Data-Driven Ai- and Bi-Soliton of the Cylindrical Korteweg–de Vries Equation via Prior-Information Physics-Informed Neural Networks

Shifang Tian(田十方)[1], Biao Li(李彪)[1*], and Zhao Zhang(张钊)[2]

[1]*School of Mathematics and Statistics, Ningbo University, Ningbo 315211, China*
[2]*Guangdong Provincial Key Laboratory of Nanophotonic Functional Materials and Devices, South China Normal University, Guangzhou 510631, China*

(Received 9 January 2024; accepted manuscript online 4 February 2024)

By the modifying loss function MSE and training area of physics-informed neural networks (PINNs), we propose a neural networks model, namely prior-information PINNs (PIPINNs). We demonstrate the advantages of PIPINNs by simulating Ai- and Bi-soliton solutions of the cylindrical Korteweg–de Vries (cKdV) equation. Numerical experiments show that our proposed model is able not only to simulate these solitons using the cKdV equation, but also to significantly improve its simulation capability. Compared with the original PINNs, the prediction accuracy of our proposed model is improved by one to three orders of magnitude. Moreover, the accuracy of the PIPINNs is further improved by adding the restriction of conservation of energy.

DOI: 10.1088/0256-307X/41/3/030201

The cylindrical Korteweg–de Vries (cKdV) equation is a shallow water wave equation that becomes important in water wave problems due to possible circular disturbances caused by point sources such as underwater earthquakes, volcanoes, atmospheric pressure, and falling meteorites. In addition, in some basins, many quasi-cylindrical internal waves are generated due to water intrusion (for example, a large number of satellite images show internal waves generated by tides when Atlantic water enters the Mediterranean). The cKdV equation first proposed by Maxon and Viecelli,[1] and its specific format is as follows:

$$u_t + 6uu_x + u_{xxx} + \frac{u}{2\,t} = 0.$$

Soliton solutions are one of the hottest research topics in the field of mathematics. At present, more and more scholars start to study the Ai- and Bi-solitons of the cKdV equation.[2–6] Here, we first introduce the Ai-soliton:

$$\begin{aligned} u_{\rm Ai} &= 2\,[\ln H_{\rm Ai}(x,\,t)]_{xx},\\ H_{\rm Ai} &= c + \frac{\rho^2}{12\sqrt[3]{t}}\Big([Ai'(Z)]^2 - ZAi^2(Z)\Big), \end{aligned} \tag{1}$$

where $Z = (x - x_0)/(12\sqrt[3]{t})$.

Then, the Bi-soliton is as follows:

$$\begin{aligned} u_{\rm Bi} &= 2[\ln H_{\rm Bi}(x,\,t)]_{xx},\\ H_{\rm Bi} &= c + \frac{\rho^2}{12\sqrt[3]{t}}\Big(ZBi^2(Z) - \big[Bi'(Z)\big]^2\Big), \end{aligned} \tag{2}$$

where $Z = (x - x_0)/(12\sqrt[3]{t})$.

The asymptotic behaviors of Ai-solitons (1) and Bi-solitons (2) are expressed as

$$u_{\rm Ai} \sim \begin{cases} -\dfrac{\sqrt{-Z}\cos\left(\frac{4(-Z)^{3/2}}{3}\right)18^{2/3}}{18Zt^{2/3}}, & Z \to -\infty,\\ \dfrac{\rho^2 e^{-\frac{4Z^{3/2}}{3}}}{12\,t\pi}, & Z \to +\infty; \end{cases}$$

$$u_{\rm Bi} \sim \begin{cases} -\dfrac{\sqrt{-Z}\cos\left(\frac{4(-Z)^{3/2}}{3}\right)18^{2/3}}{18Zt^{2/3}}, & Z \to -\infty,\\ \dfrac{12^{1/3}}{6t^{2/3}\sqrt{Z}}, & Z \to +\infty. \end{cases}$$

