ORIGINAL PAPER

Data-driven forward-inverse problems of the 2-coupled mixed derivative nonlinear Schrödinger equation using deep learning

Wei-Xin Qiu · Kai-Li Geng · Bo-Wei Zhu · Wei Liu · Ji-Tao Li · Chao-Qing Dai

Received: 10 July 2023 / Accepted: 1 February 2024 / Published online: 4 May 2024

Abstract In recent years, generative adversarial networks(GAN) has achieved great success in generating realistic images. However, the instability of GAN and the lower accuracy of physics-informed neural networks(PINN) in solving highly complex partial differential equations make training models extremely challenging. This paper proposes a novel physics-informed GAN with gradient penalty (PIGAN-GP) and applies it to predict solutions of the 2-coupled mixed derivative nonlinear Schrödinger. The PIGAN-GP integrates PINN as part of the generator in the GAN framework, namely, utilizes PINN to solve the physical equation and generate predictions for the soliton positions and shapes. We predict the positions and shapes of nondegenerate solitons by the real and predicted solutions to demonstrate the high accuracy and stability of this PIGAN-GP network. Additionally, we also discuss the influence of noise levels and different initializations on the model parameter discovery using the PINN.

Keywords Nondegenerate soliton · Physics-informed generative adversarial networks · 2-coupled mixed derivative nonlinear Schrödinger equation

1 Introduction

A soliton is a special type of nonlinear wave that can maintain its shape and energy without dispersing or deforming, possessing unique characteristics during its propagation. Optical solitons, for instance, exhibit stable propagation in optical fibers. The study of the dynamical behavior of solitons can enhance the efficiency and stability of communication systems [1]. In the realm of nonlinear wave theory, the investigation of solitons helps us gain a better understanding of the behavior of waves in nonlinear systems and the influence of these waves on materials and media. The collision between vector solitons can lead to the exchange of energy between components [2], which has wide applications in fields such as physics, optics, acoustics, etc. However, due to the nonlinear characteristics of solitons, the prediction of their collision behavior [3] has always been a challenge. In recent years, the development of deep learning technology has provided new ideas for solving soliton prediction problems [4].

W.-X. Qiu · K.-L. Geng · B.-W. Zhu ·
W. Liu (✉) · C.-Q. Dai (✉)
Mechanical and Electrical Engineering, College of Optical, Zhejiang A&F University, Lin'an 311300, China
e-mail: liuwei@zafu.edu.cn

C.-Q. Dai
e-mail: dcq424@126.com

J.-T. Li (✉)
School of Physics and Telecommunications Engineering, Zhoukou Normal University, Zhoukou 466001, China
e-mail: lijitao@zknu.edu.cn

In the past 50 years, great progress has been made in understanding different applications of multi-scale physics from Geophysics to Biophysics by using finite difference, finite element, spectral and even meshless methods to numerically solve partial differential equations [5]. Despite continuous progress, the use of classical analysis or computational tools to simulate and predict the evolution of nonlinear multiscale systems with non-uniform cascades inevitably faces severe challenges and introduces high costs and multiple sources of uncertainty [6]. For complicated nonlinear problems, it is difficult to find accurate solutions. Researchers have proposed to predict the solutions of nonlinear problems via the physics-informed neural networks(PINN) [7], which combines neural networks and physical equations to solve the problems of efficiency and accuracy of partial differential equations(PDEs). By incorporating physical constraints, such as PDEs and physical laws, into the loss function and utilizing observational data for neural network training, the model can adapt to the system's behavior. During training, the optimization algorithm adjusts the neural network parameters by minimizing the loss function, representing the discrepancy between model predictions and actual data. Upon completion of training, the neural network becomes capable of predicting solutions to PDEs, enabling the retrieval of the system's solutions at any given time and position without the necessity of explicitly solving the PDEs.

In subsequent studies, Raissi et al. [8, 9] utilized the nonlinear fitting ability of neural networks to approximate the solutions of physical equations. This method can improve the computational efficiency and accuracy of physical simulation processes. After Raissi et al. presented the PINN method in 2019 [7, 9], Fang et al. embedded the conservation laws such as energy conservation into PINN [10] and proposed a subnet structure for physical neural networks [11]. Chao et al. introduced the local adaptive activation function of neurons into the PINN network to improve the performance of the neural network [12]. Li et al. presented an adaptive search algorithm and mixed the training prior information to enhance the approximation ability of the network [13]. Chen et al. incorporated two kinds of Miura transformation constraints into neural networks to solve nonlinear PDEs for unsupervised learning [14]. Up to now, PINN, through a data-driven approach, eliminates the need for explicitly solving PDEs and learning the behavior of PDEs from actual samples. Compared with traditional numerical methods, PINN, despite being trained on a relatively small number of data points, still yields robust numerical solutions and is applied to handle complex geometric and multi-physics scenarios. Its efficient neural network approximation endows it with significant practical value in solving real-time or large-scale problem. However, the mechanism of PINN is not suitable for using additional information samples to improve the network, it is often inefficient when processing additional information. Consequently, for some complex PDEs, the efficiency of training PINN tends to be low. For different equations, PINN requires specialized design and modification of hyperparameters [15, 16] and its simple extensions cannot completely solve different problems. For example, when solving the problem of 2-CMDNLSE [17], the network proposed by Raissi et al. alone to find the solutions of the coupled equations [7] can not get satisfactory results, and the prediction will have large errors or deformed results and lead to the poor fitting effect. If the number of neurons and network width is increased, the training frequency and training time will significantly add, and the fitting effect has not been effectively improved [7].

Recently, Generative Adversarial Network (GAN) has become very popular. GAN is also a kind of deep learning network. Because GAN has the function of learning data distribution and unsupervised learning, it has been widely used in image generation, video generation, speech synthesis and other related fields. Since Goodfellow and others put forward GAN [18] in 2014, Wasserstein GAN [19], CycleGAN [20], StyleGAN [21] and other extensions have emerged. GAN has a wide range of applicability, and is able to learn the distribution characteristics of data from unlabeled data, thus it provides an understanding of the hidden structure behind the data and generates new samples that conform to the distribution. Due to its ability to generate and synthesize data samples to expand the training set, it is very useful in situations where data is scarce or requires a large number of samples. Because GANs can accurately approximate data distributions even in the presence of scarce samples, this provides us with an approach to enhance networks using a limited set of labeled samples. GAN also has its drawbacks, namely its unstable training and difficulty in regulation. In the adversarial training process of GAN, there is a balance point between the generator and discriminator, making it difficult to optimize both networks simultaneously, which may lead to unstable training [22]. Although GAN can generate high-quality images, training GAN may be difficult due to its unstable and sensitive nature. GAN often suffers from pattern collapse [23], which generates a set of images but does not capture the full diversity of training data. In addition, GAN is very sensitive to hyperparameters and initialization, which makes GAN training more challenging. The method for training GAN is progressive growth technology, where the resolution of the generator and discriminator gradually increases during the training process [18]. This method has been proven to be effective in generating high-resolution images, but it still faces the aforementioned issues of traditional GAN architecture.

