

RESEARCH ARTICLE | MAY 06 2022

Data driven soliton solution of the nonlinear Schrödinger equation with certain $\mathcal{PT}$ -symmetric potentials via deep learning

J. Meiyazhagan; K. Manikandan ; J. B. Sudharsan; M. Senthilvelan  

Chaos 32, 053115 (2022)

<https://doi.org/10.1063/5.0086038>

APL Machine Learning

2023 Papers with Best Practices in Data Sharing and Comprehensive Background

Read Now



Data driven soliton solution of the nonlinear Schrödinger equation with certain $\mathcal{PT}$ -symmetric potentials via deep learning

Cite as: Chaos 32, 053115 (2022); doi: 10.1063/5.0086038

Submitted: 21 January 2022 · Accepted: 18 April 2022 ·

Published Online: 6 May 2022



View Online



Export Citation



CrossMark

J. Meiyazhagan,¹ K. Manikandan,²  J. B. Sudharsan,² and M. Senthilvelan^{1,a)} 

AFFILIATIONS

¹Department of Nonlinear Dynamics, Bharathidasan University, Tiruchirappalli 620 024, Tamil Nadu, India

²Centre for Nonlinear Systems, Chennai Institute of Technology, Chennai 600 069, Tamil Nadu, India

^{a)}Author to whom correspondence should be addressed: velan@cniid.bdu.ac.in

ABSTRACT

We investigate the physics informed neural network method, a deep learning approach, to approximate soliton solution of the nonlinear Schrödinger equation with parity time symmetric potentials. We consider three different parity time symmetric potentials, namely, Gaussian, periodic, and Rosen–Morse potentials. We use the physics informed neural network to solve the considered nonlinear partial differential equation with the above three potentials. We compare the predicted result with the actual result and analyze the ability of deep learning in solving the considered partial differential equation. We check the ability of deep learning in approximating the soliton solution by taking the squared error between real and predicted values. Further, we examine the factors that affect the performance of the considered deep learning method with different activation functions, namely, ReLU, sigmoid, and tanh. We also use a new activation function, namely, sech, which is not used in the field of deep learning, and analyze whether this new activation function is suitable for the prediction of soliton solution of the nonlinear Schrödinger equation for the aforementioned parity time symmetric potentials. In addition to the above, we present how the network's structure and the size of the training data influence the performance of the physics informed neural network. Our results show that the constructed deep learning model successfully approximates the soliton solution of the considered equation with high accuracy.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0086038>

The soliton in complex parity time ($\mathcal{PT}$) symmetric media has attained an undeniable advantage of gain and loss distribution in its dynamics. Such dynamical studies on solitons have been widely analyzed in mode-locked lasers, ultrashort pulse optics, optical solitons in communication systems, and atomic lasers. Solving partial differential equations (PDEs) using deep learning approaches have become popular because the method depends on the optimization techniques. Physics Informed Neural Networks (PINNs) are one kind of optimization method, which is used to solve a wide class of PDEs, including the nonlinear Schrödinger (NLS) equation. We utilize the PINN method to approximate the soliton solution of the NLS equation with three different $\mathcal{PT}$ -symmetric potentials, namely, Gaussian, periodic, and Rosen–Morse potentials. The PINN method accurately approximates the soliton solution of the considered NLS equation for all three potentials.

I. INTRODUCTION

For the past four decades, soliton and its applications have been studied in depth in several branches of optics. In particular, the demand for harnessing the fruitfulness of solitons in nonlinear fiber optics and communication systems has attracted a plethora of interest.¹ Mathematically, the dynamics of such optical soliton pulses can be described by the nonlinear Schrödinger (NLS) equation. By properly managing the dispersion and nonlinearity parameters in the NLS equation, one can generate a stable soliton.

It has also been shown that by introducing a proper complex parity time ($\mathcal{PT}$) symmetric potential in the NLS equation, one can gain more access to the optical soliton pulse propagation. Even though the complex $\mathcal{PT}$ -symmetric potential is non-Hermitian in nature, the underlying system admits real eigenspectra² and it also supports a continuous range of stable optical solitons.^{3–5} The dynamical behaviors of $\mathcal{PT}$ -symmetric optical solitons have

been investigated in many optical experiments and theoretical models.^{6–14}

Nowadays, Machine Learning (ML) and Deep Learning (DL) approaches have become important tools in the prediction task in various fields of physics.^{15–18} In the field of nonlinear dynamics, ML methods have been used for the replication of chaotic attractors,¹⁹ prediction of chaotic laser pulses amplitude,²⁰ detection of unstable periodic orbits,²¹ chaotic signals separation,²² network classification from symbolic time series,²³ identification of chimera states,^{24–26} and also in the study of extreme events.^{27–31}

The rapid growth in the field of DL enables us to solve linear and nonlinear partial differential equations (PDEs) by an approximation technique, namely, Physics Informed Neural Network (PINN), which was introduced by Raissi *et al.*³² For the past couple of years, PINNs have been widely used to solve NLS equation and its generalizations.^{33–37} In this direction, quite recently, the logarithmic NLS equation with $\mathcal{PT}$ -symmetric harmonic potential and Scarf-II potential has been solved through PINN approach.^{38,39} In the present work, we consider NLS equation with three different $\mathcal{PT}$ -symmetric potentials, namely, Gaussian, periodic, and Rosen–Morse potentials and approximate the soliton solution of all three cases with the PINN approach. In our study, we introduce a new activation function, namely, sech and test the ability of this new function by comparing it with the other activation functions that are being used in the literature. To the best of our knowledge, this is the first time wherein the PINN approach is being used to solve the NLS equation with the above said $\mathcal{PT}$ -symmetric potentials and this is also the first time sech is used as an activation function in this approach.

We organize our presentation as follows. In Sec. II, we present the methodologies involved in the PINN approach and the general way of solving the considered NLS equation with PINN method. The data driven soliton solution of the NLS equation with all three considered potentials, a comparison with exact solution, and the error occurring in this approximation are given in Sec. III. A comparative study on factors that affect the performance of the PINN is discussed in Sec. IV. We present our conclusions in Sec. V.

II. PINN AND THE NLS EQUATION WITH $\mathcal{PT}$ -SYMMETRIC POTENTIAL

A. The scheme of PINN

Usually, the PINNs have been used for solving nonlinear PDEs that have the general form³²

$$u_t - \mathcal{N}[u(x, t); \lambda] = 0, \quad x \in \Omega, \quad t \in [0, T]. \quad (1)$$

In this work, we consider the complex nonlinear PDEs with the following initial and boundary conditions, that is,

$$\begin{cases} iu_t = \mathcal{N}[u(x, t); \lambda_0], & x \in \Omega, \quad t \in [0, T], \\ I[u(x, t)]|_{t=0} = u_I(x), & x \in \Omega \text{ (initial condition),} \\ B[u(x, t)]|_{x \in \partial\Omega} = u_B(t), & t \in [0, T] \text{ (boundary conditions),} \end{cases} \quad (2)$$

where $u(x, t)$ is the solution of the PDE, $\mathcal{N}[\cdot, \lambda]$ is the combination of linear and nonlinear operators, which are parameterized by the initial vector λ_0 , $[0, T]$ represents the lower and upper boundary of the time variable t , Ω and $\partial\Omega$ denote the spatial variable range and

the boundary of that domain, respectively, I and B are operators corresponding to initial and boundary values, $I[u(x, t)]|_{t=0} = u_I(x)$ and $B[u(x, t)]|_{x \in \partial\Omega} = u_B(t)$, respectively, represent the initial and boundary conditions. We define a complex-valued physics model $f(x, t)$ as follows:

$$f(x, t) := iu_t - \mathcal{N}[u; \lambda_0]. \quad (3)$$

We can differentiate the latent solution $u(x, t)$ with respect to time variable t and spatial variable x using the derivative technique, namely, Automatic Differentiation (AD)^{40,41} based on the chain rule, which is used to make backpropagation (BP)⁴² in Artificial Neural Networks (ANNs). For the implementation of BP, AD, and other optimization steps involved in the complex-valued PINN, we use Tensorflow,⁴³ which is a well known open-source software library used for AD and DL computations. We use four different kinds of activation functions for the activation of neurons in the ANN (a comparative study on the activation functions is given in Sec. IV A). However, for the main study, we choose tanh as the nonlinear activation function, which is being used in the current literature in the form³²

$$Z_l = \tanh(w_l Z_{l-1} + b_l), \quad l = 1, 2, 3, \dots, n, \quad (4)$$

where w_l is the $\dim(Z_l) \times \dim(Z_{l-1})$ weight matrix and b_l is the $\dim(Z_l)$ bias vector. We define the loss function for the training process as

$$\begin{aligned} L_{Train} = & \frac{1}{N_I} \sum_{j=1}^{N_I} |I[u(x_I^j, t)]|_{t=0} - u_I(x_I^j)|^2 \\ & + \frac{1}{N_B} \sum_{j=1}^{N_B} |B[u(x_B^j, t_B^j)]|_{x_B \in \partial D} - u_B(t_B^j)|^2 \\ & + \frac{1}{N_C} \sum_{j=1}^{N_C} |f(x_C^j, t_C^j)|^2, \end{aligned} \quad (5)$$

where $\{x_I^j, u_I^j\}_{j=1}^{N_I}$ and $\{t_B^j, u_B^j\}_{j=1}^{N_B}$ are, respectively, represent the initial and boundary conditions and $\{x_C^j, t_C^j, f(x_C^j, t_C^j)\}_{j=1}^{N_C}$ denotes the collocation points of $f(x, t)$. We create the sample points using Latin Hypercube Sampling (LHS) algorithm⁴⁴ and the optimization for the loss function by the Limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm.⁴⁵

The major steps involved in solving the PDE (1), with initial and boundary conditions (2), using the PINN method, are given below:

- (i) Defining the structure of ANN, which is described by a fixed number of layers and a fixed number of neurons.
- (ii) Preparing three training sets, namely, (i) the initial condition set, (ii) boundary conditions sets, and (iii) the random collocation points using the LHS technique.⁴⁴
- (iii) Getting the training loss function L_{Train} given in (5) by adding weighted L^2 -norm errors of the initial, boundary condition residuals, and $f(x, t)$.
- (iv) Train the ANN in order to get suitable values of $\{\hat{w}, \hat{b}\}$ to minimize the L_{Train} using the L-BFGS algorithm.