In the last decade, due to its strong "learning" ability, deep learning has had a remarkable influence in many fields, such as computer vision, recommendation systems, and natural language processing.[7–10] An important reason behind its success is that the neural network used for deep learning is the key to numerically approaching complex models. The outstanding characteristics of deep learning have led to new numerical solution methods for solving complex models, such as nonlinear partial differential equations (NPDEs). In recent years, the use of deep neural networks to solve NPDEs has attracted extensive attention and some models optimized on the basis of them have emerged, greatly promoting the development of this field. In these works, Raissi *et al.*[11] proposed physics-informed neural networks (PINNs), which have attracted a great deal attention because of their good generalization ability and prediction performance in solving NPDEs. Although there have been some positive results, PINNs continue to encounter some unexpected difficulties in approaching complex potential solutions. Recently, some neural network methods based on PINNs have been proposed, featuring modified or optimized neural network structures or loss functions such as MSE.[12,13] Jagtap *et al.*[14] simulated an inverse problem in supersonic flows by using an improved PINN model. Chen and his team proposed an improved PINN, obtained the data-driven localized wave solutions of the N-double poles solutions of the nonlocal Hirota equation with nonzero boundary conditions[15] and simulated the data-driven vector localized waves of the Manakov system[16] by improved PINNs. They also introduced two-stage PINNs[17] and simulated the data-driven soliton molecule, M-shape double-peak

*Corresponding author. Email: libiao@nbu.edu.cn
© 2024 Chinese Physical Society and IOP Publishing Ltd

soliton, plateau soliton and interaction solution of the Sawada–Kotera equation, and some coupled equations. On the basis of PINNs, Li and his team proposed gradient-optimized PINNs,[18,19] mix-training PINNs and prior information mix-training PINNs,[20,21] these models further improved the accuracy of the potential solutions.

Based on the above existing research results, we modified the loss function MSE, and combined the initial sampling region I and boundary sampling region B in the original PINN model into one mix-training sampling region IB. As the prior information, the sample points in the sharp region of the equation solution are added to mix-training sampling region IB to form a new sampling region IBP and then prior information PINNs (PIPINNs) are proposed.

In this Letter, we first introduce the prior-information mix-training model PIPINNs. Then, we use PINNs and PIPINNs to simulate the Ai- and Bi-solitons of the cKdV equation. We also add the law of conservation of energy to the loss function $\mathrm{MSE}_{\mathrm{PIPINNs}}$ of PIPINNs and carry out numerical experiments. Then, by checking the numerical results, we find that PIPINNs demonstrate better performance in simulating Ai- and Bi-solitons compared with PINNs. Finally, we give the conclusion and discussion.

Model Setup. When simulating the existing research results, we find that sometimes the sampling area is too random, so many points with large errors are not selected as training points. Therefore, we consider whether it is possible to add constraint conditions for selecting training points, so that some points with large errors can be selected for training. At this time, we list the sampling points of the sharp area of the solution in advance and then add them to the previous IB training domain to generate a new region IBP. By sampling again, it will be carried out in a new area, so that the method can be further optimized. This idea employs the idea of the adaptive grid used in the finite difference method. The idea of adaptive grid is to reduce the number of grid nodes in the area with small error and densify the grid in the area with large error. The new loss function is defined as

$$\begin{aligned}\mathrm{MSE}_{\mathrm{PIPINNs}} &= \mathrm{MSf} + \mathrm{MSIBP} \\ &= \frac{1}{N_{\mathrm{f}}}\sum_{i=1}^{N_{\mathrm{f}}}[f(x^i,\, t^i)]^2 \\ &\quad + \frac{1}{N_{\mathcal{I}BP}}\sum_{i=1}^{N_{\mathcal{I}BP}}[u^i_{\mathcal{I}BP} - u(x^i_{\mathcal{I}BP},\, t^i_{\mathcal{I}BP})]^2.\end{aligned}$$

The loss function MSE is composed of two parts, namely, equation residuals MSf and loss function MSIB. Here the mixed region IB is composed of initial conditions and boundary conditions; $u^i_{\mathcal{I}BP}$ is our predicted solution, $u(x^i_{\mathcal{I}BP},\, t^i_{\mathcal{I}BP})$ is the true solution of the equation; N_{f} is the number of sampling points in the whole region, and $N_{\mathcal{I}BP}$ is the number of sampling points in the mixed region. Similarly, we order $N_{\mathcal{I}BP} = N_{\mathcal{I}} + N_{\mathcal{B}}$. In this way, a kind of prior-information mix-training PIPINNs model is generated by adding the sampling points of the sharp area of the solution to the training domain in advance. The structure of the PIPINNs model is shown in Fig. 1.