PINN is highly efficient for solving PDEs, but its mechanism is not suitable for using additional information samples to improve the network, resulting in low efficiency in processing additional information. Therefore, for some complex PDEs, the training efficiency is often low. However, GAN can accurately approximate the data distribution under scarce samples, but lacks stability. In order to solve these problems and effectively solve PDEs, we combined the GAN [24] and the PINN. We propose a novel GAN architecture, whose generator is composed of the PINN [23]. GAN is composed of two neural networks with a generator and a discriminator, which compete with each other in a minimax game. The generator will attempt to generate realistic images that can deceive the discriminator, while the discriminator will attempt to distinguish the generated fake images from real images. This architecture can be used to address the instability of traditional GAN networks and the limitations of PINN mechanisms and thus improves the accuracy of PINN and the generalization ability of the model and its adaptability to real-world situations. GAN can provide additional supervised signals as auxiliary training for PINN, and the adversarial training between the generator and discriminator can provide additional information about the behavior of the physical system, which helps improve the performance of PINN and makes the new network more robust when facing diverse physical contexts. We will use this network to predict the nondegenerate single and double soliton solutions of the coupled mixed derivative nonlinear Schrödinger Eq. (2-CMDNLSE) [25], and compare this method with the traditional PINN [26].

2 Physics-informed generative adversarial networks

As shown in Fig. 1, this method is a GAN network composed of a generator and a discriminator. We replace the generator in the traditional GAN by the PINN. We take x and t as the input of the generator to replace the original noise input, and get the output $G(x,t)$ after experiencing the neuron and activation function tanh. Then $G(x,t)$ will enter PDE processing to obtain the Loss function L_{PINN} of PINN, where $G(x,t)$ is used as input to the discriminator along with the real sample $u_{\min i}(x,t)$, and this processing allow the discriminator to evaluate the image generated by the generator against the real image. The score of the generated image is added with the scores of L_{PINN} and the real image to respectively get the loss functions of the generator and discriminator, whose values decline by training the neural network with the optimizer. This process makes the discriminator have a stronger ability to distinguish between real and generated samples, and the generator has a stronger ability to generate realistic samples, until the generator can "cheat" the discriminator into Nash equilibrium, and the training finishes.

First, we introduce the loss functions of the generator and discriminator. In an ideal state, we hope that the discriminator evaluates the images generated by the generator image(fake) and the real image(real) with scores equal to 0 and 1, respectively. Since the use of logarithmic function can improve the stability of numerical calculation and avoid the numerical instability and the underflow or overflow problems in the Floating-point calculation, we will use the logarithmic function in loss function. We hope that the generator has a stronger ability to generate realistic samples, namely $D[G(x_T^q, t_T^q)]$ tends to 1, so the formula is written as $\{1 - D[G(x_T^q, t_T^q)]\}$ in loss function. However, we hope that the discriminator can distinguish between the real and generated samples, that is $D[G(x_T^q, t_T^q)]$ and $D[u_{\min i}(x_W^q, t_W^q)]$ tend to 0 and 1 respectively, so the formula is written as $\{1 - D[u_{\min i}(x_W^q, t_W^q)]\}$ and $D[G(x_T^q, t_T^q)]$ in the loss

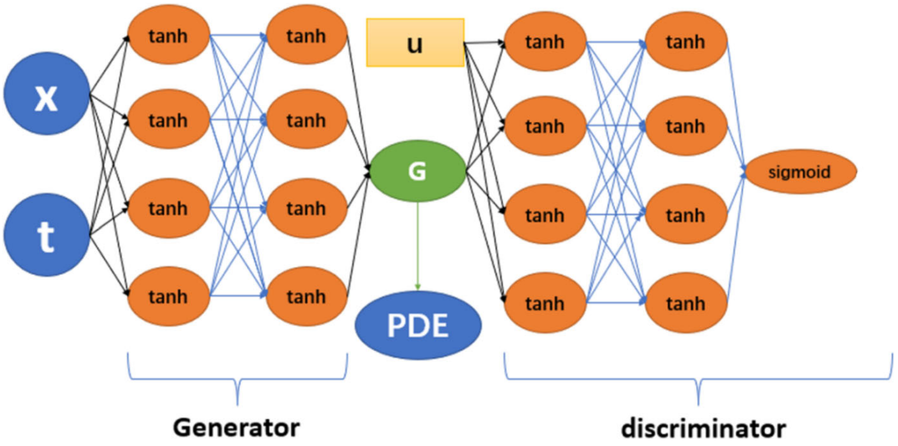

Fig. 1 The Network Structure of the PIGAN-GP

function. Therefore, loss functions of the generator G and discriminator D become

$$L_G = \lambda_1 L_{PINN} + \frac{1}{q}\sum_{q}^{1} \log\{1-D[G(x_T^q, t_T^q)]\} \tag{1}$$

$$L_D = \frac{1}{q}\sum_{q}^{1} \log\{1-D[u_{\min i}(x_W^q, t_W^q)]\} + D[G(x_T^q, t_T^q)] + \lambda_2 GP \tag{2}$$

where the real number λ_1 and λ_2 represent the coefficients of L_{PINN} and gradient penalty [24], respectively. By calculating the gradient of the discriminator output relative to the input sample, we use the norm of these gradients as a penalty term and add them into the discriminator's loss function to encourage the generation of smooth distributions and improve the training stability. Punishing gradients in each optimization iteration ensure that the discriminator's gradient remains within a reasonable range, which helps prevent gradient explosion or disappearance, and thereby enhances the robustness of the model.

Then, we introduce the situation of the discriminator, which takes a small data sample $u_{\min i}(x,t)$ on the output results $G(x,t)$ and the total area Ω of the dataset of the generator as its input, namely fake and real images. The discriminator will score the image generated by the generator and the real image and feed them back to the loss function. The small data sample is similar to the input image data of the traditional GAN, where we consider the actual coordinate (x,t) as the corresponding (x,y) in the horizontal and vertical directions of the pixel, and the amplitude of the corresponding soliton as the pixel. In this way, small data samples can be input into the discriminator as labeled data.

The numbers of network layers and neurons of the PINN network are basically the same as that of the discriminator, which is composed of ordinary linear layers and activation function tanh, while the last discriminator is the sigmoid function, which will output a scalar value to indicate whether the input is the accurate solution of the PDE. In this way, the discriminator can distinguish between the false samples generated by the generator and the samples in the real dataset, which makes the prediction results of the generator no longer solely be guided by PINN, and ultimately the output of the generator more approximate to exact solution. However, when there is a significant difference in the abilities of generators and discriminators, the training of neural networks may encounter difficulties. In order to improve the stability of training and the quality of generated samples, we introduce the gradient penalty, which is the gradient penalty in Wasserstein GAN [24]. The aim is to constrain the gradient of the discriminator to change more smoothly and reduce extreme response to the input space, which helps to generate more realistic and high-quality samples.

Next, we introduce the generator, which is obtained through PINN processing of PDE. We not only feed the output $G(x,t)$ of the generator into the discriminator, but also into the PDE. In PDE, we will have the following processing: we use the random sampling for the initial and Dirichlet boundaries of the data set to obtain MSE_{bc} and MSE_{ic} by minimizing the Mean squared error between the real and predicted values. Because Latin hypercube sampling has lower computational costs and ensures uniform distribution of sampling across all dimensions, which thereby enhances coverage of the entire input space. The selection of sampling strategy needs to strike a balance between accuracy and computational efficiency. Latin hypercube sampling meets our requirement for a relatively uniform sampling of the input space. So we perform the Latin hypercube sampling on the coordinates in the dataset [27], and then apply partial differentiation to the predicted value corresponding to the coordinate position, which is then taken into the physical equation. MSE_f is obtained by the difference between the previous and subsequent iterations. Then put the sum L_{PINN} of the above minimum Mean squared error into the loss function of the generator as the regularization mechanism.

