Informed Reconstruction-Oriented Numerical Networks: A More Efficient Method for Solving Multiple Unknown Parameters of Manakov Equations

Peng-Fei Wang(王鹏飞)[1], Yun-Zhou Sun(孙运周)[1*], Houria Triki[2], and Qin Zhou(周勤)[1,3*]

[1]*Research Center of Nonlinear Optical Science and Quantum Technology, Hubei Engineering Research Center for Wide Bandgap Semiconductor Materials and Devices, School of Microelectronics, Wuhan Textile University, Wuhan 430200, China*

[2]*Radiation Physics Laboratory, Department of Physics, Faculty of Sciences, Badji Mokhtar University, Annaba, P.O. Box 12, 23000, Algeria*

[3]*State Key Laboratory of New Textile Materials and Advanced Processing, Wuhan Textile University, Wuhan 430200, China*

(Received 13 February 2026; accepted manuscript online 13 March 2026)

By simultaneously introducing a finite-difference-based numerical loss term and a clustering–reconstruction mechanism, we propose an enhanced physics-informed neural network named the informed reconstruction-oriented numerical network (IRON-Net) and subsequently apply it to the Manakov equations—a well-known two-component nonlinear physical model. Numerical experiments are conducted on a dataset containing eight analytical solutions with noise. The results indicate that, compared to conventional PINNs and other mainstream algorithms, IRON-Net demonstrates significant advantages in training accuracy, convergence rate, and robustness, achieving a stepwise improvement in the neural network's ability to enforce physical constraints. Additional ablation experiments further confirm the necessity of the consistency constraint within IRON-Net. This study provides an effective approach for modeling and parameter identification in complex nonlinear optical systems as well as other nonlinear physical scenarios.

DOI: 10.1088/0256-307X/43/4/040004 CSTR: 32039.14.0256-307X.43.4.040004

1. Introduction. Manakov equations, as an important extension of the integrable scalar nonlinear Schrödinger equation, can be used to describe the propagation of two mutually coupled waves in nonlinear physical systems. In nonlinear optics, this model effectively characterizes the propagation of dual-polarization optical pulses in weakly birefringent fibers.[1–3] In condensed matter physics, it can describe the evolution behavior of two spin states or two different types of ultracold atoms.[4,5] In plasma physics, it is used to analyze nonlinear interactions among charge-density waves of different modes.[6,7] Due to their integrability in mathematics, the Manakov equations not only provide a unified modeling framework for multicomponent nonlinear physical systems but also offer a solid theoretical foundation and practical tools for understanding and controlling complex coupled-wave phenomena.

With the rapid development of parameter identification and inverse problem studies, increasing attention has been devoted to the determination of multiple unknown parameters in nonlinear evolution equations. Among existing methodologies, physics-informed neural networks (PINNs) have emerged as robust function approximators, harnessing the expressive power of deep learning to address both forward and inverse problems governed by partial differential equations.[8–13] A defining feature of PINNs is their mesh-free nature; by training on scattered collocation points, they circumvent the computational bottleneck of mesh generation typically required for complex or high-dimensional domains.[14–18] Consequently, this versatile paradigm has been successfully deployed across a broad spectrum of scientific and engineering disciplines.[19–23] However, in practical applications, PINNs are prone to overfitting when confronted with noisy observational data, or may suffer from gradient pathologies due to excessive reliance on a single residual loss term.[24–26] Therefore, various extensions have been proposed to enhance physical consistency and improve convergence and robustness.[27–32] For example, gradient-enhanced physics-informed neural networks (gPINNs) smooth local residuals through gradient-enhanced loss terms;[33] coupled physics-informed neural networks (cPINNs) and extended physics-informed neural networks (XPINNs) employ domain decomposition strategies with continuity constraints imposed at subdomain interfaces;[34,35] geometry-aware methods introduce strong geometric constraints by constructing exact trial functions that satisfy prescribed boundary conditions;[36] two-stage PINNs explicitly incorporate conservation laws as penalty terms in the loss function;[37] the physics structure-informed neural network (Psi-NN) extracts and embeds deep physical structural constraints via knowledge distillation, clustering, and parameter reconstruction;[38]