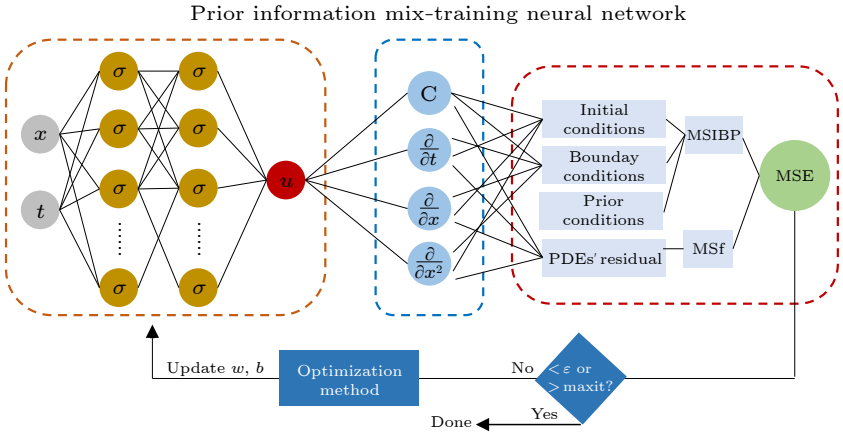

Fig. 1. Prior-information physics-informed neural network (PIPINN) model.

As shown in Fig. 1, this neural network is composed of an input layer, hidden layer, output layer, and activation function σ. The total error consists of an initial error, boundary condition error, and function residuals. The potential solution $u(x,\, t)$ obtained by this neural networks model is optimized by two optimizers, L-BFG-S and Adam. In addition, ε and *maxit* are the conditions for whether the loop continues. When one of the two conditions is met, the loop is complete and the final potential solution $u(x,\, t)$ we simulated is output.

The above sampling points are random sampling in a specific area. Finally, the L-BFG-S and Adam optimizers use the gradient-descent method to minimize the loss function MSE until it reaches convergence. The above process is what we call the *training process*.

We conduct the numerical experiments. In all cases, without loss of generality, using the deep neural networks with 4, 6 hidden layers and 32, 64, 128 neurons in six cases, we define the hyperbolic tangent function tanh as the nonlinear activation equation of the model. By default, the L-BFG-S algorithm is used to train the network in 20000 steps and then Adam optimizer with a default learning rate is used to optimize the network in 20000 steps. The weight w and b deviation values in all mod-

els are initialized with the Xavier method.[11] No other additional regularization techniques are used. It is worth noting that we use $N_{\mathrm{f}} = 10000$, $N_{\mathcal{I}} = 150$, and $N_{\mathcal{B}} = 150$ sample points, respectively, on the equation residuals, initial conditions, and boundary conditions in the test. For consistency and convenience of subsequent comparison of method performance, we use $N_{\mathcal{IBP}} = 300$ sample points on the mixed dataset from the sharp regions and Dirichlet boundary conditions.

We consider the following cKdV equation with Dirichlet boundary conditions:

$$\begin{aligned} &u_t + 6uu_x + u_{xxx} + \frac{u}{2t} = 0, \\ &u(x,\, t) = \mathcal{I}(x,\, t), \quad x \in \Omega, \\ &u(x,\, t) = \mathcal{B}(x,\, t), \quad (x,\, t) \in \partial\Omega \times T, \end{aligned} \tag{3}$$

where $u(x,\, t)$ contains variables x and t, which are space and time variables, respectively.