So, we can get that the loss functions of the neural network generator are

$$MSE_{ic} = \frac{1}{N_{ic}} \sum_{i}^{N_{ic}} \left(\left| r_1(x^i, t^i) - r_1^i \right|^2 + \left| m_1(x^i, t^i) - m_1^i \right|^2 + \left| r_2(x^i, t^i) - r_2^j \right|^2 + \left| m_2(x^i, t^i) - m_2^j \right|^2 \right) \tag{3}$$

$$MSE_{bc} = \frac{1}{N_{bc}} \sum_{j}^{N_{bc}} \left(\left| r_1(x^j, t^j) - r_1^j \right|^2 + \left| m_1(x^j, t^j) - m_1^j \right|^2 . + \left| r_2(x^j, t^j) - r_2^j \right|^2 + \left| m_2(x^j, t^j) - m_2^j \right|^2 \right) \tag{4}$$

$$MSE_f = \frac{1}{N_f} \sum_{k}^{N_f} \left(\left| f_{r1}(x^k, t^k) \right|^2 + \left| f_{r2}(x^k, t^k) \right|^2 . + \left| f_{m1}(x^k, t^m) \right|^2 + \left| f_{m2}(x^m, t^m) \right|^2 \right) \tag{5}$$

$$L_{PINN} = MSE_{bc} + MSE_{ic} + MSE_f \tag{6}$$

The predicted values $G(x,t)$ of the generator are composed of the real parts $r_1(x,t)$, $r_2(x,t)$ and imaginary parts $m_1(x,t)$, $m_2(x,t)$ of the complex functions u_1 and u_2, $\{r_1^i, r_2^i, m_1^i, m_2^i\}_{i=1}^{N_{ic}}$ and $\{r_1^i, r_2^i, m_1^i, m_2^i\}_{i=1}^{N_{bc}}$ represent the initial and boundary values of u_1 and u_2, $\{x^k, t^k\}_{k=1}^{N_f}$, and $f(x,t)$ is the residual calculated by substituting the selected configuration points from the total area of the dataset Ω into the physical equation.

In this article, initial points $N_{ic} = 100$, boundary points $N_{bc} = 100$, and configuration points $N_{f^-} = 10{,}000$ are used. When $G(x,t)$ continuously approaches to exact solution $u(x)$, the final trained result can to some extent meet the physical laws.

After the above modeling work is completed, we first update the weights of the discriminator, and then sequentially update the weights of the generator. The optimizers used for the two networks are Adam and SGD, respectively. The Adam optimizer can adaptively adjust the learning rate of different weights, while SGD updates parameters each time by randomly selecting samples to avoid falling into local optima [28]. Generators and discriminators are both very weak from the beginning, so they generally do not experience significant fluctuations in their loss functions at the beginning of training. After a period of stable training, the losses of both the generator and discriminator should fluctuate within a small area without a significant continuous upward or downward trend. After reaching the Nash equilibrium, the training finishes. If the generator's loss function continues to increase significantly, it indicates that it is unable to learn how to deceive the discriminator, which is reflected in the result of starting to generate noise. If the value of the discriminator's loss function continues to rise significantly, it means that it cannot learn how to recognize the generator. The result is that the generator may generate consistent, meaningless images that can deceive the discriminator, such as directly outputting samples from the training set.

We will use the PIGAN-GP to predict nondegenerate one-soliton and two-soliton solutions [25] and compare this method with the traditional PINN [7]. The positive problem in this article is programmed using the Python 3.10 and Tensorflow 2.10.1, while the inverse problem is programmed using Tensorflow 1.15. The data reported in this article are all from a 2060 graphics card, 2.10 GHz, 12th Gen Intel (R) Core (TM) i7-12,700 processor, running on a computer with 16 GB of memory.

3 Data-driven optical soliton solutions

Recently, Geng et al. [29] obtained nondegenerate one-soliton and two-soliton solutions of the 2-CMDNLSE via Hirota bilinear method [30]. This unique multimodal coupled system [31] is always accompanied by the energy conversion, which is conducive to the research of the dense data information transmission. The physical characteristics of the energy conversion of collision solitons can be used to design logic gates and fiber coupling directions [32].

In this paper, a new network structure is proposed to predict the data-driven solutions and equation parameters of 2-CMDNLSE [31]

$$iu_{1t} + u_{1xx} + \mu(|u_1|^2+|u_2|^2)u_1 + i\gamma[(|u_1|^2+|u_2|^2)u_1]_x = 0 \tag{7}$$

$$iu_{2t} + u_{2xx} + \mu(|u_1|^2+|u_2|^2)u_2 + i\gamma[(|u_1|^2+|u_2|^2)u_2]_x = 0 \tag{8}$$

Equations (7) and (8) are models that describe the propagation of ultrashort pulses in birefringent fibers. The amplitudes u_1 and u_2 of the two polarizations are related to normalized distance x and time t. μ and γ represent the real constants of the third-order and derivative third-order nonlinearity intensities, respectively.

3.1 Nondegenerate one-soliton solution

Using the PIGAN-GP and PINN, we obtain the predictive solution of nondegenerate one-soliton for 2-CMDNLSE. The exact solution of nondegenerate one-soliton [29] is

$$\begin{aligned} u_1 &= \frac{[\alpha_1 e^{\eta_1} + A_1 e^{\eta_1+\xi_1+\xi_1^*}]}{D_1}, \\ u_2 &= \frac{[\alpha_2 e^{\eta_1} + A_2 e^{\eta_1+\xi_1+\eta_1^*}]}{D_1}, \end{aligned} \tag{9}$$

where

$$\eta_1 = \kappa_1 x + \sigma_1 t, \xi_1 = \iota_1 x + \rho_1 t, \sigma_1 = i\kappa_1^2, \rho_1 = i\iota_1^2,$$

$$D_1 = 1 + C_1 e^{\eta_1+\eta_1^*} + C_2 e^{\xi_1+\xi_1^*} + \mathrm{B}_1 e^{\eta_1+\eta_1^*+\xi_1+\xi_1^*},$$

$$C_1 = \frac{(i\gamma\kappa_1 + \mu)|\alpha_1|^2}{2(\kappa_1 + \kappa_1^*)}, C_2 = \frac{(i\gamma\iota_1 + \mu)|\alpha_2|^2}{2(\iota_1 + \iota_1^*)},$$

$$\begin{aligned} A_1 &= \frac{(i\gamma\iota_1 + \mu)(\kappa_1 - \iota_1)\alpha_1|\alpha_2|^2}{2(\kappa_1 + \iota_1^*)(\iota_1 + \iota_1^*)^2}, A_2 \\ &= \frac{(i\gamma\kappa_1 + \mu)(\iota_1 - \kappa_1)\alpha_2|\alpha_1|^2}{2(\iota_1 + \kappa_1^*)(\kappa_1 + \kappa_1^*)^2}, \end{aligned}$$

$$B_1 = \frac{|\iota_1 - \kappa_1|^2|\alpha_1|^2|\alpha_2|^2(i\gamma\mu(\iota_1 + \kappa_1) - \gamma^2\iota_1\kappa_1 + \mu^2)}{4(\kappa_1 + \kappa_1^*)^2(\iota_1 + \iota_1^*)^2(\kappa_1^* + \iota_1)(\kappa_1 + \iota_1^*)}. \tag{10}$$

In the range of$x \in [-15, 35], t \in [0, 4]$, we choose the parameters $\lambda = 1, \mu = 1$, $\alpha_1 = 1.5, \alpha_2 = -1, k_1 = 0.5001, l_1 = 0.5$, use pseudo spectral method to obtain the data for the exact solution (9), and discretize it into [256, 201] data points to obtain the dataset. The PINN part adopts the same number of network layers $L = 7$, number of neurons $n = 100$, and training times *epoch* = 10,000.

