Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(11): 33263–33285.
DOI: 10.3934/math.20241587
Received: 05 August 2024
Revised: 16 October 2024
Accepted: 23 October 2024
Published: 21 November 2024

*Research article*

# The data-driven localized wave solutions of KdV-type equations via physics-informed neural networks with a priori information

**Zhi-Ying Feng**[1], **Xiang-Hua Meng**[1,*] **and Xiao-Ge Xu**[2]

[1] School of Applied Science, Beijing Information Science and Technology University, Beijing, 100192, China

[2] College of Science and Technology, Beijing Open University, Beijing, 100081, China

* **Correspondence:** Email: xhmeng@bistu.edu.cn.

**Abstract:** In the application of physics-informed neural networks (PINNs) for solutions of partial differential equations, the optimizer may fall into a bad local optimal solution during the training of the network. In this case, the shape of the desired solution may deviate from that of the real solution. To address this problem, we have combined the priori information and knowledge transfer with PINNs. The physics-informed neural networks with a priori information (pr-PINNs) were introduced here, which allow the optimizer to converge to a better solution, improve the training accuracy, and reduce the training time. For the experimental examples, different kinds of localized wave solutions for several types of Korteweg-de Vries (KdV) equations were solved using pr-PINNs. Multi-soliton solutions of the KdV equation, multi-soliton and lump solutions of the (2+1)-dimensional KdV equation, and higher-order rational solutions of the combined KdV-mKdV equation have been solved by pr-PINNs. By comparing the results of pr-PINNs with PINNs under the same configuration, pr-PINNs show higher accuracy and lower cost in solving different solutions of nonlinear evolution equations due to the combination of the priori information with PINNs, which enables the neural network to capture the characteristics of the solution during training. The good performance of the proposed method will have important potential application value for the solutions of real-world problems.

**Keywords:** pr-PINNs; data-driven; KdV-type equations; localized wave solutions
**Mathematics Subject Classification:** 35Q35, 35Q53, 68T07

## 1. Introduction

With the development of artificial intelligence, deep learning methods have been developed as a powerful tool for solving partial differential equations (PDEs) [1, 2]. The physics-informed neural networks (PINNs) approach proposed in 2019 has successfully demonstrated its capabilities for solving

PDEs through several classical cases [3,4]. The PINNs incorporate PDE residuals into the loss function to satisfy physical laws, requiring minimal labeled data by leveraging known equations. They can solve forward problems and inverse problems without a mesh, using automatic differentiation to approximate differential operators. However, in some practical applications of solving PDEs by PINNs, problems arise such as the deviation of the simulated results from the real solution, poor solution accuracy, and excessive training time, so researchers proposed different improved methods for specific problems. Researchers utilized curriculum regularization and posed a sequence-to-sequence learning task to avoid the network failing to learn relevant physical phenomena of the problems [5]. Combining with the weighting strategy, generative adversarial PINNs were proposed to improve the training efficiency of PINNs [6]. A conservative PINN [7] was introduced for models governed by conservation laws. The conservative PINN enhanced the computational efficiency by dividing the solution domain into multiple subdomains, each of which was embedded in the corresponding conservation law and conducted parallelly by the respective neural network. Extended PINNs were proposed [8] selecting neural networks based on the complexity of different domains. The complex subdomains employed deep neural networks, while the simpler and smoother subdomains utilized shallow neural networks. To improve the convergence rate and solution accuracy, the PINNs were combined with transfer learning used in the field related to wavefield solutions solving [9–11]. Researchers used hash encoding for better input representation in the network [12]. GaberPINN was proposed utilizing a Gabor multiplicative activation function [13]. Gabor was also used as a basis function in PINN [14].

The PINNs and their variants have been applied in a wide range of fields [15]. Notably, PINNs have been extensively utilized for solving nonlinear PDEs [16]. The KdV equation [17] and the mKdV equation [18] are important nonlinear partial differential equations that are extensively utilized in various physical systems. In certain scenarios, these equations can be extended to higher-dimensional equations and combined to more accurately depict complex physical phenomena. Initially proposed in the context of water wave theory, the (2+1)-dimensional KdV equation can describe the propagation of solitary waves in shallow water in two-dimensional spaces. The combined KdV-mKdV equation serves as a crucial model for describing nonlinear wave phenomena and finds broad applications across multiple disciplines, including solid-state physics, plasma physics, fluid dynamics, and quantum field theory. These KdV-type equations, which are the focus of this paper, have a variety of meaningful localized wave solutions which can describe many nonlinear phenomena in nature. Different types of localized wave solutions including soliton solutions [19, 20], lump solutions [21], and rational solutions [22] for nonlinear evolution equations are current research hotspots in integrable system theory. For the application of deep learning in integrable systems, PINNs were applied to integrable systems to solve both second-order and third-order equations, see [23,24]. PINNs were combined with Bäcklund transformations for solving nonlinear evolution equations [25]. The data-driven rogue waves of the Hirota equation were investigated using the mix-training PINNs approach [26, 27]. The initial value problem for the sine-Gordon equation and the Korteweg-de Vries (KdV) equation were studied to verify the convergence conditions for neural tangent kernels [28]. These advancements highlight wide applications of PINNs for solving nonlinear evolution equations.

In several real problems, a situation in which information is known only partially or fuzzily is sometimes encountered. During the PINNs training process, the optimizer may fall into a bad local optimal solution [29] and the PINNs method cannot achieve the desired simulation results. To address this, an improved pr-PINNs method is proposed and applied to simulate solutions of various KdV-type equations. In the proposed method, two networks are used for training. The first network with prior information is employed acting as a pre-trained model and the weights are transferred to the second network. Through knowledge transfer, the second network of pr-PINNs has more suitable initialization parameters for the corresponding problem. The pr-PINNs can capture certain features of the solution more rapidly during the training process and reduce the likelihood of falling into bad local optimal solutions. Incorporating a priori information and utilizing knowledge transfer, the proposed method enables the network to better capture the characteristics of various localized wave solutions and thereby training accuracy can be improved and training time can be shortened. Compared with the classical PINNs method, pr-PINNs have enhanced training performance and achieved better training outcomes. This information is called a priori information and is used for network training. The code used in this paper is written based on PyTorch 2.1.2 and trained with GPU P100. As application experimental examples, different kinds of localized wave solutions for the KdV equation, a higher-dimensional KdV equation, and the combined KdV-mKdV equation are solved using pr-PINNs and PINNs, respectively. By a comparison of the computation results, pr-PINNs show higher accuracy and less training time than PINNs.

The structure of this paper is as follows: In Section 2, the structural description of pr-PINNs is given. In Section 3, the simulated multi-soliton solutions of the KdV equation are derived by using PINNs and pr-PINNs, and the results are analyzed. Section 4 gives the prediction of multi-soliton and lump solutions for the (2+1)-dimensional KdV equation using PINNs and pr-PINNs, as well as an analysis of the results. Section 5 shows the performance of pr-PINNs in solving higher-order rational solutions of the combined KdV-mKdV equation. In Section 6, the conclusions and discussions are presented.

## 2. The basic concepts of physics-informed neural networks with prior information

To better simulate various kinds of localized wave solutions such as the soliton soutions and rational solutions for different KdV-type equations including the KdV equation, higher-dimensional KdV equation, and KdV-mKdV equation with quadric and cubic nonlinear terms, the pr-PINNs method is proposed with a two-part neural network. The nonlinear PDE in general form can be written as

$$u_t + N_u(u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, ...) = 0. \tag{2.1}$$

The residual can be defined as

$$f := u_t + N_u(u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, ...). \tag{2.2}$$

The PINNs are structured as multilayer perceptron networks, with the inputs given according to the dimensionality of the problem. The number of layers and neurons in the intermediate network is determined based on user requirements. PINNs leverage the embedding of physical laws into the loss function of neural networks to learn complex relationships between inputs and outputs. During the training process, the networks fit not only initial and boundary condition constraints but also the physical laws of PDEs.

In the proposed pr-PINNs method, the first network is purely data-driven and the weights are transferred to the second network, while the second network is both data-driven and physical-driven. In order to obtain appropriate initial parameters including weights and the accelerate network convergence speed, a data-driven pre-training is conducted before PINNs training. In the pre-training, the discrete values based on the solution after Gaussian blur are used as training data and are called prior information. The introduction of prior information enables the network to capture the characteristics of the solution more quickly, thereby providing appropriate initial weights and other parameters for the training of the second network. For the pre-training in the first network, the mean squared error (MSE) is selected as the loss function, which is defined as

$$\mathrm{MSE}_1 = \mathrm{MSE}_u \doteq \frac{1}{N_u}\sum_{i=1}^{N_u}\left|u\left(t_u^i, x_u^i\right) - u^i\right|^2. \tag{2.3}$$

Here $\mathrm{MSE}_1$ represents the loss function of the first network in the gray box (seen in Figure 1) and $\mathrm{MSE}_u$ stands for the data-driven loss. $\left\{t_u^i, x_u^i, u^i\right\}_{i=1}^{N_u}$ defines the corresponding training data points and values. $u(t_u^i, x_u^i)$ is the discrete data points extracted from the analytical solution after Gaussian blurring. The Gaussian blurring process is given by the convolution of the data set $u(t_u^i, x_u^i)$ with a Gaussian kernel obtained by the normal distribution

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2}^N} e^{-r^2/(2\sigma^2)}, \tag{2.4}$$

where $r$ is the blurring radius. For a two-dimensional case, $r^2 = x^2 + y^2$ with $x$ as the distance from the origin in the horizontal axis and $y$ as the distance from the origin in the vertical axis. The standard deviation of the Gaussian distribution $\sigma$ is set to 10 in this paper. The output of $G(r)$ is the Gaussian kernel [30]. The solution after Gaussian blurring becomes smoother and has some fuzzy features of the original exact analytical solution. These datasets with fuzzy features are used for pre-training of the first network. The weights obtained from pre-training are transferred to the second network, so that the PINNs obtain more suitable initial parameters and converge faster.

