


Complex-valued neural-operator-assisted soliton identificationMing Zhang,¹ Qi Meng,^{2,*} Deng Zhang,³ Yue Wang,² Guanghui Wang,¹ Zhiming Ma,⁴ Li Chen ^{5,†} and Tie-Yan Liu^{2,‡}¹*School of Mathematics, Shandong University, Jinan 250100, China*²*Microsoft Research, Beijing 100080, China*³*School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China*⁴*Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100190, China*⁵*Institute of Theoretical Physics, State Key Laboratory of Quantum Optics and Quantum Optics Devices, Shanxi University, Taiyuan 030006, China*

(Received 7 September 2022; revised 4 May 2023; accepted 31 July 2023; published 14 August 2023)

The numerical determination of solitary states is an important topic for such research areas as Bose-Einstein condensates, nonlinear optics, plasma physics, and so on. In this paper, we propose a data-driven approach for identifying solitons based on dynamical solutions of real-time differential equations. Our approach combines a machine-learning architecture called the complex-valued neural operator (CNO) with an energy-restricted gradient optimization. The CNO serves as a generalization of the traditional neural operator to the complex domain, and constructs a smooth mapping between the initial and final states; the energy-restricted optimization facilitates the search for solitons by constraining the energy space. We concretely demonstrate this approach on the quasi-one-dimensional Bose-Einstein condensate with homogeneous and inhomogeneous nonlinearities. Our work offers an idea for data-driven effective modeling and studies of solitary waves in nonlinear physical systems.

DOI: [10.1103/PhysRevE.108.025305](https://doi.org/10.1103/PhysRevE.108.025305)**I. INTRODUCTION**

Steady soliton solutions of nonlinear partial differential equations (PDEs) arise in a wide range of contexts in physics, including Bose-Einstein condensates (BECs) [1–3], nonlinear optics [3–6], and plasma physics [7,8]. Since analytical solutions to nonlinear PDEs are generally difficult to obtain (especially for nonintegrable PDEs), the numerical identification of solitons constitutes an important subject for both theoretical inquiries and practice. From a theoretical perspective, the ground-state solitary solution can unveil the equilibrium characteristics of nonlinear systems, such as long-range order and topological structures; the lifetime of soliton solutions can offer valuable insights into the stability and response to perturbations near equilibration. From the practical point of view, the study of solitons has potential implications in long-distance communication [9–11], data transmission, and in laying the foundation for advanced photonic devices and data storage technologies [12–14].

The solitary waves that we are interested in are separable in time and space. Several traditional methods are available for finding this type of soliton [15–17]. For example, the complex evolution method extends the real-time of PDEs to complex or imaginary values [18,19]. Since complex-time evolution always reduces energy, this method is more suitable for finding ground-state solitons. Moreover, several meth-

ods were developed based on stationary equations for both ground-state and excited-state solitons, where the time dependence was eliminated using the space-time separation. For instance, the shooting method is primarily used for one-dimensional (1D) systems; the Petviashvili method [20] and its extensions [21–23] are suitable for finding higher-dimensional ground-state solitons; Newton’s method [16,17] and its advanced versions, such as the CG Newton’s method [24], iteratively search for ground-state or excited-state solitons starting from trial solutions and can be extended to higher-dimensional systems. Some machine learning methods were also applied in finding the solitons, e.g., the variational neural network ansatz [25], the deep residual [26], convolutional neural network [27,28], as well as the generative models [29]. Furthermore, there are also some dynamical PDE solvers based on machine learning [30,31]. Physics-informed neural networks were proposed as a powerful tool to approximate the dynamical solutions by incorporating the governing equations as soft constraints during the training process [32,33]. The Feynman-Kac formula-based methods [34,35] and stochastic equation-based methods [31,36] were reported as well. This class of solvers, however, is designed to find the dynamical solutions of PDEs for the given initial states, which cannot be used for determining solitary solutions directly.