The spatiotemporal dynamics of nondegenerate one-solitons is shown in Fig. 2a and b. Figure 2c and d exhibit a comparison of the predicted and exact solutions of the PIGAN-GP and PINN at three evolution time. The PIGAN-GP network takes 13 min and 52 s to achieve the prediction of relative errors L_2 = 1.869e^{-2} and 2.045e^{-2} for two components u_1 and u_2, while the PINN network takes 22 min and 45 s to achieve L_2 = 3.558e^{-1} and 3.487e^{-1} for two components u_1 and u_2. The PIGAN-GP network can indeed improve the prediction accuracy of nondegenerate one-soliton, with good accuracy in the entire spatial and temporal domains and good prediction with continuously increasing evolution time t.

In the range of$x \in [-15, 25], t \in [0, 2]$, taking a new parameter $\lambda = 1, \mu = 1$, $\alpha_1 = 0.75, \alpha_2 = -1, k_1 = 0.83, l_1 = 0.65$, we obtain the data for exact M-shaped nondegenerate one-soliton solution (9) using the pseudo spectral method and discretize it into [256, 201] data points to obtain the dataset. Figures 3c and d indicate that there are still differences in the prediction accuracy of solutions between two networks, and the PIGAN-GP network performs better in terms of accuracy. The PIGAN-GP network takes 29 min and 16 s to achieve the prediction of relative errors L_2 = 2.844e^{-2} and 2.176e^{-2} for two components u_1 and u_2, while the PINN network predicted relative errors L_2 = 3.527e^{-1} and 1.993e^{-1} for two

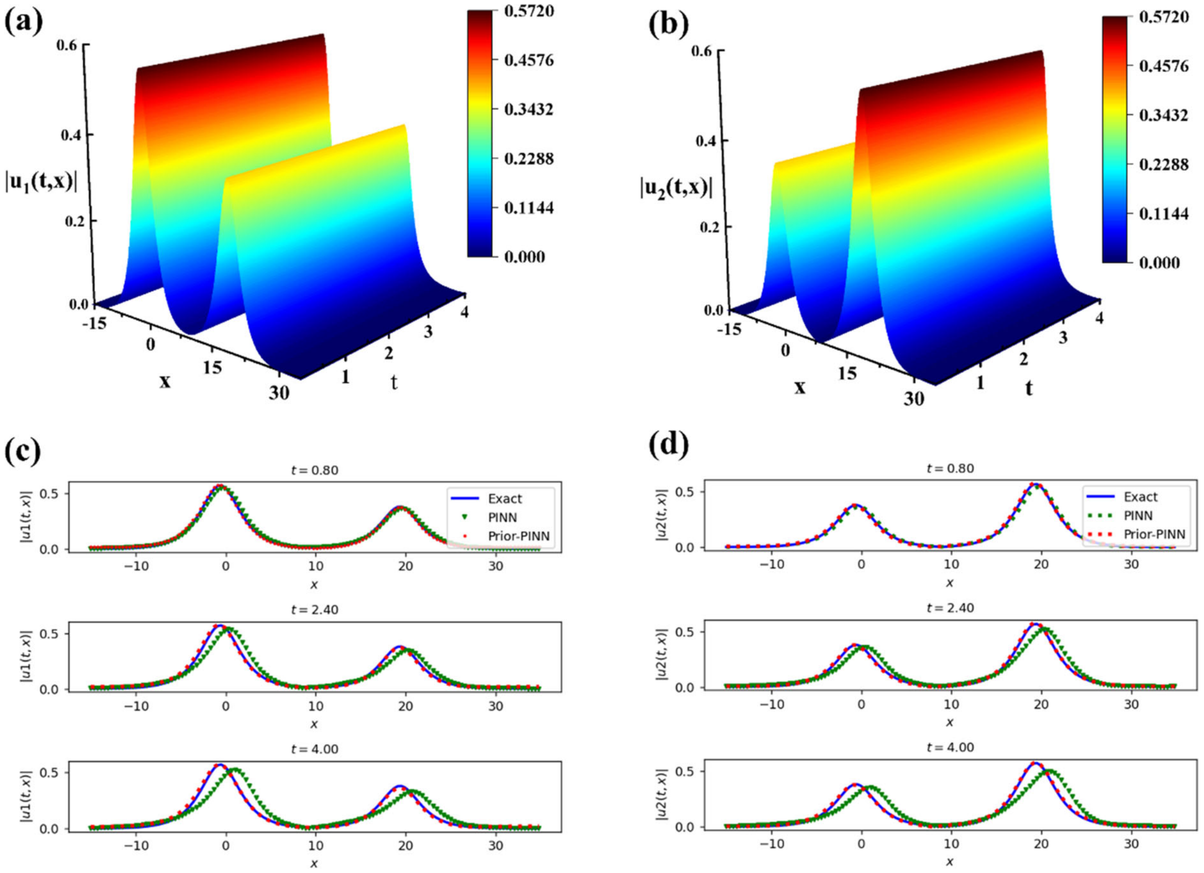

Fig. 2 Evolution of exact nondegenerate one-soliton solution for components **a** u_1 and **b** u_2. Comparison chart of predicted and exact solutions for components **c** u_1 and **d** u_2 using different networks at different evolution time

components u_1 and u_2 with a prediction time 27% less than the PIGAN-GP. Figure 3e shows the comparison of the training loss function curves of two networks only in the PINN part. It can be seen that the PIGAN-GP can reach the optimal solution more smoothly and stably, and the loss function is about $1e^{-3}$.

3.2 Nondegenerate two-soliton solution

In the given exact solution of nondegenerate double solitons in ref. [33], we take the parameters $k_1 = -1.1, l_1 = -2.1, k_2 = 1, l_2 = 2, \alpha_{11} = 1, \alpha_{12} = 1, \alpha_{21} = 1, \alpha_{22} = 1, \gamma = 1, \mu = 1$. We use the pseudo spectral method to obtain data for exact solutions and discretize them into [256, 201] data points to obtain a dataset. The PINN part adopts the number of network layer as $L = 12$, number of neurons as $n = 100$, and training frequency *epoch* = 4000. Figure 4a and b depict the boundary points and initial sampling points for sampling, and Fig. 4e is the change of loss function of two network structures. Figure 4c and d indicate that the PIGAN-GP has stability, high accuracy, and can achieve local optimal solutions earlier than PINN.

