_{\hat{c}} = \frac{1}{N_f} \sum_{k=1}^{N_f} (|f_{EC-\hat{u}}(x_{f_u}^k, t_{f_u}^k)|^2 + |f_{EC-\hat{v}}(x_{f_v}^k, t_{f_v}^k)|^2). \end{array} \right. \tag{21}$$

Here, $\{x_u^i, t_u^i, u^i\}_{i=1}^{N_0}$ and $\{x_v^j, t_v^j, v^j\}_{j=1}^{N_b}$ represent the training points sampled from the initial conditions of Eq. (2). $\{-L, t_v^j, u^j\}_{j=1}^{N_b}$, $\{L, t_v^j, u^j\}_{j=1}^{N_b}$, $\{-L, t_v^j, v^j\}_{j=1}^{N_b}$ and $\{L, t_v^j, v^j\}_{j=1}^{N_b}$ are boundary data points of Eq. (2).

$\{x_{f_u}^k, t_{f_u}^k\}_{k=1}^{N_f}$ and $\{x_{f_v}^k, t_{f_v}^k\}_{k=1}^{N_f}$ denote configuration points specified by $f_{\hat{u}}$, $f_{\hat{v}}$, $f_{EC-\hat{u}}$, and $f_{EC-\hat{v}}$. The loss function (20) consists of three parts. Specifically, $Loss_{\hat{u}}$ and $Loss_{\hat{v}}$ are the error generated by the initial and boundary data, respectively. $Loss_f$ and $Loss_{\hat{c}}$ are the errors caused by imposing Eq. (2) and its conservation law at a finite set of configuration points, respectively.

Finally, we give the network structure figure for solving the Hirota equation using the IPINN method, as shown in Fig. 1. Specifically, the solution process can be divided into three parts: (I) Input x and t , and use the deep neural network to solve \hat{u} and \hat{v} ; (II) take the training data points, Eqs. (5) and (6), and conservation laws (18) and (19) as the loss functions of the network; and (III) optimize the network parameters by employing Adam [46] and L-BFGS [47] techniques.

3 Data-driven solution for Hirota equation

In this section, we will use two different types of methods, including PINN and IPINN, to calculate the bright soliton solution and rogue wave solution of

Eq. (2), and compare them with the known Hirota equation exact solution.

In addition, in all numerical simulations, we choose a neural network with four hidden layers, and each layer has 50 neurons. Network weights are initialized using the Xavier algorithm. The activation function is hyperbolic tangent. Loss function is simply optimized by using Adam and L-BFGS algorithms. All the code in this paper is based on Python 3.7 and Tensorflow 1.15, and all numerical experiments are run on a computer with Intel(R) Core(TM) i7-4790 @ 3.60GHz processor and 16 GB memory.

3.1 Bright soliton solution

Firstly, we use the PINN method and the IPINN method to simulate the bright soliton solution of Eq. (2). Now, we give the bright soliton solution [14] of Eq. (2) as follows:

$$Q(x, t) = \operatorname{sech}(x - \beta t)e^{it}, \tag{22}$$

where β is the wave velocity. When $\beta > 0$, the wave propagates to the right; conversely, it propagates to the left. We can set $\alpha = 1$, $\beta = 0.01$ and take x and t in Eq. (2) as $[-10, 10]$ and $[0, 5]$, respectively. Therefore, we can get the initial value of Eq. (2), which is as follows:

$$Q(x, 0) = \operatorname{sech}(x). \tag{23}$$

We obtain the training data from Eq. (2) by MATLAB. Specifically, dividing space $[-10, 10]$ into

501 points and time $[0, 5]$ into 256 points, bright soliton solution $Q(x, t)$ is discretized into 256 snapshots accordingly. We try to use a small amount of data for simulation experiments. First, we randomly sample 50 points from the initial and boundary conditions, that is, $N_0 = 50$ and $N_b = 50$, respectively. Then, the Latin hypercube sampling (LHS) [48] strategy is used to select $N_f = 1000$ points from the spatio-temporal region. We minimize the loss function (20) by using 5000 steps Adam and 10,000 steps L-BFGS. After giving the training dataset, the bright soliton solution $Q(x, t)$ is successfully learned by optimizing the parameters of the neural network. The PINN method achieves a final mean squared error loss \mathbb{L}_2 of $2.7122055e-06$ at 8637 iterations. However, the IPINN model reaches a final mean squared error loss \mathbb{L}_2 of $1.8600642e-06$ at 8786 iterations.

In Fig. 2, the three-dimensional motion and density diagrams of the bright soliton solution $Q(x, t)$ with the exact, PINN, and IPINN methods are plotted, respectively, as well as the iteration number curves of the two data-driven methods. We find that the IPINN method can accurately predict the bright soliton solution. It is not difficult to see that the training loss curve in Fig. 2d shows the relationship between the number of iterations and the loss function. Obviously, the optimization effect of IPINN method is better than that of PINN method. In Fig. 3, we give the error results between the predicted solution and exact solution of PINN and IPINN methods. Moreover, the relative \mathbb{L}_2 errors of $Q(x, t)$, $u(x, t)$, and $v(x, t)$, respectively,