We conduct a numerical experiment on the Ai- and Bi-solitons of the cKdV equation. It is worth mentioning that for the accuracy of the experiment and the uniqueness of variables, the selected Ai- and Bi-solitons have the same wave crest and wave width. We show the images of these two solitons and the specific variable values of the solutions will be given in the following.

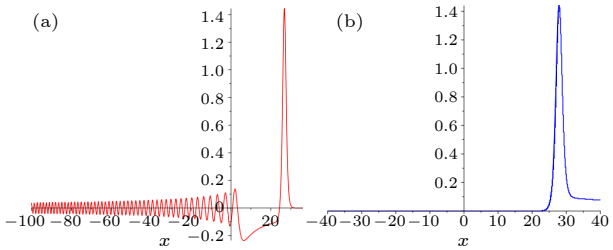

Fig. 2. (a) The Ai-soliton of cKdV Eq. (3). (b) The Bi-soliton of cKdV Eq. (3).

Firstly, we take the Ai-soliton of the cKdV equation as an example, when $c = 1$, $\rho = 10^5$, $x_0 = 10$, $T \in [1.0,\, 2.0]$, $X \in [1.1,\, 26.1]$. The three-dimensional diagrams [see Fig. 3(a)] of the Ai-soliton (1) are described in the following.

It is worth noting that we choose a neural networks with six hidden layers and 128 neurons in each layer, and add some training points in the sharp area of these solutions [see Fig. 3(b) for the new sampling area] as the prior information.

We use the PINNs and PIPINNs, combined with the training conditions and optimization methods described above, to conduct numerical experiments.

In order to find out whether the model is good or bad, we introduce a concept of relative error, that is, the relative L_2 norm between the error and the original exact solution. Consider the following condition:

$$\text{Relative error} = \frac{\|\sqrt{\text{error}_i}\|_2}{\|u(x_i,\, t)\|_2} = \frac{\sqrt{\sum_{i=1}^{N}[u_i - u(x_i,\, t)]^2}}{\sqrt{\sum_{i=1}^{N}[u(x_i,\, t)]^2}},$$

where u_i is the predictive solution of the model and $u(x_i,\, t)$ is the exact solution. Table 1 provides a more intuitive and detailed evaluation of the predictive performance of the PINNs and PIPINNs.

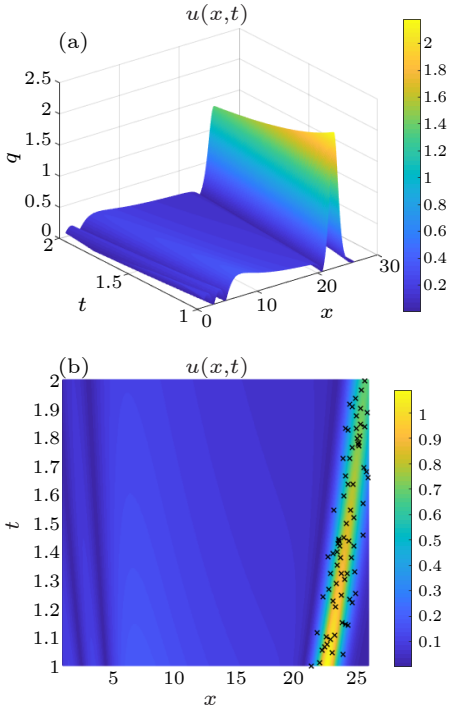

Fig. 3. (a) The three-dimensional diagram of the Ai-soliton of the cKdV equation. (b) The new sharp region of the Ai-soliton of the cKdV equation.

From Table 1, we can obtain the experimental results of the prediction performance of PINNs and PIPINNs for the Ai-soliton under all neural network conditions. Through Table 1, we find that PIPINNs are superior to PINNs when the number of neurons in the hidden layer and each layer is the same. PIPINNs not only have high accuracy, but also take less time. It is worth noting that in order to better and more clearly observe the performance of the model PIPINNs, we present the exact solution, predicted solution, and absolute error of the PIPINNs model in Fig. 4.