In the exact solution of nondegenerate double solitons given in reference [33], the parameter is taken as $k_1 = -1.1, l_1 = -2.1, k_2 = 1, l_2 = 2, \alpha_{11} = 1, \alpha_{12} = 3, \alpha_{21} = 11, \alpha_{22} = 1, \gamma = -1, \mu = 1$, and we can predict the dynamic behavior of another type of nondegenerate double solitons.

Figure 5 shows that the PIGAN-GP network can indeed stably improve the authenticity of predictions throughout the entire training process. The prediction of PINN spends 33 min and 49 s, while PIGAN-GP spends 28 min and 30 s, which takes 18.66% less time and the faster training speed than PINN. Figure 5e shows the comparison of absolute error for the nondegenerate single and double solitons under

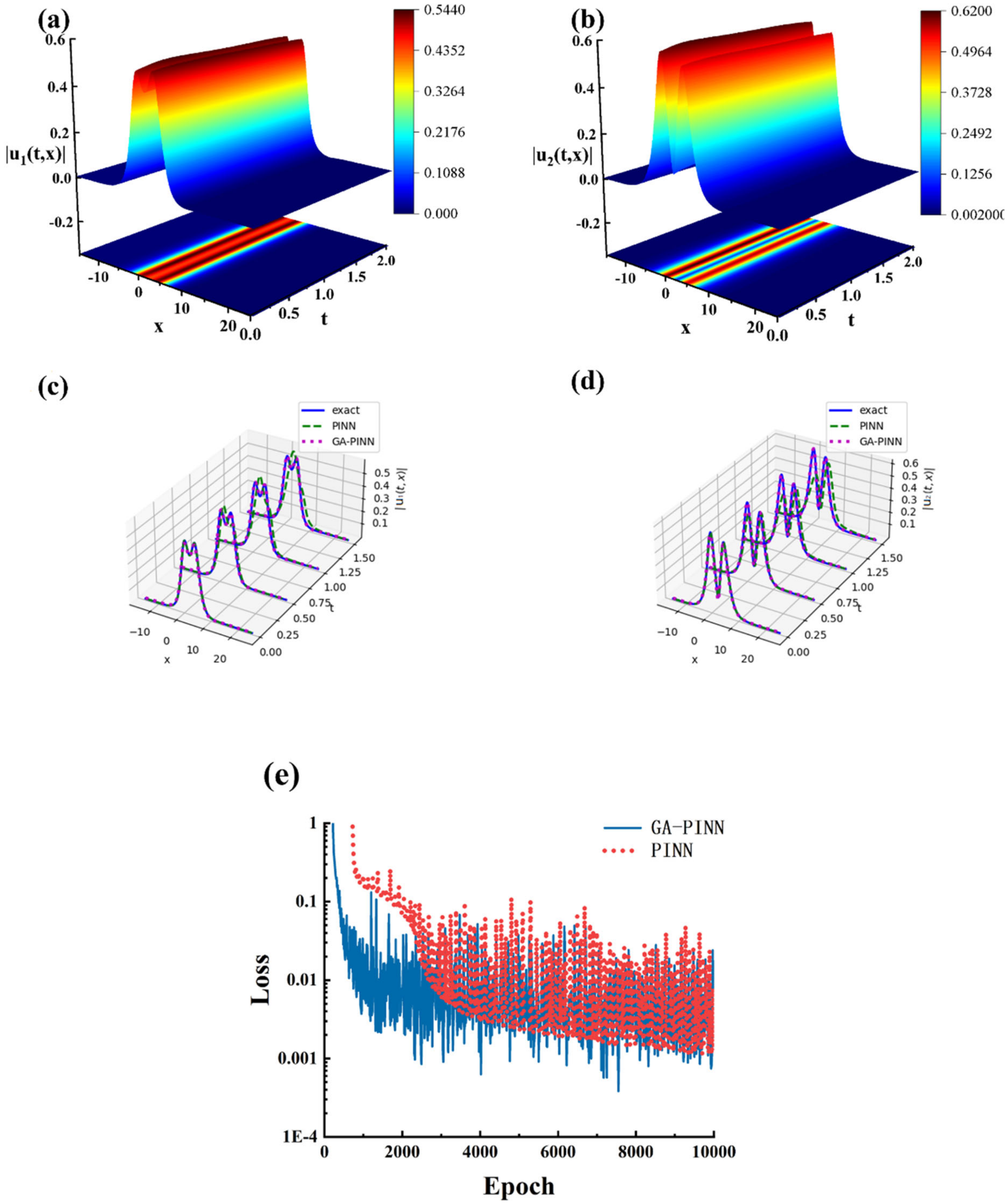

Fig. 3 Evolution of predicted nondegenerate one-soliton solution for components **a** u_1 and **b** u_2. Waterfall comparison of the evolution process of predicted and exact nondegenerate one-soliton solutions for components **c** u_1 and (d) u_2. **e** Relation curves between loss function and iteration number for different networks

