

Lax pair and symmetric difference constrained PINNs for forward and inverse problems of soliton solutions in the discrete complex mKdV equation

Pei-Kun Yang(杨培坤), Xiao-Yong Wen(闻小永), Jian-Chen Zhou(周建晨)

Citation: Chin. Phys. B. 35, 090205 (2026). DOI: 10.1088/1674-1056/ae66dc |
CSTR: 32038.14.CPB.ae66dc
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

Localized wave solutions and rogue wave solutions for the complex modified KdV equation under the elliptic function background

Xin-Yi Xu(许馨苡), Huan-He Dong(董焕河)
Chin. Phys. B. 35, 090204 (2026). DOI: 10.1088/1674-1056/ae7277 | CSTR: 32038.14.CPB.ae7277

Riemann–Hilbert approach to the reverse space–time nonlocal short pulse equation

Xue Yao(姚雪), Ning Zhang(张宁)
Chin. Phys. B. 35, 080205 (2026). DOI: 10.1088/1674-1056/ae306c | CSTR: 32038.14.CPB.ae306c

Integrable decompositions and superposed nonlinear solutions of (2+1)-dimensional Caudrey–Dodd–Gibbon–Kotera–Sawada equation

Yan Li(李岩), Ruoxia Yao(姚若侠), Senyue Lou(楼森岳)
Chin. Phys. B. 35, 040201 (2026). DOI: 10.1088/1674-1056/ae0435 | CSTR: 32038.14.CPB.ae0435

Prolongation structure and Darboux transformation of nonlinear mixed gas equations

Lixiu Wang(王立秀), Yangjie Jia(加羊杰)
Chin. Phys. B. 34, 070201 (2025). DOI: 10.1088/1674-1056/adcdf1 | CSTR: 32038.14.CPB.adcdf1

Dynamical analysis and localized waves of the n-component nonlinear Schrödinger equation with higher-order effects

Yu Lou(娄瑜), Guoan Xu(许国安)
Chin. Phys. B. 34, 030201 (2025). DOI: 10.1088/1674-1056/ada435 | CSTR: 32038.14.CPB.ada435

SPECIAL TOPIC — Nonlinear localized waves: Theory, methods and applications

Lax pair and symmetric difference constrained PINNs for forward and inverse problems of soliton solutions in the discrete complex mKdV equation

Pei-Kun Yang(杨培坤), Xiao-Yong Wen(闻小永)[†], and Jian-Chen Zhou(周建晨)

School of Applied Science, Beijing Information Science and Technology University, Beijing 100192, *China*

(Received 22 February 2026; revised manuscript received 23 April 2026; accepted manuscript online 30 April 2026)

We propose Lax pair and symmetric difference constrained physics-informed neural networks (LSD-PINNs), designed to study forward and inverse problems for soliton solutions of discrete nonlinear integrable equations (DNIEs). We take the discrete complex modified Korteweg–de Vries (mKdV) equation as an example for numerical simulations and parameter inversion. The results demonstrate that, in the forward problem, the newly proposed method can not only achieve accuracy levels of 10^{-4} for both internal and external predictions, but also successfully obtain the corresponding spectral functions and spectral parameters. Furthermore, under identical conditions, LSD-PINNs exhibits the highest simulation accuracy and stability compared to classical discrete PINNs, symmetry-enhanced deep learning (SDE)-PINNs, and Lax-pair-enhanced (LD)-PINNs. For the inverse problem, LSD-PINNs achieves inversion errors on the order of 10^{-4} for unknown parameters. In addition, under conditions of sparse data, LSD-PINNs is capable of reducing inversion errors by two orders of magnitude compared to classical discrete PINNs. This study offers a novel framework for analyzing discrete soliton solutions and spectral problems within discrete nonlinear integrable equations (DNIEs), providing new insights into integrable phenomena and contributing to the advancement of discrete integrable deep learning.

Keywords: LSD-PINNs, lax pair, symmetric difference, discrete soliton solutions, discrete complex mKdV equation

PACS: 02.30.Ik, 07.05.Mh, 05.45.Yv

DOI: 10.1088/1674-1056/ae66dc **CSTR:** 32038.14.CPB.ae66dc

1. Introduction

Recently, artificial intelligence (AI) has driven significant advancements in fields such as medical modeling,[1,2] drug development,[3] and emotion recognition.[4] Simultaneously, nonlinear partial differential equations (NPDEs) are crucial in mathematics and physics, with applications in fluid dynamics,[5,6] optical fiber communications,[6–8] and ocean engineering.[6,9] However, solving NPDEs is challenging due to their complexity and the high computational demands of numerical simulations, which often suffer from issues of cost and accuracy, particularly under high-dimensional or complex boundary conditions.[10–12] Recently, deep learning techniques, particularly physics-informed neural networks (PINNs),[13] have emerged as innovative computational tools offering a promising alternative for solving complex NPDEs. PINNs integrate data-driven learning with physical constraints by incorporating physical equations, initial and boundary conditions into the loss function, and circumvent the need for traditional mesh-based discretization. Compared to classical numerical methods, PINNs are a mesh-free approach that offers greater flexibility and computational efficiency, and are widely used in fields such as fluid mechanics,[14,15] heat transfer,[16,17] and optical communications.[18,19] However, challenges remain in solving complex problems requiring high accuracy and computational efficiency.[20]

In recent years, significant advances have been achieved in several critical dimensions of PINNs. Innovations in network architecture, such as adaptive activation functions,[21] hierarchical frameworks,[22] combining nonlinear interaction and domain decomposition techniques,[23] and gated recurrent units,[24] have collectively enhanced model expressivity and sequential feature learning. To improve computational efficiency and scalability, researchers have introduced strategies such as parallel computing,[25] hybrid frameworks that combine graph neural networks with finite volume methods,[26] extended PINNs with dual subnets,[27] and convolutional neural networks integrated with local fitting strategies.[28] In terms of training and optimization, developments such as adaptive loss balancing,[29] residual-based attention mechanisms,[30] augmenting the loss function with a mean squared error penalty on the complex conjugate term,[31] weak-form formulations with domain decomposition,[32] and parameter regularization with adjustable weighting coefficients[33] have been implemented to stabilize training and manage high-order or complex NPDEs. Moreover, PINNs have shown significant potential in the study of integrable systems, facilitating the data-

[†]Corresponding author. E-mail: xiaoyongwen@163.com

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

driven discovery of solitons,[34,35] rogue waves,[35–37] and breathers[35,38] in the context of NPDEs.

Meanwhile, integrable systems have also been studied from various perspectives: a deep learning-based polynomial extractor has been developed for the automatic identification of unusual wave modes;[39] the Lax-pair-FIND algorithm has been introduced to discover Lax pairs from sparse or noisy data without requiring prior knowledge of the equation's form;[40] compact Gramian forms and hybrid anomalous-breathing wave solutions have been derived for the complex modified Korteweg–de Vries (mKdV) equation;[41] and theoretical and numerical investigations of topologically trivial magnetic solitons in ferromagnetic nanowires have been performed to elucidate their nonlinear dynamics and controllable generation mechanisms.[42] Collectively, these findings have significantly advanced our understanding of integrable systems, whose intrinsic properties such as Lax pairs and symmetries are now increasingly exploited to improve the performance of PINNs. For instance, PINNs incorporating Miura transformations facilitate the transfer of solutions between different NPDEs,[43] while a two-stage methodology employing conservation laws improves prediction accuracy for complex dynamical systems.[44] Furthermore, Pu *et al.* introduced Lax pair-informed neural networks (LPNNs),[45] which enable high-precision simulations of localized wave phenomena and also allow for the inference of associated spectral parameters and spectral functions. Besides, they incorporated the Darboux transformation (DT) to develop DT-LPNNs to address data-driven localized wave solutions with high precision and uncover new solutions utilizing nontrivial seed solutions.[46] However, the experimental validation of these sophisticated methodologies has primarily been restricted to continuous integrable systems.

In recent research, symmetry has been primarily exploited by minimizing a loss function with additional symmetric soft constraints, thereby enhancing model learning and generalization capabilities.[47–49] A different approach was taken by Zhou *et al.*,[50,51] who introduced symmetric difference data enhancement PINNs (SDE-PINNs). By embedding the symmetric structure directly into the neural networks, their method achieves inherent symmetry and enables highly accurate simulations of discrete solitons and rogue waves. These developments not only improve computational accuracy and training efficiency but also facilitate the discovery of additional integrable phenomena. Consequently, the successful application of integrable features, particularly the Lax pair and symmetry, prompts further inquiry into whether the Lax pair can be incorporated into discrete integrable deep learning systems and whether the integration of symmetry can further enhance model performance.

The primary focus of current research is on continuous NPDEs, with relatively fewer studies investigating the forward and inverse problems related to discrete nonlinear integrable equations (DNIEs). In DNIEs, state variables are defined exclusively at discrete points, thus lacking a continuously differentiable structure, which poses significant challenges for traditional PINNs. To address this challenge, Saqlain *et al.*[52] employed discrete Laplace operators to simulate interactions between adjacent nodes, thereby facilitating difference operations. Additionally, Lin *et al.*[53] introduced pseudo-grids and replicated discrete difference operations using indexing and slicing techniques, while Zhou *et al.*[50] utilized shift operators on discrete datasets to generate neighboring points, enabling efficient computation of forward and backward differences through neural networks.

Based on these previous findings, we will incorporate the Lax pair into the loss function of discrete integrable deep learning models, thereby introducing Lax pair and difference constrained PINNs (LD-PINNs). By embedding symmetry, we further advance this framework by developing Lax pair and symmetric difference constrained PINNs (LSD-PINNs). This study seeks to address the existing paucity of research in the discrete domain and to further the integration of discrete integrable systems with deep learning methodologies.