For the second network in the green box (seen in Figure 1), the loss function $\mathrm{MSE}_2$ is defined as a sum of two parts,

$$\mathrm{MSE}_2 = \mathrm{MSE}_{ibu} + \mathrm{MSE}_f, \tag{2.5}$$

$$\mathrm{MSE}_{ibu} \doteq \frac{1}{N_u}\sum_{i=1}^{N_u}\left|u\left(t_{ibu}^i, x_{ibu}^i\right) - u_{ibu}^i\right|^2, \tag{2.6}$$

$$\mathrm{MSE}_f \doteq \frac{1}{N_f}\sum_{i=1}^{N_f}\left|f\left(t_f^i, x_f^i\right)\right|^2, \tag{2.7}$$

where $\mathrm{MSE}_{ibu}$ represents the loss of the data-driven part at the initial and boundary value points, and $\mathrm{MSE}_f$ represents the loss of the physical-driven part at the internal configuration points in the solution domain. $\left\{t_{ibu}^i, x_{ibu}^i, u_{ibu}^i\right\}_{i=1}^{N_u}$ denotes the initial and boundary training data and corresponding values. $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ represents the collocation points in the solution domain, while $f(t_f^i, x_f^i)$ denotes the PDE residuals. During training, the optimization objective is to minimize the initial and boundary loss and the PDE loss. When the loss between the network predictions and the true values is close to zero for the initial and boundary conditions, and the PDE loss is also close to zero for the internal configuration points, the network is considered to have been successfully trained. Thus the predicted solution $u_{NN}(x, t)$ from the neural network can be used to approximate the true solution $u(x, t)$ of the problem. Hence, optimizing the loss function becomes the primary objective, which effectively transforms the PDE solving problem into a loss minimization task.

The algorithmic flow of pr-PINNs is shown in Figure 1. The first step of the training process includes a priori information and it is shown in the gray box, within which the yellow box shows the structure of the fully connected neural network, and the expression of loss is given by $\mathrm{MSE}_1$. The dual-driven model consisting of the physical constraint given by the PDE and the data constraint of initial and boundary conditions is shown in the green box, where the yellow box is part of the fully connected neural network, and the blue box is the PDE information obtained via automatic differentiation, i.e., the physical constraint part. The loss in the green box is expressed by $\mathrm{MSE}_2$. In the networks, we sample the training data using latin hypercube sampling (LHS), and initialize using Xavier initialization with the activation function in both networks given by the Tanh function. After completing the first step of network training in the gray box, the knowledge is transferred to the second network in the green box.

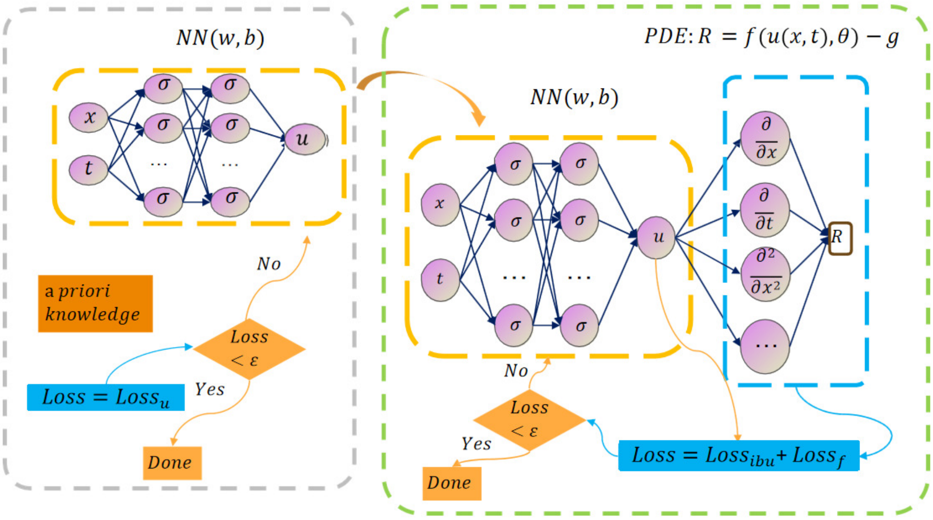


**Figure 1.** Network structure diagram of pr-PINNs.

## 3. Soliton solutions of the KdV equation

The KdV equation was originally derived in the study of nonviscous incompressible ideal fluids. Its application is particularly prominent in the analysis of surface and internal waves in shallow water. The KdV equation is an important model for the study of weakly nonlinear long waves, which arise from the intricate interplay between nonlinear and dispersion effects [31]. The basic form of the KdV equation can be expressed as

$$u_t + 6uu_x + u_{xxx} = 0. \tag{3.1}$$

In this section, the one-, two-, and three-soliton solutions of the KdV equation are simulated using pr-PINNs as well as PINNs where both have 5 layers and the number of neurons in each layer is 100. To simulate the soliton solutions for the KdV equation, the data points reflecting the basic shape characteristics of the soliton solution after Gaussian blurring [31] are considered as prior information and the training spatial-temporal region is $(x,t) \in [-10,10] \times [-10,10]$. For both the PINNs and pr-PINNs methods, the L-BFGS optimizer is used and the termination conditions of the optimizer are set as follows: The maximum number of iterations is 50,000, the maximum number of function evaluations is 50,000, the size of the history record is 50, and the gradient tolerance is $1 \times 10^{-5}$. The randomly sampled collocation points number for the initial and boundary values is 100 and the randomly sampled collocation points number inside the solution domain is 20,000 for both PINNs and pr-PINNs methods.

The multi-soliton solution of the KdV equation has been derived using the Hirota bilinear method [32]. To simulate the one-soliton solution for the KdV equation using pr-PINNs as well as PINNs, the initial condition is taken as

$$u(x,0) = \frac{1}{2}\operatorname{sech}^2\frac{x}{2}, \;\; x \in [-10,10],$$

and boundary conditions are selected as follows:

$$u(-10,t) = \frac{1}{2}\operatorname{sech}^2(5+\frac{t}{2}) \;, \;\; u(10,t) = \frac{1}{2}\operatorname{sech}^2(5-\frac{t}{2}) \;, \;\; t \in [-10,10].$$

It can be verified that the KdV equation has the followings two-soliton solution [32]:

$$u_{2ss}(x,t) = \frac{441\left(441\eta_1 + 784\eta_2 + 98\eta_1\eta_2 + 16\eta_2\eta_1^2 + 9\eta_2^2\eta_1\right)}{50(49\eta_1 + 49\eta_2 + \eta_1\eta_2 + +49)^2}, \quad \eta_1 = \mathrm{e}^{\frac{9x}{10}-\frac{729t}{1{,}000}}, \quad \eta_2 = \mathrm{e}^{\frac{6x}{5}-\frac{216t}{125}}. \quad (3.2)$$

To simulate a two-soliton solution of the KdV equation, the initial condition is given by

$$u(x,0) = u_{2ss}(x,0) \;, \;\; x \in [-10,10],$$

and boundary conditions are given as

$$u(-10,t) = u_{2ss}(-10,t), u(10,t) = u_{2ss}(10,t), \;\; t \in [-10,10].$$

The three-soliton solution for the KdV equation is [32]

$$\begin{aligned} u_{3ss}(x,t) \;\; = \;\; & -\frac{2\left(\frac{\eta_2\eta_1}{55} + \frac{37\eta_2\eta_3\eta_1}{2{,}450{,}250} + \frac{\eta_3\eta_1}{10} + \eta_1 + \frac{6\eta_2}{5} + \frac{\eta_2\eta_3}{30} + \frac{3\eta_3}{2}\right)^2}{\left(\frac{\eta_2\eta_1}{121} + \frac{\eta_2\eta_3\eta_1}{245{,}025} + \frac{\eta_3\eta_1}{25} + \eta_1 + \eta_2 + \frac{\eta_2\eta_3}{81} + \eta_3 + 1\right)^2} \\ & + \frac{2\left(\frac{\eta_2\eta_1}{25} + \frac{1{,}369\eta_2\eta_3\eta_1}{24{,}502{,}500} + \frac{\eta_3\eta_1}{4} + \eta_1 + \frac{36\eta_2}{25} + \frac{9\eta_2\eta_3}{100} + \frac{9\eta_3}{4}\right)}{\frac{\eta_2\eta_1}{121} + \frac{\eta_2\eta_3\eta_1}{245{,}025} + \frac{\eta_3\eta_1}{25} + \eta_1 + \eta_2 + \frac{\eta_2\eta_3}{81} + \eta_3 + 1}, \\ & \eta_1 = e^{x-t}, \qquad \eta_2 = e^{\frac{6x}{5}-\frac{216t}{125}}, \qquad \eta_3 = e^{\frac{3x}{2}-\frac{27t}{8}}. \end{aligned}$$

To simulate the three-soliton solution for KdV equation, the initial condition is given as

$$u(x,0) = u_{3ss}(x,0) \;, \;\; x \in [-10,10],$$

and boundary conditions are

$$u(-10,t) = u_{3ss}(-10,t), u(10,t) = u_{3ss}(10,t), \;\; t \in [-10,10].$$

The PINNs and pr-PINNs methods are used to train the one-soliton, two-soliton, and three-soliton solutions of the KdV equation, respectively, under the same parameters of the network. The compared simulated results using PINNs and pr-PINNs are shown in Figures 2–5 and Table 1. The corresponding loss curve plots are shown in Figure 4. We can see that the pr-PINNs method achieves higher accuracy and shorter time in solving the multi-soliton solution of the KdV equation compared with PINNs.

The better performance of pr-PINNs is attributed to the incorporation of a priori information, which enables the network to capture some of the physical features in advance. Specially, from Figures 3 and 5, which illustrate the two-soliton and three-soliton solutions, the predicted solutions using the PINNs method do not fit well with the desired solutions after network training due to the fact that they fall into bad local optimal solutions during the solving process. On the contrary, due to the inclusion of the priori information, pr-PINNs allow the network to capture some of the features of the solutions and prevent the network from falling into bad local optimal solutions during the training process [33]. Therefore, compared with PINNs, the pr-PINNs method can obtain more accurate simulation results under the same conditions.