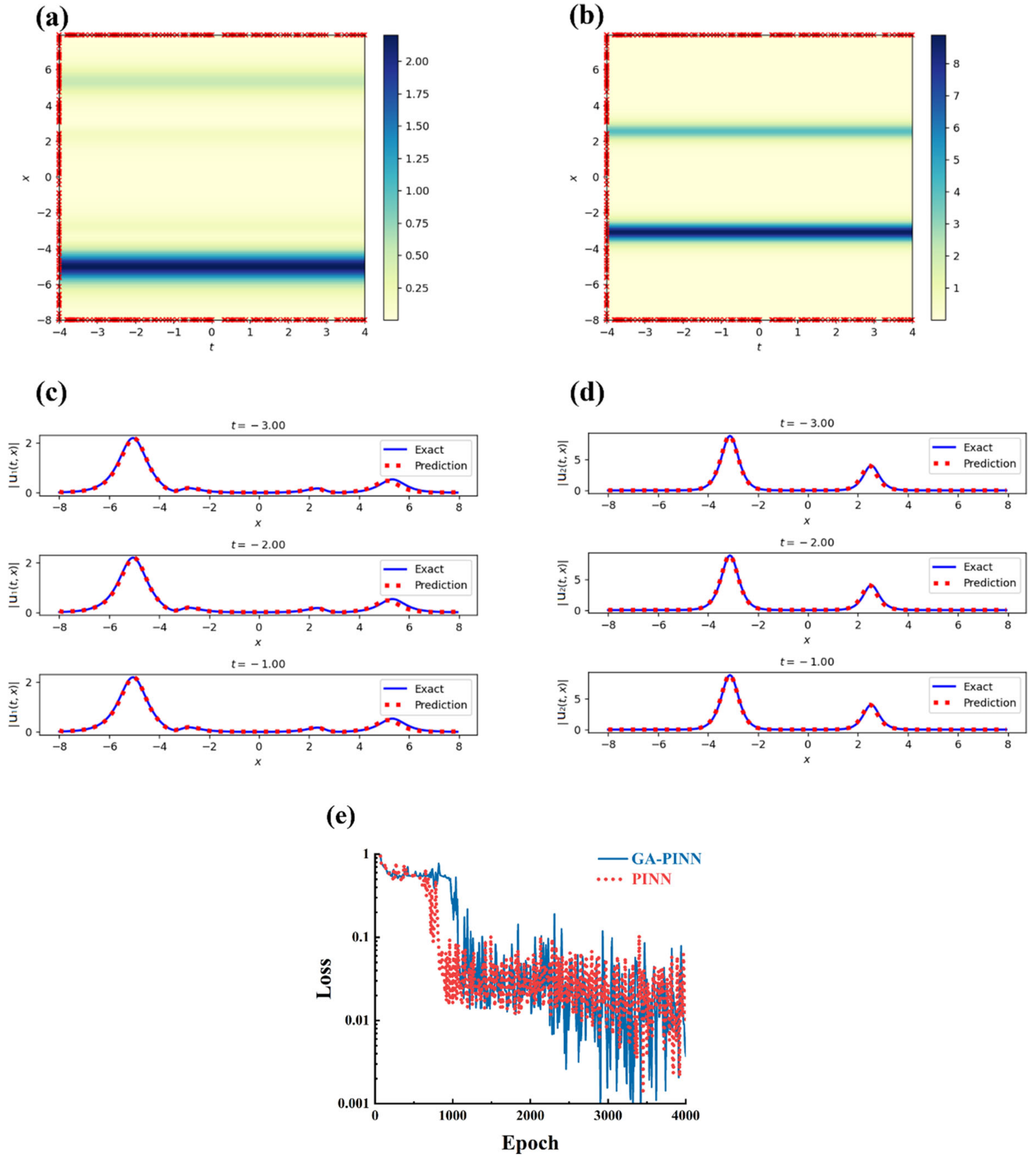

Fig. 4 Boundary and initial sampling points from exact nondegenerate double soliton solutions for components **a** u_1 and **b** u_2, cross sections of predicted and exact solutions for components **c** u_1 and **d** u_2 at different evolution time, and **e** loss functions v.s. iteration number

different parameters in Figs. 2, 3, 4, 5. Comparing the predicted values of four soliton structures from two network structures in Fig. 5e, there is no doubt that the prediction via the traditional PINN is far less accurate than the PIGAN-GP.

4 Parameter prediction of physical model

In this section, we will predict the equation parameters of 2-CMDNLSE [31], namely, treat the cubic

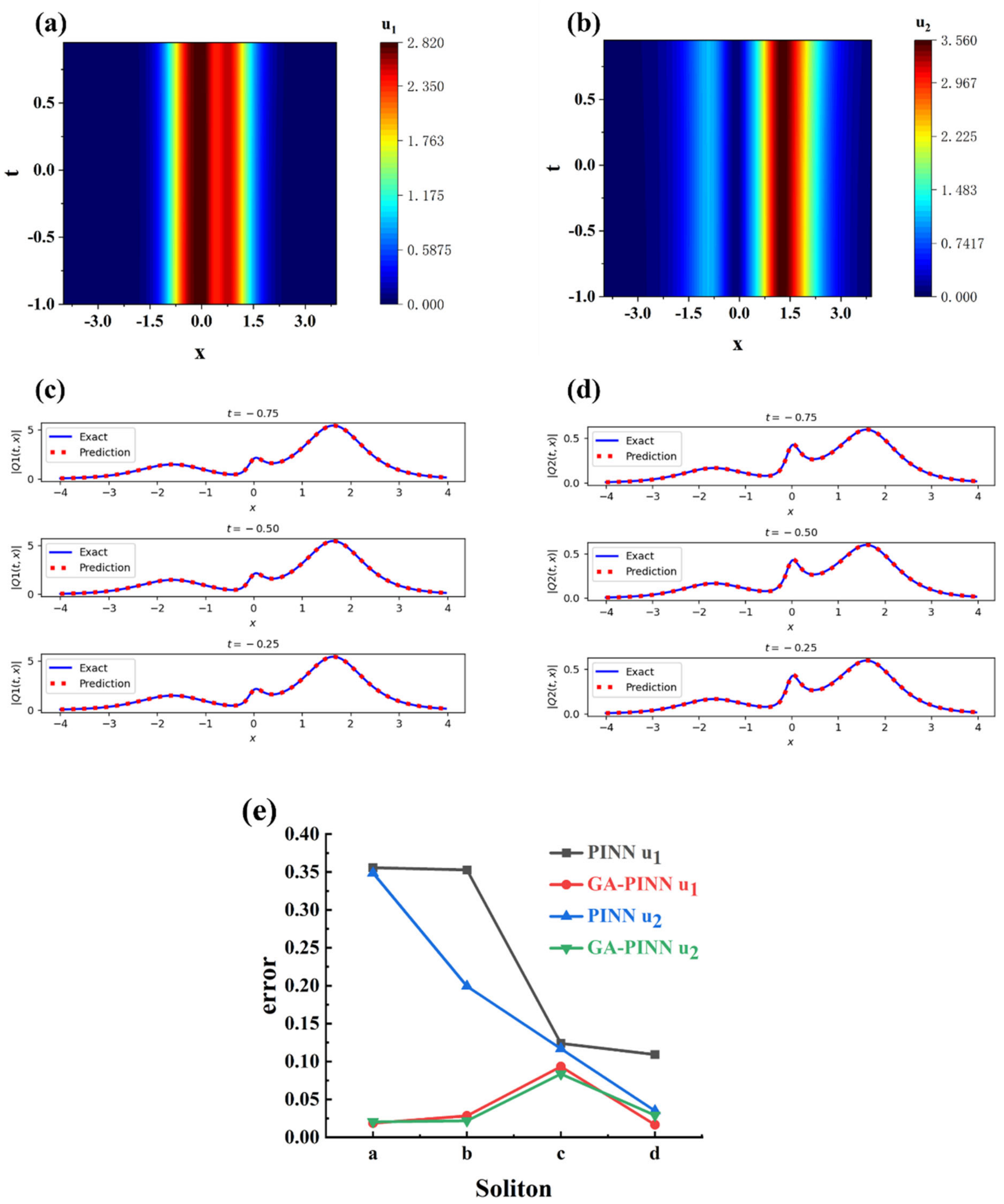

Fig. 5 Top view of predicted nondegenerate double soliton solutions for components **a** u_1 and **b** u_2. Cross sections of predicted and exact solutions for components **c** u_1 and **d** u_2 at different evolution time. **e** The comparison of absolute error for components u_1 and u_2 of nondegenerate one-soliton solitons and two-soliton solitons, where a, b, c, and d respectively correspond to cases in Figs. 2, 3, 4, 5