*Corresponding authors. Email: qinzhou@whu.edu.cn; syz@wtu.edu.cn

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

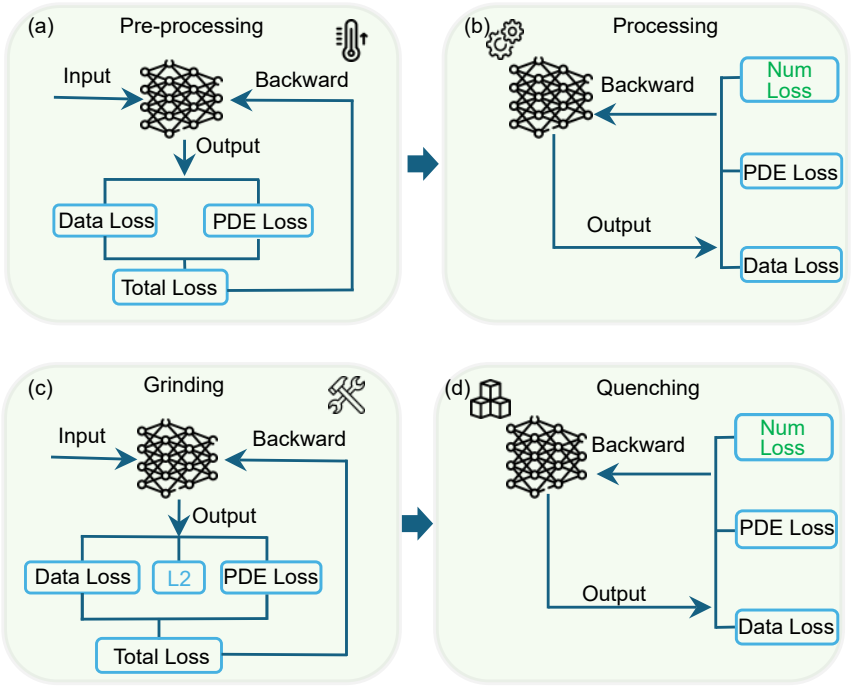

Fig. 1. Flowchart of (a) pre-processing, (b) processing, (c) grinding, and (d) quenching stages.

residual-based adaptive refinement physics-informed neural networks (RAR-PINNs) construct more reasonable residual constraints via adaptive sampling in high-residual regions;[39] and prior-information PINNs (PIPINNs) impose additional constraints derived from prior information and energy conservation.[40]

Motivated by these prior studies, we propose an enhanced PINN framework, namely the informed reconstruction-oriented numerical networks (IRON-Net). The innovation of this method lies in constructing the numerical loss constraints based on finite-difference schemes and introducing a clustering–reconstruction mechanism to effectively regulate the internal structure of neural networks. In this Letter, we demonstrate the effectiveness of the proposed approach by identifying two unknown coefficients in the Manakov equations. Through a stage-wise optimization strategy, IRON-Net successfully addresses the robustness challenges encountered by conventional PINNs in parameter inversion tasks under high-noise conditions.

2. Methods. In this section, we provide a detailed description of the algorithmic improvement process. As shown in Fig. 1, the proposed algorithm consists of four main stages: pre-processing, processing, grinding, and quenching. Figure 1 also depicts the evolution of the neural network architecture as well as the corresponding changes in the loss function across these stages. The detailed procedures and underlying mechanisms of each stage are described comprehensively in the following sections.

2.1. Pre-Processing and Processing. Neural networks typically rely on the backpropagation of the chain rule to perform automatic differentiation (AD). Assuming the input is y and the trainable parameters are θ, a neural network $u(y;\theta)$ can be represented as a composite function consisting of L layers,

$$u = f_L \circ f_{L-1} \circ \cdots \circ f_1(y). \tag{1}$$

When employing AD to compute the derivative $\mathrm{d}u/\mathrm{d}y|_{y_i}$ at the point y_i, the chain rule dictates that

$$\left.\frac{\mathrm{d}u}{\mathrm{d}y}\right|_{y_i} = \left.\frac{\partial f_L}{\partial f_{L-1}}\frac{\partial f_{L-1}}{\partial f_{L-2}}\cdots\frac{\partial f_1}{\partial y}\right|_{y_i}. \tag{2}$$

Each Jacobian matrix $\partial f_l/\partial f_{l-1}$ on the right-hand side of the equation is evaluated strictly at the single coordinate point $y = y_i$.

From a computational perspective, the calculation of $\mathrm{d}u/\mathrm{d}y|_{y_i}$ requires no prior knowledge of the function values or properties at $y_i + \Delta y$. This implies that, within the neural network framework, each training point (collocation point) is treated as a point-wise independent entity.[41] Consequently, when the observational data is noisy, the lack of local information can lead to certain errors. Although all data points share a common set of neural network parameters θ—which, to some extent, facilitates "global coupling"—the network tends to learn a smooth overall profile. In the absence of explicit local constraints, the neural network may satisfy the governing physical equations at point y_i while simultaneously generating non-physical micro-oscillations in the immediate vicinity of y_i.[24–26]

In essence, numerical differentiation formulas act as low-pass filters. For instance, the five-point finite difference method used to approximate the first-order derivative at y_i is expressed as

$$u'(y_i) \approx \frac{-u(y_{i+2}) + 8u(y_{i+1}) - 8u(y_{i-1}) + u(y_{i-2})}{12h}. \tag{3}$$