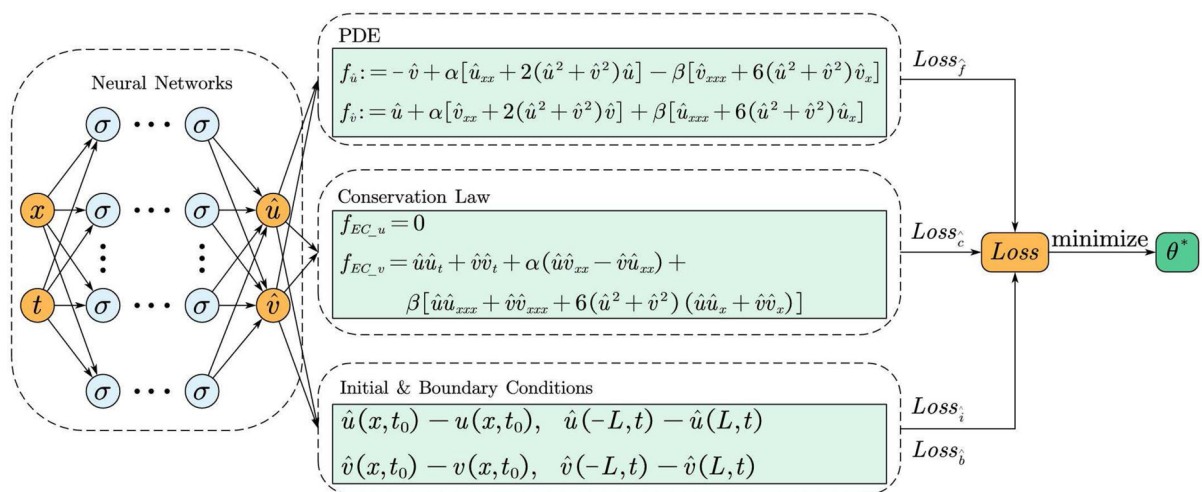


Fig. 1 Schematic of IPINN for the Hirota equation

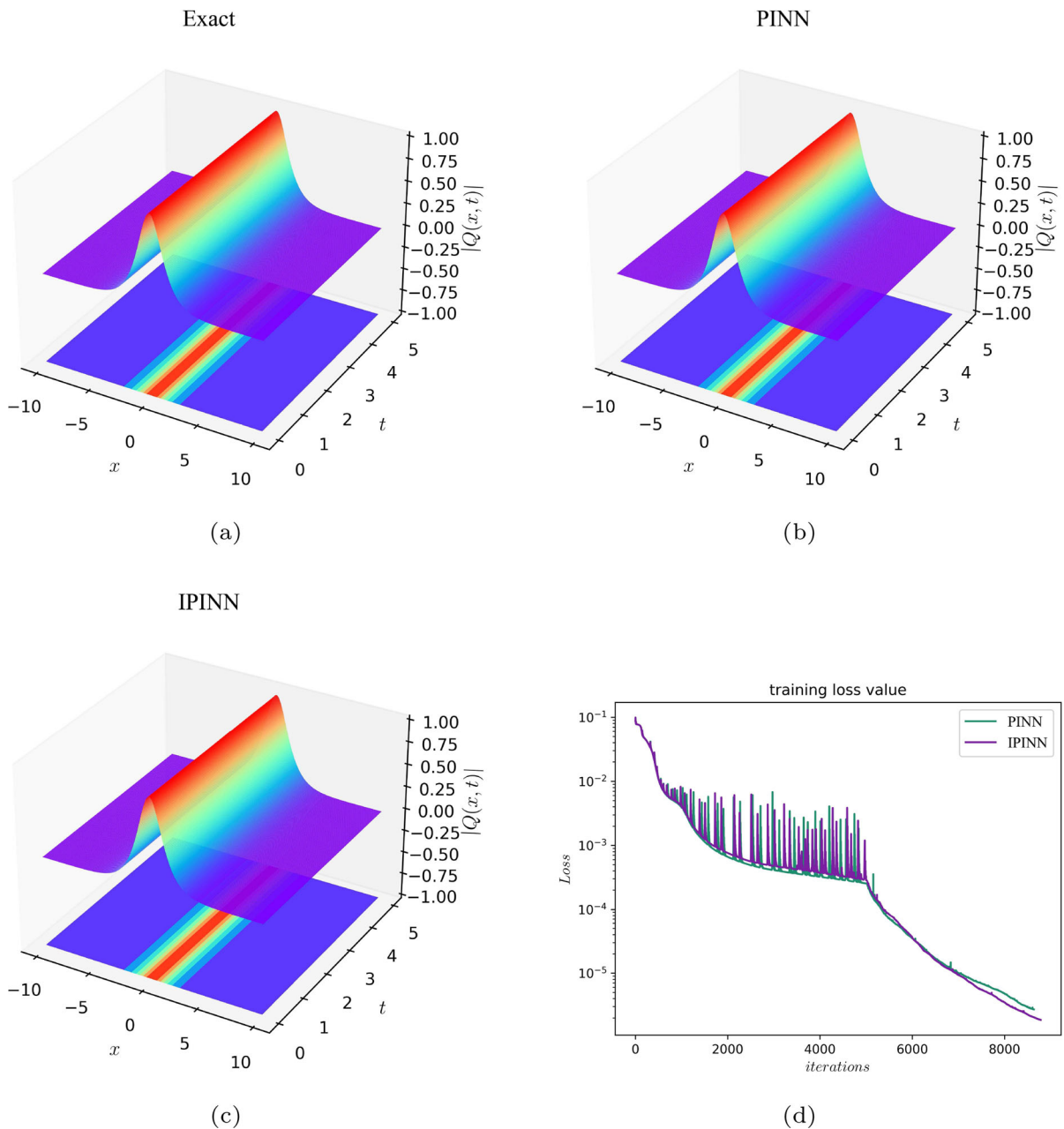


Fig. 2 Prediction of bright soliton solution (22) using PINN and IPINN. **a** Exact solution; **b** prediction solution of PINN method; **c** prediction solution of IPINN method; **d** loss curve function curves for the optimization iterations of PINN and IPINN methods

are $1.273869\text{e-}02$, $7.944935\text{e-}02$, $6.070850\text{e-}02$ in PINN method, and $6.496740\text{e-}03$, $3.690088\text{e-}02$, $2.759208\text{e-}02$ in IPINN method. In Fig. 4, we show the comparison between the exact solution and the predicted solution obtained by using PINN and IPINN methods at different times $t = 3.50, 3.92, 4.50, 4.90$. The results show that the predictions of IPINN method

match well with the evolution described by the exact solution.

3.2 Rogue wave solution

In this section, we consider using two types of methods (PINN and IPINN methods) to solve the rogue wave

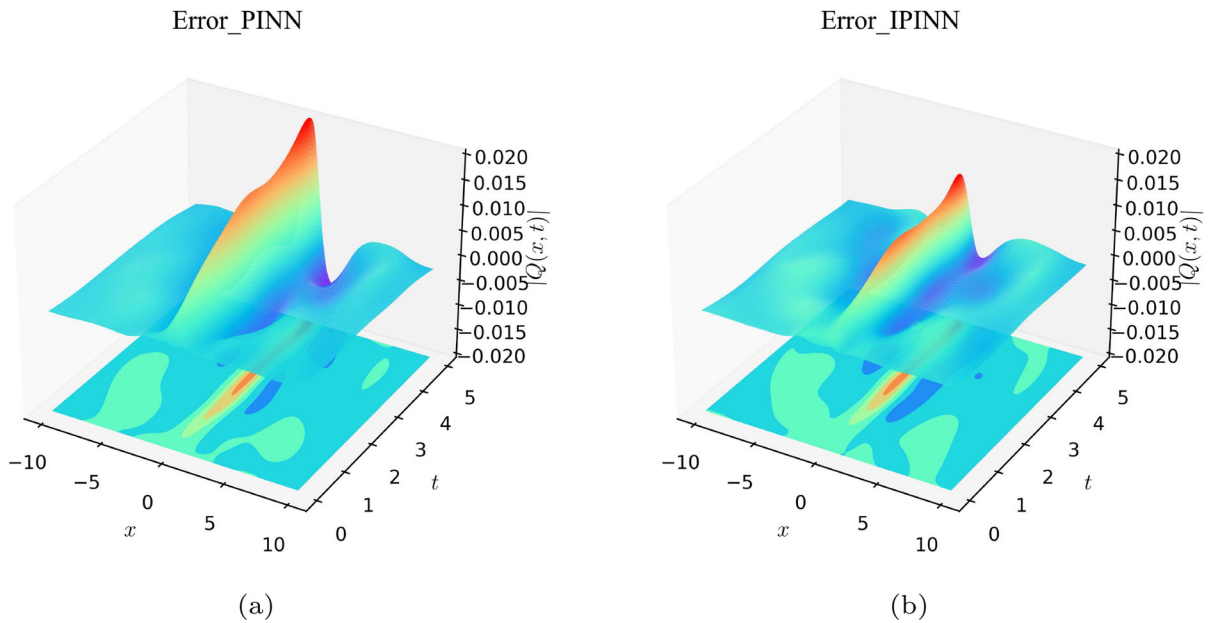


Fig. 3 The error comparison of bright soliton solution (22) is solved by PINN and IPINN methods. **a** Error of solution using PINN method; **b** error of solution using IPINN method

solution of Eq. (2). The rogue wave solution [49] of Eq. (2) can be expressed as follows:

$$Q(x, t) = \left[1 - \frac{4(1 + 4it)}{4(x - 6\beta t)^2 + 16t^2 + 1} \right] e^{2it}. \quad (24)$$