**Table 1.** Comparison of training results for the KdV equation.

| Solution type | Method | $L^2$ norm error | Pre-training time | All training time |
|---|---|---|---|---|
| One-soliton | PINNs | 3.023602e − 03 | - | 45.6 s |
| | pr-PINNs | 2.969873e − 03 | 5.54 s | 32.7 s |
| Two-soliton | PINNs | 1.270713e − 01 | - | 569 s |
| | pr-PINNs | 1.322320e − 02 | 88 s | 283 s |
| Three-soliton | PINNs | 3.251062e − 01 | - | 298 s |
| | pr-PINNs | 1.316734e − 02 | 50.8 s | 122 s |

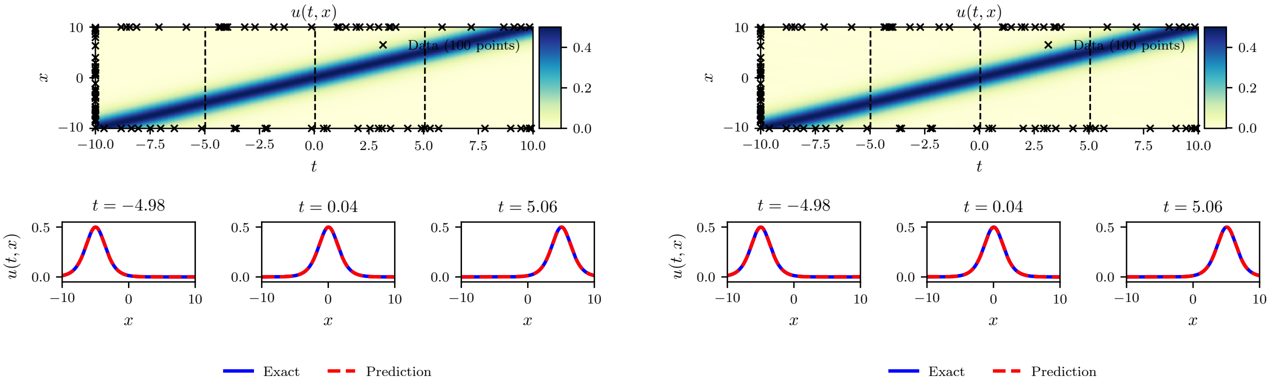
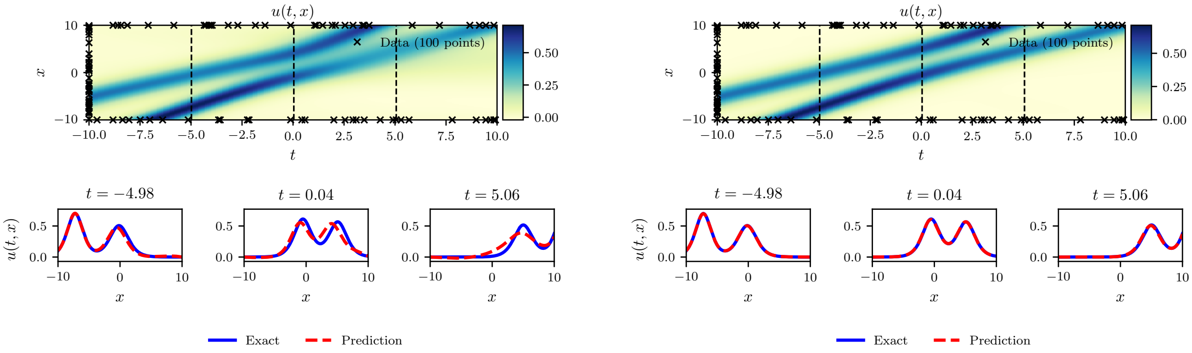
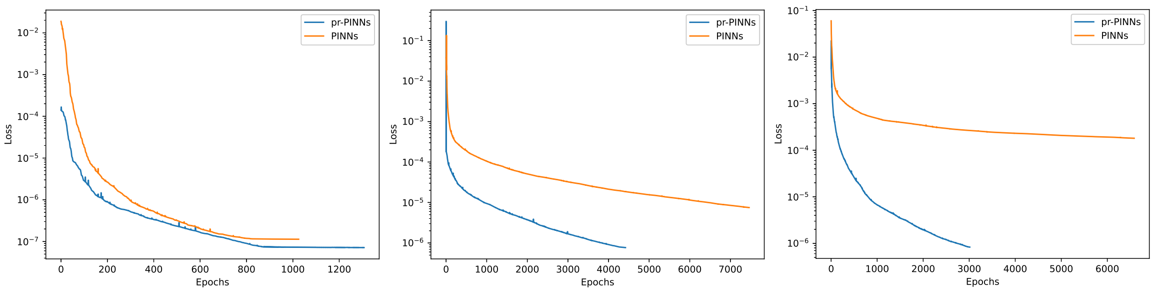


**Figure 2.** The density plots of the one-soliton solution for the KdV equation using PINNs (a) and pr-PINNs (b), and corresponding cross-sectional plots at $t = -4.98, t = 0.04$, and $t = 5.06$.



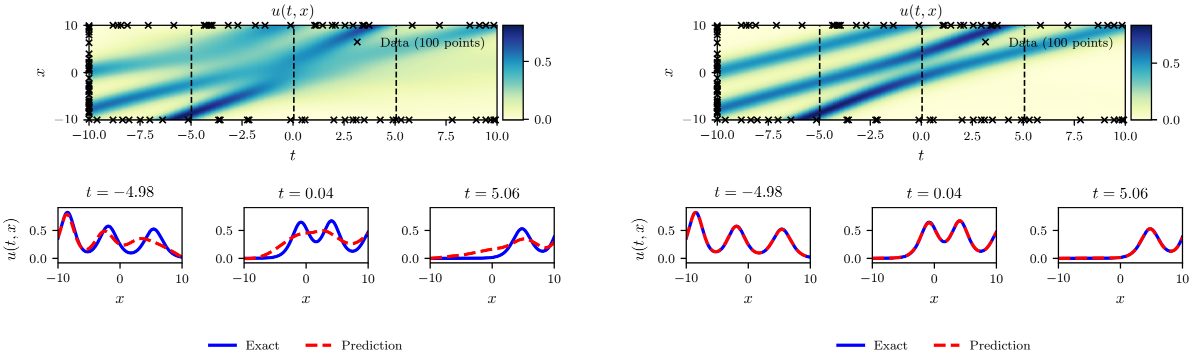
(a) (b)

**Figure 3.** The density plots of the two-soliton solution for the KdV equation using PINNs (a) and pr-PINNs (b), and corresponding cross-sectional plots at $t = -4.98, t = 0.04$, and $t = 5.06$.



(a) (b) (c)

**Figure 4.** The corresponding loss curves of the one-soliton (a), two-soliton (b), and three-soliton (c) solutions using PINNs and pr-PINNs.



(a) (b)

**Figure 5.** The density plots of the three-soliton solution for the KdV equation using PINNs (a) and pr-PINNs (b), and corresponding cross-sectional plots at $t = -4.98, t = 0.04$, and $t = 5.06$.

## 4. Soliton and lump solutions of the (2+1)-dimensional KdV equation

In this section, the effectiveness of PINNs and pr-PINNs methods for solving different kinds of localized wave solutions for a higher-dimensional equation are compared and the reasons why the two methods produce the corresponding results are analyzed. The pr-PINNs method works with a priori information, and the shape features of the soliton and lump solutions are captured by the network before the weights are transferred to the PINNs.

The (2+1)-dimensional KdV equation is given as

$$\begin{cases} u_t - u_{xxx} + 3(uv)_x = 0, \\ u_x = v_y. \end{cases} \tag{4.1}$$

In the following, the PINNs and pr-PINNs are employed to simulate the soliton and lump solutions for the (2+1)-dimensional KdV equation. To simulate solutions, the training spatial-temporal region is set as $(x, y, t) \in [-5, 5] \times [-5, 5] \times [0, 1]$. For both PINNs and pr-PINNs methods, the network has 5 layers and the number of neurons in each layer is 120. The L-BFGS optimizer is utilized and termination conditions of the optimizer are: The maximum number of iterations is set at 50,000, the maximum number of function evaluations is set at 50,000, the size of the history record is 50, and the gradient tolerance is $1 \times 10^{-6}$. The randomly sampled collocation points number for the initial and boundary area is 100 and the randomly sampled collocation points number inside the solution domain is 20,000 for both PINNs and pr-PINNs methods. The compared simulated results using PINNs and pr-PINNs are shown in Figures 6–11 and Table 2. The corresponding loss curve plots are shown in Figure 12.

To solve the one-soliton solution for a (2+1)-dimensional KdV equation, the initial condition is given as

$$u(x, y, 0) = \frac{81}{200}\text{sech}^2(\frac{9x}{20} + \frac{y}{2}),\ x \in [-5, 5],\ y \in [-5, 5],$$

and boundary conditions are

$$\begin{aligned} &u(x, -5, t) = \frac{81}{200}\text{sech}^2(-5 - \frac{729t}{2,000} + \frac{9x}{20}), \quad u(x, 5, t) = \frac{81}{200}\text{sech}^2(5 - \frac{729t}{2,000} + \frac{9x}{20}), \\ &y \in [-5, 5], \quad t \in [0, 1], \\ &u(-5, y, t) = \frac{81}{200}\text{sech}^2(-\frac{9}{2} - \frac{729t}{2,000} + \frac{y}{2}), \quad u(5, y, t) = \frac{81}{200}\text{sech}^2(-\frac{9}{2} - \frac{729t}{2,000} + \frac{y}{2}), \\ &y \in [-5, 5], \quad t \in [0, 1]. \end{aligned}$$

The (2+1)-dimensional KdV equation can be calculated to have the following exact two-soliton solution

$$\begin{aligned} &u_{2ss}(x, y, t) = \frac{189\left(16\eta_2\eta_1^2 + 9\eta_2^2\eta_1 + 91\eta_2\eta_1 + 378\eta_1 + 672\eta_2\right)}{25\,(\eta_2\eta_1 + 42\eta_1 + 42\eta_2 + 42)^{\,2}}, \\ &\eta_1 = e^{-\frac{729t}{1,000} + \frac{9x}{10} + y}, \qquad \eta_2 = e^{-\frac{216t}{125} + \frac{6x}{5} + \frac{7y}{5}}. \end{aligned} \tag{4.2}$$

To simulate the two-soliton solution of the KdV equation using PINNs and pr-PINNs methods, the initial condition is given as

$$u(x, y, 0) = u_{2ss}(x, y, 0),\ x \in [-5, 5],\ y \in [-5, 5],$$

and boundary conditions are given as

$$\begin{aligned} &u(-5,y,t) = u_{2ss}(-5,y,t),\ u(5,y,t) = u_{2ss}(5,y,t),\ y \in [-5,5],\ t \in [0,1], \\ &u(x,-5,t) = u_{2ss}(x,-5,t),\ u(x,5,t) = u_{2ss}(x,5,t),\ x \in [-5,5],\ t \in [0,1]. \end{aligned} \tag{4.3}$$