This formulation explicitly incorporates the output values

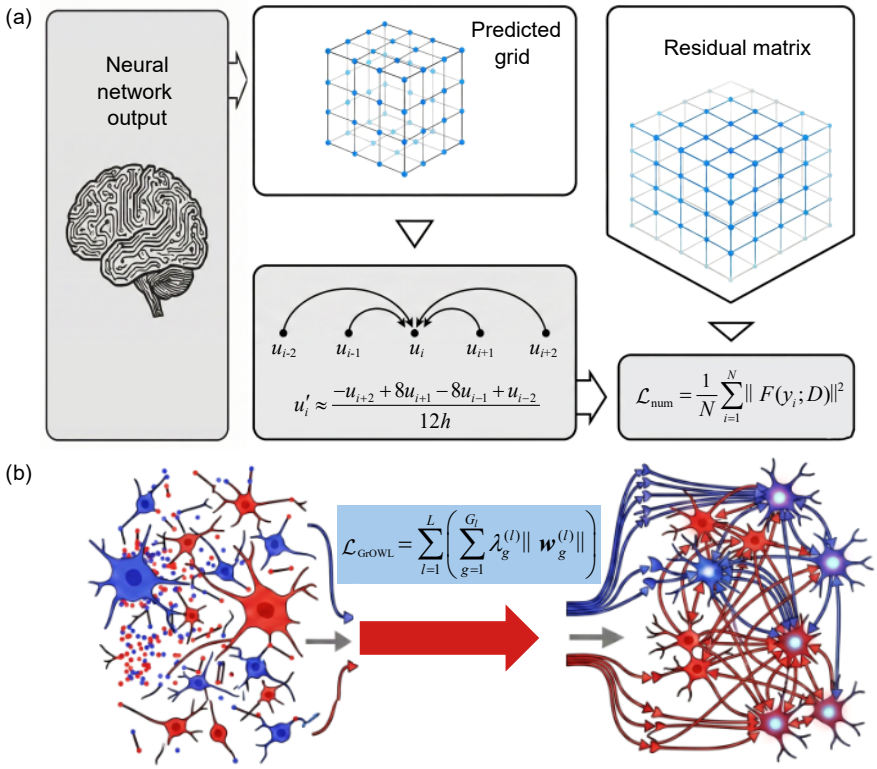

Fig. 2. The concept of (a) numerical loss and (b) GrOWL regularized loss.

of five spatial points, from $u(y_{i-2})$ to $u(y_{i+2})$, to enforce local constraints, where h represents the grid spacing.[42–44] As illustrated in Fig. 2(a), which depicts the process of embedding the five-point central difference scheme into the neural network as a soft constraint, we introduce this scheme into a network with N total outputs. By constructing the numerical differentiation terms and substituting them into the residual equation $F(x, D)$ (which contains the parameters to be inverted), we obtain the numerical residual matrix $\mathcal{R}_{\text{min}}$. This matrix is updated iteratively in tandem with the neural network parameter θ. By minimizing the mean squared error, the numerical residual matrix $\mathcal{R}_{\text{min}}$ is transformed into the numerical loss

$$\mathcal{L}_{\text{num}} = \frac{1}{N}\sum_{i=1}^{N}||F(y_i|; D)||^2. \tag{4}$$

Integrating this into the standard PINNs loss function

$$\mathcal{L}_{\text{PINNs}} = \lambda_{\text{d}}\mathcal{L}_{\text{data}} + \lambda_{\text{p}}\mathcal{L}_{\text{pde}}, \tag{5}$$

we derive the final objective function

$$\mathcal{L}_{\text{pro}} = \lambda_{\text{d}}\mathcal{L}_{\text{data}} + \lambda_{\text{p}}\mathcal{L}_{\text{pde}} + \lambda_{\text{n}}\mathcal{L}_{\text{num}}. \tag{6}$$

We define the training stage where the $\mathcal{L}_{\text{pro}}$ constraint is first introduced as the "processing stage". To further enhance the computational robustness of the finite difference operators, this method incorporates a "pre-processing stage" as a precursor task. During this stage, the model is first pre-trained using $\mathcal{L}_{\text{PINNs}}$ to guide the neural network in capturing the global features of the solution and achieving a smoothed state. Once the model has successfully adapted to the distribution of the dataset, it transitions to the processing stage, which includes the numerical differentiation constraints. This strategy effectively mitigates numerical gradient oscillations caused by random initialization in the early stages of training, thereby ensuring the accuracy and stability of the subsequent inversion process.

2.2. Grinding and Quenching. In addition to the explicit soft constraints introduced by numerical losses, we incorporate the clustering–reconstruction mechanism to serve as a physical hard constraint for the neural network. As demonstrated by Liu *et al.*,[38] this mechanism can extract symmetries and conservation laws of a system from the evolutionary process of its parameters. These physical relationships are then hard-coded into the network architecture using a relationship matrix R, thereby forming intrinsic consistency constraints. The core logic involves appending a penalty term via grouped ordered weighted LASSO (GrOWL) regularization,

$$\mathcal{L}_{\text{GrOWL}} = \sum_{l=1}^{L}\left(\sum_{g=1}^{G_l}\lambda_g^{(l)}||\boldsymbol{w}_g^{(l)}||\right). \tag{7}$$

This term induces clustering among the parameters $(\boldsymbol{w}_g^{(l)})\}_{g=1}^{G_l}$ of the neurons in the l-th layer of an L-layer neural network. By grouping parameters with similar absolute values into the same cluster, the model effectively mitigates biases caused by observational noise on individual parameters. This mechanism is introduced following the completion of the processing stage. The loss function at this stage, driven by the synergy of physical principles and parameter regularization, is defined as

$$\mathcal{L}_{\text{gri}} = \lambda_{\text{d}}\mathcal{L}_{\text{data}} + \lambda_{\text{p}}\mathcal{L}_{\text{pde}} + \lambda_{\text{g}}\mathcal{L}_{\text{GrOWL}}. \tag{8}$$