Table 1. Numerical results of the Ai-soliton simulated by PINNs and PIPINNs.

Neuron networks	PINNs		PIPINNs	
Neurons | Layers	$u(x,\, t)$	Time	$u(x,\, t)$	Time
32 | 4	4.3×10^{-1}	30500.1 s	1.1907×10^{-2}	20240.95 s
32 | 6	3.5×10^{-1}	35001.6 s	1.2481×10^{-2}	30052 s
64 | 4	4.5×10^{-1}	42040 s	1.4402×10^{-1}	31080 s
64 | 6	4.2×10^{-1}	50400 s	8.242×10^{-3}	35050 s
128 | 4	3.976×10^{-1}	67003.45 s	1.74×10^{-2}	50014.2 s
128 | 6	4.71×10^{-1}	80105.1 s	1.5187×10^{-2}	60303.44 s

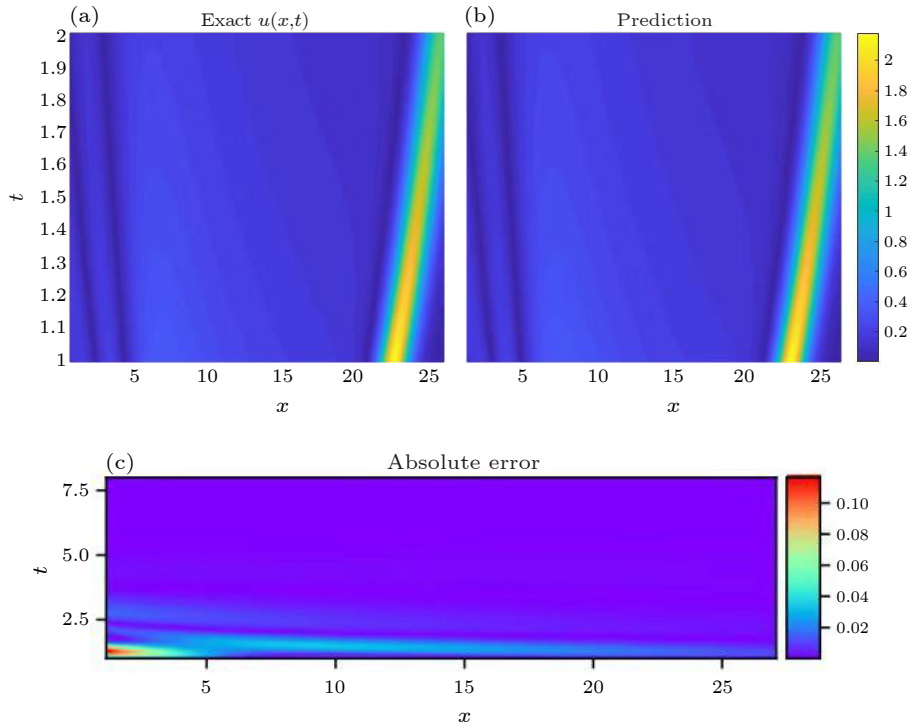

Fig. 4. (a) Exact solution of the Ai-soliton of the cKdV equation. (b) Predicted solution of the Ai-soliton of the cKdV equation. (c) Absolute error of the Ai-soliton simulated by PIPINNs (relative error is 1.5187×10^{-2}).

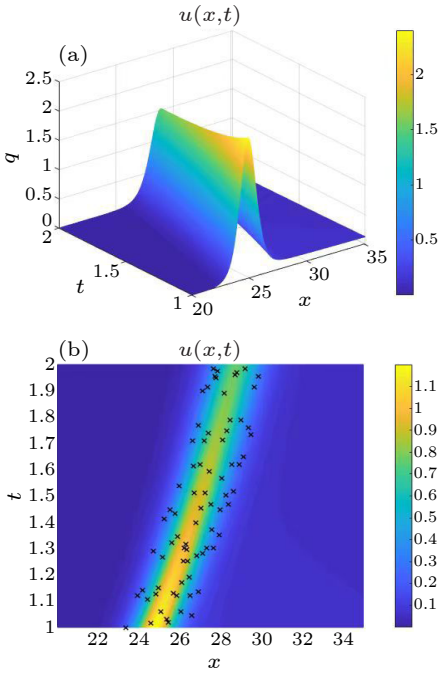

Fig. 5. (a) The three-dimensional diagram of the Bi-soliton of the cKdV equation. (b) The new sharp region of the Bi-soliton of the cKdV equation.