Using these four steps, we approximate the solution of the considered PDE.

B. PINN method for NLS equation with $\mathcal{PT}$ -symmetric potential

We consider NLS equation with a $\mathcal{PT}$ -symmetric potential in the form

$$i\psi_t + \psi_{xx} + P(x)\psi + \sigma|\psi|^2\psi = 0, \quad (6)$$

where $\psi = \psi(x, t)$ is a complex field, σ is the nonlinear coefficient corresponding to focusing and defocusing interactions, and $P(x)$ is the $\mathcal{PT}$ -symmetric potential, which has the form

$$P(x) = [V(x) + iW(x)], \quad (7)$$

where $V(x)$ and $W(x)$ are real and imaginary parts of the $\mathcal{PT}$ -symmetric potential and they should satisfy the following two conditions:

$$V(-x) = V(x), \quad W(-x) = -W(x). \quad (8)$$

In this work, to obtain the soliton solution of (6) using the above mentioned PINN method,³² we define the equation, initial and boundary conditions, respectively, as follows:

$$i\psi_t = -\psi_{xx} - P(x)\psi - \sigma|\psi|^2\psi, \quad x \in (-L, L), \quad t \in (0, T), \quad (9)$$

$$\psi(x, 0) = \psi_0(x), \quad x \in [-L, L], \quad (10a)$$

$$\psi(-L, t) = \psi(L, t), \quad t \in [0, T]. \quad (10b)$$

Since the solution $\psi(x, t)$ of Eq. (9) is complex, we consider it in the form $\psi(x, t) = u(x, t) + iv(x, t)$, where $u(x, t)$ and $v(x, t)$ are two real functions denoting real and imaginary parts of the solution ψ , respectively. Now, we use the associated complex-valued PINN as $f(x, t) = if_u(x, t) - f_v(x, t)$ with $-f_v(x, t)$ and $f_u(x, t)$ being the real and imaginary parts of $f(x, t)$, respectively. The explicit form of the functions reads

$$f(x, t) = i\psi_t + \psi_{xx} + [V(x) + iW(x)]\psi + \sigma|\psi|^2\psi, \quad (11a)$$

$$f_u(x, t) = u_t + v_{xx} + V(x)v + W(x)u + \sigma(u^2 + v^2)v, \quad (11b)$$

$$f_v(x, t) = v_t - u_{xx} - V(x)u + W(x)v - \sigma(u^2 + v^2)u, \quad (11c)$$

and, using this, we approximate $\psi(x, t)$ by a complex-valued deep neural network. The shared parameters between $\psi(x, t)$ and $f(x, t)$ can be trained by minimizing L_{Train} , which is the combination of

three mean squared errors as given below,

$$L_{Train} = L_I + L_B + L_C, \quad (12)$$

where the mean squared errors are taken in the form

$$\begin{aligned} L_I &= \frac{1}{N_I} \sum_{j=1}^{N_I} \left(\left| u(x_I^j, 0) - u_0^j \right|^2 + \left| v(x_I^j, 0) - v_0^j \right|^2 \right), \\ L_B &= \frac{1}{N_B} \sum_{j=1}^{N_B} \left(\left| u(-L, t_B^j) - u(L, t_B^j) \right|^2 \right. \\ &\quad \left. + \left| v(-L, t_B^j) - v(L, t_B^j) \right|^2 \right), \\ L_C &= \frac{1}{N_C} \sum_{j=1}^{N_C} \left(\left| f_u(x_C^j, t_C^j) \right|^2 + \left| f_v(x_C^j, t_C^j) \right|^2 \right), \end{aligned} \quad (13)$$

with $\{x_I^j, u_0^j, v_0^j\}_{j=1}^{N_I}$ denoting the initial data, $\{t_B^j, u(\pm L, t_B^j), v(\pm L, t_B^j)\}_{j=1}^{N_B}$ denoting the boundary data, and $\{x_C^j, t_C^j, f_u(x_C^j, t_C^j), f_v(x_C^j, t_C^j)\}_{j=1}^{N_C}$ denoting the collocation points on $f(x, t)$. The losses L_I , L_B , and L_C , respectively, represent the $\mathbb{L}^2$ -norm error in initial, boundary, and inside the spatiotemporal regime. Figure 1 shows the schematic diagram of the PINN. The left panel of the figure corresponds to the ANN, where we have two input neurons for space and time and two output neurons for real and imaginary parts of the solution. The right panel shows the physics information, which we give as a form of training loss function L_{Train} .

III. DATA DRIVEN SOLUTIONS OF THE NLS EQUATION WITH $\mathcal{PT}$ -SYMMETRIC POTENTIALS

In this section, we present the outcomes of DL while solving the focusing ($\sigma = 1$) NLS equation with soliton solution with three $\mathcal{PT}$ -symmetric potentials, namely, (i) Gaussian, (ii) periodic, and (iii) Rosen–Morse potential. To make the PINN to solve the considered problem, we are in need of training set data. The training set consists of $N_I = 50$ data points on initial conditions, $N_B = 100$ data points on the periodic boundary conditions (50 on the upper boundary and another 50 on the lower boundary), and $N_C = 20\,000$ collocation points, which are chosen randomly using the LHS⁴⁴ method. We choose a six-layer ANN in which the first and the last layer have two neurons, which are used for input (x, t) and output $(u(x, t), v(x, t))$. The other four hidden layers have 100 units of neurons each. The hyperbolic tangent function given in Eq. (4) is used for the activation of the neurons. The space and time interval are taken as $L = 10$ and $T = 5$, respectively. So, the limit for spatial and time points are $[-10, 10]$ and $[0, 5]$. The PINN model has been run for 40 000 optimization steps to minimize the loss function L_{Train} . To verify the outcome of PINN, we compare the solutions obtained from PINN with the numerical solutions. To generate the latter data, we use the Fourier spectral method⁴⁶ with a special Fourier discretization with 256 space modes and a fourth-order explicit Runge–Kutta temporal integrator with 201 points at the same space/time interval to solve the NLS equation (9). So, $\psi(x, t)$ is a 256×201 matrix. We note here that the solution obtained using

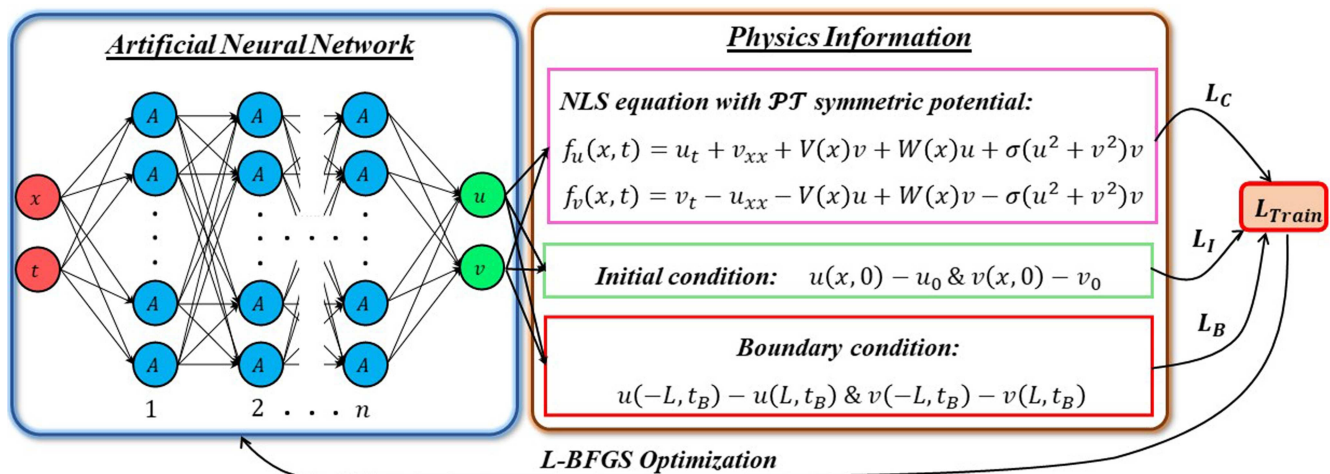


FIG. 1. Schematic diagram of PINN. The left panel shows the input and output layers with two neurons and n hidden layers. Each neuron is activated by the activation function A . The right panel corresponds to the physics information of the PINN given as a loss function of the optimization problem.

the above said numerical method is just to access the accuracy of the PINN solution. Training of the PINN itself does not require a numerical solution.

The above setup has been considered same for all three $\mathcal{PT}$ -symmetric potentials throughout this work except for the activation function which we change in each case since it will be used to study the influence in the accuracy of solving the NLS equation with $\mathcal{PT}$ -symmetric potential using the PINN method.

A. NLS equation with $\mathcal{PT}$ -symmetric Gaussian potential

To begin, we consider the NLS Eq. (9) with $\mathcal{PT}$ -symmetric Gaussian potential,⁴⁷

$$P(x) = V(x) + iW(x) = e^{-x^2} + iW_0 x e^{-x^2}, \quad (14)$$

where W_0 is the strength of the imaginary part with the value 0.1. The real ($V(x)$) and the imaginary ($W(x)$) parts of the potential $P(x)$ given in (14) satisfy the conditions given in (8). The Gaussian profile is taken as the initial profile to solve the NLS equation (9) with the above potential (14). After training, the PINN with the above mentioned setup with the Gaussian potential, the approximated solution is shown in Fig. 2. Figures 2(a) and 2(b), respectively, represent the predicted and exact magnitudes of the soliton solution $|\psi(x, t)| = \sqrt{u^2(x, t) + v^2(x, t)}$ for the NLS equation (9) with $\mathcal{PT}$ -symmetric Gaussian potential (14). The star markers in Fig. 2(a) denote the randomly chosen data points on the initial (50 points) and boundary (100 points) conditions. Figure 2(c) shows the value of squared errors between the predicted and exact values of the solution. From Fig. 2(a), we can see that the predicted soliton solution of the NLS Eq. (9) with potential (14) is similar to that of the exact solution shown in Fig. 2(b). To examine the error between these two solutions, we plot the squared error of them in Fig. 2(c). This figure infers that the error value between the predicted and exact solutions is of the order of 10^{-6} . The relative $\mathbb{L}^2$ -norm

errors of $u(x, t)$, $v(x, t)$, and $\psi(x, t)$, respectively, are 2.1856×10^{-2} , 2.8822×10^{-2} , and 3.1912×10^{-3} . These results infer that the considered PINN is enabled to approximate the soliton solution of the NLS equation with considered Gaussian potential with low error values. Figures 2(d)–2(f) show the comparisons of exact and the predicted soliton solution at different times, say, $t = 0.65$, $t = 2.64$, and $t = 4.38$. The predicted solitons at different time instants are fitted well with the exact soliton solutions. This also confirms the ability of PINN in solving the NLS equation for the given Gaussian $\mathcal{PT}$ -symmetric potential.