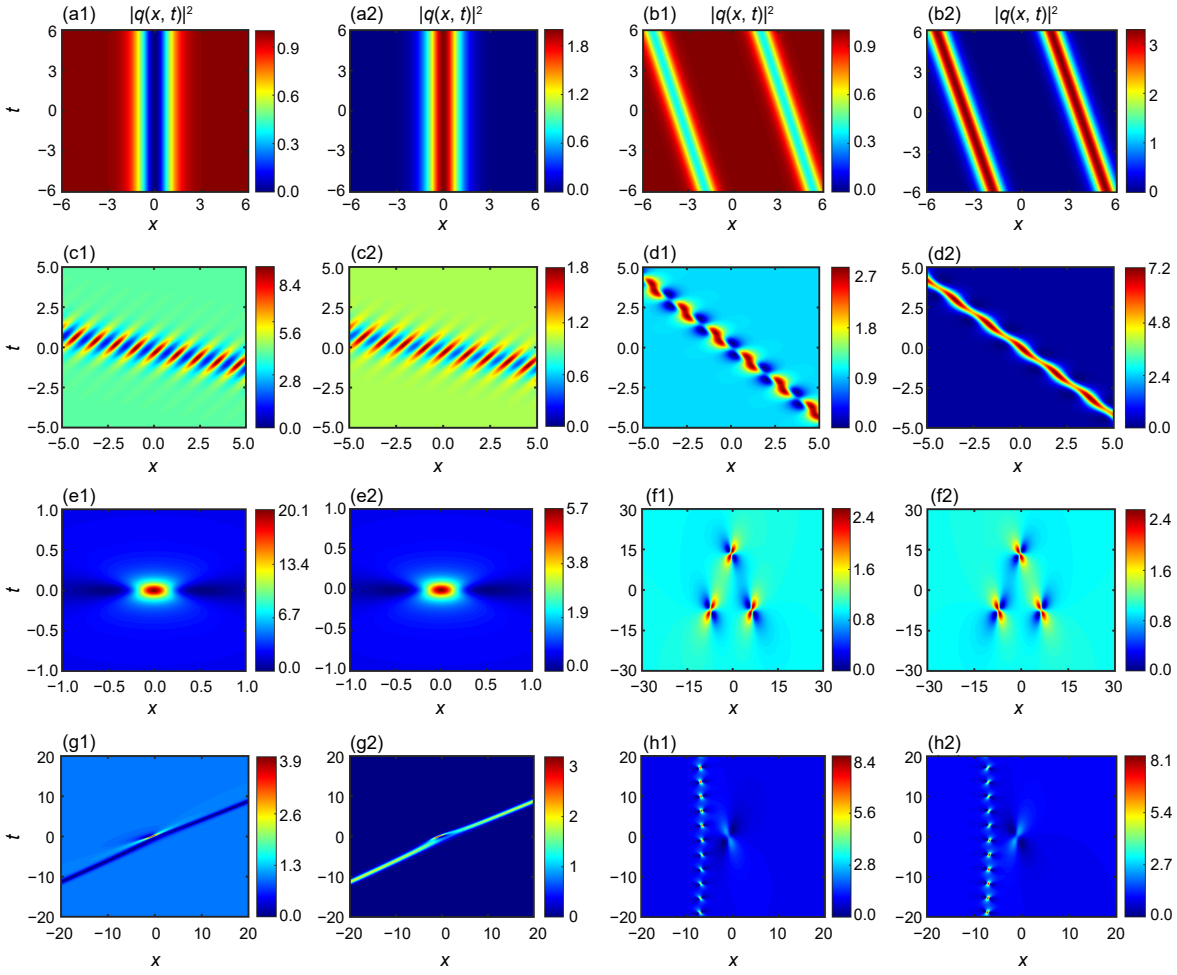

Fig. 3. The two-dimensional distribution plots of the soliton intensity for (a1) and (a2) the first-order bright and dark soliton solutions; (b1) and (b2) the second-order bright and dark soliton solutions; (c1) and (c2) the type-I breather solutions; (d1) and (d2) the type-II breather solutions; (e1) and (e2) the first-order rogue wave solutions; (f1) and (f2) the second-order rogue wave solutions; (g1) and (g2) the bright and dark soliton-rogue wave solutions; and (h1) and (h2) the breather–rogue wave solutions of the Manakov equations.

Under the refinement of this stage, cluster centers naturally emerge. As illustrated in the conceptual diagram in Fig. 2(b), GrOWL regularization clusters neurons with initially disparate parameter magnitudes.

Subsequently, these centers are extracted via the hierarchical agglomerative clustering algorithm and reconstructed into a physically meaningful low-rank relationship matrix R. As shown in Fig. 1(b), we define this process as the grinding stage. The succeeding quenching stage involves re-training the neural network under the formally embedded structural hard constraints. At this point, the trainable parameters of the model are significantly compressed—from the original high-dimensional parameter matrix to a small set of core parameters C composed of cluster centers. Through the structural mapping

$$W = R \cdot C, \tag{9}$$

and driven by the loss function

$$\mathcal{L}_{\text{que}} = \lambda_{\text{d}}\mathcal{L}_{\text{data}} + \lambda_{\text{p}}\mathcal{L}_{\text{pde}} + \lambda_{\text{n}}\mathcal{L}_{\text{num}}. \tag{10}$$

The model is optimized efficiently on a constrained physical manifold during the re-training process, subject to both explicit soft numerical constraints and structural hard physical constraints. This mechanism enables the model to output predicted solutions that are not only high in accuracy but also strictly compliant with physical logic.

3. Model. The classic Manakov equations are given by [45]

$$\begin{cases} \mathrm{i}\dfrac{\partial q_1(x,t)}{\partial t} + \alpha\dfrac{\partial^2 q_1(x,t)}{\partial x^2} \\ \quad + \beta\left[|q_1(x,t)|^2 + |q_2(x,t)|^2\right] q_1(x,t) = 0, \\ \mathrm{i}\dfrac{\partial q_2(x,t)}{\partial t} + \alpha\dfrac{\partial^2 q_2(x,t)}{\partial x^2} \\ \quad + \beta\left[|q_1(x,t)|^2 + |q_2(x,t)|^2\right] q_2(x,t) = 0, \end{cases} \tag{11}$$

where $q_i(x,t)$ $(i = 1, 2)$ are complex-valued functions, t is the time variable, and x represents the spatial coordinates. α and β are the coefficients of diffraction and nonlinearity, respectively.