In this paper, we propose a data-driven approach to search for the solitons based on a machine-learning architecture called the complex-valued neural operator (CNO). We were motivated by the question of if it is possible to identify solitary wave solutions by directly looking at the real-time PDE, rather than its variants such as the imaginary-time PDE or

*meq@microsoft.com

†lchen@sxu.edu.cn

‡tyliu@microsoft.com

the stationary equation. Since solitary waves exhibit space-time separation, in principle, we can use traditional PDE solvers [e.g., Euler or Runge-Kutta (RK) solvers] to identify solutions where the initial and final states differ by only a phase factor. There are two challenges for such a task. First, iterating through the initial states by PDE solvers is highly time-consuming. Second, finding a specific solution is an optimization problem, and hence it would be ideal to use the gradient descent. Traditional PDE solvers, however, cannot compute the derivative with respect to initial states. These two challenges can be well addressed by the neural operator (NO) [37,38], which can establish the continuous mapping between the real-value functions during training. We extend the NO to the complex domain, namely, the CNO with complex layers and complex activation functions, to accommodate the complex dynamical PDEs. We further develop an energy-restricted optimization algorithm to reduce the spaces of states during the search process. Our approach is concretely demonstrated on the one-dimensional Gross-Pitaevskii (GP) equations with both homogeneous and inhomogeneous nonlinearities. Furthermore, we show that the trained CNO can be also used for the stability analysis of solitary states.

The rest of this paper is organized as follows. In Sec. II, we present the basic idea of our approach and show the CNO architecture. In Sec. III, we use our method in both the homogeneous GP and the inhomogeneous GP equations to identify the solitons. In Sec. IV, we present the application of the CNO to the stability analysis of solitons. A brief summary and outlook can be found in Sec. V.

II. GENERAL METHOD

Let us generically consider a class of nonlinear PDEs of a one-dimensional physical system

$$f(\psi, \dot{\psi}, t) = 0, \quad (1)$$

with $\psi(x, t)$ being a complex-valued function. From the perspective of field theory, the PDEs come from the Euler-Lagrange equation $\partial_\psi L = d_t(\partial_{\dot{\psi}} L)$ with $L(\psi, \dot{\psi}, t)$ being the Lagrangian. For example, the GP equation is generated by the nonlinear Schrödinger Lagrangian, and the nonlinear Klein-Gordon equation arises from the Klein-Gordon Lagrangian with mass or high-order potentials. The solitons that we are interested in are space-time separable, i.e.,

$$\psi(x, t) = e^{-iat} \phi(x), \quad (2)$$

where $i = \sqrt{-1}$ is the imaginary unit and e^{-iat} is a time-dependent phase factor that is isolated from the spatial solitary state $\phi(x)$. Using the time-space separation of $\psi(x, t)$, the temporal degree-of-freedom of Eq. (1) can be eliminated, which leads to the stationary equation purely in terms of $\phi(x)$, solving which one can obtain the time-independent solitons. Due to the nonlinearity of f , the stationary equation is generally not a linear eigenstate equation, and various iterative methods were developed for such stationary equations [15,16].

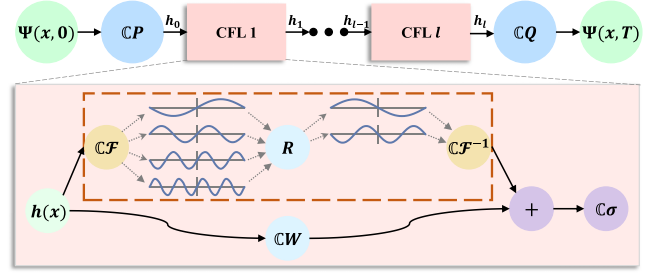


FIG. 1. Upper panel: Architecture of a CNO where initial $\psi(x, 0)$ and final $\psi(x, T)$ states serve as the input and output, $\mathbb{C}P$ and $\mathbb{C}Q$ correspond to the complex embedding and decoding layers, and $h(x)$ are the hidden states. Lower panel: Detailed structure of a CFL with $\mathbb{C}F$ and $\mathbb{C}F^{-1}$, respectively, the complex Fourier and inverse Fourier transformations, R the low-pass filter layer, $\mathbb{C}W$ a local complex transformation, and $\mathbb{C}\sigma$ the complex-valued nonlinear activation.