The (2+1)-dimensional KdV equation has the following three-soliton solution based on the calculations:

$$\begin{aligned} u_{3ss}(x,y,t) \quad = \quad & -\frac{2\left(\frac{\eta_2\eta_1}{20} + \frac{17\eta_2\eta_3\eta_1}{614{,}075} + \frac{18\eta_3\eta_1}{145} + \frac{9\eta_1}{10} + \frac{6\eta_2}{5} + \frac{\eta_2\eta_3}{66} + \frac{13\eta_3}{10}\right)^2}{\left(\frac{\eta_2\eta_1}{42} + \frac{\eta_2\eta_3\eta_1}{122{,}815} + \frac{18\eta_3\eta_1}{319} + \eta_1 + \eta_2 + \frac{\eta_2\eta_3}{165} + \eta_3 + 1\right)^2} \\ & + \frac{2\left(\frac{21\eta_2\eta_1}{200} + \frac{289\eta_2\eta_3\eta_1}{3{,}070{,}375} + \frac{198\eta_3\eta_1}{725} + \frac{81\eta_1}{100} + \frac{36\eta_2}{25} + \frac{5\eta_2\eta_3}{132} + \frac{169\eta_3}{100}\right)}{\frac{\eta_2\eta_1}{42} + \frac{\eta_2\eta_3\eta_1}{122{,}815} + \frac{18\eta_3\eta_1}{319} + \eta_1 + \eta_2 + \frac{\eta_2\eta_3}{165} + \eta_3 + 1}, \\ & \eta_1 = e^{-\frac{729t}{1{,}000} + \frac{9x}{10} + y}, \qquad \eta_2 = e^{-\frac{216t}{125} + \frac{6x}{5} + \frac{7y}{5}}, \qquad \eta_3 = e^{-\frac{2{,}197t}{1{,}000} + \frac{13x}{10} + \frac{19y}{10}}. \end{aligned} \tag{4.4}$$

To simulate the three-soliton solution of the KdV equation using PINNs and pr-PINNs methods, the initial condition is taken as

$$u(x,y,0) = u_{3ss}(x,y,0),\ x \in [-5,5],\ y \in [-5,5],$$

and boundary conditions are

$$\begin{aligned} &u(-5,y,t) = u_{3ss}(-5,y,t),\ u(5,y,t) = u_{3ss}(5,y,t),\ y \in [-5,5],\ t \in [0,1], \\ &u(x,-5,t) = u_{3ss}(x,-5,t),\ u(x,5,t) = u_{3ss}(x,5,t),\ x \in [-5,5],\ t \in [0,1]. \end{aligned} \tag{4.5}$$

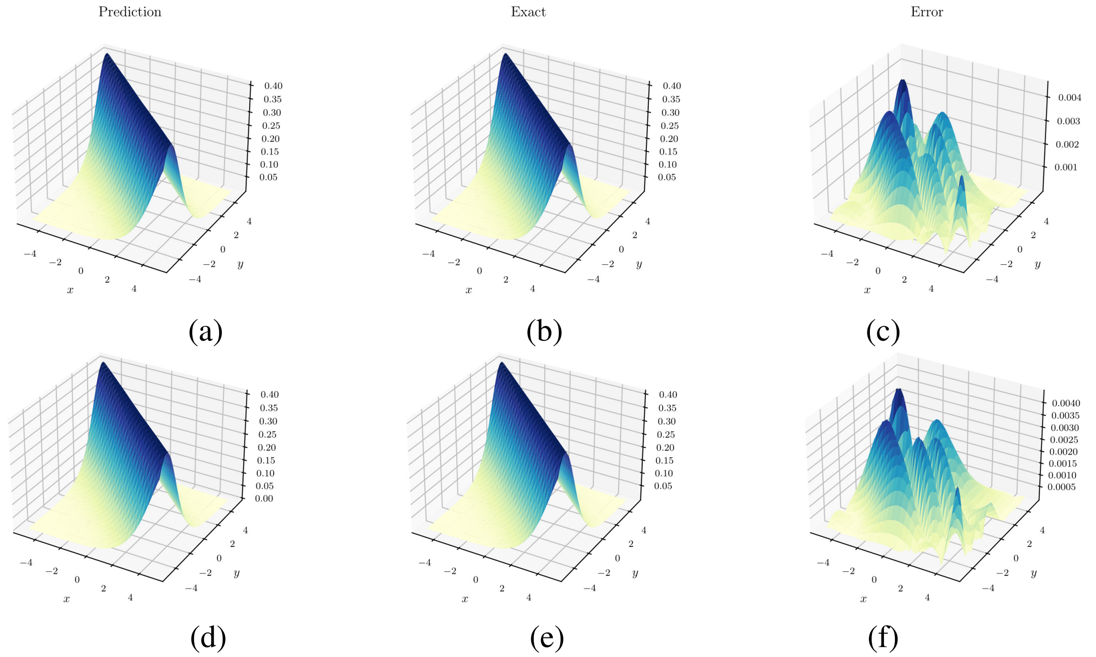


**Figure 6.** The predicted, exacted, and error plots of one-soliton solution for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$ and the corresponding error plots.

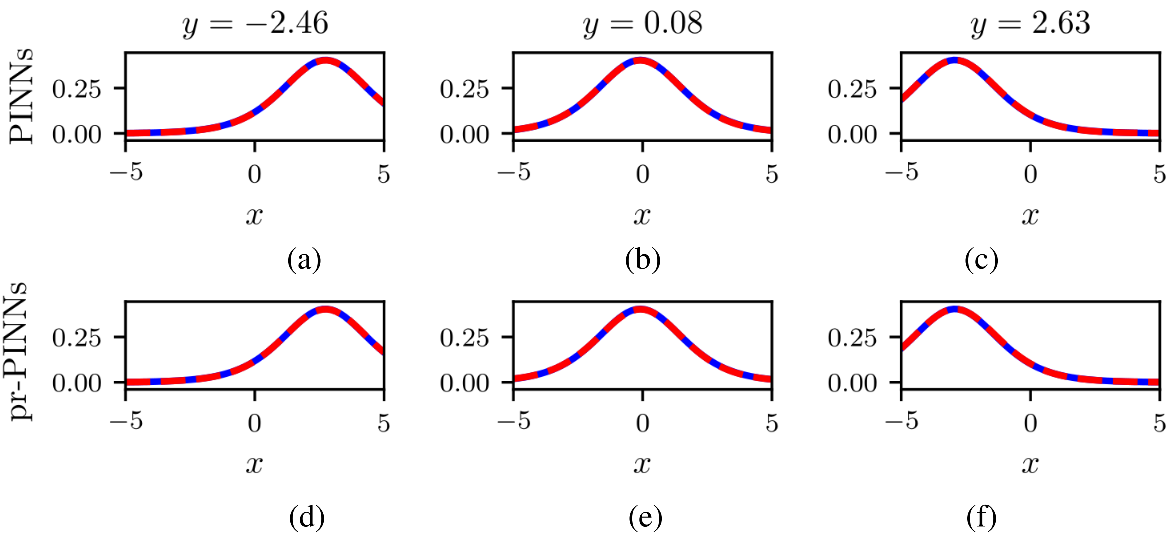


**Figure 7.** The cross-sectional plots of simulated one-soliton solution for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$. The exact solutions are given by the blue curves while the predicted solutions are given by the dashed curves. They practically overlap here.

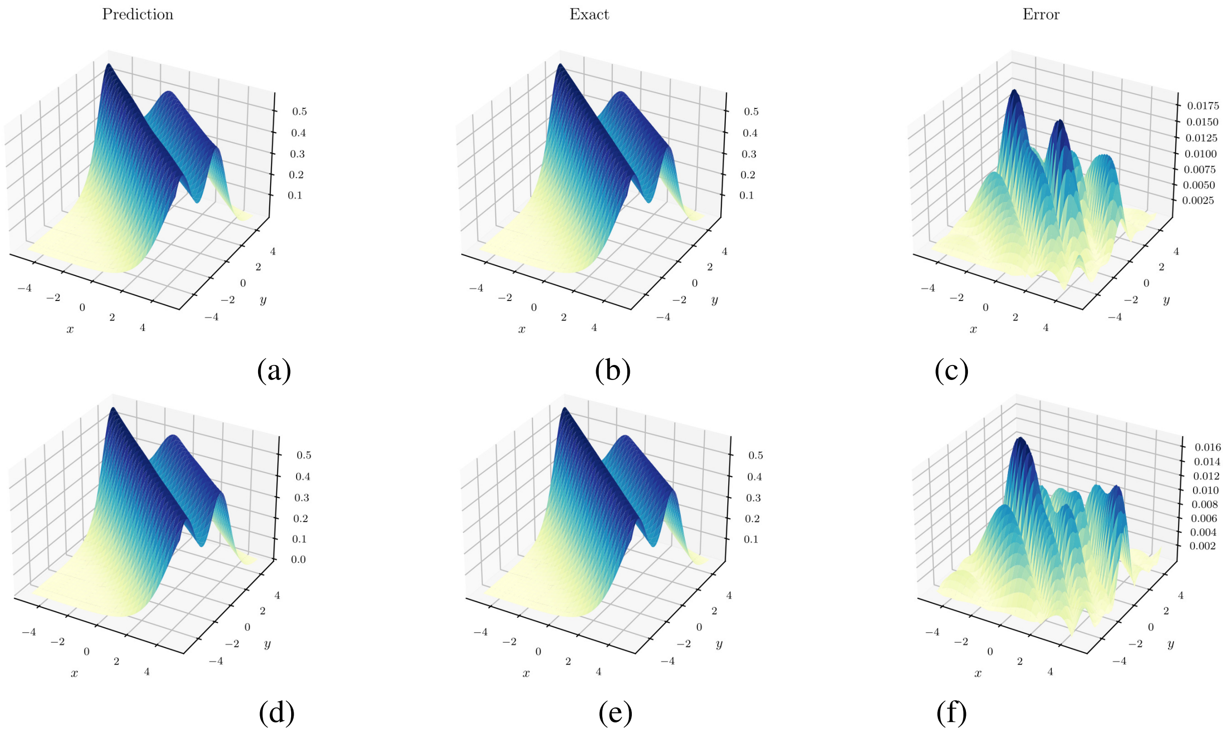
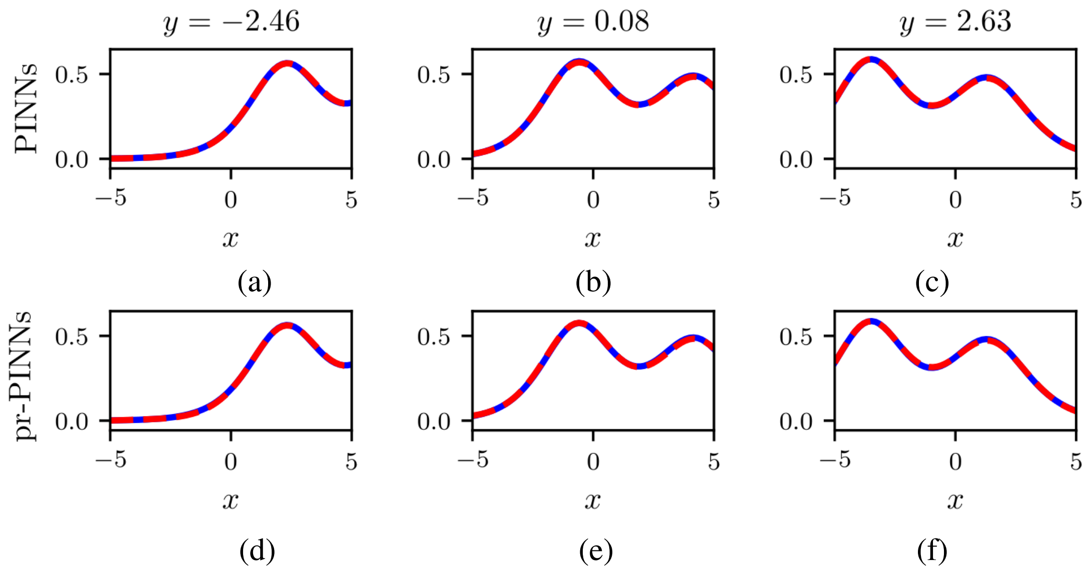