B. NLS equation with $\mathcal{PT}$ -symmetric periodic potential

Let us now consider the potential $P(x)$ in (9) in the form³

$$P(x) = V(x) + iW(x) = \cos^2 x + iW_0 \sin 2x, \quad (15)$$

where the value of strength of the imaginary part is $W_0 = 0.45$. Since the potential $P(x)$ is $\mathcal{PT}$ -symmetric, the real ($V(x)$) and imaginary ($W(x)$) parts satisfy the conditions mentioned in (8). Here also, the Gaussian profile is considered as the initial profile to solve the NLS equation (9) with $\mathcal{PT}$ -symmetric periodic potential (15). After the initial common setup made for training the PINN, as mentioned before, we obtain the soliton solution from PINN. The obtained results are reported in Fig. 3. The predicted and exact magnitudes of the soliton solution $|\psi(x, t)|$ obtained are shown in Figs. 3(a) and 3(b), respectively. The data points, which are randomly chosen on the initial and boundary conditions for the purpose of training, are denoted as stars in Fig. 3(a). The squared error values between the predicted and exact solutions are shown in Fig. 3(c). From Fig. 3(c), we can see that the error values are of the order of 10^{-5} , which confirms that our constructed PINN model succeeds in approximating the soliton solution of the NLS equation with $\mathcal{PT}$ -symmetric periodic potential (15) with high accuracy. Further, to check the correctness of the solution, we plot the solution $|\psi(x, t)|$ at different

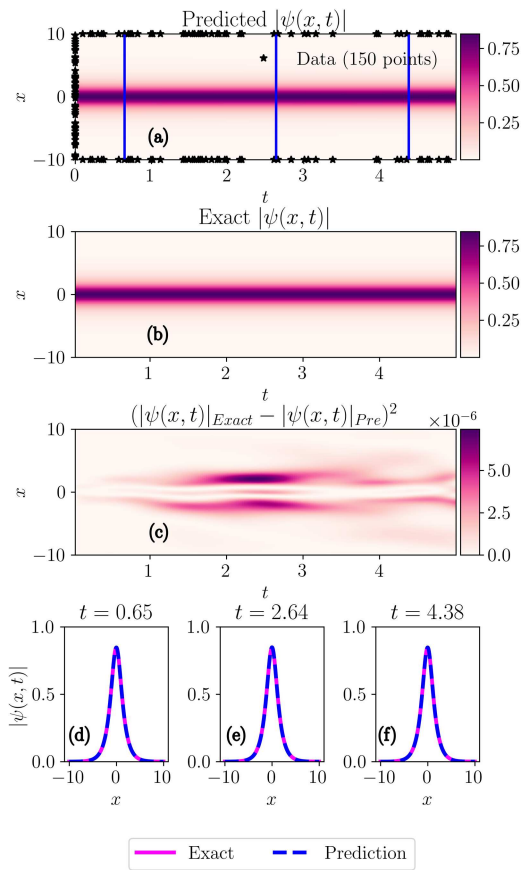


FIG. 2. Results of PINN in approximating soliton solution of NLS equation (9) with Gaussian potential (14). (a) represents the predicted values of $|\psi(x, t)|$ by PINN. The stars in (a) represent randomly selected data points on initial and boundary conditions. (b) showing the exact values of $|\psi(x, t)|$. (c) corresponds to the squared error values between the predicted and exact results. (d)–(f) A comparison of approximation done by PINN in finding soliton solution at particular time instants $t = 0.65$, $t = 2.64$, and $t = 4.38$.

instants of time, say, $t = 0.65$, $t = 2.64$, and $t = 4.38$ in Figs. 3(d), 3(e), and 3(f), respectively, and these figures also confirm that the solution obtained through PINN is accurate since the exact and predicted solutions coincide with each other. The relative $\mathbb{L}^2$ -norm error values in $u(x, t)$, $v(x, t)$ and $\psi(x, t)$ in this case are found to be 5.0925×10^{-2} , 4.9897×10^{-2} , and 4.272×10^{-3} .

C. NLS equation with $\mathcal{PT}$ -symmetric Rosen-Morse potential

Next, we consider another $\mathcal{PT}$ -symmetric potential, namely, Rosen-Morse potential, which is given by Ref. 48,

$$P(x) = V(x) + iW(x) = -a(a+1)\text{sech}^2 x + i2b \tanh x, \quad (16)$$

where a and b are parameters, which we will take as 0.1 and 0.03. The potential considered in (16) also satisfies the conditions given

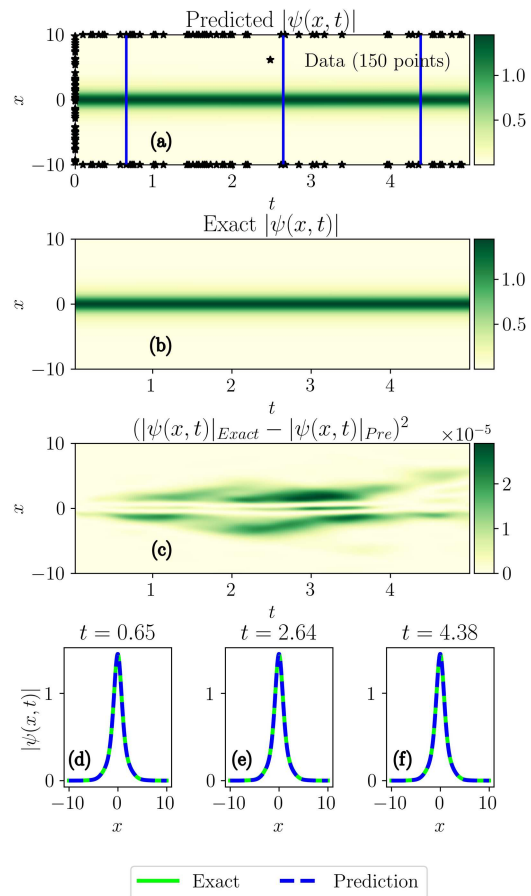


FIG. 3. Results of PINN in approximating soliton solution of NLS equation (9) with periodic potential (15). (a) represents the predicted values of $|\psi(x, t)|$ by PINN. The stars in (a) represent randomly selected data points on initial and boundary conditions. (b) shows the exact values of $|\psi(x, t)|$. (c) corresponds to the squared error values between the predicted and exact results. (d)–(f) A comparison of approximation done by PINN in finding soliton solution at particular time instants $t = 0.65$, $t = 2.64$, and $t = 4.38$.

in (8). We take the initial profile in the form⁴⁸

$$\psi(x) = \sqrt{a^2 + a + 2} \text{sech } x e^{ibx}, \quad (17)$$

which satisfies the stationary part of (9). We consider the same preliminary setup and train the PINN as in the case of Gaussian and periodic potentials and obtain the outcome which we present in Fig. 4. Figures 4(a) and 4(b), respectively, correspond to the predicted and exact magnitudes of the soliton solution of the NLS equation (9) with Rosen-Morse potential (16). The stars at the boundary of Fig. 4(a) denote the data points taken in the initial and boundary conditions. The squared error values between exact and the predicted ones are in the order of 10^{-4} , which is reported in Fig. 4(c). Figures 4(a)–4(c) reveal that PINN succeeds in approximating the soliton solution of the Rosen-Morse potential as well. The relative $\mathbb{L}^2$ -norm error values of $u(x, t)$, $v(x, t)$, and $\psi(x, t)$

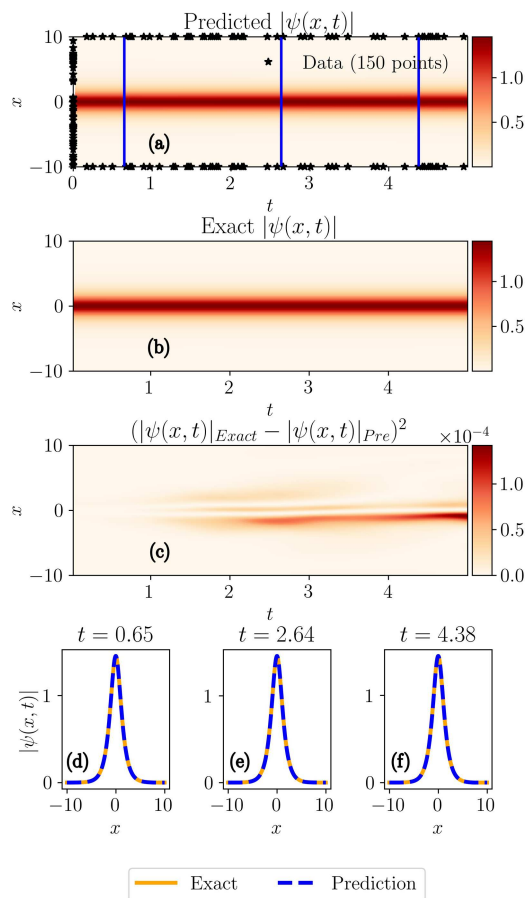


FIG. 4. Results of PINN in approximating soliton solution of NLS equation (9) with Rosen–Morse potential (16). (a) represents the predicted values of $|\psi(x, t)|$ by PINN. The stars in (a) represent randomly selected data points on initial and boundary conditions. (b) shows the exact values of $|\psi(x, t)|$. (c) corresponding to the squared error values between the predicted and exact results. (d)–(f) A comparison of approximation done by PINN in finding soliton solution at particular time instants $t = 0.65$, $t = 2.64$, and $t = 4.38$.

for this case are found to be 3.7277×10^{-2} , 3.2468×10^{-2} , and 5.9112×10^{-3} . In Figs. 4(d)–4(f), the exact and the predicted solitons are plotted one over the other at different time instants, say, for example, $t = 0.65$, $t = 2.64$, and $t = 4.38$ in order to check whether the predicted result is accurate or not. From these figures, we can see that the exact and predicted solutions fit well one over the other, indicating that the predicted result is accurate.