A. Basic idea

We aim to identify solitary solutions by directly looking at the real-time dynamical PDEs. In such a context of real-time dynamics, the PDE [Eq. (1)] maps the initial states to the final states, i.e., $G: \psi(x, 0) \mapsto \psi(x, T)$, and the solitary solutions correspond to the fixed points of the mapping with unit fidelity $F = 1$, where

$$F = \frac{1}{N^2} \left| \int dx \psi^*(x, T) \psi(x, 0) \right|^2, \quad (3)$$

and $N = \int dx |\phi(x)|^2$ is the total particle number and $\psi^*(x, T)$ is the complex conjugation of $\psi(x, T)$. Considering the fact that the space of states is continuous and quite large, it is not feasible to locate the solitons by directly iterating through the initial states $\psi(x, 0)$ using PDE solvers. To address this issue, we developed the CNO and a restricted searching algorithm. Our algorithm mainly consists of two steps: (i) Learn the initial-final state mapping G by training the CNO based on a finite-size dataset \mathcal{D} ; (ii) Perform restricted gradient optimization to find the solitons within a specific energy range. Below we explain in detail.

As is schematically shown in Fig. 1, CNO is a trainable machine-learning architecture that is able to learn the relation between the initial state $\psi(x, 0)$ (as the input) and the final state $\psi(x, T)$ (as the output) based on a finite-size dataset \mathcal{D} . More detailed descriptions of CNO can be found in Sec. II B. The dataset \mathcal{D} is composed of the pairs of initial-final states, i.e.,

$$\mathcal{D} = \{[\psi_k(x, 0), \psi_k(x, T)] | k = 1, 2, \dots, M\}, \quad (4)$$

with $\psi_k(x, 0)$ being random differentiable functions and M being the total number of data points. Practically, $\psi_k(x, T)$ can be generated by the numerical propagating Eq. (1) by certain PDE solvers, e.g., the Euler or the RK solvers. CNO learns the mapping G by minimizing the loss function \mathcal{L} , which refers to the training process. The loss function \mathcal{L} is defined to characterize the effective distance between the network prediction $\psi_{\text{pred}}(x, T)$ and the solutions $\psi(x, T)$ in the dataset \mathcal{D} . Practically, in the following discussion we use the L_2 -norm

loss function

$$\mathcal{L} = \frac{1}{M} \sum_k \int dx \frac{|\psi_{k,\text{pred}}(x, T) - \psi_k(x, T)|}{|\psi_k(x, T)|}. \quad (5)$$

Once the mapping G is learned by the CNO, the trained CNO can be used to find the solitons by minimizing $\Delta F = |F - 1|$ with respect to the initial state $\psi(x, 0)$. This is an optimization process, which can be carried out using the gradient descent method. Here, we emphasize that the CNO-based soliton search has two major advantages. First, the forward propagation of CNO is much faster than traditional PDE solvers. The detailed discussion on the computational complexity will be shown in Sec. II B. In Sec. IV, we additionally show a concrete running-time comparison between CNO and traditional PDE solvers in the context of the stability analysis of solitons. Furthermore, the output of the CNO is continuously derivable to its input, which thus naturally facilitates the gradient-based optimization of ΔF .

Although the trained CNO can provide several computational conveniences as mentioned above, the vast space of the initial states remains a significant obstacle in our practical pursuit of solitons. This motivated us to reduce the search space, which is based on Noether's theorem. Noether's theorem establishes a relationship between symmetries and conserved charges, with the charges being of help in space reduction. For example, in systems with time translation symmetry, such as the GP equation or nonlinear Klein-Gordon equation, the energy (or the Hamiltonian) $E = \int dx \Pi \dot{\psi} - L$ is conserved, with $\Pi = \partial_{\dot{\psi}} L$ being the conjugate field. Therefore, we can perform the energy-restricted optimization by adding an inequality constraint into the minimization of ΔF . To be more specific, if we set an energy upper bound $E \leq E_{\text{max}}$, the optimization problem is reformed as

$$\min_{\psi(x,0)} \Delta F \quad \text{subject to} \quad E_{\psi(x,0)} \leq E_{\text{max}}. \quad (6)$$

Using the augmented Lagrangian method [39], the above optimization problem can be rewritten as

$$\min_{\psi(x,0), \lambda, \gamma} \Delta F + \frac{\gamma}{2} \left(\max \left\{ \frac{\lambda}{\gamma} + E_{\psi(x,0)} - E_{\text{max}}, 0 \right\}^2 - \frac{\lambda^2}{\gamma^2} \right), \quad (7)$$

where $E_{\psi(x,0)}$ is the energy of the input $\psi(x, 0)$, λ is the Lagrange multiplier, and γ is the penalty factor.