Here, we still assume $\alpha = 1, \beta = 0.01$ and choose the spatio-temporal region of Eq. (2) as $(x, t) \in [-2.5, 2.5] \times [-0.5, 0.5]$. Thus, the initial solution of Eq. (2) can be obtained as

$$Q(x, -0.5) = \left[1 - \frac{4(1 - 2i)}{4(x + 0.03)^2 + 5} \right] e^{-i}. \quad (25)$$

Next, we divide time $[-0.5, 0.5]$ into 201 points and space $[-2.5, 2.5]$ into 256 points. The rogue wave solution $Q(x, t)$ is discretized into 201 snapshots accordingly by MATLAB. It is worth noting that we do not select boundary data. The sampling number of initial condition and solution region, and the training number, is the same as in Sect. 3.1. Based on the above selection, we use PINN and IPINN methods to numerically simulate the rogue wave solution of Eq. (2). This is the case for the PINN method, which achieves a final mean squared error loss \mathbb{L}_2 of $1.0907666e-04$ and a number of iterations of 10358. Nevertheless, by using the IPINN method, it reaches a

final mean squared error loss \mathbb{L}_2 of $1.7176968e-05$ and a number of iterations of 10390.

Figure 5a–c shows the comparison between the predicted solution and the exact solution of the rogue wave with the help of PINN and IPINN methods, and we find that IPINN method can simulate the evolution of the rogue wave, while PINN method cannot. In Fig. 5d, we can clearly see that the loss function curve of IPINN method decreases faster and smoother at 5001 iterations compared to PINN method. Figure 6 shows the comparison of the error between the two methods when simulating the rogue wave solution and the exact solution, and it can be obviously found that the error of IPINN method is significantly lower than PINN method, and meanwhile, the relative \mathbb{L}_2 errors of $Q(x, t), u(x, t),$ and $v(x, t)$, respectively, are $4.192447e-02, 1.414866e+00, 1.676037e+00$ in PINN method, and $8.522858e-03, 1.436250e+00, 1.666065e+00$ in IPINN method. In Fig. 7, we show the comparison between the exact and predicted solutions of the rogue wave solution at different times $t = -0.355, -0.105, 0.045, 0.095$. Obviously, the predicted solution of the IPINN method is very close to the exact solution, while the PINN method is very poor.

In practical engineering applications, the data obtained by sampling are inevitably noisy, so we add

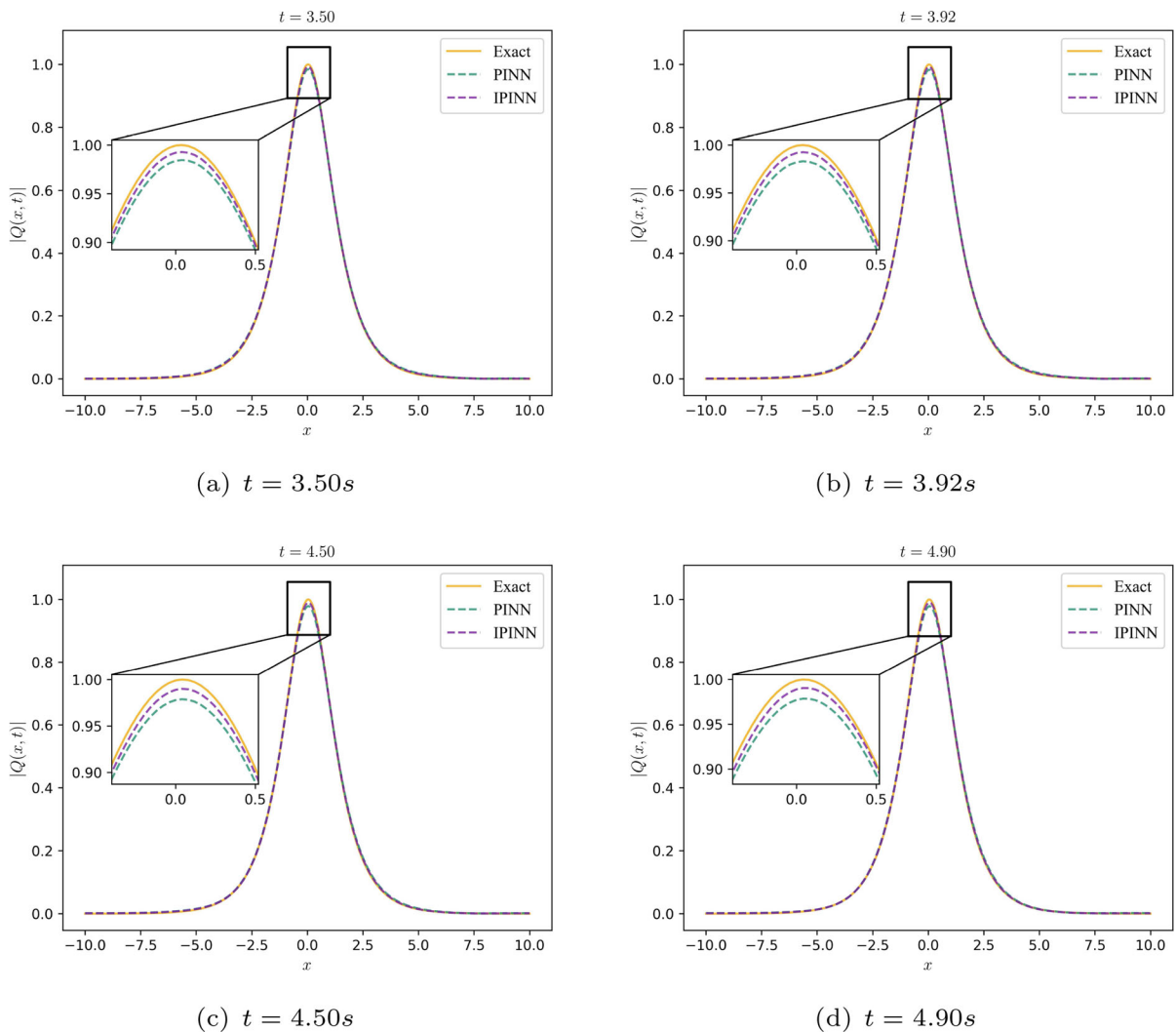


Fig. 4 Comparison of exact, PINN, and IPINN methods of the bright soliton solution (22) at different times

various degree of noises (1%, 2% Gaussian noise, as shown in Fig. 8) to the training data to verify the robustness of the IPINN method. Based on the different noisy data, we simulate the rogue wave solution (24) using the PINN and IPINN methods. The results show that for the noise as 1%, 2%, the relative \mathbb{L}_2 errors of $Q(x, t)$ are $4.311988e-02$, $4.891149e-02$ in the PINN method and $1.344709e-02$, $1.040529e-02$ in the IPINN method, respectively.

In Figs. 8, 9, and 10, we show a series of experimental results for solving the rogue wave solution (24) using the PINN method and the IPINN method under various noises. We find that the IPINN method can predict the rogue wave solution

accurately, whereas the PINN method cannot. Therefore, the IPINN method has better robustness for noisy data.