The experiment in this paper is based on the discrete complex mKdV equation, as delineated in Ref. [54]

$$\begin{aligned}\frac{\mathrm{d}q_n}{\mathrm{d}t} =\ & (1+|q_n|^2)\Big[q_{n+2}-q_{n-2}+2q_{n-1}-2q_{n+1}\\ & +q_n(q^*_{n-1}q_{n+1}-q^*_{n+1}q_{n-1})+|q_{n+1}|^2q_{n+2}\\ & -q^*_n(q^2_{n-1}-q^2_{n+1})-|q_{n-1}|^2q_{n-2}\Big],\end{aligned}\tag{1}$$

where $q_n = q(n,t)$ is a complex-valued function defined on discrete space n and continuous time t, with $|q_n|$ and q^*_n representing its modulus and complex conjugate, respectively. Equation (1) is particularly significant for investigating transverse magnetic wave propagation in nematic optical fibers, as well as in studying the dynamics of specific periodic optical pulses.[55,56] The Lax pair for Eq. (1) is given by[57]

$$\begin{aligned}&\boldsymbol{\Phi}_{n+1} = \begin{pmatrix}\lambda & q_n\\ -q^*_n & \lambda^{-1}\end{pmatrix},\\ &\boldsymbol{\Phi}_n,\ \ \boldsymbol{\Phi}_{n,t} = V_n\boldsymbol{\Phi}_n = \begin{pmatrix}V_{11} & V_{12}\\ V_{21} & V_{22}\end{pmatrix}\boldsymbol{\Phi}_n,\end{aligned}\tag{2}$$

with

$$\begin{aligned}V_{11} =\ & \lambda^{-2}(2-q^*_nq_{n-1})-2q_nq^*_{n-1}+(q_nq^*_{n-1})^2\\ & +q_nq^*_{n-2}+q_{n+1}q^*_{n-1}+q_n|q_{n-1}|^2q^*_{n-2}\\ & -\frac{3}{2}+q_{n+1}|q_n|^2q^*_{n-1}+\lambda^2q_nq^*_{n-1}+\frac{\lambda^4-3\lambda^{-4}}{4},\\ V_{12} =\ & (q_{n-2}|q_{n-1}|^2+q^2_{n-1}q^*_n+q_{n-2}-2q_{n-1})\lambda^{-1}\\ & +\lambda(q^2_nq^*_{n-1}+q_{n+1}|q_n|^2+q_{n+1}-2q_n)\\ & +q_{n-1}\lambda^{-3}+q_n\lambda^3,\end{aligned}$$

$$
\begin{aligned}
V_{21} &= -(q^*_{n-2}|q_{n-1}|^2+q^{*2}_{n-1}q_n+q^*_{n-2}-2q^*_{n-1})\lambda\\
&\quad -\lambda^{-1}(q^{*2}_n q_{n-1}+q^*_{n+1}|q_n|^2+q^*_{n+1}-2q^*_n)\\
&\quad -q^*_{n-1}\lambda^3-q^*_n\lambda^{-3},\\
V_{22} &= \lambda^2(2-q_nq^*_{n-1})-2q^*_nq_{n-1}+(q^*_nq_{n-1})^2+q^*_nq_{n-2}\\
&\quad +q^*_{n+1}q_{n-1}+q^*_n|q_{n-1}|^2q_{n-2}-\frac{3}{2}+q^*_{n+1}|q_n|^2q_{n-1}\\
&\quad +\lambda^{-2}q^*_nq_{n-1}+\frac{\lambda^{-4}-3\lambda^4}{4},
\end{aligned}
$$

where λ is a spectral parameter independent of n and t, while $\Phi_n=(\phi_{1n},\phi_{2n})^{\mathrm{T}}$ is a fundamental solution of Eq. (2). Reference [58] introduced the discrete generalized $(n,N-n)$-fold DT of Eq. (1) to derive discrete localized wave solutions and used asymptotic analysis to examine two-soliton interactions and the limits of rational soliton solutions. In addition, the related Lax pair,[57] Hamiltonian structure, infinite conservation laws,[59] and the continuum limit[60] of Eq. (1) have also been investigated. However, to the best of our knowledge, the use of PINNs for addressing discrete soliton solutions and inverse problems associated with Eq. (1) has not yet been investigated. Therefore, our method is experimentally validated using Eq. (1) as an illustrative example.

The structure of the paper is as follows: Section 2 details the integration of the Lax pair into PINNs for discrete integrable systems and further explores its combination with symmetry property, specifically involving four methods: PINNs, LD-PINNs, SDE-PINNs, and LSD-PINNs. Section 3 provides numerical simulations of the one- and two-soliton solutions of Eq. (1) and compares the four methods. Section 4 explores the inverse problem for the discrete one- and two-soliton solutions of Eq. (1) using LSD-PINNs and PINNs. Finally, Section 5 summarizes the findings of this study and presents concluding remarks.

2. Methodology

In this section, we provide a detailed introduction to the LD-PINNs and LSD-PINNs methods, which consist of three components.

2.1. Discrete difference operator

The general form of DNIEs can be expressed as follows:

$$
\begin{cases}
F(n,t)=\frac{\partial Q_n}{\partial t}-\mathcal{N}_n[Q_{n-1},Q_n,Q_{n+1},\ldots]=0,\\
\quad (n,t)\in\mathcal{D}\times T,\\
Q(n,t_I)=I(n),\quad n\in\mathcal{D},\\
Q(n,t)=B(n,t),\quad (n,t)\in\partial\mathcal{D}\times T,
\end{cases}
\tag{3}
$$

where $\mathcal{D}$ is the spatial domain of the system, and T is the temporal domain, typically $[0,T_{\max}]$. The quantity $Q(n,t)$ depends on the spatial position n and time t, with $\frac{\partial Q_n}{\partial t}$ representing its temporal change. The operator $\mathcal{N}_n$ describes spatial coupling and local dynamics, affecting the evolution of $Q_n(t)$. The initial condition $I(n)$ specifies $Q(n,t)$ at t_I, and the boundary condition $B(n,t)$ defines $Q(n,t)$ on the boundary $\partial\mathcal{D}$, ensuring $Q(n,t)=B(n,t)$.

In contrast to continuous NPDEs, Eq. (3) is defined at discrete spacetime grid points, with its physical equations composed of discrete operators rather than continuous differential operators. Consequently, the mechanism of constructing equation residuals based on automatic differentiation in traditional PINNs is no longer applicable, making it difficult to directly solve Eq. (3). To overcome this limitation, we employ a two-dimensional Hammersley sequence[61] to randomly sample interior data points throughout the entire spatio-temporal domain $\mathcal{D}\times T$. Subsequently, we generate the shifted points $(N-2,T)$, $(N-1,T)$, $(N+1,T)$, and $(N+2,T)$ through spatial shifts. These points are then fed into the neural network, and the resulting outputs $\hat{Q}(N\pm i,T)$ are utilized to perform discrete difference operations, denoted by $E\pm i$ and defined as

$$
E\pm i=\hat{Q}(N\pm i,T)-\hat{Q}(N,T),\quad i=1,2,\ldots. \tag{4}
$$

Thus, by incorporating discrete difference operations, traditional PINNs can be effectively applied to the study of Eq. (3).

2.2. LD-PINNs

In 1968, Lax introduced the notion of the Lax pair and developed a general method for representing integrable systems using Lax pairs.[62] As a defining characteristic of integrable systems, the Lax pair provides a crucial framework for exploring their integrability, conservation laws, and classification. In the theory of integrable systems, the spectral parameter λ and the associated spectral function Φ_n are fundamental objects in the Lax formulation. They determine the scattering data for the Lax eigenvalue problem and form the basis of the inverse scattering transform (IST), which solves the Cauchy problem by linearising the time evolution. Moreover, they play a central role in the DT, which uses them as the key ingredients to construct multi-soliton, breather, and rogue wave solutions from a given seed solution. In spatially semi-discrete integrable systems, the Lax pair presented in this paper is expressed in the following matrix form:

$$
\begin{cases}
\Phi_{n+1}=L_n(\lambda)\Phi_n,\\
\dfrac{\mathrm{d}\Phi_n}{\mathrm{d}t}=M_n(\lambda)\Phi_n,
\end{cases}
\tag{5}
$$

where λ is a spectral parameter independent of n and t. The matrices $L_n(\lambda)$ and $M_n(\lambda)$ are $n\times n$ matrices depending on λ and the potential $q(x,t)$. The vector $\Phi_n=(\phi_{1n}(t),\phi_{2n}(t),\phi_{3n}(t),\ldots)^{\mathrm{T}}$ is an n-dimensional column vector. Moreover, starting from Eq. (5), if $n=2$, one can derive the

associated DNIEs as shown in Eq. (3) by using the following zero-curvature equation

$$L_{n,t} = M_{n+1}L_n - L_nM_n, \tag{6}$$

where $L_{n,t}$ represents the time derivative of L_n, i.e., $\partial L_n/\partial t$.

In Eq. (3), PINNs integrate initial and boundary conditions with physical constraints to predict the solution $\hat{Q}(n,t)$ by minimizing a designated loss function. Building upon this framework, we introduce a novel approach termed LD-PINNs, as illustrated in Fig. 1, which outputs the eigenfunction $\Phi_n = (\phi_{1n}, \phi_{2n})^{\mathrm{T}}$ via the neural network. Furthermore, we incorporate the spectral parameter λ, along with its associated Lax pair, as an additional constraint denoted by $F_{\mathrm{Lp}}(n,t;\theta,\lambda)$ within the loss function. By optimizing the network parameters θ and spectral parameter λ, the predicted solution $Q(n,t)$ and spectral function are obtained. The loss function is composed of three components: the DNIE residual, denoted by $\mathcal{L}_{\mathrm{DNIE}}(\theta)$; the Lax pair residual, denoted by $\mathcal{L}_{\mathrm{Lp}}(\theta,\lambda)$; and the initial and boundary data loss, denoted by $\mathcal{L}_{\mathrm{Data}}(\theta)$, and is defined as follows:

$$\mathcal{L}_{\mathrm{LSD-PINNs}}(\theta,\lambda) = \mathcal{L}_{\mathrm{DNIE}}(\theta) + \mathcal{L}_{\mathrm{Lp}}(\theta,\lambda) + \mathcal{L}_{\mathrm{Data}}(\theta), \tag{7}$$

with

$$\begin{cases} \mathcal{L}_{\mathrm{DNIE}}(\theta) = \dfrac{1}{N_{\mathrm{F}}}\displaystyle\sum_{i=1}^{N_{\mathrm{F}}}\left[\hat{F}_{\mathrm{Eq}}\left(n_{\mathrm{F}}^i, t_{\mathrm{F}}^i;\theta\right)\right]^2, \\ \mathcal{L}_{\mathrm{Lp}}(\theta) = \dfrac{1}{N_{\mathrm{F}}}\displaystyle\sum_{i=1}^{N_{\mathrm{F}}}\left[\hat{F}_{\mathrm{Lp}}\left(n_{\mathrm{F}}^i, t_{\mathrm{F}}^i;\theta,\lambda\right)\right]^2, \\ \mathcal{L}_{\mathrm{Data}}(\theta) = \dfrac{1}{N_{\mathrm{IB}}}\displaystyle\sum_{i=1}^{N_{\mathrm{IB}}}\left[\hat{Q}\left(n_{\mathrm{IB}}^i, t_{\mathrm{IB}}^i;\theta\right) - Q\left(n_{\mathrm{IB}}^i, t_{\mathrm{IB}}^i\right)\right]^2, \end{cases} \tag{8}$$

where θ represents the weight and bias parameters of the neural network. N_{F} and N_{IB} denote the total numbers of internal train points, initial and boundary points, respectively. The sets $\{n_{\mathrm{F}}^i, t_{\mathrm{F}}^i\}_{i=1}^{N_{\mathrm{F}}}$ comprise points that are randomly sampled within the spatial and temporal domains, while $\{n_{\mathrm{IB}}^i, t_{\mathrm{IB}}^i\}_{i=1}^{N_{\mathrm{IB}}}$ pertain to the datasets associated with the initial and boundary conditions.