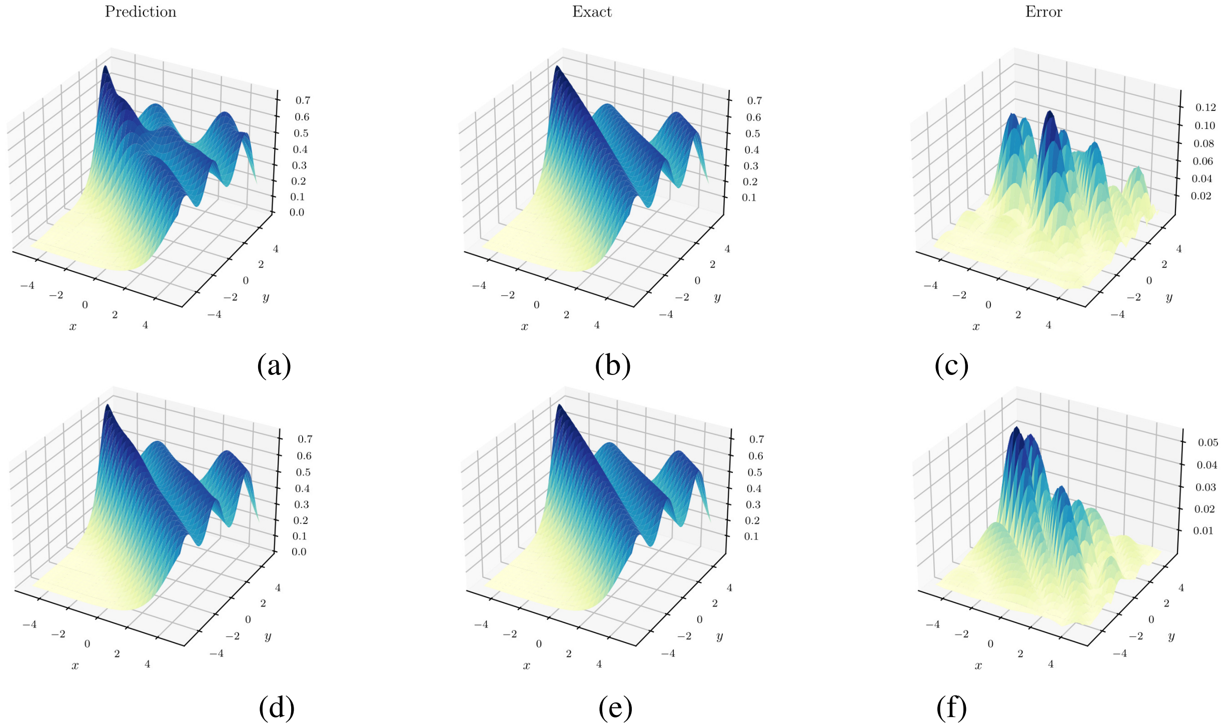
**Figure 8.** The predicted, exacted, and error plots of two-soliton solution for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$ and the corresponding error plots.



(a) (b) (c)

(d) (e) (f)

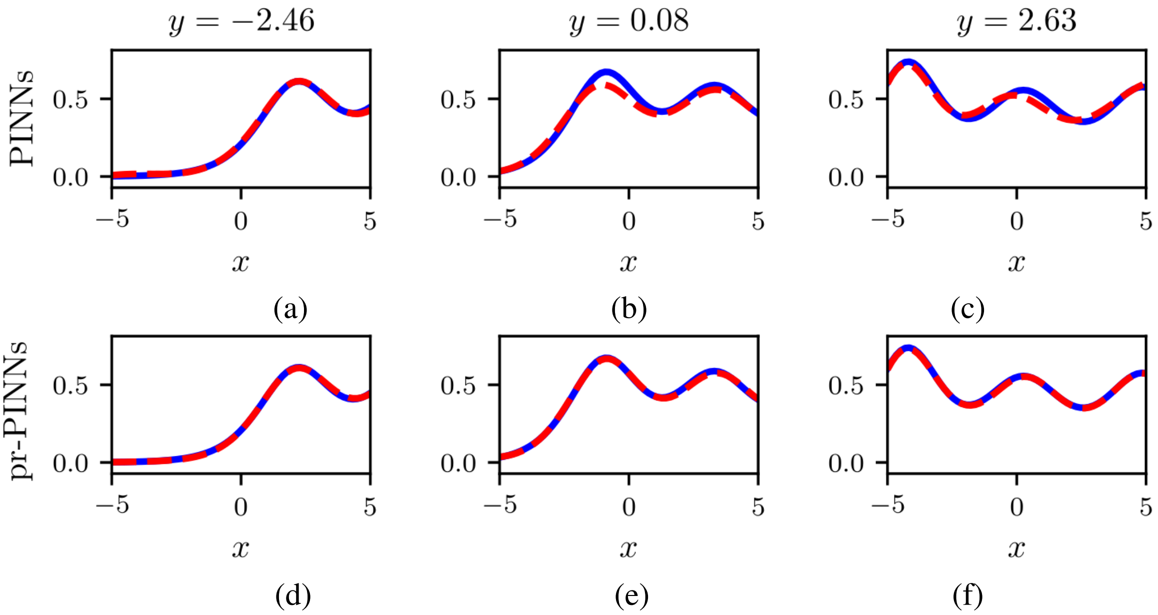
**Figure 9.** The cross-sectional plots of simulated two-soliton solution for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$. The exact solutions are given by the blue curves while the predicted solutions are given by the dashed curves. They practically overlap here.



(a) (b) (c)

(d) (e) (f)

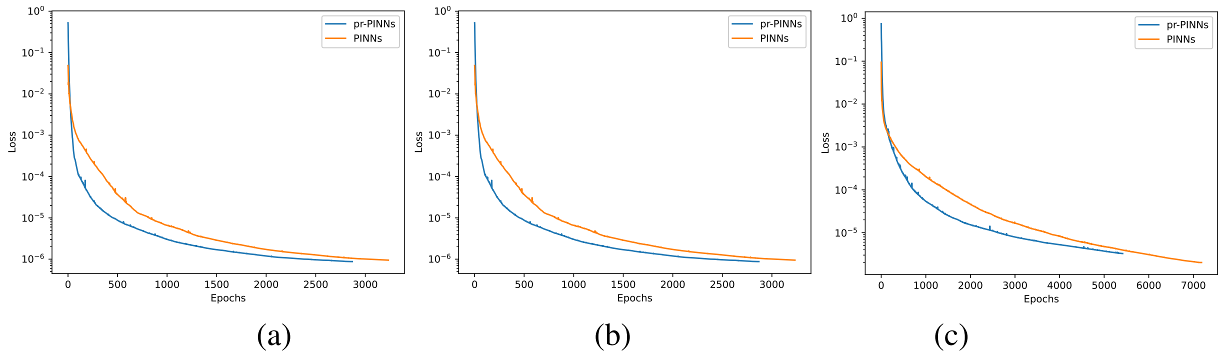
**Figure 10.** The predicted, exacted, and error plots of three-soliton solution for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$ and the corresponding error plots.



(a) (b) (c)

(d) (e) (f)

**Figure 11.** The cross-sectional plots of simulated three-soliton solution for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$. The exact solutions are given by the blue curves while the predicted solutions are given by the dashed curves.



(a) (b) (c)

**Figure 12.** The corresponding loss curves of the one-soliton (a), two-soliton (b), and three-soliton (c) solutions using PINNs and pr-PINNs.

The lump solution is a special structure of localized wave solutions for high-dimensional nonlinear PDEs which decay exponentially in the spatial dimensions. This kind of solution holds significant potential applications in nonlinear wave theory, including water wave propagation, optics, and plasma physics. Thus, we test the accuracy and efficiency of pr-PINNs for simulating the lump solution of the (2+1)-dimensional KdV equation. For the simulated lump solution, the expression for the initial conditions is given in the following form:

$$u(x, y, 0) = -\frac{1}{2} - \frac{8\left(-5 + 4x^2 + 20xy + 5y^2\right)}{(5 + 4x^2 + 4xy + 5y^2)^2}, x \in [-5, 5], \ y \in [-5, 5],$$

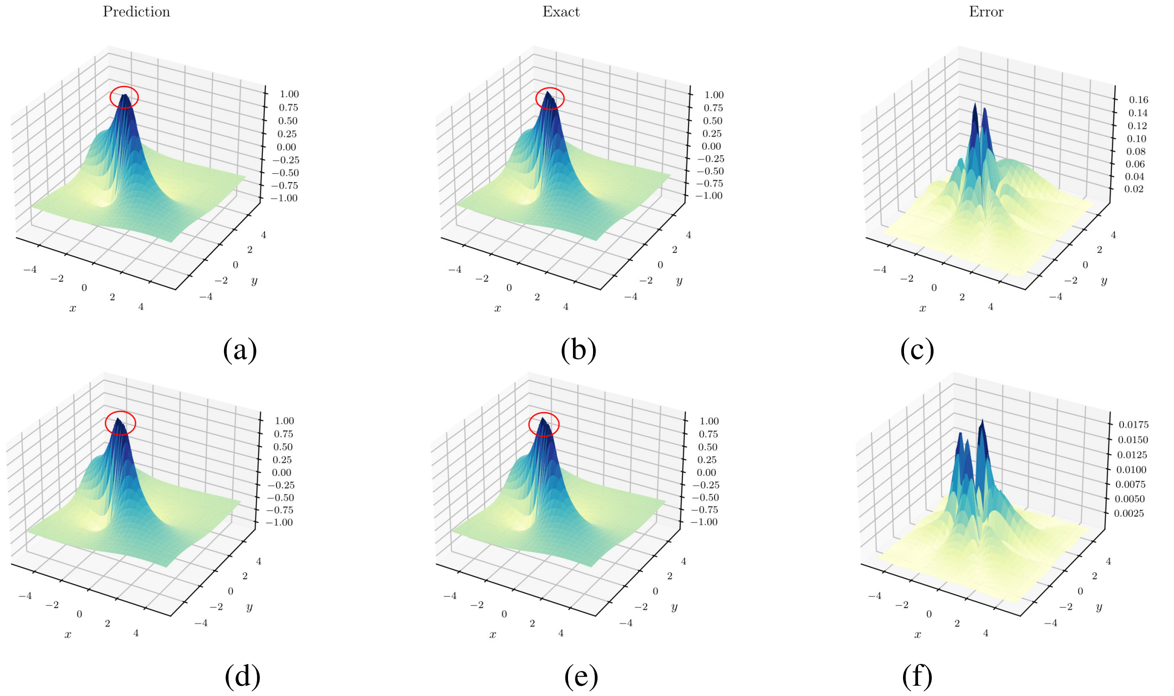
and boundary conditions are given in the following form:

$$
\begin{aligned}
&u(-5,y,t) = -\frac{1}{2} + \frac{8\left(-9,875 - 600t + 711t^2 + 5,000y - 1,050ty - 125y^2\right)}{(2,025 - 840t + 117t^2 - 200y - 6ty + 25y^2)^2},\\
&y \in [-5,5],\ t \in [0,1],\\
&u(5,y,t) = -\frac{1}{2} + \frac{8\left(-9,875 + 600t + 711t^2 - 5,000y - 1,050ty - 125y^2\right)}{(2,025 + 840t + 117t^2 + 200y - 6ty + 25y^2)^2},\\
&y \in [-5,5],\ t \in [0,1],\\
&u(x,-5,t) = -\frac{1}{2} + \frac{8\left(-1,2375 + 10,500t + 711t^2 + 5,000x + 60tx - 100x^2\right)}{(2,525 + 60t + 117t^2 - 200x + 84tx + 20x^2)^2},\\
&x \in [-5,5],\ t \in [0,1],\\
&u(x,5,t) = -\frac{1}{2} + \frac{8\left(-12,375 - 10,500t + 711t^2 - 5,000x + 60tx - 100x^2\right)}{(2,525 - 60t + 117t^2 + 200x + 84tx + 20x^2)^2},\\
&x \in [-5,5],\ t \in [0,1].
\end{aligned}
\tag{4.6}
$$

The lump solution is simulated for the (2+1)-dimensional KdV equation using the PINNs and pr-PINNs methods, respectively, and the error associated with PINNs is $5.566199 \times 10^{-2}$ while the error obtained using the pr-PINNs method is $6.365303 \times 10^{-3}$ as seen in Table 2. The compared results between the two methods are shown in Figures 13, 14, and 15. This demonstrates that the pr-PINNs method gets an order of magnitude improvement in solving lump solutions compared with the PINNs method. A comparison of the difference at the peaks circled in red shows the significant advantages and recognizes validity of the pr-PINNs method for solving the lump solution of the (2+1)-dimensional KdV equation.

**Table 2.** Comparison of training results for the (2+1)-dimensional KdV equation.

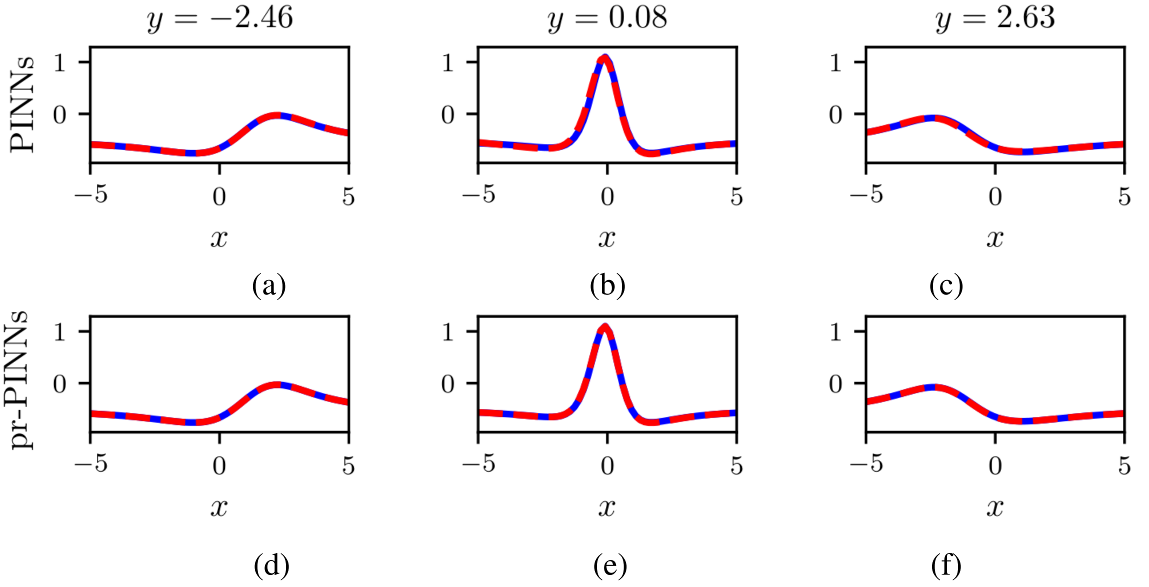
| Solution type | Method | $L^2$ norm error | Pre-training time | All training time |
|---|---|---|---|---|
| One-soliton | PINNs | 6.019869e − 03 | - | 41.8 s |
| | pr-PINNs | 4.588799e − 03 | 4.73 s | 31.6 s |
| Two-soliton | PINNs | 7.795944e − 03 | - | 104.94 s |
| | pr-PINNs | 6.412547e − 03 | 7.21 s | 73.45 s |
| Three-soliton | PINNs | 8.637403e − 02 | - | 732 s |
| | pr-PINNs | 3.985691e − 02 | 129 s | 724 s |
| Lump | PINNs | 5.566199e − 02 | - | 922 s |
| | pr-PINNs | 6.365303e − 03 | 16.2 s | 451.2 s |



(a) (b) (c)

(d) (e) (f)

**Figure 13.** The predicted, exacted, and error plots of lump solution for the (2+1)-dimensional KdV equation using PINNs ((a),(b),(c)) and pr-PINNs ((d),(e),(f)) compared with the exact solution at $t = 0$ and the corresponding error plots.



(a) (b) (c)

(d) (e) (f)

**Figure 14.** The cross-sectional plots of simulated lump solutions for the (2+1)-dimensional KdV equation using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)) compared with the exact solution at $t = 0$. The exact solutions are given by the blue curves while the predicted solutions are given by the dashed curves. They practically overlap here.

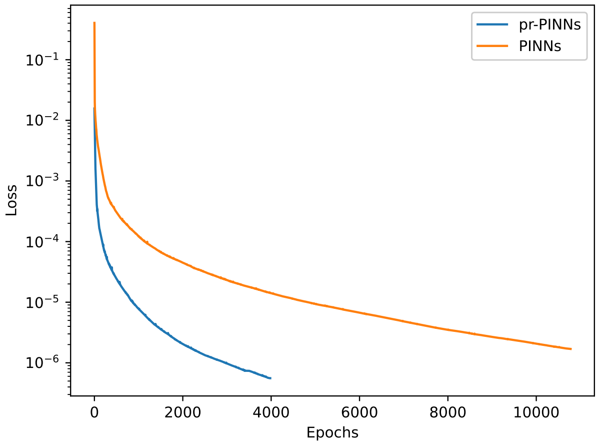


**Figure 15.** The corresponding loss curves of lump solutions using PINNs and pr-PINNs.

## 5. The rational solution of the combined KdV-mKdV equation

The combined KdV-mKdV equation given by [22]

$$u_t + 6uu_x + 6u^2u_x + u_{xxx} = 0, \tag{5.1}$$

can describe nonlinear phenomena in various fields such as fluid dynamics, plasma physics, and so on [29]. The combined KdV-mKdV equation contains quadratic and cubic nonlinear terms. In the following, rational solutions of the combined KdV-mKdV equation are simulated using the PINNs and pr-PINNs methods, respectively. The initial and boundary conditions for the second-order rational solution are given by [22],

$$\begin{aligned}
u(x,0) &= \frac{\frac{12{,}987}{40} + \frac{1431x}{10} - 405x^2 - 108x^3 - 54x^4}{\left(\frac{48}{5} - \frac{27x}{4} + \frac{9x^2}{2} + 3x^3\right)^2 + \left(9 + \left(\frac{3}{2} + 3x\right)^2\right)^2}, \quad x \in [-10, 10],\\
u(-10,t) &= \frac{-\frac{18{,}944{,}253}{40} - 12312t}{\frac{10{,}791{,}225}{16} + \left(-\frac{24{,}729}{10} + 36t\right)^2}, \quad t \in [-10, 10],\\
u(10,t) &= \frac{-\frac{27{,}469{,}773}{40} + 13{,}608t}{\frac{16{,}040{,}025}{16} + \left(\frac{33{,}921}{10} + 36t\right)^2}, \quad t \in [-10, 10].
\end{aligned} \tag{5.2}$$

The initial and boundary conditions for the third-order rational solution are given by [34],

$$\begin{aligned}
u(x,0) = {}& \frac{72\left(x^{10} + 90x^8 + 7{,}560x^6 - 97{,}200x^4 - 874{,}800x^2 + 5{,}248{,}800\right)}{x^{12} + 36x^{10} + 4{,}860x^8 + 505{,}440x^6 + 4{,}374{,}000x^4 + 94{,}478{,}400x^2 + 94{,}478{,}400} - 1,\\
& x \in [-10, 10],\\
u(-10,t) = {}& -\frac{324t^4 + 63{,}720t^3 + 212{,}274t^2 - 72{,}322{,}140t + 355{,}191{,}853}{324t^4 + 44{,}280t^3 + 2{,}394{,}414t^2 - 65{,}595{,}900t + 1{,}502{,}951{,}449},\\
& t \in [-10, 10],
\end{aligned}$$

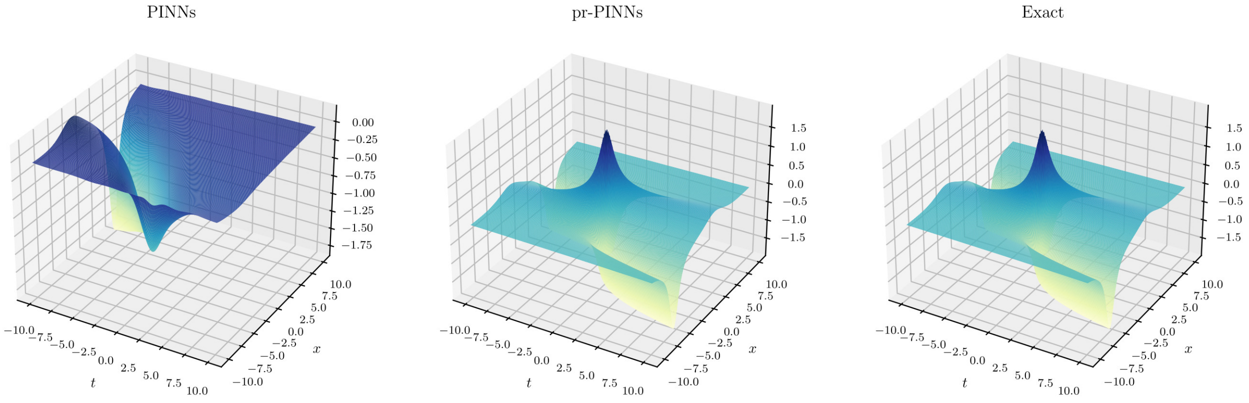
$$u(10,t) = -\frac{324t^4 - 63,720t^3 + 212,274t^2 + 72,322,140t + 355,191,853}{324t^4 - 44,280t^3 + 2,394,414t^2 + 65,595,900t + 1,502,951,449},$$
$$t \in [-10, 10]. \qquad (5.3)$$

The training spatial-temporal region is $(x, t) \in [-10, 10] \times [-10, 10]$. For both PINNs and pr-PINNs methods, the network has 7 layers and the number of neurons in each layer is 100. The L-BFGS optimizer is selected and termination conditions of the optimizer are set as follows: The maximum number of iterations is 50,000, the maximum number of function evaluations is 50,000, the size of the history record is 50, and the gradient tolerance is $1 \times 10^{-6}$. The randomly sampled collocation points number for the initial and boundary area is 100 and the randomly sampled collocation points number inside the solution domain is $25,000$.