D. Non-stationary solution of NLS equation with PT -symmetric Rosen–Morse potential

Finally, we analyze the ability of PINN in predicting non-stationary solutions of the NLS equation. Let us consider an initial profile of a non-stationary solution to the NLS equation for the

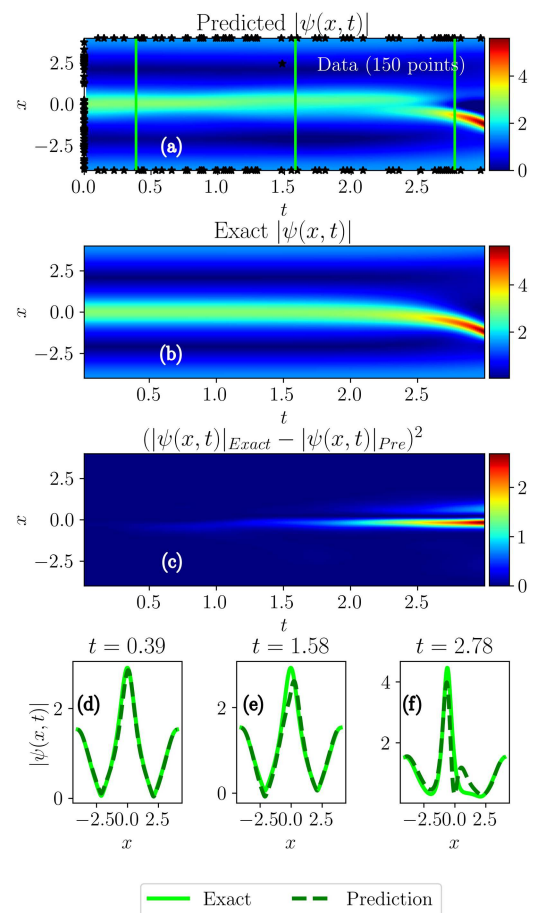


FIG. 5. Results of PINN in approximating non-stationary solution of NLS equation (9) with Rosen–Morse potential (15). (a) represents the predicted values of $|\psi(x, t)|$ by PINN. The stars in (a) represent randomly selected data points on initial and boundary conditions. (b) shows the exact values of $|\psi(x, t)|$. (c) corresponds to the squared error values between predicted and exact results. (d)–(f) A comparison of approximation done by PINN in finding soliton solution at particular time instants $t = 0.39$, $t = 1.58$, and $t = 2.78$.

potential (16) in the form

$$\psi(x) = \sqrt{a^2 + 1} \operatorname{sech} x e^{-ibx}, \quad (18)$$

which does not satisfy the stationary part of (9). Let us fix the parameters as $a = 1.75$ and $b = 0.35$. The obtained results after training the PINN with the same preliminary setup considered earlier are reported in Fig. 5. The predicted and exact magnitudes of $\psi(x, t)$ are presented, respectively, in Figs. 5(a) and 5(b). From these two figures, we observe that PINN successfully predicts the non-stationary solution as well. We have examined the error values between the predicted and exact results and plot the outcome in Fig. 5(c). We come across the relative L^2 -norm error values of $u(x, t)$, $v(x, t)$, and $\psi(x, t)$ for this case as 5.4182×10^{-1} , 5.5958×10^{-1} , and 2.8065×10^{-1} ,

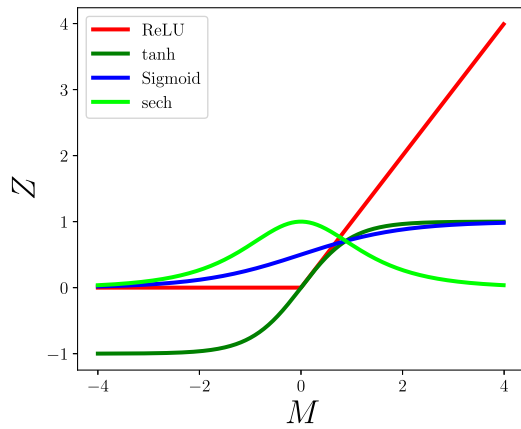


FIG. 6. Different kinds of activation functions.

respectively. To verify the obtained solution, we also plot the solution at different time instants, say, $t = 0.39$, $t = 1.58$, and $t = 2.78$ in Figs. 5(d), 5(e), and 5(f), respectively. The exact and predicted magnitudes of the predicted solution are fitted well in figures (d) and (e) but in Fig. 5(f), we can observe that the magnitude of the solution not fitted well with the exact one. This is due to the time-dependent nature of the solution. Our investigations reveal that one can solve the considered problem with less accuracy using PINN.

IV. FACTORS AFFECTING THE PERFORMANCE OF PINN

A. Effect of activation functions

In our main study, we have chosen tanh as the activation function [see Eq. (4)] because it gives us the solution with a low error value. To study the effect of other activation functions in approximating the soliton solution of the $\mathcal{PT}$ -symmetric potentials, we consider three other functions, namely, Rectified Linear Unit

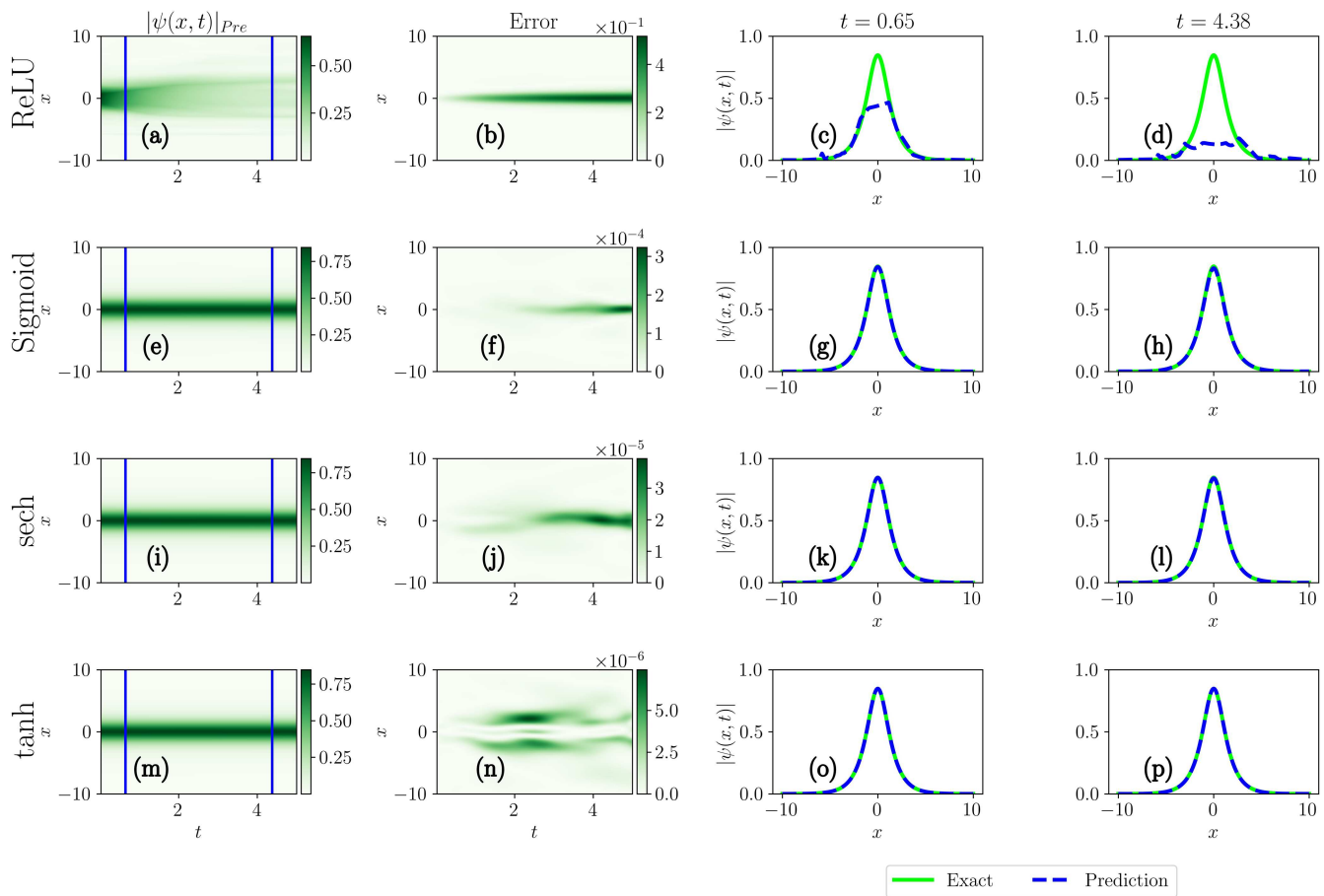


FIG. 7. PINN results for the Gaussian potential case with different activation functions. Rows one to four, respectively, represent the activation functions ReLU, sigmoid, sech, and tanh. (a), (e), (i), and (m) represent the predicted magnitude of soliton solutions. (b), (f), (j), and (n) represent the error values in magnitude of soliton solutions. (c), (g), (k), and (o) in the third column and (d), (h), (l), and (p) in fourth column correspond to the soliton solution at particular time instants $t = 0.65$ and $t = 4.38$, respectively.