As shown in Fig. 3, we utilize eight classic analytical solutions in the case of Eq. (11) with $\alpha = 0.5$ and $\beta = 1.0$ specifically, these include first-order bright–dark soliton solutions, [46] second-order bright–dark soliton

Table 1. The results of α and β predicted by different algorithms across various control groups.

	PINNs		gPINNs		XPINNs		IRON-Net	
	α	β	α	β	α	β	α	β
1	0.4934	0.9957	0.4966	0.9929	0.5010	1.0705	0.4992	1.0020
2	0.4897	0.9922	0.4430	0.9252	0.4912	0.9932	0.5010	1.0047
3	−0.3516	0.4208	0.3283	0.8789	−0.3144	0.4624	0.4751	0.9756
4	0.0955	1.1630	0.4854	0.9835	0.5665	1.0150	0.4977	0.9987
5	−0.0858	0.8183	0.4898	0.9947	0.4971	1.0928	0.4980	0.9993
6	0.0428	0.9590	0.0754	0.9602	−0.2894	0.9218	0.4914	0.9955
7	0.2645	0.6606	0.4793	0.9363	0.4912	0.9895	0.4981	0.9941
8	0.4885	0.9976	0.4858	0.9957	0.4891	1.0797	0.4938	0.9982

solutions,[47] type-I breather solutions,[48] type-II breather solutions,[49] first-order rogue wave solutions,[50] second-order rogue wave solutions,[51] bright soliton-rogue wave and dark soliton-rogue wave solutions,[52] and breather–rogue wave solutions.[53] Observational data were generated by performing random Latin hypercube sampling within the space–time domain $(x,t) \in [-5,5]$[54] and incorporating 5% random Gaussian noise. The data corresponding to these eight categories of solutions serve as control groups 1–8, respectively, with α and β in Eq. (11) treated as the parameters to be inverted.

4. Experiments and Results. Hyperparameter selection was informed by relevant literature and refined through sensitivity analysis conducted over multiple pre-experiments.[55–60] In order to maintain the fairness of the comparison, the majority of hyperparameters for these methods remained consistent: each utilized 4 hidden layers with 64 neurons per layer, and the AdamW optimizer was chosen with an initial learning rate set to 0.001.[61] To minimize the impact of randomness on experimental outcomes, each group of experiments was independently repeated multiple times to ensure the statistical reliability of the conclusions.

We selected several current mainstream algorithms as benchmarks, such as PINNs, gPINNs, and XPINNs, to serve as references for testing our algorithm. Table 1 presents the prediction results for α and β by the four algorithms under the aforementioned experimental settings. Based on the baseline condition of $\alpha = 0.5$ and $\beta = 1.0$, Table 1 objectively demonstrates the precision of IRON-Net predictions relative to the PINNs, XPINNs, and gPINNs algorithms.

To further evaluate algorithmic accuracy through additional metrics, in the comparative analysis of parameter inversion results, this study employs the relative error (RE)

$$\mathrm{RE}(D) = \left| \frac{D_{\mathrm{pred}} - D_{\mathrm{true}}}{D_{\mathrm{true}}} \right| \times 100\%, \tag{12}$$

between the predicted parameter value D_{pred} and the true value D_{true}. This metric serves as the core indicator for performance evaluation.

Figure 4(a) illustrates the RE of the four algorithms across the aforementioned control groups. The experimental results indicate that, driven by the eight categories of datasets employed, the traditional PINNs algorithm demonstrates limited generalization capability in only a few experimental groups, reflecting a lack of robustness. Although gPINNs and XPINNs improve prediction accuracy to some extent by incorporating additional physical

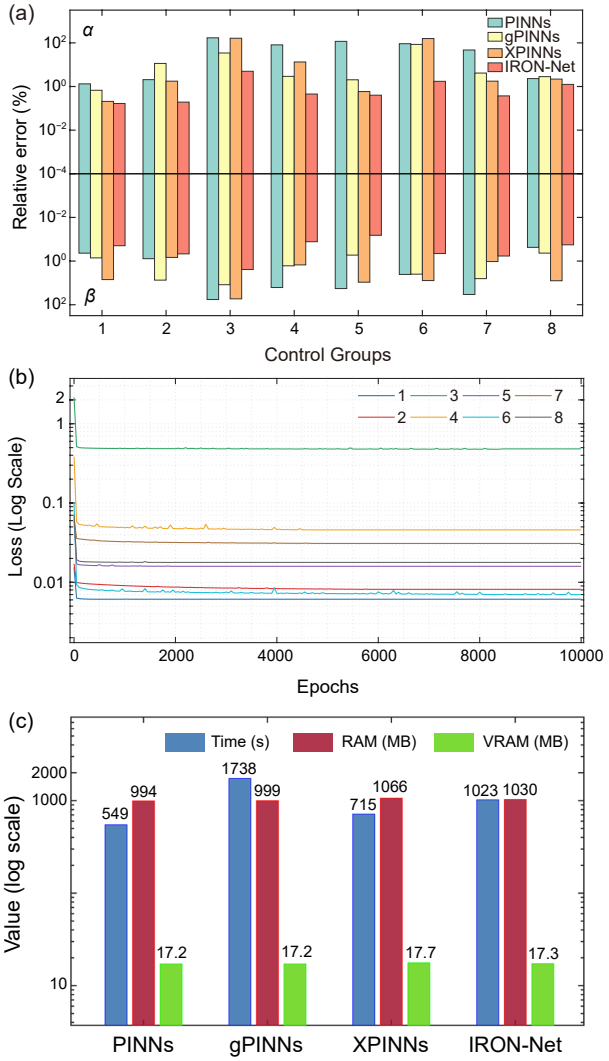

Fig. 4. The diagram of comparison of (a) RE for α and β across different algorithms and control groups; (b) the loss evolution of IRON-Net over the first 10000 rounds across different control groups; (c) the average memory and GPU memory usage across different algorithms, and the time consumption by different algorithms on the relevant control groups.