Furthermore, in the case of a four-layer neural network and 50 neurons, we discuss the effect of the number of initial condition sampling points N_0 and space-time domain sampling points N_f on solving the rogue wave of the Hirota equation by the IPINN algorithm, as shown in Table 1. The results show that, overall, when N_f is fixed, the larger the number of N_0 , the smaller the relative \mathbb{L}_2 error. When N_0 is fixed, the error does not decrease with the increase of N_f . To sum up, we can conclude that the relative \mathbb{L}_2 error is mainly determined by the initial condition sampling points

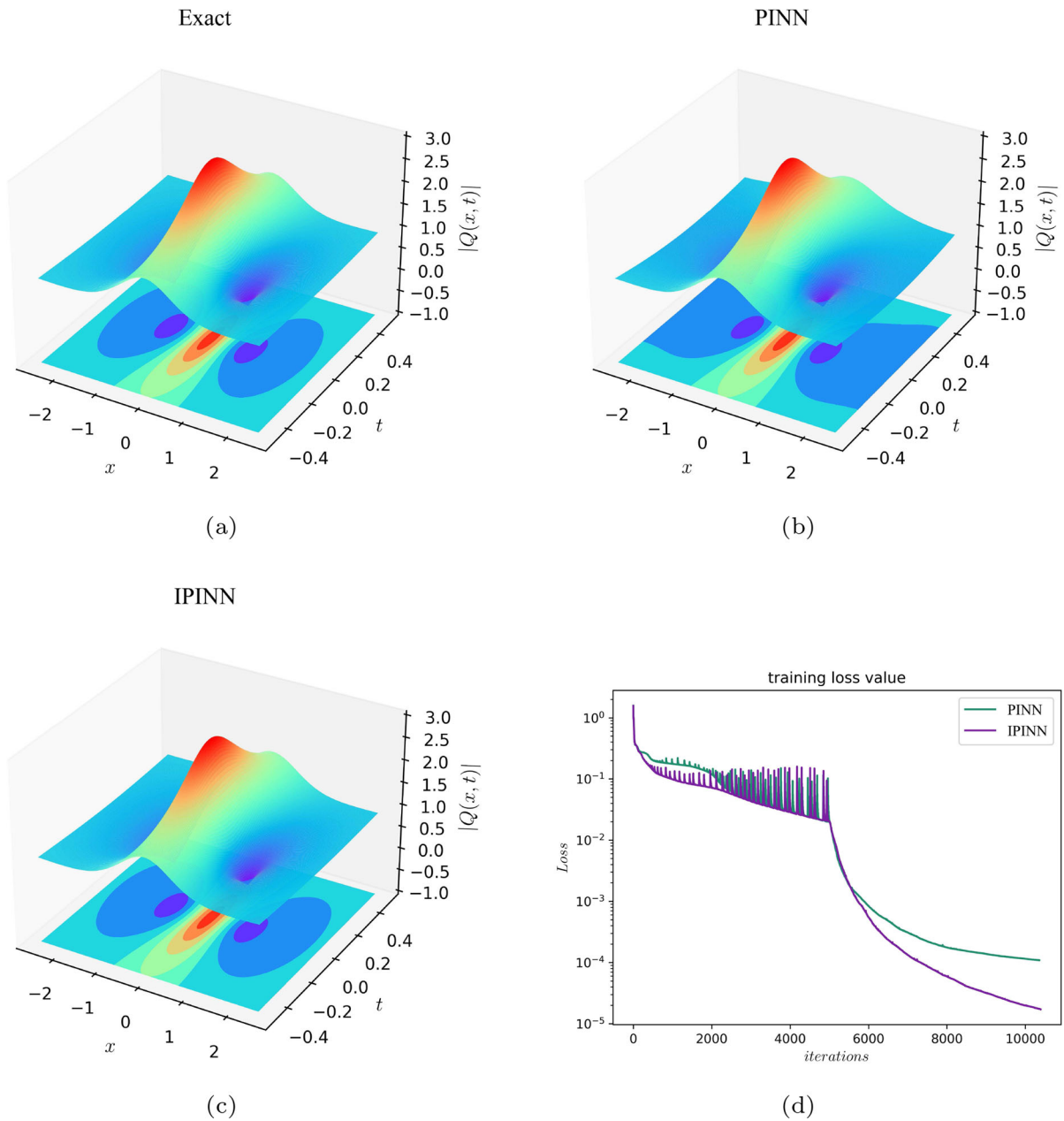


Fig. 5 Prediction of rogue wave solution (24) using PINN and IPINN. **a** Exact solution; **b** prediction solution of PINN method; **c** prediction solution of IPINN method; **d** loss curve function curves for the optimization iterations of PINN and IPINN

N_0 . Similarly, assume that $N_0 = 50$ and $N_f = 1000$ remain constant. Table 2 shows the effect of the number of network layers and neurons on the performance of the IPINN method. The results show that when the number of network layers is fixed, the

relative \mathbb{L}_2 error generally decreases with the increase of the number of neurons. When the number of neurons is fixed, the influence of the number of network layers on the relative \mathbb{L}_2 error is not obvious.

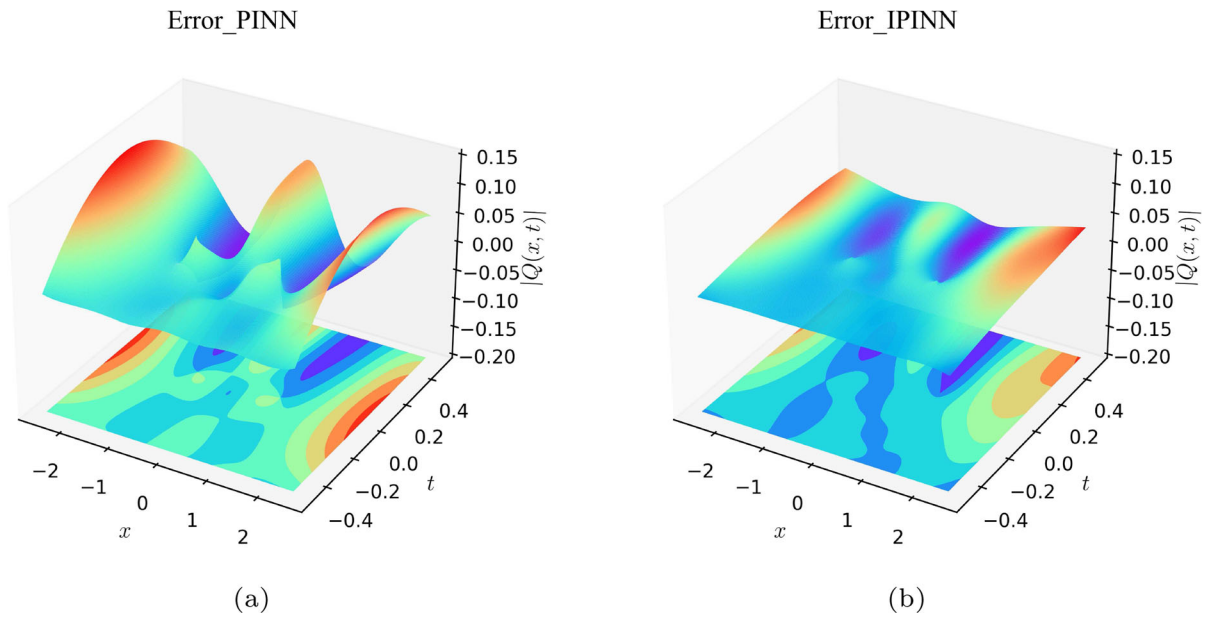


Fig. 6 The error comparison of rogue wave solution (24) is solved by PINN and IPINN. **a** Error of solution using PINN method; **b** error of solution using IPINN method

4 Data-driven parameter discovery for Hirota equation

Next, we will consider the using of the IPINN method to simulate the parameter discovery of Eq. (2). The IPINN method is first used to identify the parameters α and β in Eq. (2). In addition, we use this method to predict the parameters of the higher-order terms of Eq. (2).