The second-order and third-order rational solutions for the combined KdV-mKdV equation trained by PINNs and pr-PINNs are shown in Figures 16 and 18. The cross-sectional Figures 17 and 19 show that the pr-PINNs method gets higher accuracy results while the solutions simulated by the PINNs method deviate from the true solutions more. The simulated results of the solutions solved by the PINNs and pr-PINNs methods, respectively, are displayed in Figures 17 and 19. From the compared results of the loss and error between the two methods shown in Figure 20 and Table 3, it can be seen that pr-PINNs method performs better than the PINNs method with better accuracy and shorter training time in simulating the high-order rational solutions for the combined KdV-mKdV equation.

**Table 3.** Comparison of training results for the combined KdV-mKdV equation.

| Solution type | Method | $L^2$ norm error | Pre-training time | All training time |
|---|---|---|---|---|
| second-order rational | PINNs | 8.160418e − 01 | - | 522 s |
| | pr-PINNs | 4.816645e − 02 | 66 s | 392 s |
| third-order rational | PINNs | 8.379468e − 01 | - | 292 s |
| | pr-PINNs | 6.392167e − 02 | 132 s | 272 s |



(a) (b) (c)

**Figure 16.** The 3D plots of the simulated second-order rational solutions for the combined KdV-mKdV equation using PINNs (a) and pr-PINNs (b) as well as the exact solution (c).

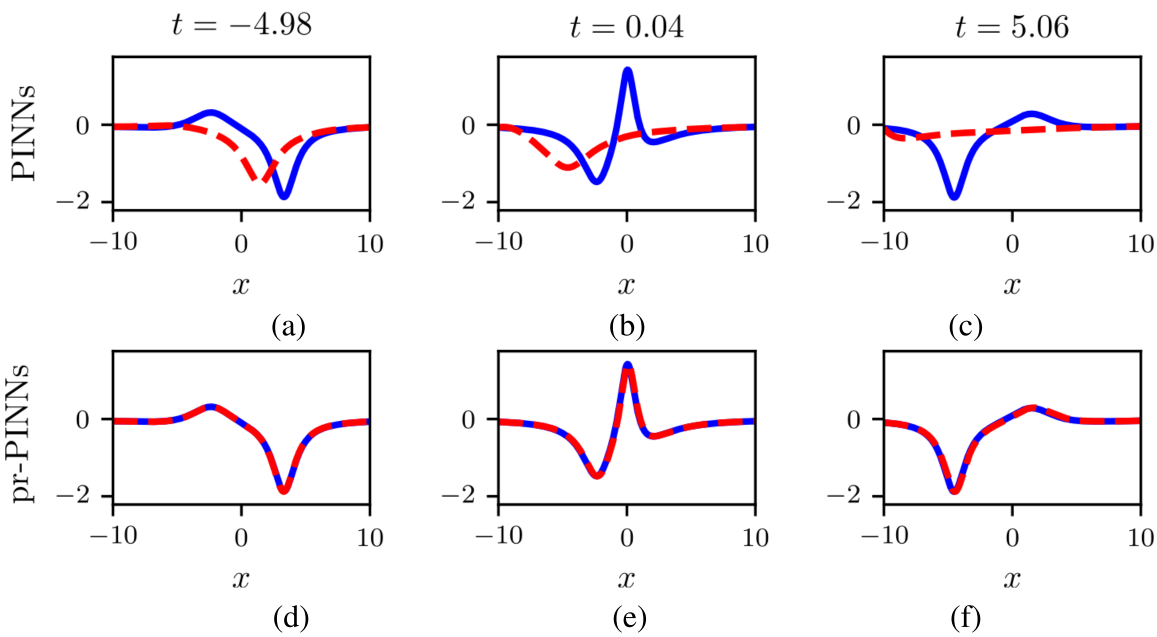


**Figure 17.** Cross-sectional plots of the second-order rational solutions corresponding to Figure 16 at $t = -4.98, t = 0.04$, and $t = 5.06$ using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)). The exact solutions are given by the blue curves while the predicted solutions are given by the dashed curves.

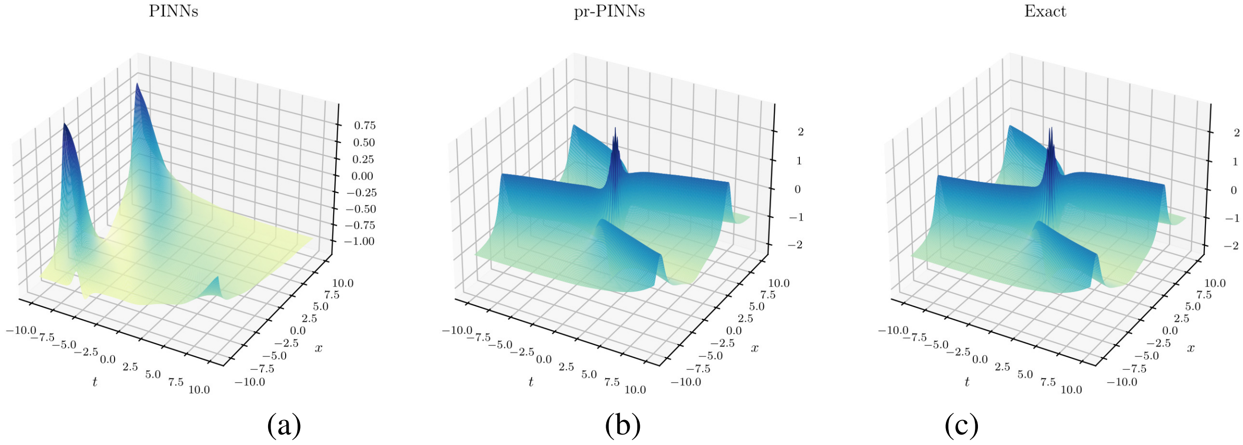
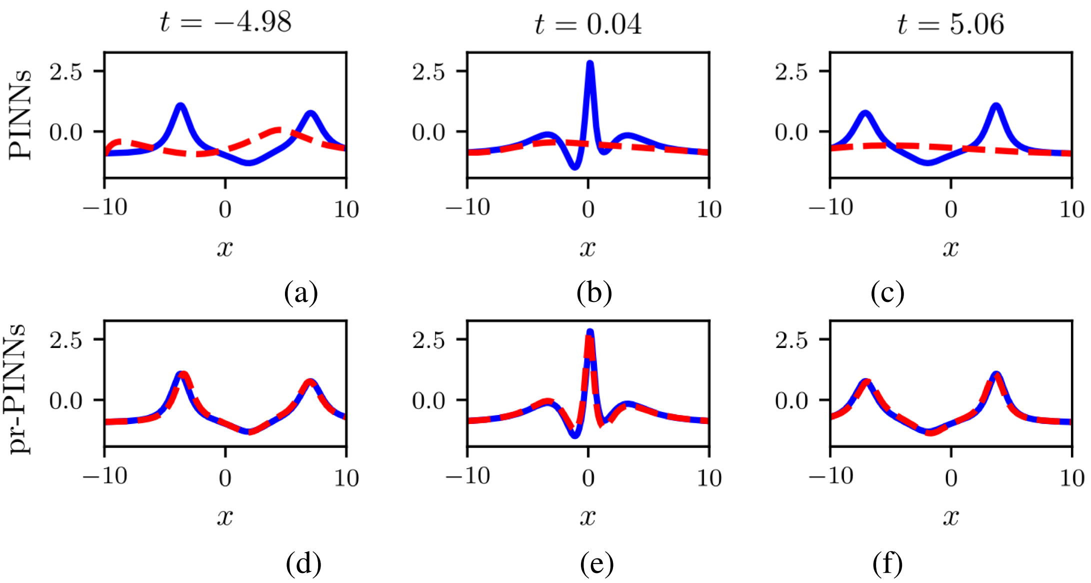


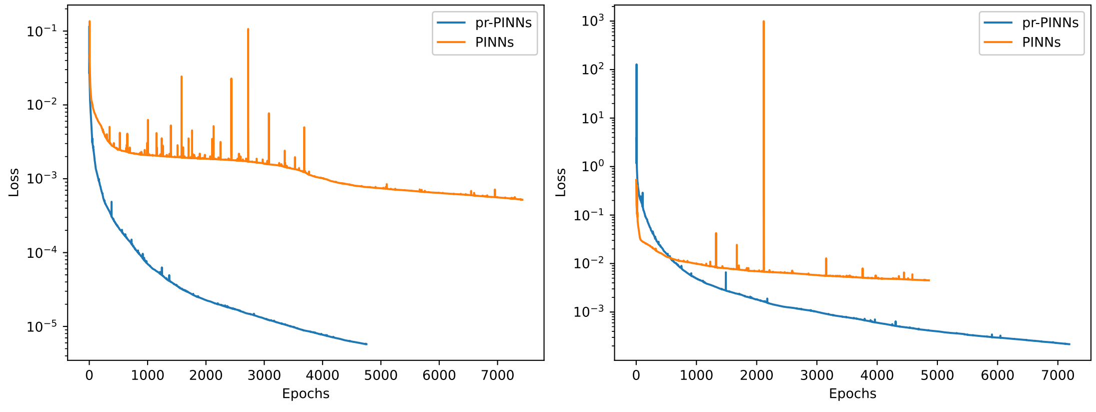
**Figure 18.** The 3D plots of the simulated third-order rational solutions for the combined KdV-mKdV equation using PINNs (a) and pr-PINNs (b) as well as the exact solution (c).