Table 2. The number of runs with a mean RE below 5% and the average standard deviations across 40 independent runs.

Method	Average standard deviations	Number of runs with MRE $< 5\%$
PINNs	0.015	15
XPINNs	0.023	20
gPINNs	1.311×10^{-2}	20
IRON-Net	7.234×10^{-4}	40

constraints or domain decomposition mechanisms, their overall performance under different dynamical solutions still exhibits significant fluctuations.

In contrast, the IRON-Net proposed in this study demonstrates substantial superiority in global performance, with the RE for α and β remaining at the lowest levels across all control groups. Particularly in experiments involving complex solutions such as the second-order breather, the bright soliton-rogue wave, and the dark soliton-rogue wave solutions, the prediction accuracy of IRON-Net for single parameters is improved by four orders of magnitude compared to traditional PINNs.

Table 2 summarizes, across eight control groups each with five repetitions, the number of runs with a mean relative error (MRE)

$$\mathrm{MRE}(\alpha, \beta) = \frac{\mathrm{RE}(\alpha) + \mathrm{RE}(\beta)}{2}, \tag{13}$$

below 5% and the standard deviations averaged over the control groups. The average standard deviation for each model is computed by first calculating the standard deviation within each control group (based on its five repetitions) and then averaging these eight values. Results were obtained from 40 independent runs in total. A distinct performance stratification is observed: IRON-Net is the only model that consistently achieves an MRE below 5% across all 40 trials with relatively small standard deviations. In contrast, other improved algorithms meet this accuracy standard in only approximately 50% of the cases, highlighting the superior robustness and stability of the proposed framework.

In the analysis of specific dynamical structures, we observe that the inversion performance of IRON-Net on the type-I breather dataset is marginally less effective than in other groups. Considering the inherent nature of inverse problems, parameter identification relies on the extraction of underlying physical laws from the evolution trajectories of observed waveforms. The type-I breather exhibits relatively simple periodic oscillations, resulting in a sparse distribution of physical feature points across the space–time plane. If the spatial sampling points fail to accurately capture the regions where nonlinear interactions are most intense within the oscillation phase, the sensitivity of the loss function to physical parameters will decrease significantly. Although the type-II breather shares similar characteristics, it demonstrates more complex coherent interference patterns, such as interlaced double peaks or distorted envelopes in the space–time plane. This complex nonlinear superposition mechanism exhibits an extremely strong dependence on parameters α and β, implying that the physical equations possess a higher stiffness for these parameters. When the parameters deviate slightly from their true values, the interference fringes of the type-II breather undergo significant topological misalignment, generating substantial physical residuals that compel the network to undergo rapid correction.

The loss function evolution in Fig. 4(b) further validates this inference.By analyzing the statistics of the loss metrics during the first 10,000 iterations of the quenching stage (where missing values are handled via terminal backfilling), defined as

$$\mathcal{L}_{\mathrm{Record}} = \mathcal{L}_{\mathrm{data}} + \mathcal{L}_{\mathrm{pde}}. \tag{14}$$

It can be observed that the convergence slope for the type-I breather group is relatively gradual with higher residual losses. In contrast, the type-II breather exhibits superior convergence performance, while solutions with highly distinct features, such as bright–dark soliton and rogue wave, demonstrate exceptional convergence characteristics. [62,63]

Meanwhile, this study conducted a comprehensive evaluation of the computational resource utilization and time efficiency for each algorithm. As illustrated in Fig. 4(c), which summarizes the memory (RAM), video memory (VRAM), and training time consumption of the four algorithms across the aforementioned control groups, we monitored and recorded the hardware resource allocation at fixed intervals. The PINNs algorithms and their variants exhibited a high degree of similarity in their VRAM and RAM footprints, with the differences in resource overhead being essentially negligible. In terms of training time comparison, to eliminate statistical bias caused by some algorithms triggering early stopping due to gradient oscillations, we selected three control groups where all four algorithms achieved stable convergence—specifically, the bright–dark soliton, first-order bright–dark soliton, and breather–rogue wave groups—to calculate average time consumption statistics. The results indicate that while PINNs possess the fastest execution speed in certain simple tasks due to having the most minimalist physical constraint terms, the training time for IRON-Net falls within a reasonable intermediate range. Furthermore, IRON-Net demonstrates superior flexibility, as an optimal balance between prediction accuracy and computational cost can be achieved by tuning the number of iterations in each training stage. In contrast, gPINNs significantly increase computational overhead due to the introduction of gradient-enhanced loss terms, which require higher-order AD of the residual equations. This results in the most protracted training cycles among all tested algorithms. IRON-Net ensures computational economy and respectable time efficiency while maintaining exceptionally high inversion accuracy.