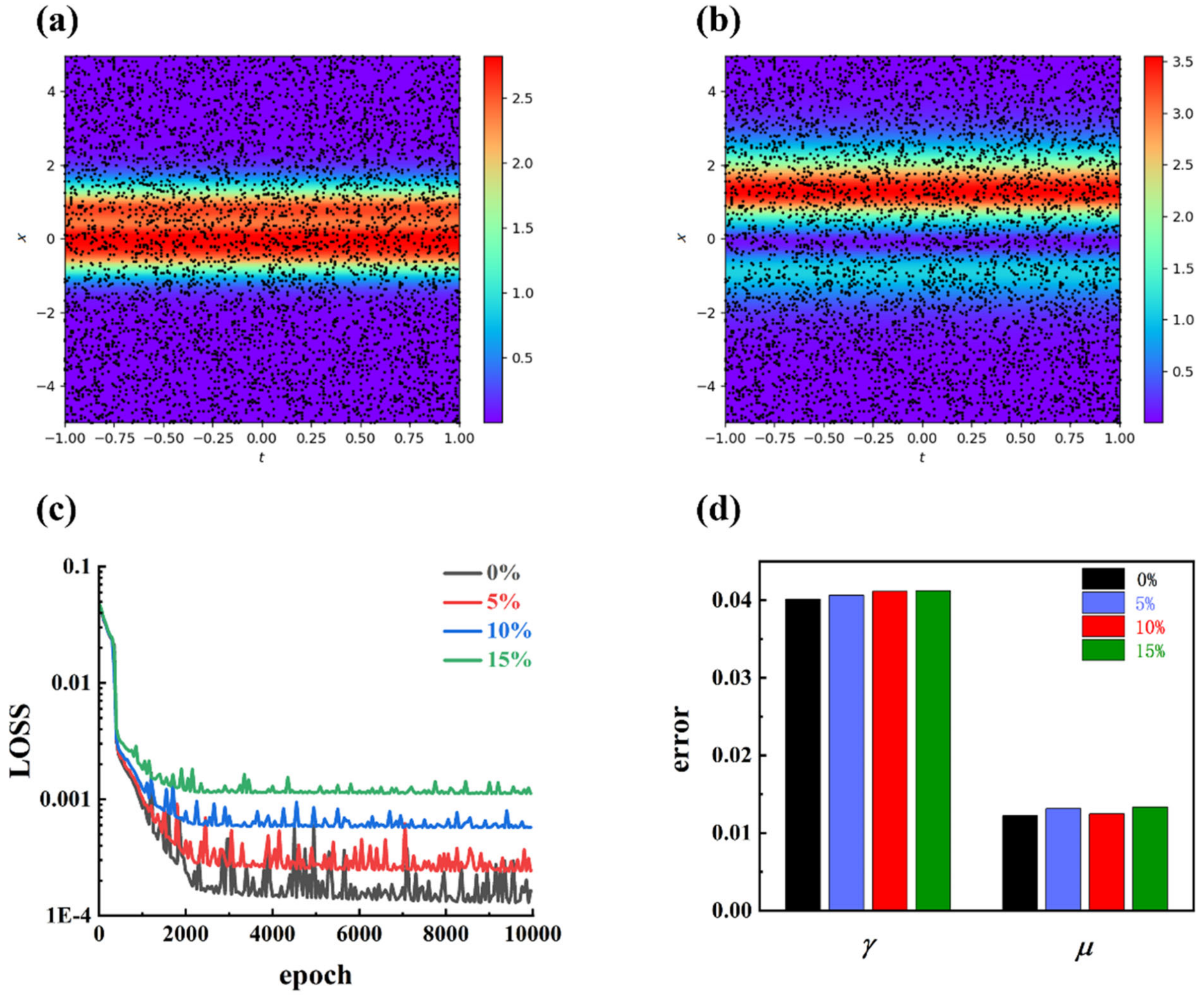

Fig. 6 Top views of predicted nondegenerate one-soliton for components **a** u_1 and **b** u_2 with sampling points. **c** Loss function and **d** error after adding different noises

Table 1 2-CMDNLSE obtained by learning unknown parameters

Item\CMDNLSE	Model	Relative error	
		μ	γ
Correct equations	$iu_{1x} + u_{1xx} + \mu(\|u_1\|^2+\|u_2\|^2)u_1 + i\gamma[(\|u_1\|^2+\|u_2\|^2)u_1]_x = 0$ $iu_{2x} + u_{2xx} + \mu(\|u_1\|^2+\|u_2\|^2)u_2 + i\gamma[(\|u_1\|^2+\|u_2\|^2)u_2]_x = 0$	–	–
Identified equations	$iu_{1x} + u_{1xx} + 0.98776293\,(\|u_1\|^2+\|u_2\|^2)u_1 + 1.0401235\,i[(\|u_1\|^2+\|u_2\|^2)u_1]_x = 0$ $iu_{2x} + u_{2xx} + 0.98776293\,(\|u_1\|^2+\|u_2\|^2)u_2 + 1.0401235\,i[(\|u_1\|^2+\|u_2\|^2)u_2]_x = 0$	0.01223707	0.0401235

nonlinear strength μ and derivative cubic nonlinear strength γ in the equation as unknown parameters.

Envelopes u_1 and u_2 are composed of real and imaginary parts as

$$u_1 = r_1 + i \cdot m_1 \tag{11}$$

$$u_2 = r_2 + i \cdot m_2 \tag{12}$$

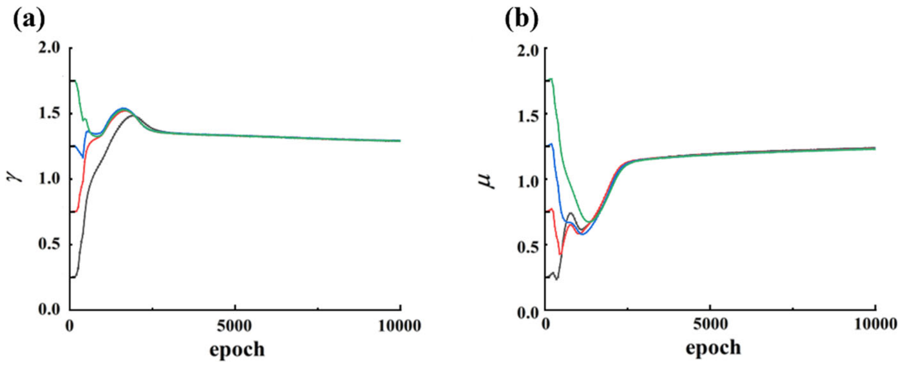

Fig. 7 Curve of variation for parameters **a** γ and **b** μ with training times for different initial values

Inserting Eqs. (11), (12) into Eq. (9), and performing the separation of real and imaginary parts, we can minimize the Mean squared error via the PINN method to obtain the approximate values of the unknown parameters. The Mean squared errors of sampling points and residuals read

$$MSE_1 = \frac{1}{N_s}\sum_{p}^{N_s}\left(\left|r_1(x^p,t^p) - r_1^p\right|^2 + \left|m_1(x^p,t^p) - m_1^p\right|^2. + \left|r_2(x^p,t^p) - r_2^p\right|^2 + \left|m_2(x^p,t^p) - m_2^p\right|^2\right) \tag{13}$$

$$MSE_2 = \frac{1}{N_s}\sum_{p}^{N_s}\left(\left|f_{r1}(x^p,t^p)\right|^2 + \left|f_{r2}(x^p,t^p)\right|^2. + \left|f_{m1}(x^p,t^p)\right|^2 + \left|f_{m2}(x^p,t^p)\right|^2\right) \tag{14}$$

$$Loss = MSE_1 + MSE_2 \tag{15}$$

The real values of unknown parameters in 2-CMDNLSE correspond to γ = 1, μ = 1. We discretize nondegenerate one-soliton by the pseudo spectral method into [256, 201] data points to obtain the dataset. Figure 6a and b display the situation of sampling points. In order to predict unknown parameters, we chose to sample points N_s = 5000 and used a neural network with the number n = 50 of neurons and a depth L = 6 of layers. Table 1 shows the training results (predicted values and relative errors) of these unknown coefficients. From this table, it can be seen that the effect of the prediction is good.