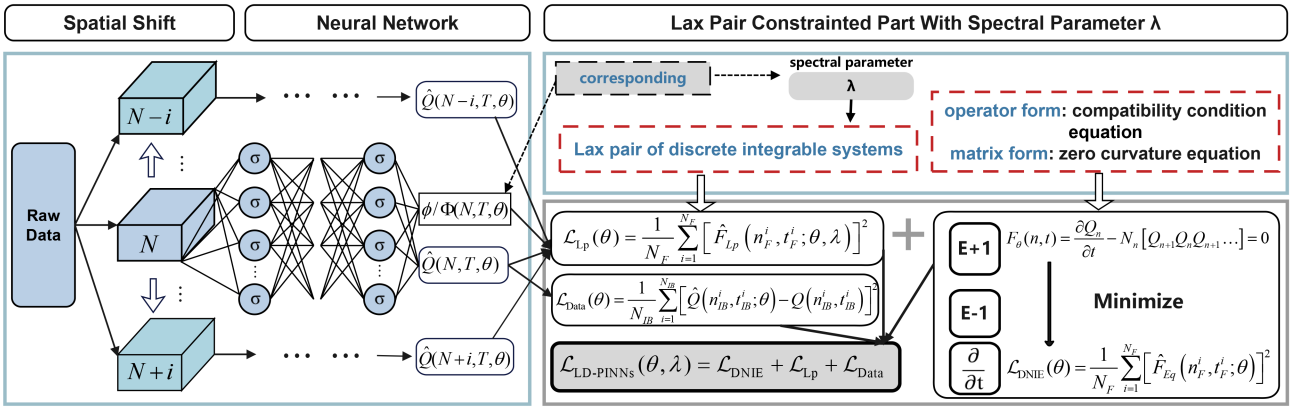


Fig. 1. The network architecture of LD-PINNs. Building upon spatially shifted raw data, the Lax pair is successfully incorporated into discrete integrable deep learning by adding the transformed Lax pair $F_{\mathrm{Lp}}(n,t;\theta,\lambda)$ to the loss function.

2.3. LSD-PINNs

According to Ref. [51], we integrate symmetry by combining outputs at symmetric points at the network's output layer. For a point $P(n_0,t_0)$, where n_0 and t_0 represent its spatial and temporal coordinates, respectively, the symmetric counterpart of this point can be described as $(2n_0 - N, 2t_0 - T)$ in the presence of both spatial and temporal symmetry. In contrast, if only time-reversal symmetry is present, the symmetric point is represented as $(n_0, -t_0)$. To effectively utilize this symmetry information, the network's outputs are adjusted to maintain consistency with the symmetry around $P(n_0,t_0)$, resulting in the following modified data points (N,T):

$$\hat{Q}(N,T;\theta) = \frac{L(N,T;\theta) + L(2n_0 - N, 2t_0 - T;\theta)}{2}, \tag{9}$$

where L denotes the output of the neural network, and the network is designed to enforce symmetry about the point $P(n_0,t_0)$.

Furthermore, the influence of initial symmetric point selection on the performance of the method is comprehensively examined in Ref. [51]. In this paper, the midpoint between the training boundary and the true symmetric point is selected as the initial symmetric point. Moreover, the existing SDE-PINNs primarily rely on global symmetry. Future studies should explore local and superimposed symmetry to further enhance and optimize the method. Therefore, based on the symmetric architecture described in Ref. [51], we introduce a novel approach termed LSD-PINNs by combining symmetry with LD-PINNs, as illustrated in Fig. 2. Firstly, a differential dataset is constructed using $E\pm i$ applied to the raw data, while a symmetric dataset is generated according to Eq. (9). Both datasets, along with spectral functions, serve as outputs for the neural network. The training of the neural network is subsequently conducted by minimizing Eq. (7).

The precise handling of boundary conditions is essen-

tial for generating reliable solution predictions, as highlighted in previous studies.[63,64] In our experimental analysis, the boundary loss converges to a minimal constant value. Consequently, in alignment with the methodology outlined in Ref. [51], we apply soft and hard constraints to the boundary conditions, respectively, and subsequently analyze the effects of these two approaches.

The network comprises a two-dimensional (2D) input layer, followed by three fully connected hidden layers, each consisting of 64 neurons. It uses the Tanh activation function and Xavier initialization for weight initialization.[65] Based on the optimization concept presented in Ref. [66], the training process utilizes both the Adam and L-BFGS optimizers to enhance accuracy and accelerate convergence.

In addition, the efficacy of these methods is generally assessed through validation points that are uniformly distributed across the entire spatiotemporal domain. The predictive accuracy of the methods is evaluated using the relative error based on the L^2-norm, which measures the deviation between the predicted outcomes and the reference solutions, defined as

$$\mathrm{RE} = \frac{\|\hat{Q}(n,t) - Q(n,t)\|_2}{\|Q(n,t)\|_2}, \tag{10}$$

where $\hat{Q}(n,t)$ denotes the predicted value at the point (n,t), while $Q(n,t)$ denotes the true value. Furthermore, to evaluate the model's predictive performance more comprehensively, we define the relative error of internal prediction ($\mathrm{RE_{Ip}}$) and the relative error of external prediction ($\mathrm{RE_{Ep}}$) in the training and untrained regions, respectively.

The code in this article is implemented in Python 3.12.9 and PyTorch 2.6.0. The numerical experiments are conducted on a high-performance computer equipped with an NVIDIA GeForce RTX 4070 GPU.

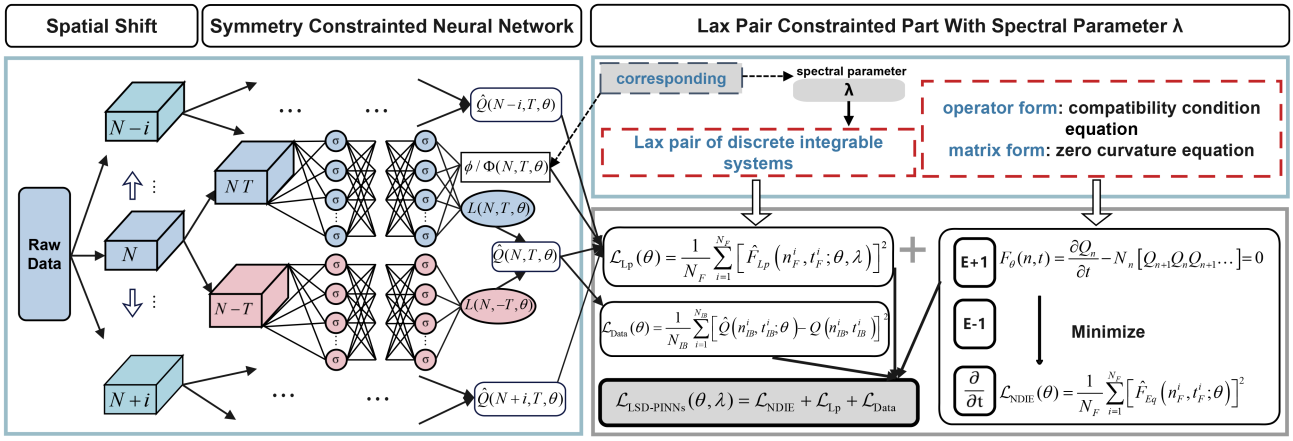


Fig. 2. The network architecture of LSD-PINNs. Symmetry is successfully incorporated into LD-PINNs through symmetric outputs at the final layer of the neural network.

3. Numerical simulation of Eq. (1) via LSD-PINNs

In this section, we employ PINNs,[13] LD-PINNs, SDE-PINNs,[51] and LSD-PINNs to conduct numerical simulations of the discrete one- and two-soliton solutions of Eq. (1). The learning performance of each method is assessed using 20 different random seeds. Furthermore, the training process and dynamic visualization are illustrated using the results obtained from the LSD-PINNs method.

3.1. Approximation of one-soliton solution

In the study conducted by Ref. [58], the precise one-soliton solution associated with Eq. (1) is presented as

$$q_n = \frac{1-|\lambda_1|^4}{2|\lambda_1|\lambda_1^*}\,\mathrm{sech}\left(\xi_1 + 2\ln\lambda_1 - \ln|\lambda_1|\right), \tag{11}$$

where $\xi_1 = (2\ln\lambda_1)n + \frac{\lambda_1^8 - 2\lambda_1^6 + 2\lambda_1^2 - 1}{\lambda_1^4}t$ $(\lambda_1 > 0)$.

In this study, we choose $\lambda_1 = \frac{3}{4}$ and define $R = \mathcal{D} \times T = [-15,15] \times [-3.5,3.5]$. The processes of training data sampling, model training, and prediction are all executed within the region $R_{\mathrm{in}} = [-15,15] \times [-3,3]$. The training data points are generated using a two-dimensional Hammersley sampling technique. In contrast, the region $R_{\mathrm{out}} = [-15,15] \times ([-3.5,-3] \cup [3,3.5])$ is employed to assess the method's generalization performance. The loss function is subsequently formulated by concurrently integrating the initial condition $I(n)$, the boundary condition $B(n,t)$, and the system of residual equations $\{F_{\mathrm{Eq}}(n,t;\theta)$ and $F_{\mathrm{Lp}}(n,t;\theta,\lambda)\}$. Here, $I(n)$ is derived from Eq. (11) evaluated at $t = -3.0$, while $B(n,t)$ is determined by applying Eq. (11) at the spatial boundaries $n = \pm 15$. The symmetric point P_{s} is initialized at coordinates $(-0.50, -0.50)$, while its temporal component t_{s} is allowed to evolve adaptively. The spectral parameter λ is initially set to 1.0, with the numbers of internal sampling points and initial-boundary points designated as $N_{\mathrm{F}} = 5000$ and $N_{\mathrm{IB}} = 150$, respectively.

Initially, we evaluate the performance of individual loss

components under both hard and soft constraints, as illustrated in Fig. 3. The training process for each loss component exhibits similar patterns under both constraint conditions; however, the hard constraint facilitates more rapid convergence. Specifically, the method achieves convergence after 12 iterations with a computational time of 23.17 s under the hard constraint. In contrast, the soft constraint requires 15 iterations and 27.62 s. The magnified view of the evolution of the equation residual $F_{\mathrm{Eq}}(n,t;\theta)$ shows that, under the soft constraint, Loss F continues to update beyond the 12th epoch, although the final convergence outcomes are comparable.

Under soft constraints, the learned one-soliton solution achieves $\mathrm{RE_{Ip}} = 3.30849 \times 10^{-4}$ and $\mathrm{RE_{Ep}} = 6.15725 \times 10^{-4}$, when compared to the exact solution within the designated spatiotemporal domain. In contrast, under hard constraints, the respective errors are $\mathrm{RE_{Ip}} = 3.70988 \times 10^{-4}$ and $\mathrm{RE_{Ep}} = 6.53604 \times 10^{-4}$. These values are nearly equivalent across both constraint types, suggesting that the primary impact of the constraint choice lies in the method's convergence rate. Consequently, hard constraints are consistently employed in subsequent experiments. The dynamical behavior associated with the one-soliton numerical simulation is illustrated in Fig. 4.