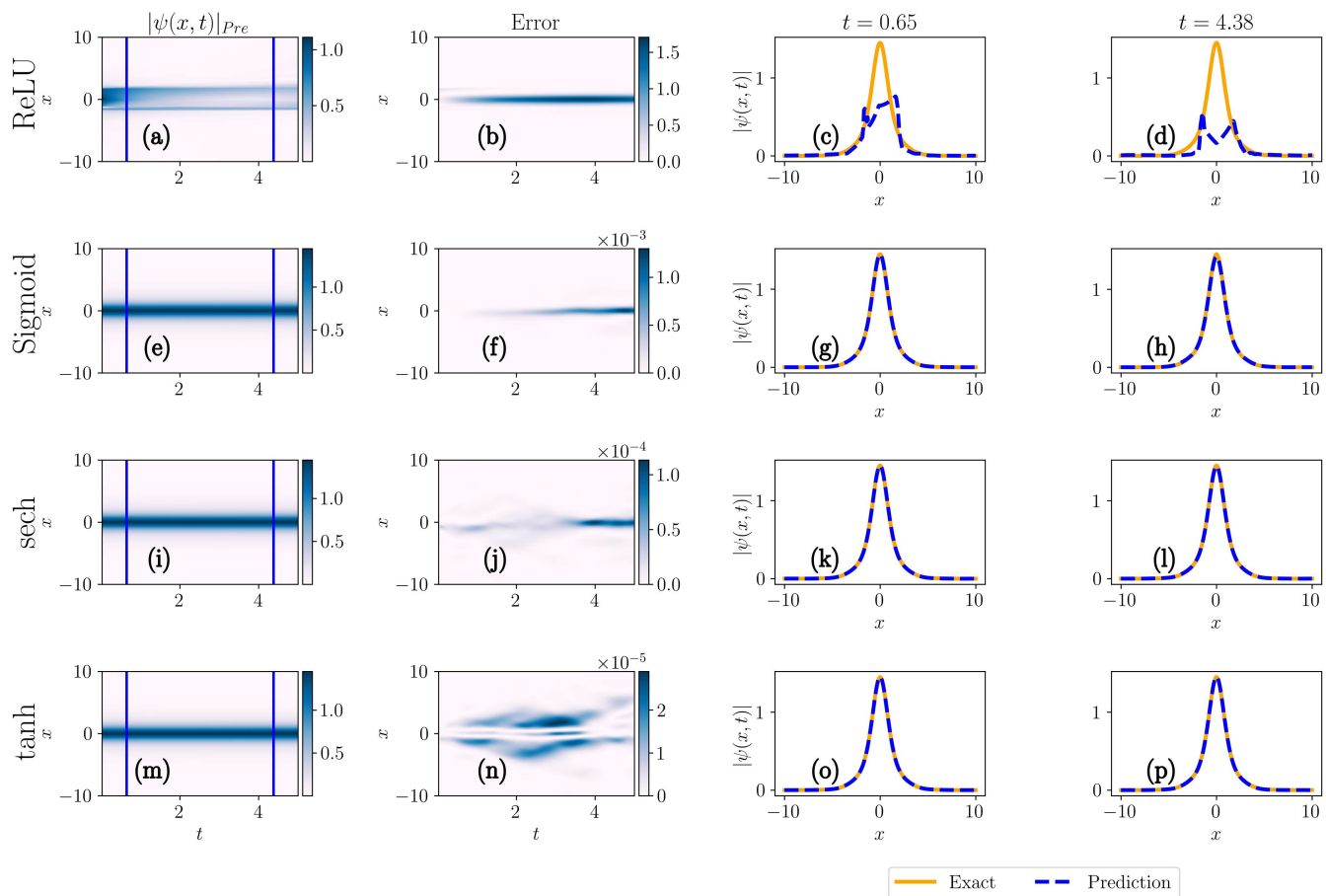


FIG. 8. PINN results for the periodic potential case with different activation functions. Rows one to four, respectively, represent the activation functions ReLU, sigmoid, sech, and tanh. (a), (e), (i), and (m) represent the predicted magnitude of soliton solutions. (b), (f), (j), and (n) represent the error values in magnitude of soliton solutions. (c), (g), (k), and (o) in the third column and (d), (h), (l), and (p) in the fourth column correspond to the soliton solution at particular time instants $t = 0.65$ and $t = 4.38$, respectively.

(ReLU), sigmoid, and sech. Since the problem under consideration is approximating the soliton solution of the NLS equation with various $\mathcal{PT}$ -symmetric potentials, we intend to use a new activation function, namely, sech, which has not been used in the field of DL. We consider the general form of the activation functions as given below,

$$(i) Z_j = \text{ReLU}(M) = \max(0, M), \quad (19a)$$

$$(ii) Z_j = \text{sigmoid}(M) = \frac{1}{1 + e^{-M}}, \quad (19b)$$

$$(iii) Z_j = \text{sech}(M), \quad (19c)$$

$$(iv) Z_j = \tanh(M), \quad (19d)$$

where $M = w_j Z_{j-1} + b_j$ as taken in (4). The functionality of each activation function can be visualized with the help of Fig. 6.

The predicted values and the error values in the case of NLS equation with $\mathcal{PT}$ -symmetric Gaussian potential in approximating the soliton solution with all four different activation functions are reported in Fig. 7. Figure 7(a) reveals that the function ReLU fails in approximating the soliton solution. The squared error value between the predicted and the actual magnitude of soliton solution comes out in the order of 10^{-1} only [Fig. 7(b)]. The comparison between the predicted and exact solutions at two other instants of time, say, at $t = 0.65$ and $t = 4.38$ are presented in Figs. 7(c) and 7(d), which also confirms that by using ReLU as the activation function, a good approximation cannot be obtained for Gaussian potential. The results for the other three activation functions, namely, sigmoid, sech, and tanh are plotted in Figs. 7(e)–7(h), 7(i)–7(l), and 7(m)–7(p), respectively. From the outcome, we can infer that the prediction done by PINN with tanh as activation function gives an accurate result when compared with the other three. As far as the Gaussian potential is concerned, we come across squared error values that are of the order of 10^{-1} , 10^{-4} , 10^{-5} , and

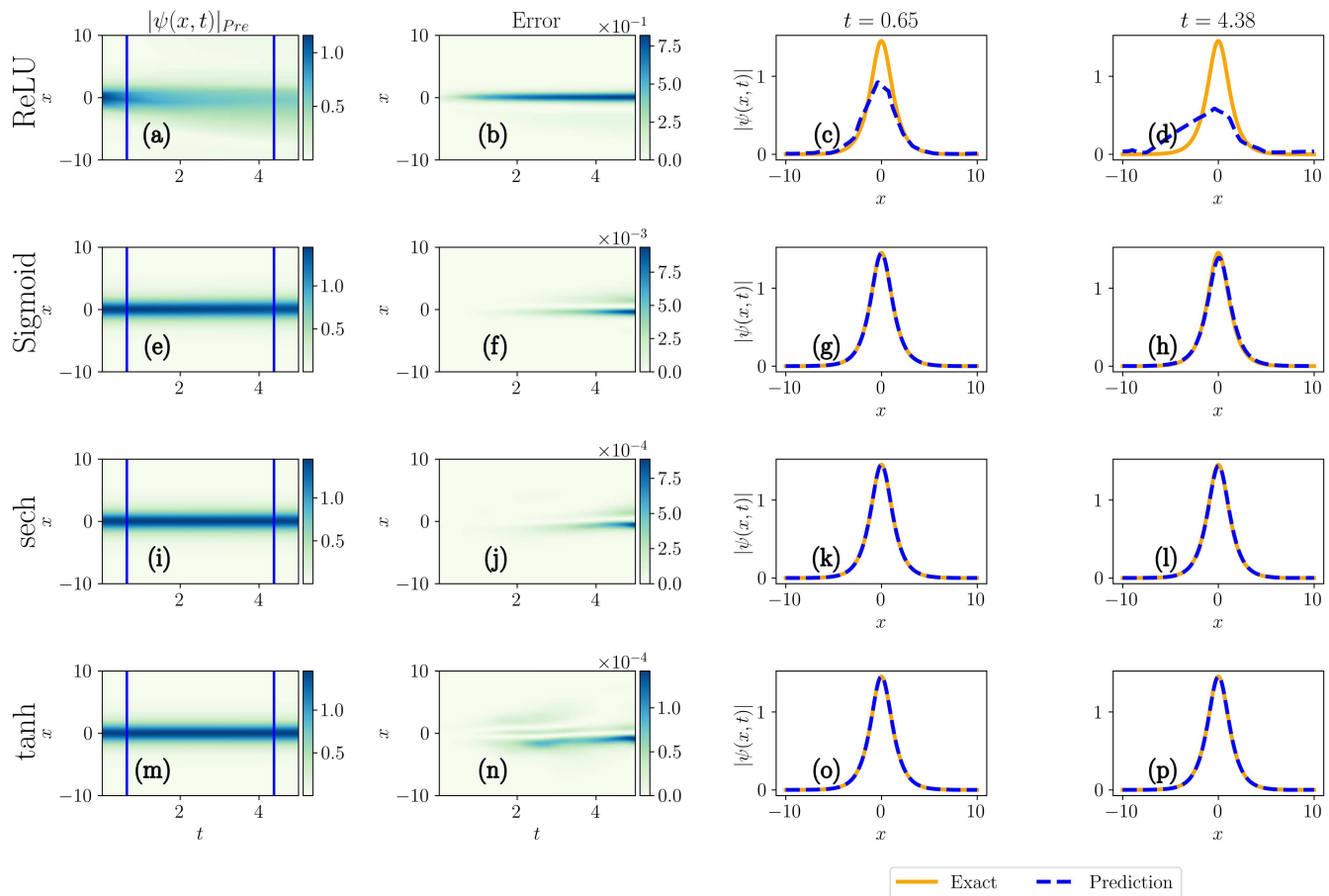


FIG. 9. PINN results for the Rosen–Morse potential with different activation functions. Rows one to four, respectively, represent the activation functions ReLU, sigmoid, sech, and tanh. (a), (e), (i), and (m) represent the predicted magnitude of soliton solutions. (b), (f), (j), and (n) represent the error values in magnitude of soliton solutions. (c), (g), (k), and (o) in the third column and (d), (h), (l), and (p) in the fourth column correspond to the soliton solution at particular time instants $t = 0.65$ and $t = 4.38$, respectively.

10^{-6} for the ReLU, sigmoid, sech, and tanh activation functions, respectively.