B. Complex-valued neural operator

Here, we discuss in detail the CNO. Neural operators (NO) are machine-learning models proposed to learn mappings between functions [37,38]. Since the mapping G here is generally complex, we have to extend the NO from the real to the complex domain. Taking the Fourier neural operator as a backbone [37], we propose the complex-valued NO, i.e., the CNO. There have been some recent studies on the complex-valued neural networks [40–44]. One known challenge lies in the fact that complex-valued activation functions are not simultaneously complex-differentiable and bounded [40,41], which leads to complex neural networks still being an open and active research topic.

Our CNO, as shown in Fig. 1, is formulated as a multilayer architecture with the initial state $\psi(x, 0)$ being the input and the final state $\psi(x, T)$ the output, i.e.,

$$\psi(x, T) = \mathbb{C}Q\{\text{CFL}^l[\dots(\text{CFL}^1\{\mathbb{C}P[\psi(x, 0)]\})]\}. \quad (8)$$

Here, $\mathbb{C}P[\psi(x, 0)]$ and $\mathbb{C}Q(h_l)$ layers are the complex embedding and complex decoding layers, where the first can lift the input to a higher-dimensional hidden space to ensure the expressiveness of the model, while the second works oppositely to make the output have the same dimension as the input.

Feature learning takes place in the hidden space, which consists of several complex-valued Fourier layers (CFLs). In CFL $j + 1$ ($j = 0, 1, \dots, l - 1$), the hidden state h_j are projected to h_{j+1} as

$$h_{j+1}(x) = \mathbb{C}\sigma(\mathbb{C}W h_j(x) + \mathbb{C}\mathcal{F}^{-1}\{R \cdot \mathbb{C}\mathcal{F}[h_j(x)]\}), \quad (9)$$

where σ denotes the complex-valued element-wise nonlinear activation, $\mathbb{C}W$ is a complex-valued convolutional network which implements the linear transformation on h_j , $\mathbb{C}\mathcal{F}$ and $\mathbb{C}\mathcal{F}^{-1}$ are the complex Fourier transformation and inverse Fourier transformation, and R denotes the low-pass filter defined on the frequency space. The main points in the construction of CFLs lie in the complex-valued convolution $\mathbb{C}W h_j(x)$ and the complex-valued activation $\mathbb{C}\sigma(\cdot)$. Below, we will discuss them one by one.

(1) Complex convolution. To perform the equivalent operation of traditional real-valued two-dimensional (2D) convolution in the complex domain, we convolve the hidden complex vector $h = a + ib$ with the complex filter matrix $\mathbb{C}W = A + iB$, where A and B are real matrices and a and b are real vectors. Since the convolution operator is distributive, convolving the vector h by the filter $\mathbb{C}W$ can be simply expressed by

$$\mathbb{C}W * h = (A * a - B * b) + i(B * a + A * b), \quad (10)$$

with $*$ denoting the convolutional operation.

(2) Complex-value activation. We generalize the real-valued Gaussian error linear unit (GELU) [45] to its complex counterpart, namely, $\mathbb{C}\text{GELU}$. The $\mathbb{C}\text{GELU}$ is defined as

$$\mathbb{C}\text{GELU}(z) = \text{GELU}[\mathcal{R}(z)] + i\text{GELU}[\mathcal{I}(z)], \quad (11)$$

where both the real $\mathcal{R}(z)$ and imaginary $\mathcal{I}(z)$ parts of a neuron are activated by GELU, with

$$\text{GELU}(x) = x \cdot \frac{1}{2} [1 + \text{erf}(x/\sqrt{2})], \quad (12)$$

$\text{erf}(\cdot)$ the Gauss error function and x being a real number. It is known that GELU is nonconvex and nonmonotonic, and has been practically applied in many large language models (e.g., OpenAI's GPT [46] and Google AI's BERT models [47]) where it outperforms the convex and monotonic ReLU. Note that we may have some other extensions, such as activating the norm of z while keeping its phase factor unchanged, i.e., $\mathbb{C}\text{GELU}(x) = \text{GELU}(|z| + z_0) \exp(i \arg z)$ with z_0 being a real learnable parameter. The performance of various extensions remains to be further studied, but this is not the focus of this article. Therefore, in the following, we consistently adopt the $\mathbb{C}\text{GELU}$ defined in Eq. (11).