Figures 4(a)–4(d) illustrate the solution density and its associated error maps, the propagation patterns of a data-driven one-soliton at various temporal instances, the loss function and t_{s} variation curve, and a three-dimensional visualization, respectively. As illustrated in Fig. 4(b), the predicted values at three distinct temporal points exhibit a high degree of congruence with the true values. Furthermore, as illustrated in Fig. 4(c), following the initial 10 iterations using the Adam

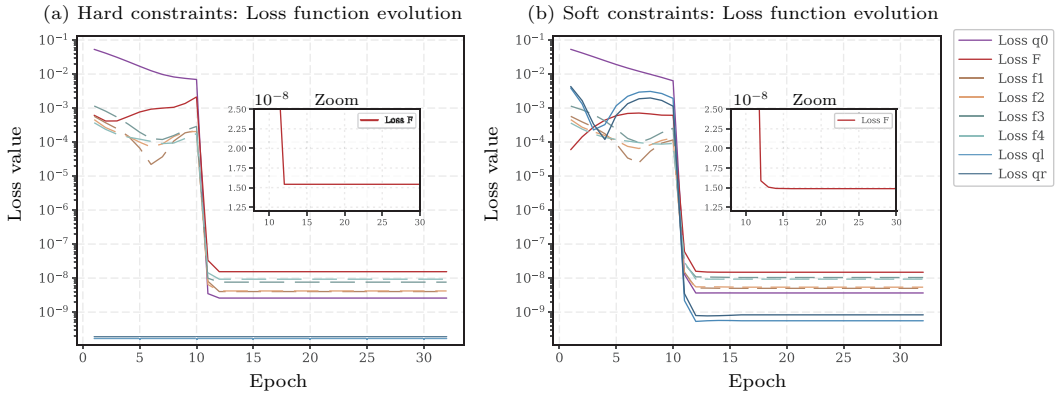


Fig. 3. (a) and (b) show the changes in different loss components of the one-soliton solution under hard and soft constraints for Eq. (1), respectively. The loss function includes the initial condition (Loss q0), boundary conditions (Loss ql and Loss qr), and residual equation losses (Loss F for $F_{\mathrm{Eq}}(n,t;\theta)$ and Loss $\{f_k | k \in \{1,2,3,4\}\}$ for $F_{\mathrm{Lp}}(n,t)$).

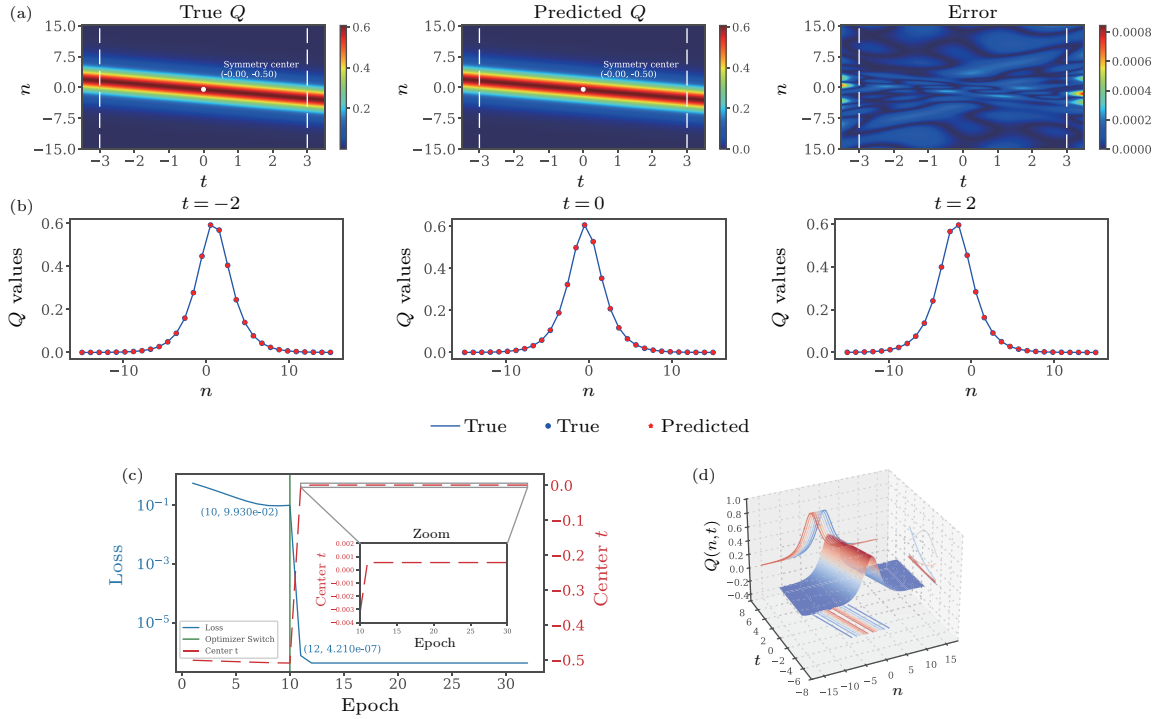


Fig. 4. The comprehensive numerical simulation results for the one-soliton solution of Eq. (1): (a) The visualization of the solution alongside the error density; (b) The propagation curves at three selected time points; (c) The evolution of the loss function and t_{s}; (d) A 3D representation of the solution.

optimizer, the loss function stabilizes at 9.92995×10^{-1}. Subsequently, upon transitioning to the L-BFGS optimizer, the loss exhibits a rapid decline to 4.21043×10^{-7}. Simultaneously, the temporal component t_s of the symmetric point is observed to converge toward the true value $(t = 0)$, a detail that can be further scrutinized in the magnified view. Ultimately, the symmetric point evolves to $P_s(-0.50, 5.50 \times 10^{-4})$. These findings substantiate the method's efficacy in accurately capturing complex nonlinear evolution.

LSD-PINNs not only achieve a high-precision solution for the one-soliton solution to Eq. (1), but also enable the numerical determination of the spectral parameter and the corresponding spectral functions, which are typically inaccessible through PINNs. As illustrated in Fig. 5(a), the estimated λ is 0.9926. Subfigures 5(b1) and (5b2) present the associated learned spectral functions, denoted as ϕ_{1n} and ϕ_{2n}, respectively.

As a defining characteristic of integrable systems, the Lax pair provides a crucial framework for exploring their integrability, conservation laws, and classification. The numerically obtained spectral parameter and spectral functions are integral to classic theories such as the IST, DT, and algebraic-geometric theory. This method has the potential to yield new insights within these classical methods and establish a vital foundation for the discovery of new Lax pairs and physical equations, thereby advancing the integration of discrete integrable systems with deep learning.

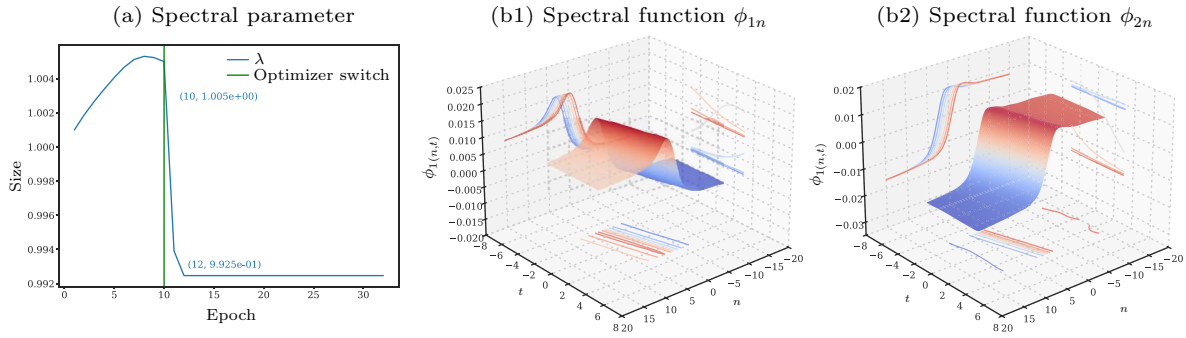


Fig. 5. Learned spectral parameter and functions: (a) The variation of λ; (b1) and (b2) The three-dimensional plots with contour maps of ϕ_{1n} and ϕ_{2n}.

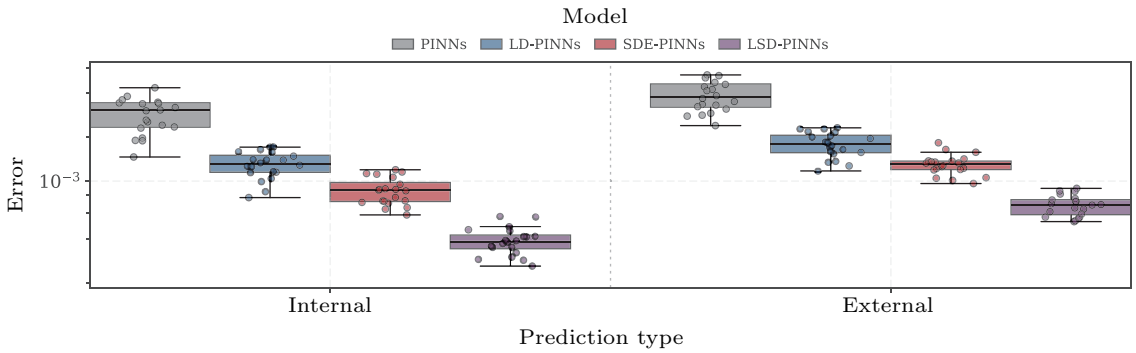


Fig. 6. Performance comparison boxplot of the four methods for solving the one-soliton solution of Eq. (1).

Table 1. Performance comparison of the four methods for obtaining the one-soliton solution of Eq. (1).

Model	L2 relative error $\times 10^{-4}$ (mean $\pm$ std)			Time (s)
	RE_{Ip}	RE_{Ep}	RE_{Total}	
PINNs	**33.53 ± 22.69**	**45.94 ± 33.60**	**35.72 ± 24.56**	**14.94**
LD-PINNs	13.06 ± 2.63	17.93 ± 3.44	13.94 ± 2.57	41.11
SDE-PINNs	8.62 ± 1.81	13.0 ± 2.08	9.46 ± 1.58	8.70
LSD-PINNs	**3.92 ± 0.83**	**6.84 ± 1.09**	**4.50 ± 0.74**	**29.39**

To assess the learning performance of the four methods (PINNs, LD-PINNs, SDE-PINNs, and LSD-PINNs) for the one-soliton solution of Eq. (1), we employ the same $\mathcal{D} \times T$ and training configuration as previously established. We conduct 20 experiments with different random seeds for each method. Table 1 presents the mean and standard deviation of RE_{Ip}, RE_{Ep}, and the total relative error RE_{Total} for each method across all experiments, as well as the average computation time for each method.

As listed in Table 1, although the training duration for LSD-PINNs is slightly longer than that of PINNs, the training accuracy exhibits an improvement by an order of magnitude across the ranges of R_{in} and R_{out}. Notably, the standard deviation of RE_{Ip} for LSD-PINNs is reduced by 50% compared to that of PINNs. Conversely, while LD-PINNs reduce the evaluation error across all ranges relative to PINNs, they require the most computational time. In contrast, SDE-PINNs accelerate convergence but yield approximately 50% lower prediction accuracy in R_{in} and R_{out} compared to LSD-PINNs. These

findings are also supported by Fig. 6. In conclusion, LSD-PINNs offer superior predictive performance and enhanced stability with only a marginal increase in training time, highlighting their potential utility in high-accuracy scientific computing and engineering simulations, and further affirming the significance of our research.