The results coming out from PINN with four different activation functions for the $\mathcal{PT}$ -symmetric periodic potential are reported in Fig. 8. The rows one to four represent the results coming out from PINN with the activation functions ReLU, sigmoid, sech, and tanh, respectively. From Figs. 8(c), and 8(d), we can see that the approximation done by PINN with ReLU as an activation function is not fitting well with the original result. But in the case of sech, the prediction is better while comparing with the prediction done by ReLU and sigmoid functions since the squared error value comes out less. From the plots given in the second column of Fig. 8, we infer that the squared error values of the cases ReLU, sigmoid, sech, and tanh are of the order of 10^0 , 10^{-3} , 10^{-4} , and 10^{-5} , respectively.

Prediction results of PINN with different activation functions for the NLS equation with Rosen–Morse potential are shown in Fig. 9. The results corresponding to PINN with ReLU as the activation function are shown in Figs. 9(a)–9(d). From Fig. 9(a) we can

see that the approximation done by the PINN is not accurate. From the squared error plots shown in the second column in Fig. 9, we can see that the error value is low in the case of tanh function. The plots on the third and fourth columns of Fig. 9 confirm that the predicted and exact results are fitted well with each other while we compare tanh activation function with the other activation functions.

The overall outcome is presented in Table I. For better comparison, we take the $\mathbb{L}^2$ -norm error value in approximating u , v , and ψ for all the three cases and also for four activation functions. $\mathbb{L}^2$ -norm error values of the PINN with ReLU as activation function are very high in approximating u , v , and ψ for all three potentials when compared with the other three activation functions. This is due to the piecewise linearity of the ReLU function. In the case of sigmoid activation function, the value of $\mathbb{L}^2$ -norm error is of the order of 10^{-2} . While comparing the outcome of the PINN with sech and tanh functions, it is clear that the error value is slightly low in the case of tanh while approximating the function ψ . But for the approximation of real (u) and imaginary (v) parts of the solution the PINNs with sech

TABLE I. $\mathbb{L}^2$ -norm error values in u , v , and ψ of the PINN results approximated with different activation functions for all three considered potentials.

$\mathcal{PT}$ -symmetric potentials	$\mathbb{L}^2$ -norm error	Activation functions			
		ReLU	sigmoid	sech	tanh
Gaussian	u	8.2281×10^{-1}	1.6449×10^{-2}	1.6841×10^{-2}	2.1856×10^{-2}
	v	7.2510×10^{-1}	2.0385×10^{-2}	1.8659×10^{-2}	2.8822×10^{-2}
	ψ	6.4185×10^{-1}	1.1915×10^{-2}	5.8949×10^{-3}	3.1912×10^{-3}
Periodic	u	1.0433×10^0	4.7092×10^{-2}	2.9961×10^{-2}	5.0925×10^{-2}
	v	1.0236×10^0	4.7094×10^{-2}	2.9019×10^{-2}	4.9897×10^{-2}
	ψ	6.9093×10^{-1}	1.3379×10^{-2}	5.2880×10^{-3}	4.2720×10^{-3}
Rosen–Morse	u	1.1185×10^0	1.1149×10^{-1}	2.4371×10^{-2}	3.7277×10^{-2}
	v	1.1529×10^0	8.2902×10^{-2}	2.4424×10^{-2}	3.2468×10^{-2}
	ψ	4.8206×10^{-1}	3.3237×10^{-2}	9.8817×10^{-3}	5.9112×10^{-3}

TABLE II. $\mathbb{L}^2$ -norm error values in u , v , and ψ of the PINN results approximated with different number of hidden layers of ANN for all three considered potentials.

$\mathcal{PT}$ -symmetric potentials	$\mathbb{L}^2$ -norm error	Number of hidden layers			
		1	2	3	4
Gaussian	u	4.0425×10^{-2}	2.0164×10^{-2}	2.3489×10^{-2}	2.1856×10^{-2}
	v	6.9994×10^{-2}	2.4169×10^{-2}	3.1864×10^{-2}	2.8822×10^{-2}
	ψ	2.8418×10^{-2}	5.5559×10^{-3}	3.5918×10^{-3}	3.1912×10^{-3}
Periodic	u	7.9002×10^{-1}	3.2177×10^{-2}	4.8171×10^{-2}	5.0925×10^{-2}
	v	9.7382×10^{-1}	3.1851×10^{-2}	4.7245×10^{-2}	4.9897×10^{-2}
	ψ	1.8630×10^{-1}	7.6633×10^{-3}	4.0443×10^{-3}	4.2720×10^{-3}
Rosen–Morse	u	3.9728×10^{-1}	2.1559×10^{-2}	3.7352×10^{-2}	3.7277×10^{-2}
	v	3.7073×10^{-1}	2.1047×10^{-2}	3.2467×10^{-2}	3.2468×10^{-2}
	ψ	8.3272×10^{-2}	6.4215×10^{-3}	5.8233×10^{-3}	5.9112×10^{-3}

TABLE III. $\mathbb{L}^2$ -norm error values in u , v , and ψ of the PINN results approximated with different number of neurons (10–50) in each hidden layer of ANN for all three considered potentials.

$\mathcal{PT}$ -symmetric potentials	$\mathbb{L}^2$ -norm error	Number of neurons in the hidden layers				
		10	20	30	40	50
Gaussian	u	2.1027×10^{-2}	1.9056×10^{-2}	2.1658×10^{-2}	2.0601×10^{-2}	1.9279×10^{-2}
	v	2.7296×10^{-2}	2.2187×10^{-2}	2.7912×10^{-2}	2.5865×10^{-2}	2.3295×10^{-2}
	ψ	4.4595×10^{-3}	5.7220×10^{-3}	3.8554×10^{-3}	4.1573×10^{-3}	4.5314×10^{-3}
Periodic	u	5.3167×10^{-1}	3.7814×10^{-2}	4.2063×10^{-2}	4.5505×10^{-2}	4.6104×10^{-2}
	v	7.0910×10^{-1}	3.6862×10^{-2}	4.1328×10^{-2}	4.4600×10^{-2}	4.5129×10^{-2}
	ψ	1.3475×10^{-1}	4.3729×10^{-3}	4.1834×10^{-3}	4.0019×10^{-3}	3.7271×10^{-3}
Rosen–Morse	u	1.4408×10^{-1}	2.5117×10^{-2}	3.9776×10^{-2}	2.7235×10^{-2}	3.1777×10^{-2}
	v	1.0567×10^{-1}	2.3972×10^{-2}	3.3787×10^{-2}	2.5332×10^{-2}	2.7809×10^{-2}
	ψ	3.2571×10^{-2}	7.2073×10^{-3}	5.2887×10^{-3}	5.1823×10^{-3}	5.0052×10^{-3}

TABLE IV. $\mathbb{L}^2$ -norm error values in u , v , and ψ of the PINN results approximated with different number of neurons (60–100) in each hidden layer of ANN for all three considered potentials.

$\mathcal{PT}$ -symmetric potentials	$\mathbb{L}^2$ -norm error	Number of neurons in the hidden layers				
		60	70	80	90	100
Gaussian	u	1.8949×10^{-2}	1.9171×10^{-2}	1.8510×10^{-2}	2.0515×10^{-2}	2.1856×10^{-2}
	v	2.2847×10^{-2}	2.3018×10^{-2}	2.1814×10^{-2}	2.5669×10^{-2}	2.8822×10^{-2}
	ψ	4.3099×10^{-3}	4.6188×10^{-3}	4.9689×10^{-3}	4.0189×10^{-3}	3.1912×10^{-3}
Periodic	u	4.8924×10^{-2}	4.3401×10^{-2}	5.2175×10^{-2}	5.0627×10^{-2}	5.0925×10^{-2}
	v	4.7821×10^{-2}	4.2629×10^{-2}	5.0985×10^{-2}	4.9554×10^{-2}	4.9897×10^{-2}
	ψ	3.8299×10^{-3}	3.8624×10^{-3}	4.0359×10^{-3}	3.9781×10^{-3}	4.2720×10^{-3}
Rosen–Morse	u	3.0926×10^{-2}	3.7381×10^{-2}	4.4708×10^{-2}	4.1246×10^{-2}	3.7277×10^{-2}
	v	2.7566×10^{-2}	3.2622×10^{-2}	3.7069×10^{-2}	3.5090×10^{-2}	3.2468×10^{-2}
	ψ	4.9355×10^{-3}	5.8375×10^{-3}	5.3917×10^{-3}	5.6393×10^{-3}	5.9112×10^{-3}

TABLE V. $\mathbb{L}^2$ -norm error values in u , v , and ψ of the PINN results approximated with different number of collocation points for all three considered potentials.

$\mathcal{PT}$ -symmetric potentials	$\mathbb{L}^2$ -norm error	Number of collocation points			
		5000	10 000	15 000	20 000
Gaussian	u	2.0919×10^{-2}	1.9999×10^{-2}	1.9238×10^{-2}	2.1856×10^{-2}
	v	2.6675×10^{-2}	2.4926×10^{-2}	2.3326×10^{-2}	2.8822×10^{-2}
	ψ	3.6122×10^{-3}	4.0438×10^{-3}	4.1420×10^{-3}	3.1912×10^{-3}
Periodic	u	4.5870×10^{-2}	5.1518×10^{-2}	4.2062×10^{-2}	5.0925×10^{-2}
	v	4.4983×10^{-2}	5.0452×10^{-2}	4.1056×10^{-2}	4.9897×10^{-2}
	ψ	3.8983×10^{-3}	3.8867×10^{-3}	3.8140×10^{-3}	4.2720×10^{-3}
Rosen–Morse	u	3.8685×10^{-2}	3.7610×10^{-2}	4.6277×10^{-2}	3.7277×10^{-2}
	v	3.3381×10^{-2}	3.2416×10^{-2}	3.8593×10^{-2}	3.2468×10^{-2}
	ψ	6.0374×10^{-3}	5.6554×10^{-3}	5.8951×10^{-3}	5.9112×10^{-3}

TABLE VI. $\mathbb{L}^2$ -norm error values in u , v , and ψ of the PINN results approximated with different number of initial and boundary points of the considered domain for all three considered potentials.