Now, we discuss the complexity of CNO. In a CFL, the time complexity majorly comes from the Fourier and the inverse Fourier transforms which provides a complexity $O(n \log n)$ with n being the dimension of the input. Hence, the total complexity of a CNO with l CFLs is $O(ln \log n)$. On the other hand, traditional explicit PDE solvers based on the finite difference and the pseudospectrum are well known to exhibit complexity $O(t_s n^2)$ and $O(t_s n \log n)$, respectively, with t_s being the time steps. Hence, the forward propagation of CNO should be significantly faster than traditional explicit solvers as $l \ll t_s$. This property not only facilitates our soliton identification algorithm mentioned above, but also speeds up the stability analysis that will be shown in Sec. IV. We will numerically compare the propagation efficiency of the CNO and traditional PDE solvers in Sec. IV.

III. SOLITON IDENTIFICATION

We demonstrate our approach on the one-dimensional GP equation

$$i\partial_t \psi = \left[-\frac{\partial_x^2}{2m} + V(x) + g(x)|\psi|^2 \right] \psi, \quad (13)$$

which is generated by the Lagrangian (setting $\hbar = 1$)

$$L = \int dx \left[i\psi^* \dot{\psi} - \frac{|\partial_x \psi|^2}{2m} - V(x)|\psi|^2 - \frac{g(x)}{2} |\psi|^4 \right]. \quad (14)$$

It describes a quasi-one-dimensional BEC with N atoms being tightly confined in the transverse y - z directions, with $\psi(x, t)$ being the longitudinal field satisfying $\int dx |\psi(x, t)|^2 = N$. The first, second, and third terms on the right-hand side (R.H.S.) of Eq. (13), respectively, denote the kinetic energy, the longitudinal potential along x , and the nonlinear interaction arising from the two-body s -wave collision of atoms with $g = 2a_s/m\ell_\perp^2$ the reduced nonlinearity strength, a_s the scattering length, ℓ_\perp the transverse confinement length, and m the atomic mass. g is commonly set positive to stabilize the BEC [48,49]. Particularly, an equation with a space-independent interaction $g(x) = g_0$ is known as the homogeneous GP equation, whereas if $g(x)$ is space dependent, the equation is called the inhomogeneous GP equation [50,51]. Practically, spatial inhomogeneity $g(x)$ can be achieved using the confinement-induced resonance technique [52,53], i.e., properly engineering the transverse confinement near the orbital resonance point to induce spatial inhomogeneity in the scattering length. The energy of the BEC is simply the Hamiltonian of the Schrödinger field, i.e.,

$$E = \int dx \left[\frac{|\partial_x \psi|^2}{2m} + V(x)|\psi|^2 + \frac{g(x)}{2} |\psi|^4 \right], \quad (15)$$

which is conserved during the time evolution, as mentioned before.

It is known that almost all solitary solutions of the 1D GP equation can be anticipated using Newton's method or other eigensolvers, based on trial solutions constructed from the eigenstates of the linear Schrödinger equation (with the nonlinearity turned off) or their variations. These solitons provide the ground truth to verify the effectiveness of our algorithm, which is the main objective of this paper. On the other hand,

the trade-off is that there are rarely unexpected solutions in the 1D system and hence generalizing our algorithm to higher-dimensional systems would be more practically valuable. This generalization is not so straightforward as will be discussed in Sec. V.