3.2. Approximation of two-soliton solution

Similar to Subsection 3.1, the precise two-soliton solution of Eq. (1) is presented in Ref. [58] as

$$q_n = \frac{M_1 \cosh!\left(\xi_1 + \ln|\lambda_1| + \ln\frac{\lambda_1^2-\lambda_2^2}{|\lambda_1^2-\lambda_2^2|} + \ln\frac{|\lambda_1^2\lambda_2^{*2}-1|}{\lambda_1^2\lambda_2^{*2}-1}\right) - M_2 \cosh!\left(\xi_2 + \ln|\lambda_2| + \ln\frac{\lambda_1^2-\lambda_2^2}{|\lambda_1^2-\lambda_2^2|} - \ln\frac{|\lambda_1^2\lambda_2^{*2}-1|}{\lambda_1^2\lambda_2^{*2}-1}\right)}{N_1\cosh(N_2+\xi_2+2\ln\lambda_2-\ln|\lambda_2|)+N_3\cosh(N_2-\xi_2-2\ln\lambda_2+\ln|\lambda_2|)+N_4}, \tag{12}$$

where

$$\begin{aligned}
\xi_i &= (2\ln\lambda_i)n + \frac{\lambda_i^8 - 2\lambda_i^6 + 2\lambda_i^2 - 1}{\lambda_i^4}t \quad (i=1,2),\\
M_1 &= \lambda_1|\lambda_1|(|\lambda_2|^4-1)|\lambda_1^2\lambda_2^{*2}-1||\lambda_1^2-\lambda_2^2|,\\
M_2 &= \lambda_2|\lambda_2|(|\lambda_1|^4-1)|\lambda_2^2\lambda_1^{*2}-1||\lambda_1^2-\lambda_2^2|,\\
N_1 &= -\lambda_1^*\lambda_2^*|\lambda_1||\lambda_2||\lambda_1^2-\lambda_2^2|^2,\\
N_2 &= \xi_1 + 2\ln\lambda_1 - \ln|\lambda_1|,\\
N_3 &= -\lambda_1^*\lambda_2^*|\lambda_1||\lambda_2||\lambda_1^2\lambda_2^{*2}-1|^2,\\
N_4 &= \frac{1}{2}(\lambda_1^2\lambda_1^{*2}-1)(\lambda_2^2\lambda_2^{*2}-1)\left(|\lambda_1|^2|\lambda_2|^2+\lambda_1^{*2}\lambda_2^{*2}\right).
\end{aligned}$$

In this study, we choose $\lambda_1 = \frac{2}{3}$, $\lambda_2 = 2$ and define $R = \mathcal{D}\times T = [-10,10]\times[-1.2,1.2]$. The processes of training data sampling, model training, and prediction are all executed within the region $R_{\text{in}} = [-10,10]\times[-1.0,1.0]$. The training data points are generated using a two-dimensional Hammersley sampling technique. In contrast, the region $R_{\text{out}} = [-10,10]\times([-1.2,-1.0]\cup[1.0,1.2])$ is employed to assess the method's generalization performance. The loss function is subsequently formulated by concurrently integrating the initial condition $I(n)$, the boundary condition $B(n,t)$, and the system of residual equations $\{F_{\text{Eq}}(n,t;\theta), F_{\text{Lp}}(n,t;\theta,\lambda)\}$. Here, $I(n)$ is derived from Eq. (12) evaluated at $t=-0.01$, while $B(n,t)$ is determined by applying Eq. (12) at the spatial boundaries $n=\pm10$. The symmetric point P_{s} is initialized at coordinates $(-0.50,-0.50)$, while its temporal component t_{s} is allowed to evolve adaptively. The spectral parameter λ is initially set to 1.0, with the numbers of internal sampling points and initial-boundary points designated as $N_{\text{F}} = 5000$ and $N_{\text{IB}} = 150$, respectively.

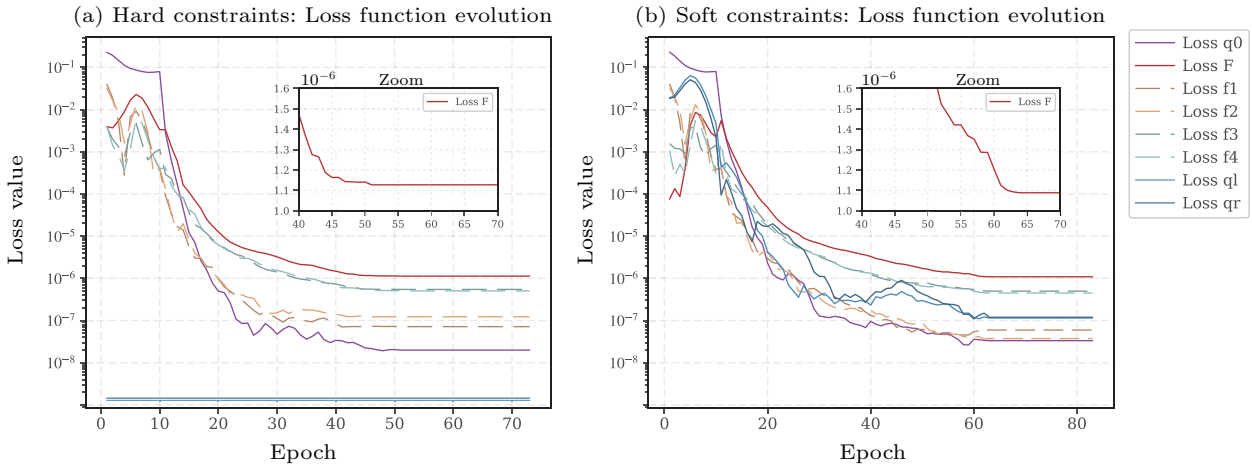


Fig. 7. (a) and (b) show the changes in different loss components of the two-soliton solution under hard and soft constraints for Eq. (1), respectively. The loss function includes the initial condition (Loss q0), boundary conditions (Loss ql and Loss qr), and residual equation losses (Loss F for $F_{\text{Eq}}(n,t;\theta)$ and Loss $\{f_k|k\in\{1,2,3,4\}\}$ for $F_{\text{Lp}}(n,t)$).

Similar to Subsection 3.1, we assess each loss component under hard and soft constraints in Fig. 7. The training processes for each loss component exhibit similar patterns across both constraint conditions; however, the hard constraint facilitates a more rapid convergence. Specifically, the method achieves convergence after 52 iterations with a computational time of 417.61 s under the hard constraint. In contrast, the soft constraint requires 63 iterations and 473.20 s. To analyze the variation in $F_{\text{Eq}}(n,t;\theta)$ in greater detail, a closer inspection clearly shows that the method under the hard constraint converges faster, yet ultimately attains results comparable to those under the soft constraint.

In the initial stages of training, distinct peaks are observable in Figs. 3 and 7. Notably, the pronounced peak at the 10th epoch is largely due to the transition from the Adam optimizer to L-BFGS. At this juncture, the Adam optimizer had

advanced the parameters to the periphery of a local minimum basin, resulting in a marked reduction in the gradient norm, despite the incomplete utilization of curvature information. In this context, L-BFGS leveraged local curvature information to formulate quasi-Newton steps,thereby altering the local optimisation landscape faced by the optimiser. Furthermore, during the early training phases, the model primarily addresses the data and Lax equation losses. As the equation loss escalates, the optimizer explores alternative parameter configurations to equilibrate all loss components, which may temporarily lead to an increase in certain losses.

Under soft constraints, the learned two-soliton solution achieves $\mathrm{RE_{Ip}} = 6.06141 \times 10^{-4}$ and $\mathrm{RE_{Ep}} = 2.642709 \times 10^{-3}$. Correspondingly, under hard constraints, the respective errors are $\mathrm{RE_{Ip}} = 5.85899 \times 10^{-4}$ and $\mathrm{RE_{Ep}} = 2.25399 \times 10^{-3}$. These values are nearly equivalent across both constraint types, suggesting that the primary impact of the constraint choice lies in the method's convergence rate. Consequently, hard constraints are consistently employed in subsequent experiments. The dynamical behavior associated with the two-soliton numerical simulation is illustrated in Fig. 8.

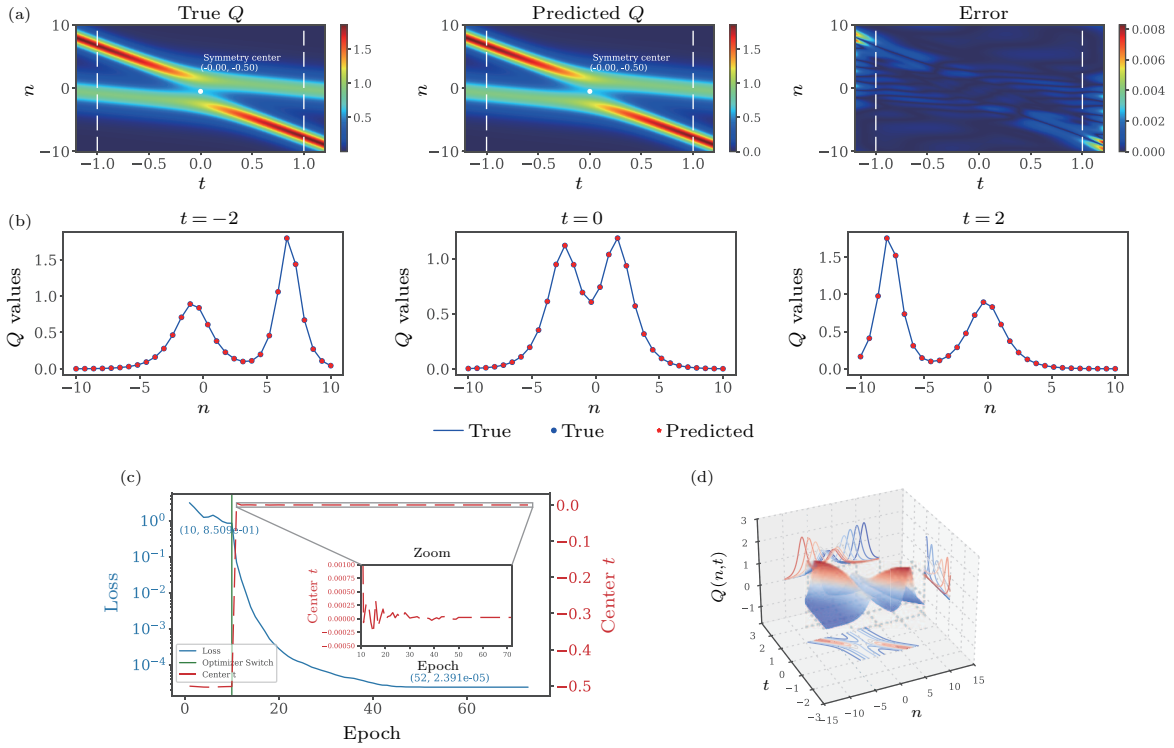


Fig. 8. The comprehensive numerical simulation results for the discrete two-soliton solution of Eq. (1): (a) The visualization of the solution alongside the error density; (b) The propagation curves at three selected time points; (c) The evolution of the loss function and t_s; (d) A 3D representation of the solution.