Then, we present the Bi-soliton of the cKdV equation when $c = 1$, $\rho = 10^{-3.8}$, $x_0 = 10$, $T \in [1.0, 2.0]$, $X \in [20.1, 35.1]$. The three-dimensional diagrams [Fig. 5(a)] of the Bi-soliton are described in the following.

To test PIPINNs more clearly, we give the sharp region of the Bi-soliton of the cKdV equation in Fig. 5(b).

We follow the same experimental method mentioned above. The experimental results for the predictive performance of PIPINNs for this Bi-soliton under all neural network conditions are listed in Table 2. From Table 2, we also find that PIPINNs are still more accurate than PINNs. It is worth noting that in order to better and more clearly observe the performance of the model PIPINNs, we present the exact solution, predicted solution, and absolute error of the PIPINNs model in Fig. 6.

According to the numerical results, we find that the performance of the PIPINNs model is still very good and the accuracy is improved again. In terms of time consumption, it is still better than that of PINNs. In conclusion, when simulating Ai- and Bi-solitons, the accuracy and performance of PIPINNs are better than PINNs.

In order to make the numerical results simulated by PIPINNs more accurate, we add the restrictions to loss function $\mathrm{MSE}_{\mathrm{PIPINNs}}$. We introduce the concept of the law of conservation of energy.

The law of conservation of energy, the first law of thermodynamics, states that the total energy in a closed (isolated) system remains constant. In general, the total energy is no longer just the sum of kinetic energy and potential energy, but the total amount of rest energy (inherent energy), kinetic energy, and potential energy. The law of conservation of energy can be stated as follows: a change in the total energy of a system can only be equal to the amount of energy passing into or out of the system. The total energy is the sum of the mechanical energy, heat energy, and any form of internal energy other than heat energy of the system.

In our work, we cite the law of conservation of energy as follows:

$$\left[\frac{1}{2}t^d u^2\right]_t + \left[t^d\left(\frac{\alpha}{3}u^3 - \frac{\beta}{2}u_x^2 + \beta u u_{xx}\right)\right]_x = 0, \quad (4)$$

where $d = 1$, $\alpha = 6$, $\beta = 1$.

Table 2. Numerical results of the Bi-soliton of the cKdV equation simulated by PINNs and PIPINNs.

Neuron networks	PINNs		PIPINNs	
Neurons | Layers	$u(x,\,t)$	Time	$u(x,\,t)$	Time
32 | 4	1.48×10^{-1}	29084 s	2.27×10^{-2}	21024 s
32 | 6	3.4×10^{-1}	38640.2 s	1.24×10^{-2}	20744.2 s
64 | 4	2.34×10^{-1}	40950.4 s	2.14×10^{-2}	30211 s
64 | 6	6.54×10^{-1}	53201.5 s	6.34×10^{-3}	37044 s
128 | 4	5.451×10^{-1}	61440 s	9.9875×10^{-4}	43214.2 s
128 | 6	3.5×10^{-1}	71643.23 s	7.54×10^{-4}	51303.44 s