A. Homogeneous GP equation

For the first example, we consider a homogeneous BEC confined in a harmonic potential $V(x) = \omega^2 x^2/2$ with fixed $g(x) = g_0 = 20\omega/N\ell$ and $\ell = (m\omega)^{-1/2}$ denoting the harmonic length. We generate a training dataset $\mathcal{D}_{\text{train}}$ of size $M_{\text{train}} = 10^4$ and a testing dataset $\mathcal{D}_{\text{test}}$ of size $M_{\text{test}} = 10^3$ for such a learning task, where the data points are collected by independently propagating the GP equation Eq. (13) using the RK finite difference method [54] from the initial states $\psi(x, 0)$ to the final state $\psi(x, T)$ with $T = \omega^{-1}$ being fixed. The initial states $\psi(x, 0)$ are randomly generated using the harmonic basis, i.e., $\psi(x, 0) = \sum_{n=1}^{n_c} c_n \xi_n$ where c_n denotes the expansion coefficients satisfying $\sum_n |c_n|^2 = 1$, $n_c = 20$ is the basis cutoff, and

$$\xi_n(x) = \frac{1}{2^n n!} \left(\frac{1}{\pi R^2} \right)^{1/4} H_n \left(\frac{x}{R} \right) e^{-\frac{x^2}{2R^2}}, \quad (16)$$

with H_n being the Hermite polynomial and $R = (3gN\ell^2/2\omega)^{1/3}$ being the Thomas-Fermi radius. Employing the harmonic basis, rather than directly randomizing the $\psi(x, 0)$ in the coordinate space is based on the physical consideration that the complex wave $\psi(x, t)$ and its first-order derivative $\partial_x \psi(x, t)$ should be continuous for a potential $V(x)$ without singularity. However, the soliton identification is still conducted in the coordinate space.

Through feeding $\mathcal{D}_{\text{train}}$ to the CNO, we carry out the training by minimizing the loss function \mathcal{L} [Eq. (5)] using the Adam optimizer [55]. Our CNO contains $l = 4$ CFLs, and the embedding dimension is 64. Figure 2(a) presents \mathcal{L} as a function of training epochs, where the solid and dashed curves indicate \mathcal{L} on the training $\mathcal{D}_{\text{train}}$ and the testing $\mathcal{D}_{\text{test}}$ datasets, respectively. The inset takes a closed look at \mathcal{L} within the range [0.04, 0.1]. It is shown that the training process converges at about 450 epochs, as indicated by the loss function \mathcal{L} reaching a broad plateau. The training error and testing error exhibit similar behavior with small quantitative differences, which indicates the trained CNO does not suffer from a severe overfitting problem.

After convergence, we perform the energy-restricted optimization [Eq. (7)] to identify the soliton states based on the trained CNO. Practically, we set the energy bound to $E_{\text{max}} = 9\omega$, and optimize Eq. (7) using the Adam optimizer from 10^3 stochastic initial states. As a result, the lowest seven solitary states can be identified. In Figs. 2(b1) to 2(b3), we typically display the amplitude of the ground-state $|\phi_0|$, the third excited-state $|\phi_3|$, and the fifth excited-state $|\phi_5|$ by solid lines, respectively. As a comparison, we also show the amplitude of solitons $|\phi_n|$ obtained by Newton's method by dashed lines, which serve as the ground truth. Both results are in good qualitative agreement. Quantitatively, the solitons found by the CNO shown in Fig. 2(b) do not perfectly match the ground truth. The mismatch can be attributed to the intrinsic error of