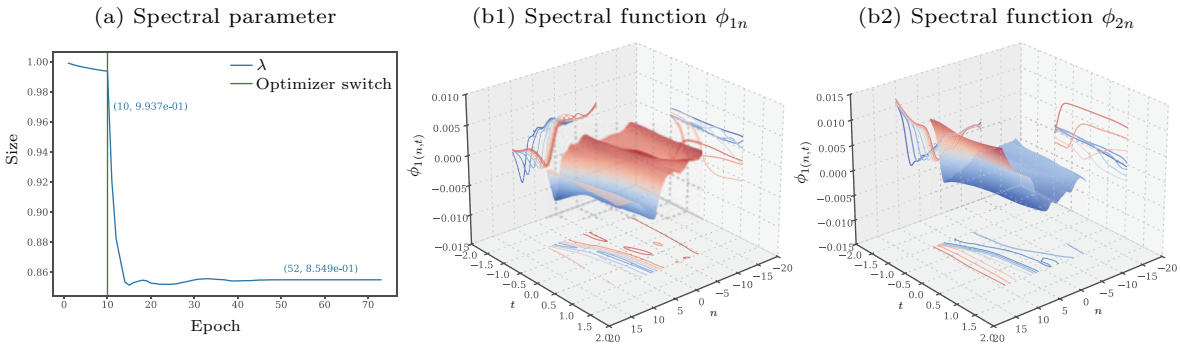


Fig. 9. Learned spectral parameter and functions: (a) The variation of λ; (b1) and (b2) The three-dimensional plots with contour maps of ϕ_{1n} and ϕ_{2n}.

Figures 8(a)–8(d) illustrate the solution density and its associated error maps, the propagation patterns of a data-driven two-soliton at various temporal instances, the loss function and t_s variation curve, and a three-dimensional visualization, respectively. As illustrated in Fig. 8(b), the predicted values at three distinct temporal points exhibit a high degree of congruence with the true values. Furthermore, as illustrated in Fig. 8(c), following the initial 10 iterations using the Adam

optimizer, the loss function stabilizes at 8.50889×10^{-1}. Subsequently, upon transitioning to the L-BFGS optimizer, the loss exhibits a rapid decline to 2.39092×10^{-5}. Simultaneously, the temporal component t_s of the symmetric point is observed to converge toward the true value ($t = 0$), a detail that can be further scrutinized in the magnified view. Ultimately, the symmetric point evolves to $P_s(-0.50, 2.24 \times 10^{-5})$. LSD-PINNs also enable the numerical determination of the spectral parameter and the corresponding spectral functions. As illustrated in Fig. 9(a), the estimated λ is 0.8486. Subfigures 9(b1) and 9(b2) present the associated learned spectral functions, denoted as ϕ_{1n} and ϕ_{2n}, respectively. These findings substantiate the method's efficacy in accurately capturing complex nonlinear evolution.

Similar to Subsection 3.1, a comprehensive evaluation of the learning performance of the four methods (PINNs, LD-PINNs, SDE-PINNs, and LSD-PINNs) on the two-soliton solution of Eq. (1) is conducted through 20 independent runs with randomized seeds. Table 2 presents the mean and standard deviation of $\mathrm{RE_{Ip}}$, $\mathrm{RE_{Ep}}$, and $\mathrm{RE_{Total}}$ for each method across all experiments, as well as the average computation time per method.

As listed in Table 2, the training duration of LSD-PINNs exhibits a substantial increase compared to that of PINNs, while the training accuracy in the regions of R_{in} and R_{out} demonstrates an improvement by an order of magnitude. In terms of stability, as measured by the $\mathrm{RE_{Ip}}$ metric, LSD-PINNs enhances training accuracy by approximately 38%, even when predictions from both LSD-PINNs and PINNs are evaluated at the two extremes. Although SDE-PINNs continues to exhibit faster convergence, it falls short of LSD-PINNs in both predictive performance and stability, as indicated by the $\mathrm{RE_{Ip}}$ and $\mathrm{RE_{Ep}}$ metrics. These findings are also supported by Fig. 10.

Table 2. Performance comparison boxplot of the four methods for obtaining the two-soliton solution of Eq. (1).

Model	L2 relative error $\times 10^{-4}$ (mean $\pm$ std)			Time (s)
	$\mathrm{RE_{Ip}}$	$\mathrm{RE_{Ep}}$	$\mathrm{RE_{Total}}$	
PINNs	**20.71 ± 8.64**	**55.64 ± 21.31**	**30.18 ± 11.72**	**168.93**
LD-PINNs	31.30 ± 14.08	71.94 ± 36.08	41.69 ± 19.68	361.80
SDE-PINNs	13.06 ± 6.83	36.49 ± 15.48	19.50 ± 8.87	137.70
LSD-PINNs	**6.34 ± 1.17**	**20.50 ± 6.10**	**10.49 ± 2.42**	**396.21**

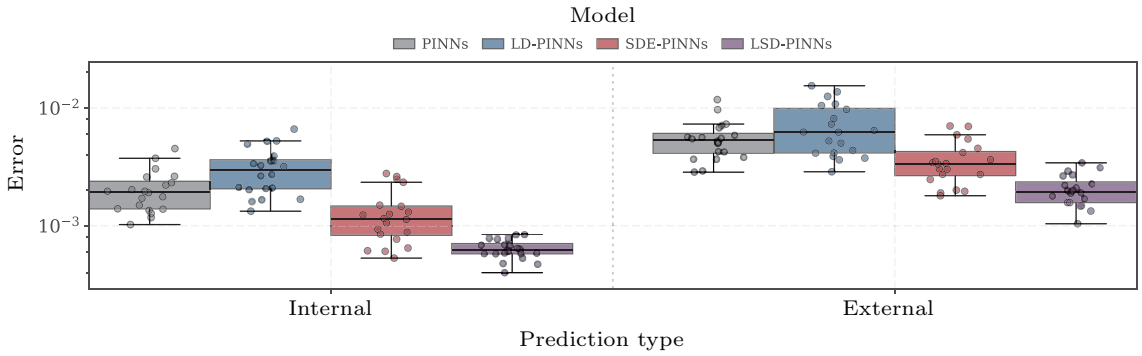


Fig. 10. Performance comparison boxplot of the four methods for solving the two-soliton solution of Eq. (1).

Furthermore, it has been observed that while LD-PINNs demonstrate superior performance in solving one-soliton solutions compared to PINNs, their effectiveness diminishes when addressing two-soliton solutions. This decline in performance can be attributed to the integration of the Lax pair into $F_{\mathrm{Lp}}(n,t;\theta,\lambda)$ and its subsequent incorporation into $\mathcal{L}_{\mathrm{LSD-PINNs}}(\theta,\lambda)$. This process imposes stringent physical constraints derived from integrable system theory on PINNs, thereby restricting the functional search space. As the complexity of the solution space increases, the optimization objective defined by $\mathcal{L}_{\mathrm{LSD-PINNs}}(\theta,\lambda)$ may encompass a larger number of local minima. In regions of heightened complexity, such as the two-soliton interaction zone, the presence of competing gradients among different loss terms can complicate the optimization process, making it challenging to achieve equilibrium. In contrast, PINNs, with their simpler loss function $\mathcal{L}_{\mathrm{PINNs}}(\theta)$, may offer a more straightforward optimization trajectory, leading to improved convergence. This observation further underscores the utility of integrating Lax pair information with symmetry in LSD-PINNs.

4. Parameter inversion for Eq. (1)

In this section, we examine the inverse problem of determining the unknown parameters in Eq. (1), which is subsequently updated as follows:

$$\begin{aligned}\frac{\mathrm{d}q_n}{\mathrm{d}t} = {} & (1+|q_n|^2)\Big\{w_1(q_{n+2}-q_{n-2}+2q_{n-1}-2q_{n+1}) \\ & +w_2[q_n(q^*_{n-1}q_{n+1}-q^*_{n+1}q_{n-1}) \\ & +|q_{n+1}|^2q_{n+2}-q^*_n(q^2_{n-1}-q^2_{n+1}) \\ & -|q_{n-1}|^2q_{n-2}]\Big\},\end{aligned} \tag{13}$$

where the parameters w_1 and w_2 are the primary focus for estimation. The experimental data are generated utilizing the

Hammersley sampling method, with varying levels of Gaussian noise applied. To mitigate the effects of noise, both the labeled data and the unknown parameters are incorporated into the loss function. Specifically, w_1 and w_2 replace the two coefficients in Eq. (1), which have true values of 1.0, and are incorporated into the modified loss function. As a result, the loss function is optimized through iterative training, during which the unknown parameters are concurrently estimated. The loss function is updated as follows:

$$\mathcal{L}(\theta) = \mathcal{L} * \mathrm{F}(\theta, \lambda, w_1, w_2) + \mathcal{L} * \mathrm{D}(\theta, w_1, w_2), \tag{14}$$

where $\mathcal{L} * \mathrm{F}(\theta, \lambda, w_1, w_2)$ denotes the combined DNIE and Lax pair residual losses, $\mathcal{L} * \mathrm{DNIE}(\theta)$ and $\mathcal{L} * \mathrm{Lp}(\theta, \lambda)$, respectively, while $\mathcal{L} * \mathrm{D}(\theta, w_1, w_2)$ represents the data loss, including initial, boundary, and internal point losses. Their specific definitions are as follows:

$$\begin{aligned}
&\mathcal{L} * \mathrm{F}(\theta, \lambda, w_1, w_2) = \\
&\quad \frac{1}{N * \mathrm{F}} \sum_{i=1}^{N_{\mathrm{F}}} \Bigg(\left[\hat{F} * \mathrm{Eq}\left(n * F^i, t_{\mathrm{F}}^i; \theta, w_1, w_2\right) \right]^2 \\
&\quad + \left[\hat{F} * \mathrm{Lp}\left(n_{\mathrm{F}}^i, t_{\mathrm{F}}^i; \theta, \lambda, w_1, w_2\right) \right]^2 \Bigg),
\end{aligned}$$

$$\begin{aligned}
\mathcal{L} * \mathrm{D}(\theta, w_1, w_2) = \frac{1}{N_D} \sum * i = 1^{N_D} \Big[& \hat{Q}\left(n_D^i, t_D^i; \theta, w_1, w_2\right) \\
& - \tilde{Q}\left(n * D^i, t_D^i\right) \Big]^2,
\end{aligned} \tag{15}$$

in which $\tilde{Q}(n_{\mathcal{D}}^i, t_D^i)$ denotes the noisy data

$$\tilde{Q}(n_D^i, t_D^i) = Q(n_D^i, t_D^i)(1 + \mathcal{N}(0, \sigma)), \tag{16}$$

where $\mathcal{N}(0, \sigma)$ is a Gaussian-distributed random variable characterized by a mean of zero and standard deviation σ. The parameter σ is expressed as a percentage relative to the magnitude of the original data values. Different noise levels are evaluated, specifically including 0%, 2.0%, 5.0%, and 10.0%.

During the training process, λ is initialized to 1.0, with point P_{s} fixed at the true symmetric point $(-0.50, 0.00)$. The neural network parameters θ, together with the unknown parameters w_1 and w_2, are jointly optimized to ensure that the method not only fits the observed data but also satisfies the given physical constraints.

Table 3. The average inversion results for the parameters w_1 and w_2 in both one- and two-soliton solutions of Eq. (13), based on five sets of random seeds, are obtained using PINNs and LSD-PINNs.