4.1 Parameter discovery for α and β

In this section, we use two different methods to identify the coefficients α and β of the second-order and third-order terms in the Hirota equation as follows:

$$iQ_t + \alpha(Q_{xx} + 2|Q|^2Q) + i\beta(Q_{xxx} + 6|Q|^2Q_x) = 0, \tag{26}$$

where α, β are unknown real constants. We consider the bright soliton solution of Eq. (26) and set $\alpha = 1, \beta = 0.1, L = 10, t_0 = 0, t_1 = 5$. We will divide the space $[-10, 10]$ into 501 points, time $[0, 5]$ into 256 points, and discrete the bright soliton solution $Q(x, t)$ into 256 snapshots. The LHS strategy is used to select $(x, t) \in [-10, 10] \times [0, 5]$ data points from the spatio-

temporal domain for parameter discovery. The rest of the settings are the same as in Sect. 3.1.

The results show that the final mean squared error loss \mathbb{L}_2 of IPINN is $1.485e-04$, and the relative \mathbb{L}_2 errors of $Q(x, t), u(x, t),$ and $v(x, t)$ are $1.928966e-04, 2.932726e-04,$ and $2.912337e-04$, respectively. However, the final mean squared error loss \mathbb{L}_2 of PINN is $1.763e-04$, and the relative \mathbb{L}_2 errors of $Q(x, t), u(x, t),$ and $v(x, t)$ are $2.137185e-04, 3.144962e-04,$ and $3.041743e-04$, respectively. Table 3 exhibits the learning parameters α, β in Eq. (26) under the cases of PINN and IPINN, and their errors of α, β are $3.600e-05, 2.932e-04$ and $9.800e-06, 1.756e-04$, respectively.

4.2 Parameter discovery for μ and ν

In this section, we will use the PINN and IPINN methods to predict the parameters of the higher-order terms in the Hirota equation, and we consider the Hirota equation of the form with two parameters, as following

$$iQ_t + Q_{xx} + 2|Q|^2Q + \frac{i}{2}(\mu Q_{xxx} + \nu|Q|^2Q_x) = 0, \tag{27}$$

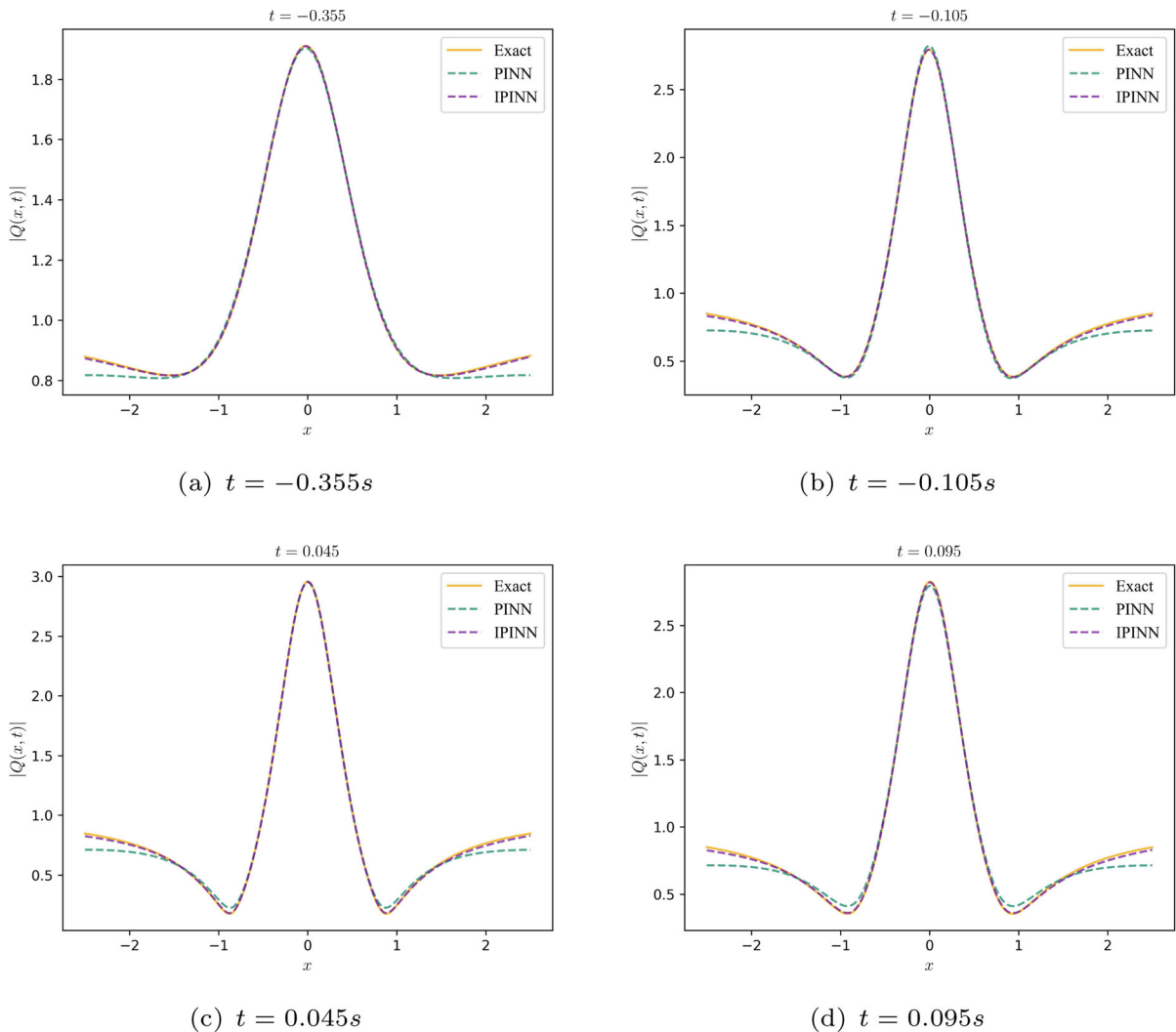


Fig. 7 Comparison of exact, PINN, and IPINN of the rogue wave solution (24) at different times

where μ is the third-order dispersion coefficient. v is related to the time-delay correction for the cubic nonlinearity term $|Q|^2Q$. We assume $\mu = 1, v = 6$. At the same time, we still consider the bright soliton solution of Eq. (27), and all the experimental settings are the same as Sect. 4.1.