$\mathcal{PT}$ -symmetric potentials	$\mathbb{L}^2$ -norm error	Number of initial and boundary points				
		10	20	30	40	50
Gaussian	u	1.7453×10^{-2}	1.7899×10^{-2}	1.6184×10^{-2}	2.1619×10^{-2}	2.1856×10^{-2}
	v	2.0003×10^{-2}	2.0321×10^{-2}	1.7222×10^{-2}	2.8040×10^{-2}	2.8822×10^{-2}
	ψ	5.5589×10^{-3}	4.9169×10^{-3}	6.3697×10^{-3}	3.6132×10^{-3}	3.1912×10^{-3}
Periodic	u	2.5785×10^{-2}	4.5283×10^{-2}	3.3628×10^{-2}	4.9036×10^{-2}	5.0925×10^{-2}
	v	2.4970×10^{-2}	4.4399×10^{-2}	3.2921×10^{-2}	4.8044×10^{-2}	4.9897×10^{-2}
	ψ	6.9655×10^{-3}	4.0064×10^{-3}	4.3703×10^{-3}	3.9586×10^{-3}	4.2720×10^{-3}
Rosen–Morse	u	2.1818×10^{-2}	3.6917×10^{-2}	3.4343×10^{-2}	3.9939×10^{-2}	3.7277×10^{-2}
	v	2.2776×10^{-2}	3.1811×10^{-2}	2.9756×10^{-2}	3.4444×10^{-2}	3.2468×10^{-2}
	ψ	1.0650×10^{-2}	5.2218×10^{-3}	4.9235×10^{-3}	5.0712×10^{-3}	5.9112×10^{-3}

function have low error values for all three considered potentials. Finally, we note that the time taken for the training of the PINN with sech activation function is higher when compared to the other three activation functions.

B. Effect of structure of the network

ANNs have large amount of parameters like weight and bias matrices, which change randomly to minimize the given loss function during the process of optimization. So, the structure of the ANN influences the accuracy of the PINN for the considered task. There are hyper-parameters that describe the structure of the ANN, namely, width of the network (number of hidden layers) and depth of the network (number of units in each layer). We test the impact of these hyper-parameters for all three potentials with tanh as an activation function and present the outcomes in Tables II, III, and IV. Table II corresponds to the $\mathbb{L}^2$ -norm error values of the PINNs with number of hidden layers varying from 1 to 4. In this study, we fix the number of units equal to 100. It is clear from this table that the performance of the PINN with single hidden layer is very low as compared with the other PINNs with more number of hidden layers. Further, when increasing the number of layers, we observe that the performance of PINN for all three potentials getting increased. In other words, the $\mathbb{L}^2$ -norm error values decrease. However, in the case of periodic and Rosen–Morse potentials, the error values of the PINN with four hidden layers are slightly high when compared to the error values of the PINN with three hidden layers. The difference between these error values is considerably low. We need a model that performs well in finding solution of the NLS equation for all three considered potentials. So, in our study, we fixed the number of hidden layers equal to 4.

Next, we examine how the performance of PINN is affected by the number of neurons in the hidden layers. For this, we consider the PINN with tanh activation function and four hidden layers. Now, we vary the number of neurons from 10 to 100 and present the results of the PINN with the number of neurons 10–50 in Table III and for the number of neurons 60–100 in Table IV, respectively. From these two tables, we can see that $\mathbb{L}^2$ -norm error values are very high for the case of PINN with ten neurons. Further, increasing the number of neurons the error values are decreasing and for some cases they are oscillating between low and high values because while increasing the number of neurons automatically increases the size of the weight and bias matrices and the model needs to optimize the more number of parameters. The error value of the solution of the NLS equation with the Gaussian potential gives a low value only when the PINN is trained with 100 neurons. So, we fixed the number of neurons in each hidden layer as 100.

C. Effect of sampling points

We use the sampling points which are sampled from LHS⁴⁴ for the input to the PINN model. These training data points also influence the performance of the PINN in solving the considered problem. The results with different number of collocation points are presented in Table V. For this study, we use tanh activation function, four hidden layers each with 100 neurons and 50 initial points, and 50 points each on upper and lower boundaries. From Table V, it is clear that $\mathbb{L}^2$ -norm error values are changing with respect to the

change in the number of collocation points. When the number of collocation points is 20 000, the error value is completely low for all cases especially for the Gaussian potential case. Our aim is to construct a DL model, which is good enough to make the solution to the NLS equation for all three considered potentials. So, it is better to have a more number of collocation points inside the considered domain so that the model can train with more points which lead to high accurate solution.

Finally, we experiment the PINN by varying the number of initial and boundary sampling points and the $\mathbb{L}^2$ -norm error values in u , v , and ψ for all three potentials presented in Table VI. In this table, we vary the number of points from 10 to 50 in both initial and boundary regions. Here, the number on the boundary denotes the number of points taken for both the upper and lower boundary. For example, the number 10 denotes that there are 10 points on the initial and also 10 points each on both the upper and lower boundary regions so that there are totally 20 points on the boundary of the considered domain. Here also, we use PINN with tanh activation function with four hidden layers each with 100 neurons and 20 000 collocation points inside the domain. From the results that are shown in Table VI, we observe that in most cases, the error values become low when we increase the number of points, and also in some cases, the error values vary between low and high values; particularly in the case of NLS equation with Rosen–Morse potential, the error value is very high when the PINN trained with low number of points on the initial and boundary regions, say, 10. As discussed in the earlier cases, here also, we fix the number of points on the initial and boundary regions as equal to 50 because the PINN with this setup has considerable low values for the $\mathbb{L}^2$ -norm error for all three considered potentials.

It is worth noting that all the above presented results may vary in the repeated learning processes because of the stochastic nature of the sampling technique and algorithm.

V. CONCLUSION

In this work, we have considered the NLS equation with three $\mathcal{PT}$ -symmetric potentials, namely, Gaussian, periodic, and Rosen–Morse and approximated the soliton solution of the NLS equation with the help of a DL approach, the so called PINN. For this purpose, we have considered a complex-valued PINN with tanh as an activation function. The PINN solves the given equation for the prescribed initial and boundary conditions by minimizing the mean squared error loss. We have considered 20 000 collocation points by LHS,⁴⁴ 50 points, and 100 points on initial and boundary data, respectively. The predicted, exact, and squared error in the magnitude of the soliton solution for the considered three different potentials are evaluated and plotted. Further, we have also plotted the exact and predicted magnitudes of the soliton solution one over the other for various instants of time. From the results, we conclude that our constructed PINN can approximate the soliton solution for the given NLS equation for all three potentials precisely. The squared errors are found to be very low of the order of 10^{-6} , 10^{-5} , and 10^{-4} , respectively, for the Gaussian, periodic, and Rosen–Morse potentials.

To visualize the performance of the PINN with tanh as the activation function, we also present the scatterplot of actual

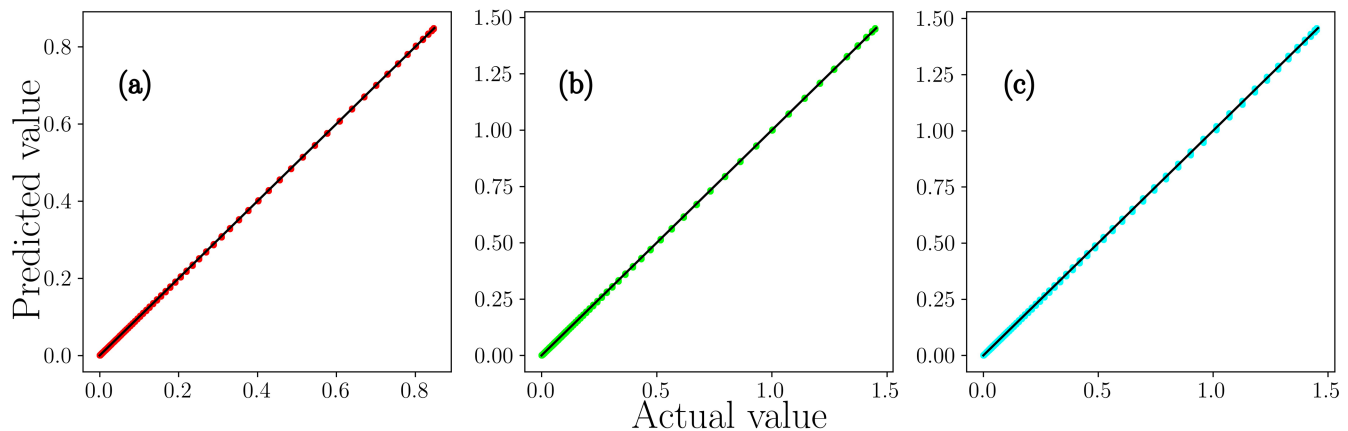


FIG. 10. Scatterplots showing the performance of the PINN. (a)–(c) correspond to the result of NLS equation with Gaussian, periodic, and Rosen–Morse, respectively.

vs the predicted data for all three considered potentials in Fig. 10. The scatterplots of the NLS equation with Gaussian, periodic, and Rosen–Morse potentials are, respectively, shown in Figs. 10(a)–10(c). The scatterplots confirm that the considered PINN accurately predicts the soliton solution in all three cases. Further, to analyze the factors that influence the performance of the PINN, we tested the effect against the activation functions, network structure, and sampling points. First, we have considered three functions, namely, ReLU, sigmoid, and tanh along with a new activation function sech. The PINNs with ReLU and sigmoid as the activation functions approximated the soliton solution with less accuracy when compared to the PINNs with sech and tanh as activation functions. We have also examined the ability of these different PINNs by calculating the $\mathbb{L}^2$ -norm error values for real (u) and imaginary (v) parts of the solution (ψ) for all three considered potentials. From the results, we conclude that the PINN can approximate the soliton solution of the NLS equation for the considered $\mathcal{PT}$ -symmetric potentials with tanh and sech as the activation function. We have also examined the effect on the performance due to the width and the depth of the PINN. From the obtained results, we fixed the number of hidden layers equal to 4 and 100 neurons in each layer. Finally, we have also done an experiment on the number of sampling points and initial and boundary regions. From the outcomes, we found that the amount of training data should be 20 000 collocation points, 50 initial points, and 50 boundary points in order to get a high accurate solution for the considered problem. One can use the considered DL model, namely, PINN for solving the NLS equation with $\mathcal{PT}$ -symmetric potentials.