Domain points	Soliton	One-soliton		Two-soliton	
	Model	PINNs	LSD-PINNs	PINNs	LSD-PINNs
	Condition	RE(%) of (w_1, w_2)		RE(%) of (w_1, w_2)	
$N = 20$	Clean data	(0.370, 0.168)	(**0.079**, **0.025**)	(13.142, 7.232)	(**0.113**, **0.079**)
	2% noise	(3.057, 2.880)	(**2.676**, **2.605**)	(11.251, 7.030)	(**4.684**, **2.871**)
	5% noise	(11.780, 9.406)	(**6.197**, **6.147**)	(11.211, 8.294)	(**7.347**, **5.102**)
	10% noise	(34.729, 23.687)	(**16.523**, **14.152**)	(14.317, 11.510)	(**12.154**, **10.272**)
$N = 50$	Clean data	(0.127, 0.073)	(**0.051**, **0.026**)	(0.097, 0.074)	(**0.072**, **0.031**)
	2% noise	(2.197, 1.843)	(**1.879**, **1.689**)	(1.039, 0.810)	(**0.961**, **0.656**)
	5% noise	(6.298, 5.152)	(**4.733**, **4.199**)	(2.514, 2.032)	(**2.441**, **1.499**)
	10% noise	(20.601, 14.576)	(**10.482**, **8.036**)	(4.915, 3.895)	(**4.637**, **2.913**)
$N = 100$	Clean data	(0.148, 0.095)	(**0.055**, **0.031**)	(0.068, 0.046)	(**0.049**, **0.030**)
	2% noise	(1.636, 1.408)	(**1.079**, **0.820**)	(0.567, 0.460)	(**0.539**, **0.458**)
	5% noise	(4.185, 3.577)	(**3.545**, **2.286**)	(1.244, 1.091)	(**1.091**, **1.017**)
	10% noise	(12.914, 8.016)	(**8.809**, **7.519**)	(2.485, 2.270)	(**2.314**, **2.028**)
$N = 200$	Clean data	(0.170, 0.084)	(**0.057**, **0.028**)	(0.060, 0.048)	(**0.048**, **0.047**)
	2% noise	(1.050, **1.052**)	(**1.011**, 1.234)	(0.311, 0.339)	(**0.301**, **0.299**)
	5% noise	(2.635, **2.447**)	(**2.493**, 3.078)	(**0.655**, **0.829**)	(0.689, 0.859)
	10% noise	(11.135, 6.315)	(**5.587**, **6.198**)	(1.386, 1.750)	(**1.322**, **1.681**)
$N = 450$	Clean data	(0.221, 0.103)	(**0.066**, **0.034**)	(0.094, 0.039)	(**0.062**, **0.034**)
	2% noise	(0.783, 1.130)	(**0.729**, **0.704**)	(0.190, **0.205**)	(0.121, **0.211**)
	5% noise	(2.297, 3.001)	(**1.756**, **1.756**)	(0.355, 0.683)	(**0.347**, **0.552**)
	10% noise	(5.198, 6.233)	(**3.067**, **3.718**)	(0.748, 1.305)	(**0.684**, **1.210**)

In this study, we preserve the network configuration, initial-boundary conditions, and iteration logic for the spatiotemporal domain of the forward problem as delineated in Subsections 3.1 and 3.2. Employing Hammersley sampling, we sequentially generate internal labeled points with $N = 20, 50, 100, 200, 500$. To ensure the comprehensiveness and reliability of the experiments, the unknown parameters w_1 and w_2 are estimated under five distinct random seeds, accounting for various noise levels and different numbers of labeled points. Additionally, we document the performance

of PINNs under the same conditions. Subsequently, the relative errors in parameter inversion under identical noise and labeled-point configurations are averaged across different random seeds. The results of the two methods are compared accordingly, with the data corresponding to the method exhibiting smaller inversion errors emphasized in bold. The final results are presented in Table 3. To intuitively illustrate the regularity of the inversion results, the inverted parameters w_1 and w_2 in Eq. (13) obtained using the two methods for the one- and two-soliton solutions are illustrated in Figs. 11 and 12, respectively.

Table 3 reveals that the LSD-PINNs method generally exhibits superior parameter inversion capabilities compared to standard PINNs under most experimental conditions, although exceptions are observed (e.g., when $N \geq 200$ and noise > 0). This observation is largely attributed to the integration of the Lax pair as an additional physical constraint, denoted as $F_{\mathrm{Lp}}(n,t;\theta,\lambda)$, within the loss function. When combined with the symmetric input merging operation described in Eq. (9), this method effectively reduces the parameter search space, particularly when N is small, thereby compensating for limited data through physical regularization. However, as N increases and noise is introduced, the neural network outputs may exhibit varying error patterns. In such cases, the symmetric output merging in Eq. (9) may inadvertently smooth out valuable information while propagating erroneous signals to $F_{\mathrm{Lp}}(n,t;\theta,\lambda)$. This can distort gradient directions, introduce conflicts among constraints, and ultimately diminish the accuracy of the parameter inversion for w_1 and w_2.

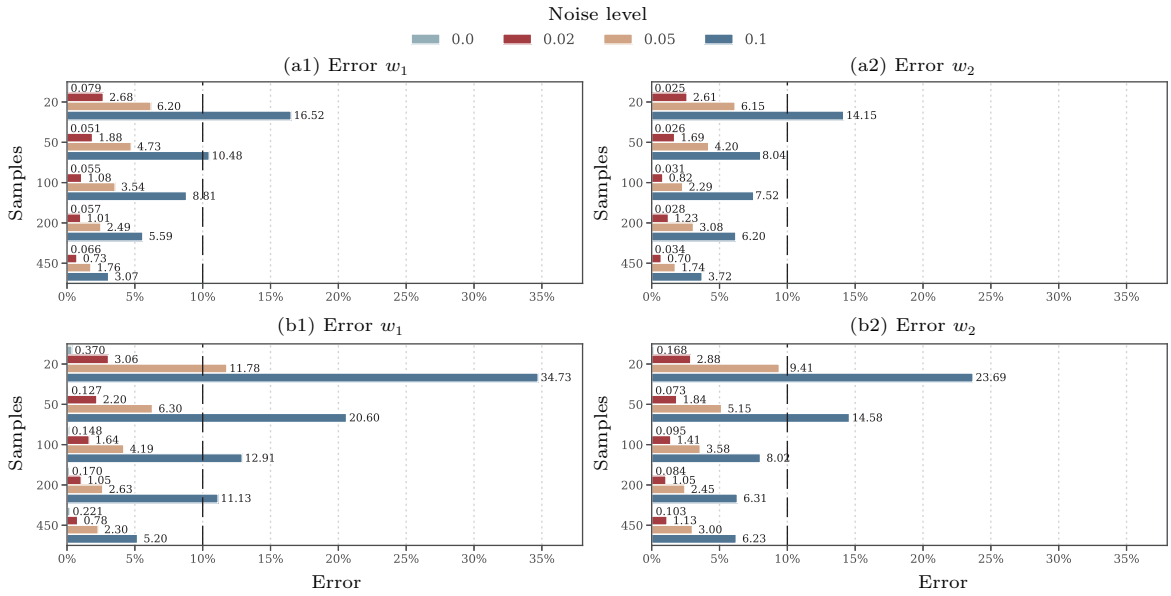


Fig. 11. Bar chart of inversion results for the parameters w_1 and w_2 for the one-soliton solution of Eq. (13): (a1) and (a2) correspond to w_1 and w_2 obtained using LSD-PINNs, respectively, while (b1) and (b2) correspond to those obtained using PINNs.

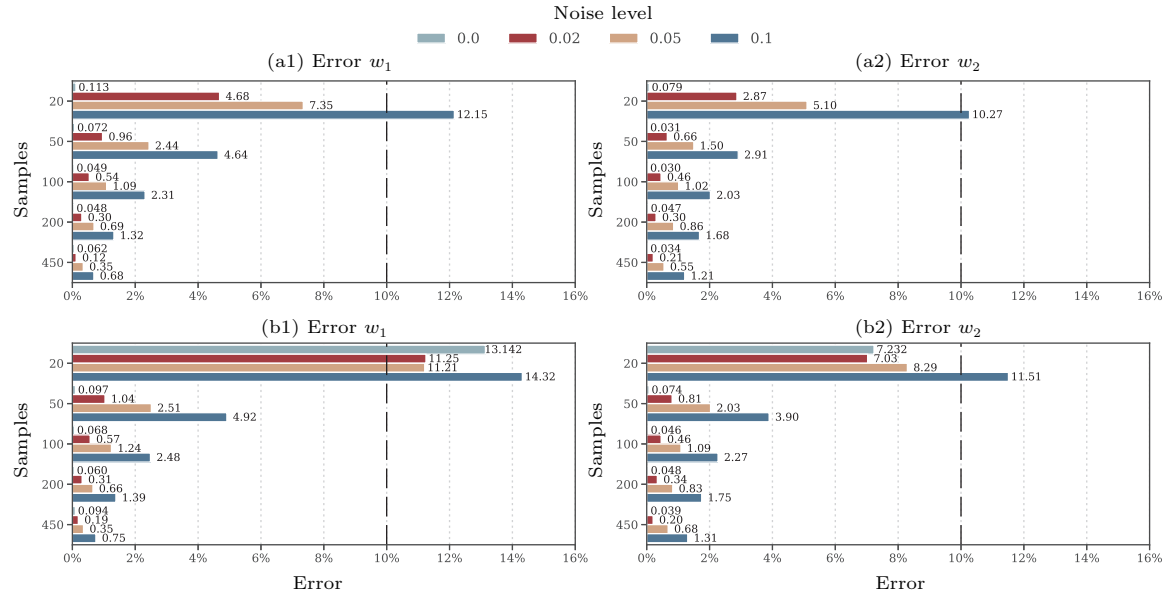


Fig. 12. Bar chart of inversion results for the parameters w_1 and w_2 for the two-soliton solution of Eq. (13): (a1) and (a2) correspond to w_1 and w_2 obtained using LSD-PINNs, respectively, while (b1) and (b2) correspond to those obtained using PINNs.

In the context of a one-soliton scenario, the inversion errors for the unknown parameters obtained using LSD-PINNs are on the order of 10^{-4}, specifically 0.079% for w_1 and 0.025% for w_2. When compared with the results obtained using PINNs under identical conditions, which yield errors of 0.370% for w_1 and 0.168% for w_2, LSD-PINNs demonstrate a reduction in error by approximately one order of magnitude. In the two-soliton scenario, under the same conditions, the inversion accuracy of LSD-PINNs improves by two orders of magnitude. This improvement is visually depicted in Fig. 12. Additionally, as shown in Figs. 11 and 12, the RE for parameter inversion using the LSD-PINNs method exceeds 10% only when $N = 20$ and the noise level reaches 10%, with the maximum error recorded at 16.52% for w_1. Conversely, under identical conditions, the RE for PINNs exceeds 10% for all tested values of N when subjected to 10% noise in both soliton scenarios. These results suggest that LSD-PINNs demonstrate enhanced parameter inversion capabilities and robustness in high-noise environments.

Furthermore, Figs. 11 and 12 indicate that, for a fixed noise level, larger values of N lead to lower inversion errors in w_1 and w_2; conversely, for a fixed N, higher noise levels result in greater inversion errors. Both observations are consistent with fundamental principles of parameter inversion: more data points provide stronger constraints on unknown parameters, while increased noise corrupts the constraining information. These findings collectively validate the rationale of our experimental design and the reliability of the inversion results.