Finally, to verify the necessity of each stage in the proposed algorithmic framework and to demonstrate that the numerical loss and the clustering–reconstruction mechanism operate synergistically rather than independently, we

Table 3. The MRE statistics for different algorithms in the ablation study.

Method	MRE
PINNs	114.12%
IRON-Net (without num loss)	7.29%
IRON-Net (without reconstruction)	8.19%
IRON-Net	3.71%

conducted a series of additional ablation experiments. As summarized in Table 3, the experiments were performed on the first-order breather dataset, on which conventional PINNs exhibit their poorest performance, thereby providing a challenging benchmark. With the pretraining stage consistently retained, we compared four models: standard PINNs, the complete IRON-Net, IRON-Net without the numerical loss, and IRON-Net without the clustering–reconstruction mechanism, in terms of their MREs. The results indicate that variants lacking either component exhibit error levels approximately twice those of the full IRON-Net, although they still outperform conventional PINNs. These findings provide strong evidence that the integration of soft constraints (numerical loss) and hard constraints (clustering–reconstruction) in IRON-Net is essential. This synergistic design not only maximizes the enforcement of physical consistency but also effectively avoids the optimization redundancy that may arise from merely stacking multiple soft constraints.

5. Conclusion. This study proposes a stage-wise optimized PINN framework. By integrating a numerical loss term constructed via finite-difference schemes, the proposed method alleviates the loss of locality inherent in AD. Meanwhile, a clustering–reconstruction mechanism is introduced to explicitly model the implicit structures induced by physical equation constraints, thereby significantly enhancing the physical consistency of the neural network outputs. This structured improvement strategy effectively overcomes the limited generalization capability of conventional PINNs caused by weak physical constraints, enabling more stable solution performance in complex nonlinear systems.

Based on eight representative classes of solutions to the Manakov system, systematic and comprehensive numerical validations are conducted in this study. The results demonstrate that, even in the presence of rogue-wave solutions with strong singularities and highly nonuniform spatiotemporal distributions, as well as breather solutions exhibiting strongly nonlinear dynamical behaviors, the proposed method is able to maintain high parameter inversion accuracy and robust numerical stability. In addition, the method exhibits favorable performance in terms of computational time efficiency. Meanwhile, a series of ablation experiments are carried out to verify the necessity of each stage in the algorithmic framework and to further elucidate the critical role of the consistency constraints in IRON-Net. On this basis, the proposed algorithm is further extended to a variety of other nonlinear systems, including higher-dimensional ones. However, due to space limitations, the detailed experimental results are not included in this paper.

The proposed method not only provides a more reliable numerical tool for modeling complex wave propagation phenomena in nonlinear optics, but also offers new methodological insights into hybrid modeling paradigms that integrate physics-driven and data-driven approaches. Future work will further explore its applicability to fields such as fluid dynamics, quantum systems, and biophysics, and investigate its integration with multiscale modeling and adaptive sampling strategies to further enhance its practical value in complex scientific and engineering problems.

Acknowledgements. This work was supported by the Hubei Provincial Natural Science Foundation (Grant No. 2023AFB873) and the National Natural Science Foundation of China (Grant Nos. 12505006, 11975172, 12261131495, and 12381240286).

References

[1] Marcuse D, Manyuk C R, and Wai P K A 1997 *J. Lightwave Technol.* **15** 1735
[2] Mumtaz S, Essiambre R J, and Agrawal G P 2012 *J. Lightwave Technol.* **31** 398
[3] Yang J 2002 *Phys. Rev. E* **65** 036606
[4] Kamchatnov A M and Sokolov V V 2015 *Phys. Rev. A* **91** 043621
[5] Kuetche V K, Bouetou T B, and Kofane T C 2010 *Phys. Rev. A* **82** 053619
[6] Lazarides N, Veldes G P, Frantzeskakis D J, and Kourakis I 2024 *Sci. Rep.* **14** 2150
[7] He J, Guo L, Zhang Y, and Chabchoub A 2014 *Proc. R. Soc. A-Math. Phys. Eng. Sci.* **470** 20140318
[8] Raissi M, Perdikaris P, and Karniadakis G E 2019 *J. Comput. Phys.* **378** 686
[9] Jagtap A D, Mao Z, Adams N, and Karniadakis G E 2022 *J. Comput. Phys.* **466** 111402
[10] Baldan M, Di Barba P, and Lowther D A 2023 *IEEE Trans. Magn.* **59** 1
[11] Lu L, Pestourie R, Yao W, Wang Z, Verdugo F, and Johnson S G 2021 *SIAM J. Sci. Comput.* **43** B1105
[12] Gao H, Zahr M J, and Wang J X 2022 *Comput. Meth. Appl. Mech. Eng.* **390** 114502
[13] Zheng H, Huang Y, Huang Z, Hao W, and Liu G 2024 *J. Comput. Phys.* **500** 112751
[14] Liu X M, Zhang Z Y, and Liu W J 2023 *Chin. Phys. Lett.* **40** 070501
[15] Hu C, Zhang Z, Liu M, Xiang L, Wu H, Liu W, and He Z 2024 *Chin. Phys. Lett.* **41** 120303
[16] Wu G Z, Fang Y, Kudryashov N A, Wang Y Y, and Dai C Q 2022 *Chaos Solitons Fractals* **159** 112143
[17] Jiang J H, Si Z Z, Dai C Q, and Wu B 2024 *Chaos Solitons Fractals.* **180** 114581
[18] Fang Y, Wu G Z, Wen X K, Wang Y Y, and Dai C Q 2022 *Opt. Laser Technol.* **155** 108428
[19] Peng W Q and Chen Y 2024 *Chaos* **34** 043124
[20] Miao Z W and Chen Y 2022 *Mod. Phys. Lett. B* **36** 2150531
[21] Sun J, Chen Y, and Tang X 2024 *J. Syst. Sci. Complex.* **37** 545
[22] Pu J, Li J, and Chen Y 2021 *Nonlinear Dyn.* **105** 1723
[23] Pu J C, Li J, and Chen Y 2021 *Chin. Phys. B* **30** 060202
[24] Wang S, Teng Y, and Perdikaris P 2021 *SIAM J. Sci. Comput.* **43** A3055
[25] Luo K, Zhao J, Wang Y, Li J, Liang L, Soekmadji H, and Liao S 2025 *Artif. Intell. Rev.* **58** 323