This is the case for the PINN method, which achieves a final mean squared error loss \mathbb{L}_2 of $2.563e-04$. Nevertheless, by using the IPINN method, it reaches a final mean squared error loss \mathbb{L}_2 of $2.177e-04$. The relative \mathbb{L}_2 errors of $Q(x, t), u(x, t)$ and $v(x, t)$, respectively, are $2.846415e-04, 3.969635e-04, 3.787746e-04$ in PINN method, and $2.594493e-04, 3.892033e-04, 3.499485e-04$ in

IPINN method. Table 4 illustrates the learning parameters μ, v in Eq. (27) under the cases of PINN and IPINN, and their errors of μ, v are $6.248e-04, 5.342e-04$ and $3.704e-04, 3.297e-04$, respectively.

5 Conclusion

In this paper, we proposed a new PINN method embedded with the conservation law, called IPINN, to solve the classical integrable Hirota equation. This algorithm improves the performance of the traditional PINN with less training data by adding the conservation laws of the equations to the neural network and

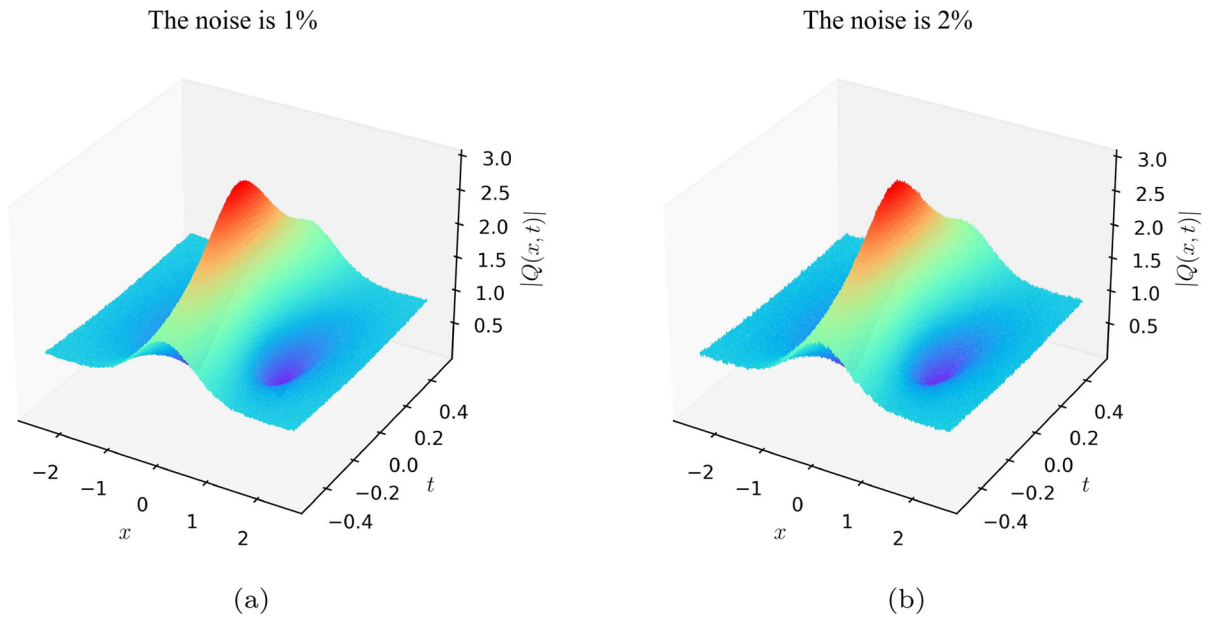


Fig. 8 Training data with different noises. **a** The noise is 1%; **b** the noise is 2%

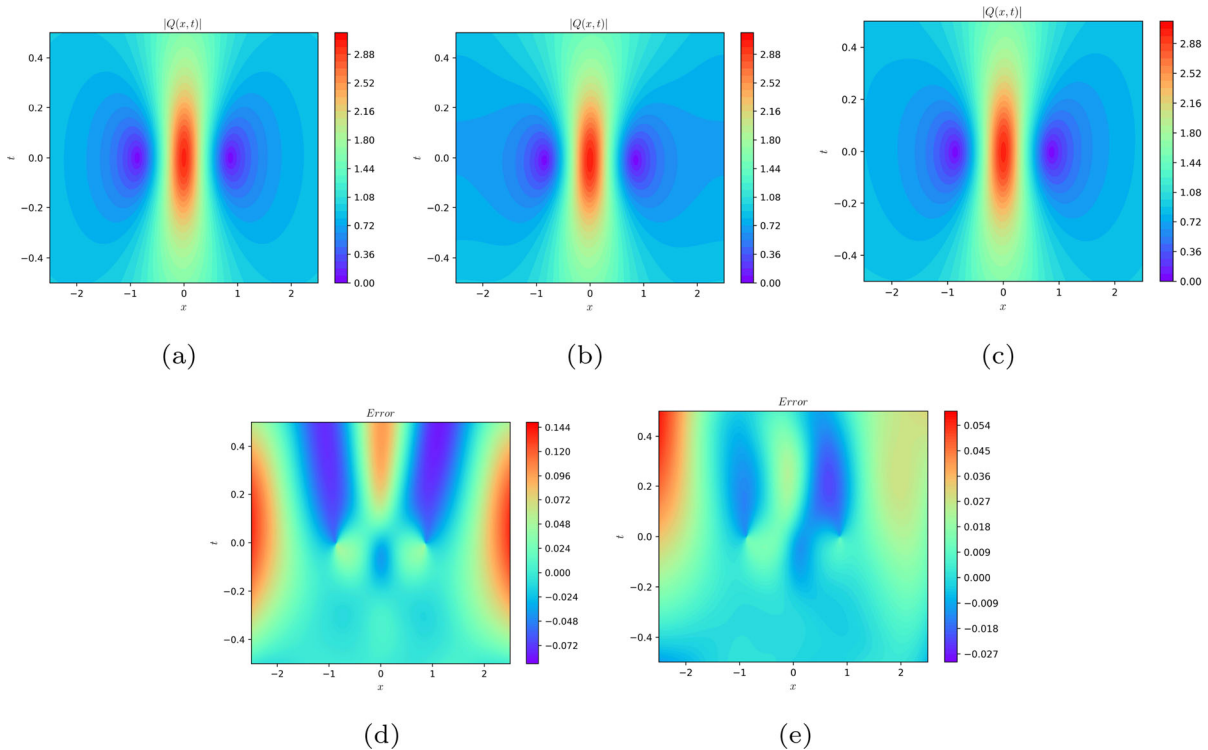


Fig. 9 Density and error diagrams of the PINN and IPINN methods for predicting rogue wave solution (24) when the noise is 1%. **a** Exact solution; **b** prediction solution of PINN method;

c prediction solution of IPINN method; **d** error of PINN method; **e** error of IPINN method