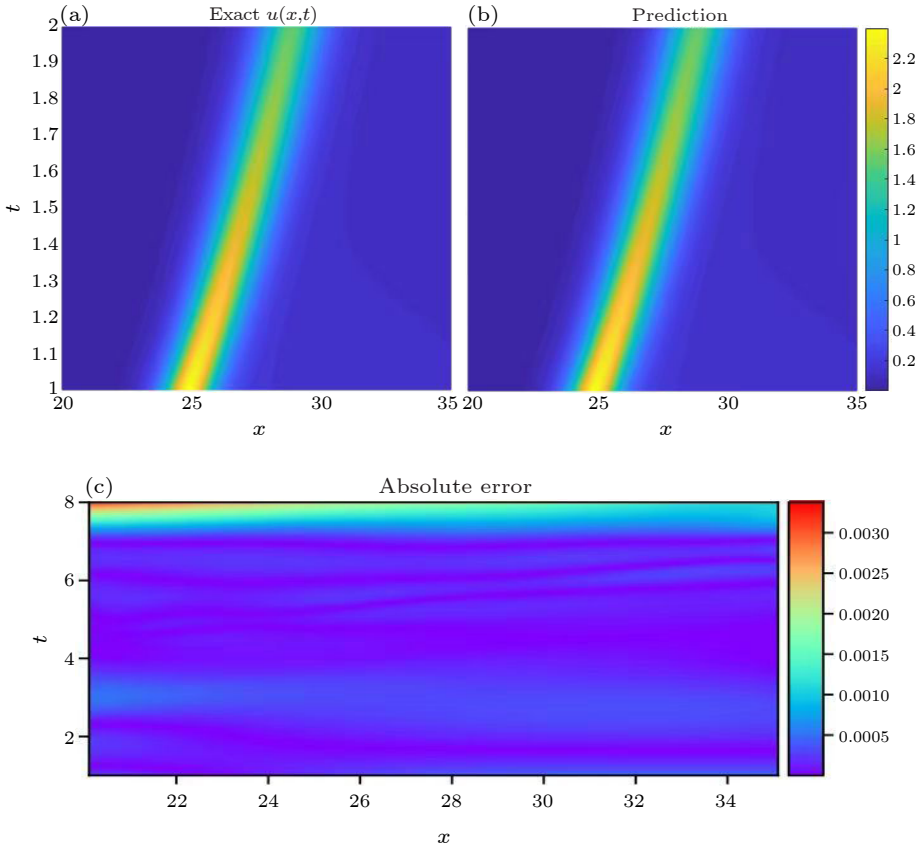

Fig. 6. (a) The exact solution to the Bi-soliton of the cKdV equation. (b) The predicted solution of the Bi-soliton of the cKdV equation by PIPINNs. (c) The absolute errors of the Bi-soliton of the cKdV equation by PIPINNs in the new sampling area, where the relative error is 7.54×10^{-4}.

Table 3. Numerical results of the Ai-soliton of the cKdV equation simulated by the new PIPINNs.

Neuron networks	PIPINNs		New PIPINNs	
Neurons | Layers	$u(x,\,t)$	Time	$u(x,\,t)$	Time
128 | 6	1.5187×10^{-2}	60303.44 s	9.8173×10^{-3}	68550 s

Table 4. Numerical results of the Bi-soliton of the cKdV equation simulated by the new PIPINNs.

Neuron networks	PIPINNs		New PIPINNs	
Neurons | Layers	$u(x,\,t)$	Time	$u(x,\,t)$	Time
128 | 6	7.54×10^{-4}	51303.44 s	2.31×10^{-4}	62333 s

To facilitate numerical experiments, we set

$$c=\left[\frac{1}{2}t^d u^2\right]_t+\left[t^d\left(\frac{\alpha}{3}u^3-\frac{\beta}{2}u_x^2+\beta u u_{xx}\right)\right]_x,$$

after adding conservation law (4), the new loss function $\mathrm{MSE}_{\mathrm{PIPINNs}}$ becomes

$$\begin{aligned}\mathrm{MSE}_{\mathrm{PIPINNs}}&=\mathrm{MSf}+\mathrm{MSc}+\mathrm{MSIBP}\\&=\frac{1}{N_{\mathrm{f}}}\sum_{i=1}^{N_{\mathrm{f}}}[f(x^i,\,t^i)]^2+\frac{1}{N_c}\sum_{i=1}^{N_c}[c(x^i,\,t^i)]^2\\&\quad+\frac{1}{N_{\mathcal{IBP}}}\sum_{i=1}^{N_{\mathcal{IBP}}}[u^i_{\mathcal{IBP}}-u(x^i_{\mathcal{IBP}},\,t^i_{\mathcal{IBP}})]^2.\end{aligned}$$

We use PIPINNs with the law of conservation of energy to carry out numerical experiments on the Ai- and Bi-solitons of the cKdV equation, respectively. Through numerical experiments, we present the results in the following.