[26] Wang S, Wang H, and Perdikaris P 2021 *Comput. Meth. Appl. Mech. Eng.* **384** 113938
[27] Wang P, Sun Y, Wang J, Yang A, Li N, and Li S 2025 *Nonlinear Dyn.* **113** 31541
[28] Su M, Zhang Z, Liu M, Qi F, Jiang Y, and Liu W 2026 *Nonlinear Dyn.* **114** 185
[29] Chen Y, Xiao H, Teng X, Liu W, and Lan L 2024 *Inf. Fusion* **102** 102041
[30] Xu S Y, Yang A C, and Zhou Q 2024 *Ukr. J. Phys. Opt.* **25** S1137
[31] Yang A, Xu S, Liu H, Li N, and Sun Y 2025 *Nonlinear Dyn.* **113** 1523
[32] Yang A, Sun Y, Wang J, Wang P, and Li N 2026 *Chin. J. Phys.* **99** 12
[33] Yu J, Lu L, Meng X, and Karniadakis G E 2022 *Comput. Meth. Appl. Mech. Eng.* **393** 114823
[34] Jagtap A D and Karniadakis G E 2020 *Commun. Comput. Phys.* **28** 2002
[35] Zhang W and Li J 2023 *Comput. Math. Appl.* **132** 161
[36] Sukumar N and Srivastava A 2022 *Comput. Meth. Appl. Mech. Eng.* **389** 114333
[37] Lin S and Chen Y 2022 *J. Comput. Phys.* **457** 111053
[38] Liu Z, Liu Y, Yan X, Liu W, Nie H, Guo S, and Zhang C 2025 *Nat. Commun.* **16** 9558
[39] Tian S, Yu Y, and Li B 2025 *Chin. Phys. Lett.* **42** 120001
[40] Tian S, Li B, and Zhang Z 2024 *Chin. Phys. Lett.* **41** 030201
[41] Mittal S 2020 *J. Syst. Architect.* **104** 101689
[42] Jensen P S 1972 *Comput. Struct.* **2** 17
[43] Zuazua E 2005 *SIAM Rev.* **47** 197
[44] Patidar K C 2005 *J. Differ. Equ. Appl.* **11** 735
[45] Yang J 1999 *Phys. Rev. E* **59** 2393
[46] Gao X, Meng L, and Zhao L 2025 *Phys. Rev. E* **111** 054209
[47] Che W, Liu C, and Akhmediev N 2023 *Phys. Rev. E* **107** 054206
[48] Gelash A and Raskovalov A 2023 *Stud. Appl. Math.* **150** 841
[49] Priya N V, Senthilvelan M, and Lakshmanan M 2013 *Phys. Rev. E* **88** 022918
[50] Chen S and Mihalache D 2015 *J. Phys. A-Math. Theor.* **48** 215202
[51] Zhong W, Belić M, and Malomed B 2015 *Phys. Rev. E* **92** 053201
[52] Ferreira F M, Costa C S, Sygletos S, and Ellis A D 2019 *J. Lightwave Technol.* **37** 989
[53] D'Amico A, Correia B, London E, Virgillito E, Borraccini G, Napoli A, and Curri V 2022 *J. Lightwave Technol.* **40** 3499
[54] Shields M D and Zhang J 2016 *Reliab. Eng. Syst. Saf.* **148** 96
[55] Fang Y, Wang Y Y, Liu W, and Dai C Q 2022 *Comput. Math. Appl.* **127** 144
[56] Yin Y H and Lü X 2023 *Commun. Nonlinear Sci. Numer. Simul.* **126** 107441
[57] Wu J, Wang Z, Chen R, and Li Q 2025 *J. Lightwave Technol.* **43** 1372
[58] Jiang X, Wang D, Chen X, and Zhang M 2022 *J. Lightwave Technol.* **40** 7095
[59] Penwarden M, Jagtap A D, Zhe S, Karniadakis G E, and Kirby R M 2023 *J. Comput. Phys.* **493** 112464
[60] Zhou H, Zhang Z, Liu M, Qi F, and Liu W 2025 *Chin. Phys. Lett.* **42** 070001
[61] Zhou P, Xie X, Lin Z, and Yan S 2024 *IEEE Trans. Pattern Anal. Mach. Intell.* **46** 6486
[62] Chabchoub A 2016 *Phys. Rev. Lett.* **117** 144103
[63] Liu C and Akhmediev N 2019 *Phys. Rev. E* **100** 062201