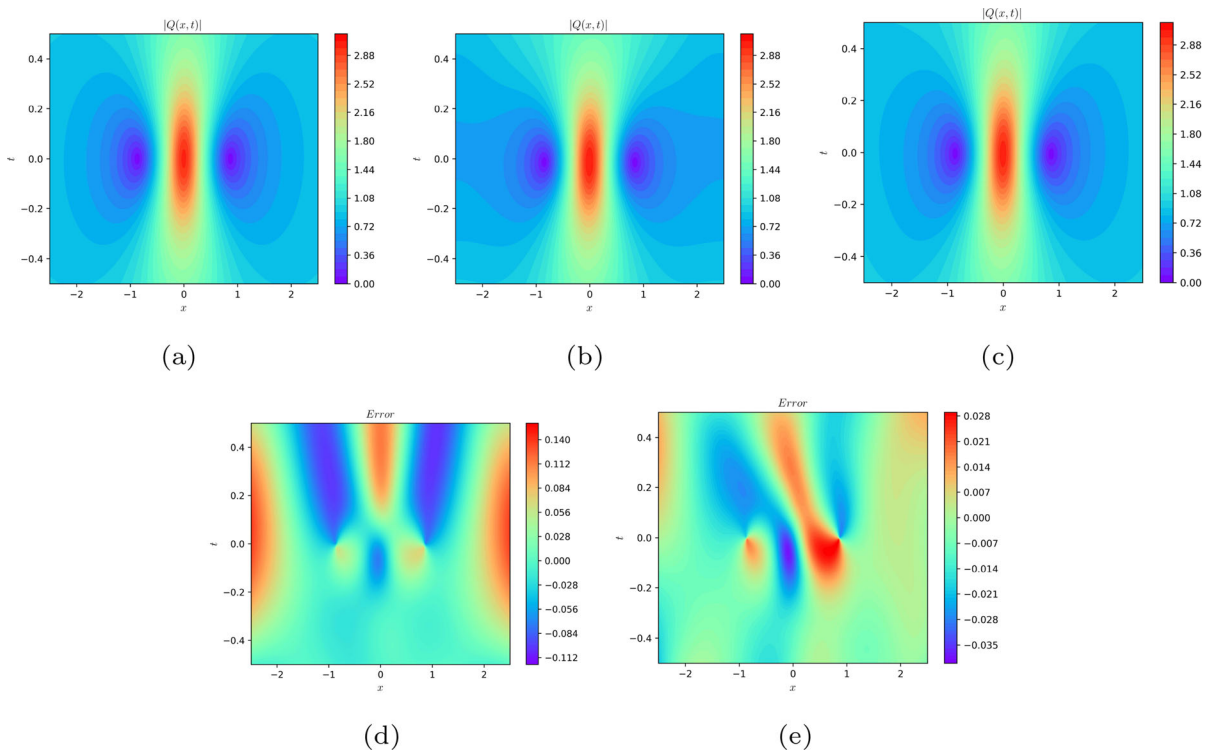


Fig. 10 Density and error diagrams of the PINN and IPINN methods for predicting rogue wave solution (24) when the noise is 2%. **a** Exact solution; **b** prediction solution of PINN method;

c prediction solution of IPINN method; **d** error of PINN method; **e** error of IPINN method

Table 1 The rogue wave solution (24) of the Hirota equation by using the IPINN: Relative \mathbb{L}_2 error of $Q(x, t)$ for different number of N_0 and N_f

N_0	N_f					
	500	1000	1500	2000	2500	3000
50	8.805079e-03	8.522858e-03	1.304595e-02	1.331060e-02	1.053535e-02	1.434573e-02
100	8.541044e-03	8.462943e-03	7.578718e-03	1.151007e-02	1.610343e-02	8.572823e-03
150	1.027909e-02	6.457162e-03	1.097187e-02	9.039189e-03	7.767210e-03	6.797012e-03
200	6.780804e-03	4.538972e-03	3.897096e-03	6.797249e-03	4.938941e-03	1.056135e-02

imposing stricter constraints on the loss function of the network.

In our numerical experiments, the bright soliton solution and the rogue wave solution of the Hirota equation are simulated by using IPINN method, and we compared the present algorithm with traditional PINN. We found that, noticeably, for the rogue wave solution, the effects of different noisy data and network structures on the IPINN method are studied, and the results show that IPINN can also predict the

evolution process of the rogue wave solution well under noisy data. Furthermore, in the inverse problem, the second-order term, the third-order term coefficient, and the parameters contained in the third-order term of the Hirota equation are identified through IPINN.

In general, numerical results show that our presented IPINN method is more effective than the PINN method in accurately recovering the different evolutionary processes and parameter discovery of the Hirota equation. The IPINN method is a promising

Table 2 The rogue wave solution (24) of the Hirota equation by using the IPINN: Relative L_2 error of $Q(x, t)$ for different number of hidden layers and neurons in each layer

Layers	Neurons				
	10	20	30	40	50
4	4.432246e-01	2.288071e-02	9.843036e-03	8.845453e-03	8.522858e-03
5	2.274656e-01	3.345982e-02	5.243493e-03	1.005136e-02	1.747532e-02
6	1.795938e-01	1.080672e-02	1.278334e-02	7.329781e-03	1.668733e-02
7	4.413522e-01	1.208218e-01	1.140209e-02	8.279967e-03	3.070231e-03

Table 3 The parameter discovery and error comparison of α and β by PINN and IPINN methods

Solution	α	Error of α	β	Error of β
Exact	1	0	0.1	0
PINN	1.000036	3.600e-05	0.099971	2.932e-04
IPINN	1.000010	9.800e-06	0.100018	1.756e-04

Table 4 The parameter discovery and error comparison of μ and ν by PINN and IPINN methods

Solution	μ	Error of μ	ν	Error of ν
Exact	1	0	6	0
PINN	1.000625	6.248e-04	6.003205	5.342e-04
IPINN	0.999630	3.704e-04	5.998022	3.297e-04

method to improve the solution of integrable equations. Since IPINN can use less data to simulate the evolution process of integrable equations, in future research, we consider combining deep learning with integrable system theory to solve more complex integrable equations.

Acknowledgements The work was supported by the National Natural Science Foundation of China (Grant No. 11601411), the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2021JM-448), and the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2023-JC-YB-063).

Data availability Data will be made available on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest concerning the publication of this manuscript.

References

- Leta, T.D., Li, J.: Dynamical behavior and exact solution in invariant manifold for a septic derivative nonlinear Schrödinger equation. *Nonlinear Dyn.* **89**(1), 509–529 (2017)
- Lovász, B., Sándor, P., Kiss, G.Z., Bánhegyi, B., Rácz, P., Pápa, Z., Budai, J., Prietl, C., Krenn, J.R., Dombi, P.: Nonadiabatic nano-optical tunneling of photoelectrons in plasmonic near-fields. *Nano Lett.* **22**(6), 2303–2308 (2022)
- Cousins, W., Sapsis, T.P.: Unsteady evolution of localized unidirectional deep-water wave groups. *Phys. Rev. E* **91**(6), 063204 (2015)
- Akhmediev, N., Ankiewicz, A., Soto-Crespo, J.M.: Rogue waves and rational solutions of the nonlinear Schrödinger equation. *Phys. Rev. E* **80**(2), 026601 (2009)
- Hull, T.E., Infeld, L.: The factorization method, hydrogen intensities, and related problems. *Phys. Rev.* **74**(8), 905 (1948)
- Mio, K., Ogino, T., Minami, K., Takeda, S.: Modified nonlinear Schrödinger equation for Alfvén waves propagating along the magnetic field in cold plasmas. *J. Phys. Soc. Jpn.* **41**(1), 265–271 (1976)
- Kaup, D.J., Newell, A.C.: An exact solution for a derivative nonlinear Schrödinger equation. *J. Math. Phys.* **19**(4), 798–801 (1978)
- Xiang, X.S., Zuo, D.W.: Breather and rogue wave solutions of coupled derivative nonlinear Schrödinger equations. *Nonlinear Dyn.* **107**, 1195–1204 (2022)
- Jia, H.X., Zuo, D.W., Li, X.H., Xiang, X.S.: Breather, soliton and rogue wave of a two-component derivative nonlinear Schrödinger equation. *Phys. Lett. A* **405**, 127426 (2021)
- Zafar, A.J., Mitra, A., Apalkov, V.: Ultrafast valley polarization of graphene nanorings. *Phys. Rev. B* **106**(15), 155147 (2022)
- Shi, D., Li, M., Huang, G., Shu, R.: Polarization-dependent characteristics of a photon-counting laser ranging system. *Opt. Commun.* **456**, 124597 (2020)