From Tables 3 and 4, it is evident that when simulating the Ai- to the Bi-solitons of the cKdV equation, the PIPINNs with the conservation of energy law are more accurate than the original PIPINNs, and the new PIPINNs can obtain better accuracy over a slightly longer time than the original PIPINNs, which is very meaningful.

In summary, in order to obtain a more accurate solution, we use the adaptive search algorithm to find the sample points of the sharp region of the solution. Then we add this prior information to the neural networks and combined random points on the Dirichlet boundary conditions for mix-training, and propose the PIPINN model. Taking the Ai- and Bi-solitons of the cKdV equation as examples, numerical experiments show that compared with the original PINNs, PIPINNs can not only simulate these solitons, but also have excellent accuracy. By adding the law of conservation of energy to the loss function $\mathrm{MSE}_{\mathrm{PMTPINNs}}$ of PIPINNs, the precision of the numerical results is improved again. It is significant that through noise experiments, we also prove that the PIPINNs have good robustness.

PIPINNs provide new insights for developing other new deep learning models in the future. However, there are still problems to explore, e.g., whether we can improve the accuracy of the model again by changing the training area and loss function or modifying the neural networks' structure, and whether these models can simulate other nonlinear integro-differential equations comprise our future work. Next, we will continue to study various neural network models and apply them to simulate complex equations.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant Nos. 12175111 and 12235007), and the K.C. Wong Magna Fund in Ningbo University.

References

[1] Maxon S and Viecelli J 1974 *Phys. Fluids* **17** 1614
[2] Stepanyants Y A 1981 *Wave Motion* **3** 335
[3] Nakamura A and Chen H H 1981 *J. Phys. Soc. Jpn.* **50** 711
[4] Hu W C, Ren J L, and Stepanyants Y 2023 *Symmetry* **15** 413
[5] Hu W C, Zhang Z, Guo Q, and Stepanyants Y 2024 *Chaos* **34** 013138
[6] Zhang Z, Hu W C, Guo Q, and Stepanyants Y 2024 *Chaos* **34** 013132
[7] He K M, Zhang X Y, Ren S Q, and Sun J 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016, pp 770–778
[8] van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, and Kavukcuoglu K 2016 arXiv:1609.03499 [cs.SD]
[9] Heaton J, Goodfellow I, Bengio Y, and Courville A 2018 *Genet. Program. Evolvable Mach.* **19** 305
[10] Alipanahi B, Delong A, Weirauch M T, and Frey B J 2015 *Nat. Biotechnol.* **33** 831
[11] Raissi M and Karniadakis G E 2018 *J. Comput. Phys.* **357** 125
[12] Lorin E and Yang X 2022 *Comput. Phys. Commun.* **280** 108474
[13] Bihlo A and Popovych R O 2022 *J. Comput. Phys.* **456** 111024
[14] Jagtap A D, Mao Z P, Adams N, and Karniadakis G E 2022 *J. Comput. Phys.* **466** 111402
[15] Peng W Q and Chen Y 2022 *Physica D* **435** 133274
[16] Pu J C and Chen Y 2022 *Chaos Solitons Fractals* **160** 112182
[17] Lin S N and Chen Y 2022 *J. Comput. Phys.* **457** 111053
[18] Tian S F and Li B 2023 *Acta Phys. Sin.* **72** 100202 (in Chinese)
[19] Li J H and Li B 2022 *Chaos Solitons Fractals* **164** 112712
[20] Tian S F, Niu Z J, and Li B 2023 *Nonlinear Dyn.* **111** 16467
[21] Tian S F, Cao C C, and Li B 2023 *Results Phys.* **52** 106842