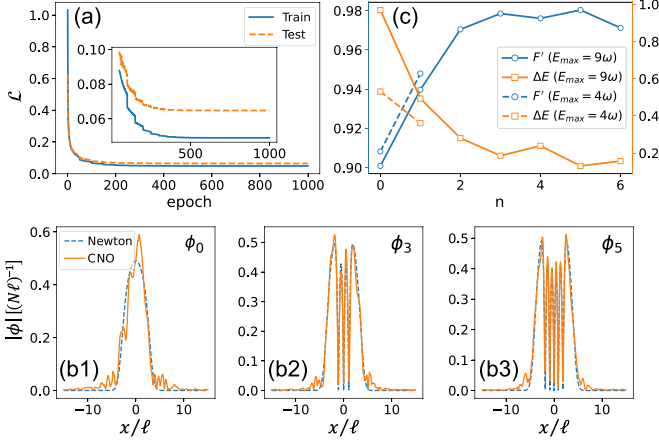


FIG. 2. Numerical results of the homogeneous GP equation. (a) Variation of the loss function \mathcal{L} on epochs during the training process, where the solid and dashed curves indicate \mathcal{L} on the training $\mathcal{D}_{\text{train}}$ and testing $\mathcal{D}_{\text{test}}$ datasets, respectively. (b) Comparison between the solitary states identified by the CNO (solid lines) and those obtained by Newton's method (dashed lines). Panels (b1), (b2), and (b3) display the amplitude of the ground-state $|\phi_0\rangle$, the third excited-state $|\phi_3\rangle$, and the fifth excited-state $|\phi_5\rangle$, respectively. (c) The overlap F' (circle) and the energy error ΔE (square) as a function of the soliton index n . Solid lines correspond to the case of $E_{\text{max}} = 9\omega$, whereas dashed lines correspond to the case of $E_{\text{max}} = 4\omega$.

CNO in learning the mapping G . In fact, training errors are inevitable for any data-driven machine-learning model.

To quantitatively measure the discrepancy between the solitons found by the CNO and Newton's method, we calculate the overlap (solid line with circles)

$$F' = \left| \int dx \phi_{n,\text{CNO}}^*(x) \phi_{n,\text{Newton}}(x) \right|^2, \quad (17)$$

and energy discrepancy (solid line with squares)

$$\Delta E = |E_n^{\text{CNO}} - E_n^{\text{Newton}}|, \quad (18)$$

as are plotted in Fig. 2(c). One can observe that the identified solitons averagely have $F' \geq 96\%$ and $\Delta E \leq 0.35$. In Fig. 2(c), ΔE roughly decreases monotonically as n increases, which means the solitons $\phi_{n,\text{CNO}}$ closer to the energy bound E_{max} exhibit a lower energy error ΔE . This phenomenon can be understood that F [Eq. (3)] is generally nonconvex near a fixed point, i.e., there are several local minima $\Delta F \gtrsim 0$ near the point $\Delta F = 0$. The energy constraint [the last term of Eq. (7)] becomes more and more important as $E_{\psi}(x, 0)$ approaches the bound E_{max} , thereby decreasing the likelihood that the algorithm will achieve a local minimum. Otherwise, if $E_{\psi}(x, 0)$ is too far away from E_{max} , the last term of Eq. (7) is simply a constant referring to no restriction. In this case, the searching algorithm may converge to a local-minimum solution giving rise to a large ΔE . A mathematical proof of this statement can be found in the Appendix. For numerical verification, we set a small bound $E_{\text{max}} = 4\omega$ such that only the lowest two solitons (ϕ_0 and ϕ_1) are allowed to be identified. The corresponding F' and ΔE are plotted by dashed lines in Fig. 2(c), from which one can clearly observe that a lower bound E_{max} is really helpful to reduce the error ΔE .

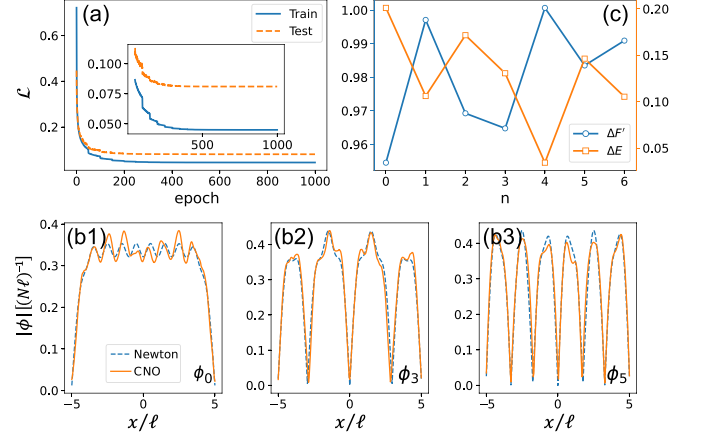


FIG. 3. Numerical results of the inhomogeneous GP equation. (a) Variation of the loss function \mathcal{L} on epochs during the training process. (b) Comparison between the solitary states identified by the CNO (solid lines) and those obtained by Newton's method (dashed lines). (c) The overlap F' (circle) and the energy error ΔE (square) as a function of the soliton index n .