To verify the stability of the neural network, we add different interference noises during the sampling process. As shown in Fig. 6c, with the increase of noise, the rate of convergence of the loss function significantly slows down, and the overall error adds. Figure 6d shows the training errors of unknown coefficients for different interference noises. We found that the PINN can accurately predict unknown coefficients, even if the sampled data is corrupted by 15% noise, and the error is still within an acceptable range.

Next, we will change the initial values of two parameters and study changes of two parameters with training times under different initial values. As shown in Fig. 7, two parameters start to evolve from different initial values. As the training times increase, their predicted values will gradually stabilize after approximately 2000–3000 training sessions, and ultimately are same to the true values. This indicates that different initial values of parameters only affect the time that it takes for parameter prediction to reach stability. If the training level meets certain requirements, the predicted values of the parameters will be close to exact values. This also implies that PINN is excellent for the stability of parameter prediction.

5 Conclusion

In summary, using the PIGAN-GP method, we predict the evolution processes of nondegenerate single and double solitons for 2-CMDNLSE and compare the results with those via the traditional PINN. In the prediction of equation parameters, we add noise to judge the stability of the neural network. The prediction errors will increase with the increase of noise, but these errors are still within a controllable range.

The PINN is less sensitive to hyperparameters and initialization, which makes training and tuning easier. Compared with the traditional PINN, the PIGAN-GP method has improved the training accuracy by about an order of magnitude, and the time cost is basically the same as that of traditional PINN. For the prediction of some soliton structures, the PIGAN-GP method achieves higher accuracy with less time, although it requires less width and length, training times for the neural network. This is also the reason why the training time cost is lower than that of traditional PINN. This network can help us better understand the significant energy transfer characteristics and behavior between two components of each vector soliton and play a positive role in future applications in logic gate and fiber directional coupler design.

Funding National Natural Science Foundation of China(Grant Nos. 12075210 and 12261131495); the Scientific Research and Developed Fund of Zhejiang A&F University(Grant No. 2021FR0009).

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have declared that no conflict of interest exists.

References

1. Si, Z.Z., Dai, C.Q., Liu, W.: Tunable three-wavelength fiber laser and transient switching between three-wavelength soliton and q-switched mode-locked states. Chin. Phys. Lett. **41**, 020502 (2024)
2. Geng, K.L., Zhu, B.W., Cao, Q.H., Dai, C.Q.: Interference phenomenon of nondegenerate solitons for nonlocal CLL equation, Appl. Math. Lett. **145**, 108793 (2023)
3. Tan, Y., Yang, J.: Resonance- and phase-induced window sequences in vector-soliton collisions. Phys. Lett. A **288**, 309–315 (2001)
4. Fang, Y., Bo, W.B., Wang, R.R., Wang, Y.Y., Dai, C.Q.: Predicting nonlinear dynamics of optical solitons in optical fiber via the SCPINN. Chaos Solitons Fractals **165**, 112908 (2022)
5. Hou, T.Y., Wu, X.H.: A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys. **134**, 169–189 (1997)
6. Higham, D.J.: An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM Rev. **43**, 525–546 (2001)
7. 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)
8. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. SIAM J. Sci. Comput. **40**, A172–A198 (2018)
9. Raissi, M., Babaee, H., Givi, P.: Deep learning of turbulent scalar mixing. Phys Rev Fluids. **4**, 124501 (2019)
10. Wu, G.Z., Fang, Y., Wang, Y.Y., Wu, G.C., Dai, C.Q.: Predicting the dynamic process and model parameters of the vector optical solitons in birefringent fibers via the modified PINN. Chaos Solitons Fractals **152**, 111393 (2021)
11. Fang, Y., Wu, G.Z., Wen, X.K., Wang, Y.Y., Dai, C.Q.: Predicting certain vector optical solitons via the conservation-law deep-learning method. Opt. Laser Technol. **155**, 108428 (2022)
12. Bai, Y., Chaolu, T., Bilige, S.: Solving Huxley equation using an improved PINN method. Nonlinear Dyn. **105**, 3439–3450 (2021)
13. Li, J., Li, B.: Mix-training physics-informed neural networks for the rogue waves of nonlinear Schrödinger equation. Chaos Solitons Fractals **164**, 112712 (2022)
14. Lin, S., Chen, Y.: Physics-informed neural network methods based on Miura transformations and discovery of new localized wave solutions. Physica D **445**, 133629 (2023)
15. Zhu, B.W., Fang, Y., Liu, W., Dai, C.Q.: Predicting the dynamic process and model parameters of vector optical solitons under coupled higher-order effects via WL-tsPINN. Chaos Solitons Fractals **162**, 112441 (2022)
16. Hou, J., Li, Y., Ying, S.: Enhancing PINNs for solving PDEs via adaptive collocation point movement and adaptive loss weighting. Nonlinear Dyn. **111**, 15233 (2023)
17. Li, M., Xiao, J.H., Liu, W.J., Wang, P., Qin, B., Tian, B.: Mixed-type vector solitons of the N-coupled mixed derivative nonlinear Schrödinger equations from optical fibers. Phys. Rev. E **87**, 032914 (2013)
18. Goodfellow, I., Abadie, J.P., Mirza, M., Xu, B., Farley, D.W., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**, 139–144 (2020)
19. Liu, M., Wang, Z., Li, H., Wu, P., Alsaadi, F.E., Zeng, N.: AA-WGAN: attention augmented Wasserstein generative adversarial network with application to fundus retinal vessel segmentation. Comput. Biol. Med. **158**, 106874 (2023)
20. Zhu, J.Y., Park, T., Isola, P. and Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232 (2017).
21. Karras, T., Laine, S. and Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410, (2019).
22. Neal, R.M.: Annealed importance sampling. Stat. Comput. **11**, 125–139 (2001)
23. Arjovsky, M. and Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862. (2017).
24. Arjovsky, M., Chintala, S. and Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223, (2017)
25. Lü, X., Tian, B.: Vector bright soliton behaviors associated with negative coherent coupling. Phys. Rev. E **85**, 026117 (2012)

26. Raissi, M., Karniadakis, G.E.: Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. **357**, 125–141 (2018)
27. Stein, M.: Large sample properties of simulations using Latin hypercube sampling. Technometrics **29**, 143–151 (1987)
28. Xue, Y., Tong, Y., Neri, F.: An ensemble of differential evolution and Adam for training feed-forward neural networks. Inf. Sci. **608**, 453–471 (2022)
29. Geng, K.L., Mou, D.S., Dai, C.Q.: Nondegenerate solitons of 2-coupled mixed derivative nonlinear Schrödinger equations. Nonlinear Dyn. **111**, 603–617 (2023)
30. Ramakrishnan, R., Stalin, S., Lakshmanan, M.: Nondegenerate solitons and their collisions in Manakov systems. Phys. Rev. E **102**, 042212 (2020)
31. Matsuno, Y.: The bright N-soliton solution of a multi-component modified nonlinear Schrödinger equation. J. Phys. A Math. Theor. **44**, 495202 (2011)
32. Cai, Y.J., Wu, J.W., Hu, L.T., Lin, J.: Nondegenerate solitons for coupled higher-order nonlinear Schrödinger equations in optical fibers. Phys. Scr. **96**, 095212 (2021)
33. Eslami, M.: Exact traveling wave solutions to the fractional coupled nonlinear Schrodinger equations. Appl. Math. Comput. **285**, 141–148 (2016)

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