(a) (b) (c)

(d) (e) (f)

**Figure 19.** Cross-sectional plots of the third-order rational solutions corresponding to Figure 16 at $t = -4.98, t = 0.04$, and $t = 5.06$ using PINNs ((a), (b), (c)) and pr-PINNs ((d), (e), (f)). The exact solutions are given by the blue curves while the predicted solutions are given by the dashed curves.



(a) (b)

**Figure 20.** The corresponding loss curves of the second-order and third-order rational solutions using PINNs and pr-PINNs.

## 6. Conclusions

The pr-PINNs method with inclusion of a prior information in the PINNs method is proposed in this paper. We used the proposed method to simulate several localized wave solutions of KdV-type equations including one-soliton, two-soliton, and three-soliton solutions, lump solutions, and second- and third-order rational solutions. The solution results of the pr-PINNs method showed higher accuracy and better convergence than that of the PINNs method, since the pr-PINNs method can better capture the characteristics of the solutions and prevent the trained results from drastically deviating from the real results.

To better illustrate the performance of the networks, more numerical experiments on the second-order rational solutions of the combined KdV-mKdV equation are carried out for different data and different numbers of network layers. The solution errors of the PINNs and pr-PINNs methods are given in Tables 4 and 5. In Table 4, a six-layer network is set with 100 neurons per layer, and other parameters in the network are set the same as the conditions in section 5. In Table 5, $N_f$ is taken to be 25,000, the number of neurons in each layer is 100, and other parameters are set the same as the conditions in section 5. We can see in both tables that the pr-PINNs method exhibits higher accuracy compared to the PINNs method under different $N_f$ and different layers. For better performance of the pr-PINNs method for simulating solutions of PDEs, it is hoped that the pr-PINNs method could be applied to more practical problems with some ambiguous information.

**Table 4.** Comparison of results with different data.

| $N_f$ | 10000 | 15000 | 20000 | 25000 |
|---|---|---|---|---|
| PINNs | 9.036961e − 01 | 6.510653e − 01 | 7.293564e − 01 | 7.248199e − 01 |
| pr-PINNs | 3.264207e − 02 | 3.748243e − 02 | 9.471193e − 02 | 3.663322e − 02 |

**Table 5.** Comparison of results with different layers.

| Layers | 4 layers | 5 layers | 6 layers | 7 layers |
|---|---|---|---|---|
| PINNs | 8.120882e − 01 | 7.2276331e − 01 | 7.248199e − 01 | 7.970699e − 01 |
| pr-PINNs | 5.326740e − 02 | 7.882531e − 02 | 3.663322e − 02 | 1.669831e − 03 |

## Author contributions

Z. Y. Feng: Methodology, software, validation, investigation, writing–original draft, visualization; X. H. Meng: Conceptualization, methodology, validation, formal analysis, writing–review and editing, supervision, funding acquisition; X. G. Xu: Conceptualization, validation, writing–review and editing. All authors have read and approved the final version of the manuscript for publication.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62071053.

## Conflict of interest

All authors declare no conflicts of interest in this paper.

## References

1. J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375** (2018), 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029

2. M. Raissi, G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, *J. Comput. Phys.*, **357** (2018), 125–141. https://doi.org/10.1016/j.jcp.2017.11.039

3. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

4. S. Lin, Y. Chen, Physics-informed neural network methods based on Miura transformations and discovery of new localized wave solutions, *Phys. D*, **445** (2023), 133629. http://dx.doi.org/10.1016/j.physd.2022.133629

5. A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, M. W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, *Adv. Neural Inform. Process. Syst.*, **34** (2021), 26548–26560.

6. W. Li, C. Zhang, C. Wang, H. Guan, D. Tao, Revisiting PINNs: Generative adversarial physics-informed neural networks and point-weighting method, *arXiv:2205.08754*, 2022. https://doi.org/10.48550/arXiv.2205.08754

7. A. D. Jagtap, E. Kharazmi, G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems, *Comput. Methods Appl. Mech. Eng.*, **365** (2020), 113028. https://doi.org/10.1016/j.cma.2020.113028

8. A. D. Jagtap, G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *Commun. Comput. Phys.*, **28** (2020), 2002–2041. http://dx.doi.org/10.4208/cicp.oa-2020-0164

9. E. L. Bourodimos, Linear and nonlinear wave motion, *Rev. Geophys.*, **6** (1968), 103–128. https://doi.org/10.1029/RG006i002p00103

10. U. Waheed, E. Haghighat, T. Haghighat, C. Song, Q. Hao, PINNeik: Eikonal solution using physics-informed neural networks, *Comput. Geosci.*, **155** (2021), 104833. https://doi.org/10.1016/j.cageo.2021.104833

11. X. Huang, T. Alkhalifah, PINNup: Robust neural network wavefield solutions using frequency upscaling and neuron splitting, *J. Geophys. Res. Solid Earth*, **127** (2022), e2021JB023703. https://doi.org/10.1029/2021JB023703

12. X. Huang, T. Alkhalifah, Efficient physics-informed neural networks using hash encoding, *J. Comput. Phys.*, **501** (2024), 112760. https://doi.org/10.1016/j.jcp.2024.112760

13. X. Huang, T. Alkhalifah, GaborPINN: Efficient physics informed neural networks using multiplicative filtered networks, *IEEE Geosci. Remote Sens. Lett.*, **20** (2023), 3003405. http://dx.doi.org/10.1109/LGRS.2023.3330774

14. T. Alkhalifah, X. Huang, Physics-informed neural wavefields with Gabor basis functions, *Neural Netw.*, **175** (2024) 106286. https://dx.doi.org/10.1016/j.neunet.2024.106286

15. M. Raissi, A. Yazdani, G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, *Science*, **367** (2020), 1026–1030. https://doi.org/10.1126/science.aaw4741

16. W. X. Qiu, Z. Z. Si, D. S. Mou, C. Q. Dai, J. T. Li, W. Liu, Data-driven vector degenerate and nondegenerate solitons of coupled nonlocal nonlinear Schrödinger equation via improved PINN algorithm, *Nonlinear Dyn.*, 2024. https://doi.org/10.1007/s11071-024-09648-y

17. J. W. Miles, The Korteweg-de Vries equation: A historical essay, *J. Fluid Mech.*, **106** (1981), 131–147. https://doi.org/10.1017/S0022112081001559

18. M. Wadati, The modified Korteweg-de Vries equation, *J. Phys. Soc. Jan.*, **34** (1973), 1289–1296. https://doi.org/10.1143/JPSJ.34.1289

19. J. C. Preisig, T. F. Duda, Coupled acoustic mode propagation through continental-shelf internal solitary waves, *IEEE J. Ocean. Eng.*, **22** (1997), 256–269. http://dx.doi.org/10.1109/48.585945

20. M. A. Guidry, D. M. Lukin, K. Y. Yang, R. Trivedi, J. Vučković, Quantum optics of soliton microcombs, *Nat. Photon.*, **16** (2022), 52–58. http://dx.doi.org/10.1038/s41566-021-00901-z

21. L. L. Bonilla, M. Carretero, F. Terragni, B. Birnir, Soliton driven angiogenesis, *Sci. Rep.*, **6** (2016), 31296. https://doi.org/10.1038/srep31296

22. R. R. Yuan, Y. Shi, S. L. Zhao, J. X. Zhao, The combined KdV-mKdV equation: Bilinear approach and rational solutions with free multi-parameters, *Results Phys.*, **55** (2023), 107188. http://dx.doi.org/10.1016/j.rinp.2023.107188

23. J. Li, Y. Chen, Solving second-order nonlinear evolution partial differential equations using deep learning, *Commun. Theor. Phys.*, **72** (2020), 105005. http://dx.doi.org/10.1088/1572-9494/aba243

24. J. Li, Y. Chen, A deep learning method for solving third-order nonlinear evolution equations, *Commun. Theor. Phys.*, **72** (2020), 115003. http://dx.doi.org/10.1088/1572-9494/abb7c8

25. Z. Zhou, L. Wang, Z. Yan, Data-driven discoveries of Bäklund transformations and soliton evolution equations via deep neural network learning schemes, *Phys. Lett. A*, **450** (2022), 128373. https://doi.org/10.1016/j.physleta.2022.128373

26. S. F. Sun, S. F. Tian, B. Li, The data-driven rogue waves of the Hirota equation by using Mix-training PINNs approach, *Phys. D*, **465** (2024), 134202. https://doi.org/10.1016/j.physd.2024.134202

27. S. F. Tian, Z. J. Niu, B. Li, Mix-training physics-informed neural networks for high-order rogue waves of cmKdV equation, *Nonlinear Dyn.*, **111** (2023), 16467–16482. https://doi.org/10.1007/s11071-023-08712-3

28. Z. Zhou, Z. Yan, Is the neural tangent kernel of PINNs deep learning general partial differential equations always convergent?, *Phys. D*, **457** (2024), 133987. https://doi.org/10.1016/j.physd.2023.133987

29. Q. Zhang, H. Gao, Z. H. Zhan, J. Li, H. Zhang, Growth optimizer: A powerful metaheuristic algorithm for solving continuous and discrete global optimization problems, *Knowl. Based Syst.*, **261** (2023), 110206. https://doi.org/10.1016/j.knosys.2022.110206

30. T. Lindeberg, Discrete approximations of Gaussian smoothing and Gaussian derivatives, *J. Math. Imaging Vis.*, **66** (2024), 759–800. https://doi.org/10.1007/s10851-024-01196-9

31. N. J. Zabusky, C. J. Galvin, Shallow-water waves, the Korteweg-deVries equation and solitons, *J. Fluid Mech.*, **47** (1971), 811–824. https://doi.org/10.1017/S0022112071001393

32. R. Hirota, *The direct method in soliton theory*, Cambridge: Cambridge University Press, 2004. https://doi.org/10.1017/CBO9780511543043

33. A. Daw, J. Bu, S. Wang, P. Perdikaris, A. Karpatne, Mitigating propagation failures in physics-informed neural networks using retain-resample-release (R3) sampling, *arXiv:2207.02338*, 2023. https://doi.org/10.48550/arXiv.2207.02338

34. M. Bokaeeyan, A. Ankiewicz, N. Akhmediev, Bright and dark rogue internal waves: The Gardner equation approach, *Phys. Rev. E*, **99** (2019), 062224. https://doi.org/10.1103/PhysRevE.99.062224