A notable observation is that the inversion results for w_2 and w_1 are consistently more accurate for the two-soliton solution than for the one-soliton solution across both methods. This disparity can be attributed to the inherent information content of the data. The one-soliton waveform is relatively simple, and its parameters may exhibit degeneracy, thereby providing insufficient constraints for accurate identification. Consequently, the corresponding data contain limited information within the spatiotemporal domain. Conversely, the intricate spatiotemporal dynamics resulting from the interaction of two solitons encapsulate richer dynamical information, imposing more independent and robust constraints on the parameters. Additionally, the symmetric output structure presented in Eq. (9) facilitates more stable and efficient utilization of gradient information from the data, aiding in avoiding optimization pitfalls and thereby enhancing inversion accuracy. These findings underscore the efficacy and promise of the LSD-PINNs method for parameter inversion in Eq. (1), demonstrating its capability to achieve stable and high-precision parameter estimation. This method provides a robust tool for data-driven modeling of complex systems.

5. Discussion and conclusion

This study presents the first integration of the Lax pair into PINNs for discrete integrable systems, introducing a novel method termed LD-PINNs. Building upon this method, we incorporate symmetry for the first time to formulate LSD-PINNs, aiming to enhance model performance. Moreover, we are the first to investigate both forward and inverse problems for discrete one- and two-soliton solutions of Eq. (1). This contributes to a better understanding of the propagation behavior of water waves and sound waves in nonlinear media. The final experimental analysis results and achievements are as follows.

In the forward problem, LD-PINNs effectively resolve both one- and two-soliton solutions of Eq. (1), while also numerically obtaining the associated spectral functions and spectral parameters. Compared to PINNs, LD-PINNs, and SDE-PINNs, LSD-PINNs exhibit superior solution accuracy and enhanced stability, with both internal and external predictions consistently remaining at the 10^{-4} order of magnitude across multiple experimental sets. Nonetheless, it is important to note that both methods incur computational costs exceeding twice those of PINNs.

In the inverse problem, our method effectively inverts parameters under sparse and low-noise conditions, with inversion errors consistently within the 10^{-4} magnitude range. Even under high noise (noise = 10%), the error remains below 10% for all cases with $N > 20$. Furthermore, under equivalent conditions, the LSD-PINNs method typically exhibits enhanced convergence and increased robustness relative to PINNs. Specifically, for the two-soliton solution with $N = 20$ under noise-free conditions, the LSD-PINNs method achieves a reduction in inversion error by two orders of magnitude.

This study has the potential to provide novel insights into established methodologies such as DT and IST, while also laying the groundwork for identifying new Lax pairs and associated physical equations. These advancements may further promote the integration of deep learning with discrete integrable systems. However, for the LSD-PINNs method proposed above, we have merely used the soliton solutions of the discrete complex mKdV equation as an illustrative example to demonstrate its effectiveness and have not yet extended this method to n-component discrete integrable systems with large n. In particular, the scenario of n-component discrete integrable systems with large n remains a highly challenging problem and will be a key focus of our future research. Therefore, a primary direction of our forthcoming work is to develop specialized neural network architectures that inherently incorporate the structural properties of Lax pairs and other physical characteristics of the equations. By ensuring that these constraints are satisfied by the architecture itself, we aim to fundamentally reduce the optimization difficulties and computa-

tional costs of the LSD-PINNs method, thereby improving the solving efficiency for large-scale discrete systems.

Acknowledgments

We are grateful to our discussion group members for their valuable comments. Project supported by the National Natural Science Foundation of China (Grant Nos. 12571263 and 12071042), the Beijing Natural Science Foundation (Grant No. 1242004), the BISTU Undergraduate Research Training Program 2025 (Grant No. 2025BKYY01), and the BISTU Starlight Fund 2025 (Grant No. XG2025ZD31).

References

[1] Raissi M, Yazdani A and Karniadakis G E 2020 *Science* **367** 1026
[2] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 *Nat. Rev. Phys.* **3** 422
[3] Wakefield L, Agarwal S and Tanner K 2023 *Cell* **186** 1792
[4] Lei J, Zhu X and Wang Y 2022 *Neural Networks* **156** 67
[5] Helal M A 2022 *Chaos Soliton. Fract.* **13** 1917
[6] Debnath L 2005 *Nonlinear Partial Differential Equations for Scientists and Engineers* (Boston: Birkhäuser)
[7] Li L, Liu J, Shen B, Malomed B A, Dai C and Wang Y 2025 *Opt. Laser Technol.* **192** 113715
[8] Kudryashov N A, Dai C Q, Zhou Q and Nifontov D R 2025 *Eur. Phys. J. Plus* **140** 757
[9] Qu G, Hu X, Miao Z, Shen S and Wang M 2021 *Results Phys.* **23** 104052
[10] Hure C, Pham H and Warin X 2020 *Math. Comput.* **89** 1547
[11] Dektor A, Rodgers A and Venturi D 2021 *J. Sci. Comput.* **88** 36
[12] Niu Z and Li B 2025 *Appl. Math. Lett.* **163** 109411
[13] Raissi M, Perdikaris P and Karniadakis GE 2019 *J. Comput. Phys.* **378** 686
[14] Kag V, Seshasayanan K and Gopinath V 2022 *Phys. Fluids* **34** 055130
[15] Cao Z, Luo K, Liu K, Cheng Y, Xing J, Jiang L and Fan J 2026 *Fuel* **405** 136475
[16] Zhu Q, Liu Z and Yan J 2021 *Comput. Mech.* **67** 619
[17] Bora A, Dai W, Wilson J P, Boyt J C and Sobolev S L 2022 *Int. J. Heat Mass Tran.* **190** 122791
[18] Jiang X, Wang D, Fan Q, Zhang M, Lu C and Lau APT 2022 *Laser Photonics Rev.* **16** 2100483
[19] Liu X, Hong Z, Liu J, Lin Y, Rodrguez-Patn A, Zou Q and Zeng X 2020 *Brief. Bioinform.* **21** 486
[20] Zhou Z and Yan Z 2024 *Physica D* **457** 133987
[21] Jagtap A D, Kawaguchi K and Karniadakis G E 2020 *J. Comput. Phys.* **404** 109136
[22] He W, Li J, Kong X and Deng L 2024 *Commun. Eng.* **3** 151
[23] Chen J, Li B and Yuen M 2025 *Commun. Nonlinear Sci.* **151** 109049
[24] Chen C 2024 *Chaos Soliton. Fract.* **185** 115136
[25] Fan D, Yang C, Sun H, Yao J, Zhang L, Jia, C and Sang Q 2025 *Phys. Fluids* **32** 014114
[26] Li T, Zou Y, Zou S, Chang X, Zhang L and Deng X 2025 *J. Comput. Phys.* **530** 113919
[27] Jiang J H, Si Z Z, Dai C Q and Wu B 2024 *Chaos Soliton. Fract.* **180** 114581
[28] Fang Z 2021 *IEEE Trans. Neural Netw. Learn. Syst.* **33** 5514
[29] Xiang Z, Peng W, Liu X and Yao W 2022 *Neurocomputing* **496** 11
[30] Anagnostopoulos S J, Toscano J D, Stergiopulos N and Karniadakis G E 2024 *Comput. Methods Appl. Mech. Eng.***421** 116805
[31] Sun J and Li B 2025 *J. Syst. Sci. Complex.* **38** 2272
[32] Chen Y and Lü X 2025 *Chaos Soliton. Fract.* **190** 115669
[33] Pu J C and Chen Y 2022 *Chaos Soliton. Fract.* **160** 112182
[34] Wang X, Wu Z, Song J, Han W and Yan Z 2024 *Chaos Soliton. Fract.* **180** 114509
[35] Pu J C, Li J and Chen Y 2021 *Chin. Phys. B* **30** 060202
[36] Zhong M, Gong S, Tian S F and Yan Z 2022 *Physica D* **439** 133430
[37] Peng W Q, Pu J C and Chen Y 2022 *Commun. Nonlinear Sci.* **105** 106067
[38] Ma P L, Pu J C and Peng W Q 2025 *Nonlinear Dyn.* **113** 26547
[39] Lin Z and Chen Y 2025 *J. Comput. Phys.* **539** 114243
[40] Lin S and Chen Y 2025 *Chaos* **35** 113120
[41] Hu Wv and Gegen H 2025 *Chin. Phys. B* **34** 070202
[42] Huang K T and Wang X S 2025 *Chin. Phys. B* **34** 107502
[43] Lin S and Chen Y 2023 *Physica D* **445** 133629
[44] Lin S and Chen Y 2022 *J. Comput. Phys.* **457** 111053
[45] Pu J C and Chen Y 2024 *J. Comput. Phys.* **10** 113090
[46] Pu J C and Chen Y 2024 *Physica D* **467** 134262
[47] Zhang Z Y, Zhang H, Zhang L S and Guo L L 2023 *J. Comput. Phys.* **492** 112415
[48] Peng W Q and Chen Y 2024 *Chaos* **34** 043124
[49] Li J Y, Zhang H, Liu Y, Guo L L, Zhang L S and Zhang Z Y 2023 *Nonlinear Dyn.* **111** 21861
[50] Zhou J C, Wen X Y and Guo M J 2025 *Commun. Theor. Phys.* **77** 065002
[51] Zhou J C, Wen X Y, Zhou P and Wei M C 2025 *Commun. Nonlinear Sci.* **151** 109046
[52] Saqlain S, Zhu W, Charalampidis E G and Kevrekidis P G 2023 *Commun. Nonlinear Sci.* **126** 109046
[53] Lin Z and Chen Y 2024 *Physica D* **468** 134304
[54] Herbst B M, Ablowitz M J and Ryan E 1991 *Comput. Phys. Commun.* **65** 137
[55] Rodríguez R F, Reyes J A, Espinosa-Cerón A, Fujioka J and Malomed B A 2003 *Phys. Rev. E* **68** 036606
[56] He J, Wang L, Li L, Porsezian K and Erdélyi R 2014 *Phys. Rev. E* **89** 062917
[57] Zhao H and Yu G 2017 *Chaos* **27** 043113
[58] Guo M J, Wen X Y and Liu X K 2024 *Wave Motion* **125** 103260
[59] Liu L, Wen X Y, Yin Y and Zhong X 2025 *Eur. Phys. J-Spec. Top.* **234** 5389
[60] Hu Y, Shen S and Zhao S 2025 *Math. Method. Appl. Sci.* **48** 5557
[61] Wu C, Zhu M, Tan Q, Kartha Y and Lu L 2023 *Comput. Methods Appl. Mech. Engrg.* **403** 115671
[62] Lax P D 1968 *Commun. Pure. Appl. Math.* **21** 467
[63] Alkhadhr S and Almekkawy M 2023 *Sensors* **23** 2792
[64] Jahani-nasab M and Bijarchi M A 2024 *Sci. Rep.* **14** 23836
[65] Glorot X and Bengio Y 2010 *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* **9** 249
[66] Lu L, Meng X, Mao Z and Karniadakis G E 2021 *SIAM Rev.* **63** 208