ACKNOWLEDGMENTS

J.M. acknowledges Ministry of Education-Rashtriya Uchchatar Shiksha Abhiyan (MoE-RUSA) 2.0 Physical Sciences, Government of India for providing a fellowship to carry out this work. K.M. and J.B.S. are funded by the Center for Nonlinear Systems, Chennai Institute of Technology, India, Funding No. CIT/CNS/2021/RP-015. The work of M.S. forms part of a research project sponsored

by National Board for Higher Mathematics (NBHM), Government of India, under the Grant No. 02011/20/2018 NBHM (R.P)/R&D II/15064. M.S. also acknowledges MoE-RUSA 2.0 Physical Sciences, Government of India for providing financial support in procuring a high-performance GPU server, which highly assisted this work.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY

The data that support the findings of this study are available within the article.

REFERENCES

- ¹B. A. Malomed and D. Mihalache, "Nonlinear waves in optical and matter-wave media: A topical survey of recent theoretical and experimental results," *Rom. J. Phys.* **64**, 106 (2019).
- ²C. M. Bender and S. Boettcher, "Real spectra in non-Hermitian Hamiltonians having $\mathcal{PT}$ symmetry," *Phys. Rev. Lett.* **80**, 5243 (1998).
- ³Z. Musslimani, K. G. Makris, R. El-Ganainy, and D. N. Christodoulides, "Optical solitons in $\mathcal{PT}$ periodic potentials," *Phys. Rev. Lett.* **100**, 030402 (2008).
- ⁴Y. Kominis, J. Cuevas-Maraver, P. G. Kevrekidis, D. J. Frantzeskakis, and A. Bountis, "Continuous families of solitary waves in non-symmetric complex potentials: A Melnikov theory approach," *Chaos Soliton. Fract.* **118**, 222–233 (2019).
- ⁵Z. Yan, Z. Wen, and C. Hang, "Spatial solitons and stability in self-focusing and defocusing Kerr nonlinear media with generalized parity-time-symmetric Scarff-II potentials," *Phys. Rev. E* **92**, 022913 (2015).
- ⁶A. Guo, G. Salamo, D. Duchesne, R. Morandotti, M. Volatier-Ravat, V. Aimez, G. Siviloglou, and D. Christodoulides, "Observation of $\mathcal{PT}$ -symmetry breaking in complex optical potentials," *Phys. Rev. Lett.* **103**, 093902 (2009).
- ⁷C. E. Rüter, K. G. Makris, R. El-Ganainy, D. N. Christodoulides, M. Segev, and D. Kip, "Observation of parity-time symmetry in optics," *Nat. Phys.* **6**, 192–195 (2010).
- ⁸A. Regensburger, M. Miri, C. Bersch, J. Näger, G. Onishchukov, D. N. Christodoulides, and U. Peschel, "Observation of defect states in $\mathcal{PT}$ -symmetric optical lattices," *Phys. Rev. Lett.* **110**, 223902 (2013).

- ⁹K. G. Makris, R. El-Ganainy, D. Christodoulides, and Z. H. Musslimani, "Beam dynamics in $\mathcal{PT}$ -symmetric optical lattices," *Phys. Rev. Lett.* **100**, 103904 (2008).
- ¹⁰Y. Chen, Z. Yan, D. Mihalache, and B. A. Malomed, "Families of stable solitons and excitations in the $\mathcal{PT}$ -symmetric nonlinear Schrödinger equations with position-dependent effective masses," *Sci. Rep.* **7**, 1–21 (2017).
- ¹¹Z. Wen and Z. Yan, "Solitons and their stability in the nonlocal nonlinear Schrödinger equation with $\mathcal{PT}$ -symmetric potentials," *Chaos* **27**, 053105 (2017).
- ¹²K. Hari, K. Manikandan, and R. Sankaranarayanan, "Dissipative optical solitons in asymmetric Rosen-Morse potential," *Phys. Lett. A* **384**, 126104 (2020).
- ¹³K. Manikandan, N. Vishnu Priya, M. Senthilvelan, and R. Sankaranarayanan, "Deformation of dark solitons in a $\mathcal{PT}$ -invariant variable coefficients nonlocal nonlinear Schrödinger equation," *Chaos* **28**, 083103 (2018).
- ¹⁴K. Manikandan, J. Sudharsan, and M. Senthilvelan, "Nonlinear tunneling of solitons in a variable coefficients nonlinear Schrödinger equation with $\mathcal{PT}$ -symmetric Rosen-Morse potential," *Eur. Phys. J. B* **94**, 1–10 (2021).
- ¹⁵G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," *Rev. Mod. Phys.* **91**, 045002 (2019).
- ¹⁶A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, "Physics-enhanced neural networks learn order and chaos," *Phys. Rev. E* **101**, 062207 (2020).
- ¹⁷S. T. Miller, J. F. Lindner, A. Choudhary, S. Sinha, and W. L. Ditto, "The scaling of physics-informed machine learning with data and dimensions," *Chaos Solitons Fractals* **5**, 100046 (2020).
- ¹⁸S. Mukhopadhyay and S. Banerjee, "Learning dynamical systems in noise using convolutional neural networks," *Chaos* **30**, 103125 (2020).
- ¹⁹J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," *Chaos* **27**, 121102 (2017).
- ²⁰P. Amil, M. C. Soriano, and C. Masoller, "Machine learning algorithms for predicting the amplitude of chaotic laser pulses," *Chaos* **29**, 113111 (2019).
- ²¹Q. Zhu, H. Ma, and W. Lin, "Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing," *Chaos* **29**, 093125 (2019).
- ²²S. Krishnagopal, M. Girvan, E. Ott, and B. R. Hunt, "Separation of chaotic signals by reservoir computing," *Chaos* **30**, 023123 (2020).
- ²³A. Panday, W. S. Lee, S. Dutta, and S. Jalan, "Machine learning assisted network classification from symbolic time-series," *Chaos* **31**, 031106 (2021).
- ²⁴G. Barmbaris, G. Neofotistos, M. Mattheakis, J. Hizanidis, G. Tsironis, and E. Kaxiras, "Robust prediction of complex spatiotemporal states through machine learning with sparse sensing," *Phys. Lett. A* **384**, 126300 (2020).
- ²⁵M. A. Ganaie, S. Ghosh, N. Mendola, M. Tanveer, and S. Jalan, "Identification of chimera using machine learning," *Chaos* **30**, 063128 (2020).
- ²⁶N. Kushwaha, N. K. Mendola, S. Ghosh, A. D. Kachhvah, and S. Jalan, "Machine learning assisted chimera and solitary states in networks," *Front. Phys.* **9**, 5 (2021).
- ²⁷M. Lellep, J. Prexl, M. Linkmann, and B. Eckhardt, "Using machine learning to predict extreme events in the Hénon map," *Chaos* **30**, 013113 (2020).
- ²⁸J. Meiyazhagan, S. Sudharsan, and M. Senthilvelan, "Model-free prediction of emergence of extreme events in a parametrically driven nonlinear dynamical system by deep learning," *Eur. Phys. J. B* **94**, 1–13 (2021).
- ²⁹S. N. Chowdhury, A. Ray, A. Mishra, and D. Ghosh, "Extreme events in globally coupled chaotic maps," *J. Phys.: Complexity* **2**, 035021 (2021).
- ³⁰A. Ray, T. Chakraborty, and D. Ghosh, "Optimized ensemble deep learning framework for scalable forecasting of dynamics containing extreme events," *arXiv:2106.08968* (2021).
- ³¹J. Meiyazhagan, S. Sudharsan, A. Venkatesan, and M. Senthilvelan, "Prediction of occurrence of extreme events using machine learning," *Eur. Phys. J. Plus* **137**, 1–20 (2022).
- ³²M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.* **378**, 686–707 (2019).
- ³³J. Pu, W. Peng, and Y. Chen, "The data-driven localized wave solutions of the derivative nonlinear Schrödinger equation by using improved PINN approach," *Wave Motion* **107**, 102823 (2021).
- ³⁴L. Wang and Z. Yan, "Data-driven rogue waves and parameter discovery in the defocusing nonlinear Schrödinger equation with a potential using the PINN deep learning," *Phys. Lett. A* **404**, 127408 (2021).
- ³⁵Z. Zhou and Z. Yan, "Deep learning neural networks for the third-order nonlinear Schrödinger equation: Solitons, breathers, and rogue waves," *arXiv:2104.14809* (2021).
- ³⁶L. Wang and Z. Yan, "Data-driven peakon and periodic peakon solutions and parameter discovery of some nonlinear dispersive equations via deep learning," *Physica D* **428**, 133037 (2021).
- ³⁷Y. Mo, L. Ling, and D. Zeng, "Data-driven vector soliton solutions of coupled nonlinear Schrödinger equation using a deep learning algorithm," *Phys. Lett. A* **421**, 127739 (2022).
- ³⁸Z. Zhou and Z. Yan, "Solving forward and inverse problems of the logarithmic nonlinear Schrödinger equation with $\mathcal{PT}$ -symmetric harmonic potential via deep learning," *Phys. Lett. A* **387**, 127010 (2021).
- ³⁹J. Li and B. Li, "Solving forward and inverse problems of the nonlinear Schrödinger equation with the generalized $\mathcal{PT}$ -symmetric Scarf-II potential via PINN deep learning," *Commun. Theor. Phys.* **73**, 125001 (2021).
- ⁴⁰A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.* **18**, 1–18 (2018).
- ⁴¹C. C. Margossian, "A review of automatic differentiation and its efficient implementation," *Wiley Interdiscipl. Rev.: Data Min. Knowl. Discovery* **9**, e1305 (2019).
- ⁴²D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature* **323**, 533–536 (1986).
- ⁴³M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (USENIX Association, 2016), pp. 265–283.
- ⁴⁴M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics* **29**, 143–151 (1987).
- ⁴⁵D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.* **45**, 503–528 (1989).
- ⁴⁶J. Yang, *Nonlinear Waves in Integrable and Nonintegrable Systems* (SIAM, 2010).
- ⁴⁷S. Hu, X. Ma, D. Lu, Z. Yang, Y. Zheng, and W. Hu, "Solitons supported by complex $\mathcal{PT}$ -symmetric Gaussian potentials," *Phys. Rev. A* **84**, 043818 (2011).
- ⁴⁸B. Midya and R. Roychoudhury, "Nonlinear localized modes in $\mathcal{PT}$ -symmetric Rosen-Morse potential wells," *Phys. Rev. A* **87**, 045803 (2013).