12. Ivanov, S.K., Kartashov, Y.V., Szameit, A., Torner, L., Konotop, V.V.: Floquet edge multicolor solitons. *Laser Photonics Rev.* **16**(3), 2100398 (2022)
13. Sheveleva, A., Andral, U., Kibler, B., Colman, P., Dudley, J. M., Finot, C.: Ideal four wave mixing dynamics in a nonlinear Schrödinger equation fibre system (2022). [arXiv: 2203.06962](https://arxiv.org/abs/2203.06962)
14. Hirota, R.: Exact envelope-soliton solutions of a nonlinear wave equation. *J. Math. Phys.* **14**(7), 805–809 (1973)
15. Zhang, H.Q., Yuan, S.S.: Dark soliton solutions of the defocusing Hirota equation by the binary Darboux transformation. *Nonlinear Dyn.* **89**(1), 531–538 (2017)
16. Zuo, D.W., Zhang, G.F.: Exact solutions of the nonlocal Hirota equations. *Appl. Math. Lett.* **93**, 66–71 (2019)
17. Hao, H.Q., Zhang, J.W.: Studies on interactions between bound solitons in the Hirota equation. *Superlattices Microstruct.* **101**, 507–511 (2017)
18. Nikolić, S.N., Aleksić, N.B., Ashour, O.A., Belić, M.R., Chin, S.A.: Systematic generation of higher-order solitons and breathers of the Hirota equation on different backgrounds. *Nonlinear Dyn.* **89**(3), 1637–1649 (2017)
19. Djoptoussia, C., Tiofack, C.G.L., Mohamadou, A., Kofané, T.C.: Ultrashort self-similar periodic waves and similaritons in an inhomogeneous optical medium with an external source and modulated coefficients. *Nonlinear Dyn.* **107**, 1–14 (2022)
20. Huang, Y., Di, J., Yao, Y.: The $\bar{\delta}$ -dressing method applied to nonlinear defocusing Hirota equation with nonzero boundary conditions. *Nonlinear Dyn.* **111**, 1–12 (2022)
21. Belyaeva, T.L., Serkin, V.N.: Nonlinear dynamics of nonautonomous solitons in external potentials expressed by time-varying power series: exactly solvable higher-order nonlinear and dispersive models. *Nonlinear Dyn.* **107**(1), 1153–1162 (2022)
22. Yin, H.M., Pan, Q., Chow, K.W.: Doubly periodic solutions and breathers of the Hirota equation: recurrence, cascading mechanism and spectral analysis. *Nonlinear Dyn.* **110**, 1–18 (2022)
23. Abdelrahim, M., Saiga, H., Maeda, N., Hossain, E., Ikeda, H., Bhandari, P.: Automated sizing of colorectal polyps using computer vision. *Gut* **71**(1), 7–9 (2022)
24. Li, H.: Deep learning for natural language processing: advantages and challenges. *Natl. Sci. Rev.* **5**(1), 24–26 (2018)
25. Tomov, M.S., Schulz, E., Gershman, S.J.: Multi-task reinforcement learning in humans. *Nat. Hum. Behav.* **5**(6), 764–773 (2021)
26. Bar-Sinai, Y., Hoyer, S., Hickey, J., Brenner, M.P.: Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci.* **116**(31), 15344–15349 (2019)
27. Panghal, S., Kumar, M.: Optimization free neural network approach for solving ordinary and partial differential equations. *Eng. Comput.* **37**(4), 2989–3002 (2021)
28. Brink, A.R., Najera-Flores, D.A., Martinez, C.: The neural network collocation method for solving partial differential equations. *Neural Comput. Appl.* **33**(11), 5591–5608 (2021)
29. Han, J., Jentzen, A., Weinan, E.: Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA* **115**(34), 8505–8510 (2018)
30. Georgoulis, E.H., Loulakis, M., Tsiourvas, A.: Discrete gradient flow approximations of high dimensional evolution partial differential equations via deep neural networks. *Commun. Nonlinear Sci.* **117**, 106893 (2023)
31. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Inferring solutions of differential equations using noisy multi-fidelity data. *J. Comput. Phys.* **335**, 736–746 (2017)
32. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM J. Sci. Comput.* **40**(1), A172–A198 (2018)
33. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations (2017). [arXiv:1711.10561](https://arxiv.org/abs/1711.10561)
34. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part II): data-driven discovery of nonlinear partial differential equations (2017). [arXiv:1711.10566](https://arxiv.org/abs/1711.10566)
35. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019)
36. Hornik, K., Stinchcombe, M., White, H.: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw.* **3**(5), 551–560 (1990)
37. Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M.: Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 1–43 (2018)
38. Arzani, A., Wang, J.X., D’Souza, R.M.: Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Phys. Fluids* **33**(7), 071905 (2021)
39. Bai, Y., Chaolu, T., Bilige, S.: The application of improved physics-informed neural network (IPINN) method in finance. *Nonlinear Dyn.* **107**(4), 3655–3667 (2022)
40. Kim, J., Lee, C.: Prediction of turbulent heat transfer using convolutional neural networks. *J. Fluid Mech.* **882**, A18 (2020)
41. Fang, Y., Wu, G.Z., Kudryashov, N.A., Wang, Y.Y., Dai, C.Q.: Data-driven soliton solutions and model parameters of nonlinear wave models via the conservation-law constrained neural network method. *Chaos Solitons Fractals* **158**, 112118 (2022)
42. Zhou, Z., Yan, Z.: Deep learning neural networks for the third-order nonlinear Schrödinger equation: bright solitons, breathers, and rogue waves. *Commun. Theor. Phys.* **73**(10), 105006 (2021)
43. Bai, Y.: A novel method for solving third-order nonlinear Schrödinger equation by deep learning. *Waves Random Complex* 1–11 (2022)
44. Ankiewicz, A., Kedziora, D.J., Chowdury, A., Bandelow, U., Akhmediev, N.: Infinite hierarchy of nonlinear Schrödinger equations and their solutions. *Phys. Rev. E* **93**(1), 012206 (2016)
45. Hisakado, M., Wadati, M.: Inhomogeneous model for DNA dynamics. *J. Phys. Soc. Jpn.* **64**(4), 1098–1103 (1995)

46. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
47. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. *Math. Program.* **45**(1), 503–528 (1989)
48. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **42**(1), 55–61 (2000)
49. Ankiewicz, A., Soto-Crespo, J.M., Akhmediev, N.: Rogue waves and rational solutions of the Hirota equation. *Phys. Rev. E* **81**(4), 046602 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.