B. Inhomogeneous GP equation

We illustrate the second example by considering a BEC carrying inhomogeneous nonlinearity $g(x) = g_0 + \delta g \sin(2\pi x)$ and being confined in a boxed potential within $x \in [-R, R]$, where the homogeneous g_0 is taken with the same value as in the first example, and $\delta g = g_0/2$ characterizes the inhomogeneous nonlinearity with $R = 5\ell$ being fixed. The data generation is similar to the first example, except that now we adopt a basis of trigonometric polynomial, i.e.,

$$\xi_n(x) = \frac{1}{R} \sin \left[\frac{n\pi}{2R} (x + R) \right], \quad (19)$$

with $n_c = 15$. The trigonometric basis ensures that the complex field $\psi(x)$ vanishes at both boundaries of the box $x = \pm R$, which is reasonable for a boxed potential with hard walls. We train the CNO based on the training dataset $\mathcal{D}_{\text{train}}$. The hyperparameters of the CNO are the same as those of the first example.

In Fig. 3(a), we show the variations of \mathcal{L} as the training processes, which indicates that the training converges at about 400 epochs. We set the upper energy bound to $E_{\text{max}} = 4$ for the soliton search, and as a consequence, the lowest seven solitons are obtained and displayed in Figs. 3(b1) to 3(b3). Again, the soliton states obtained by Newton's method are also shown in Fig. 3(b) as a reference. It can be observed that, although the potential energy is flat inside the box, the amplitude of the field exhibits strong modulations due to the inhomogeneous nonlinearity. Accordingly, in Fig. 3(c), we show the overlap F' [Eq. (17)] and the energy error ΔE [Eq. (18)] based on the solitons obtained by the CNO and Newton's method, where an averaged $F' \geq 98\%$ and $\Delta E \leq 0.13$ are indicated. Also in Fig. 3(c), ΔE shows an overall decreasing trend as n increases, which is similar to the tendency in Fig. 2(c) of the first example.

IV. CNO-BASED STABILITY ANALYSIS

The trained CNO has learned the mapping between the initial and final states, which allows it to play the role of traditional PDE solvers in the numerical analysis of solitons. Furthermore, since the forward propagation of the CNO is quite a bit faster than traditional PDE solvers as mentioned before, the numerics can be performed with higher efficiency. In this section, we demonstrate the CNO-based stability analysis on both examples above and compare its running time with that of traditional PDE solvers.

Conventionally, one adds some small perturbations into the solitary waves, i.e.,

$$\psi(x, 0) = \phi(x) + \epsilon \Delta\phi(x), \quad (20)$$

and then evaluates the responses of the system after a period of evolution, which is known as the linear stability analysis. If the perturbation is not amplified, then the soliton is said to be linearly stable. Otherwise, the soliton is said to be linearly unstable. Solitary waves can also be unstable in a nonlinear fashion, which means the instability arises from the nonlinearity as ϵ is large enough to exceed the linear response regime. To fully understand the stability of a soliton, both linear and nonlinear effects should be considered.

We carry out the stability analysis of the two examples by feeding the trained CNO with perturbed initial states $\psi(x, 0)$ [Eq. (20)], where $\Delta\phi(x) = \sum_n c_n \xi_n$ is randomly generated using the basis Eqs. (16) and (19), and ϵ denotes the overall amplitude of perturbation. After an evolution time of T , the response to perturbations is embedded in the output of CNO, i.e., $\psi_{\text{pred}}(x, T)$. We then examine several observables to quantify the responses. For the first example, we look at the center of mass of the BEC $\overline{x_{t=\{0,T\}}} = \overline{|\langle x(t) \rangle|}$ and its envelop width $\overline{\Delta x_{t=\{0,T\}}} = \sqrt{\overline{\langle x^2(t) \rangle} - \langle x(t) \rangle^2}$, where

$$\begin{aligned} \langle x(t) \rangle &= \frac{1}{N} \int dx x |\psi_{\text{pred}}(x, t)|^2, \\ \langle x^2(t) \rangle &= \frac{1}{N} \int dx x^2 |\psi_{\text{pred}}(x, t)|^2. \end{aligned} \quad (21)$$

The additional overscore on the R.H.S. of the $\overline{x_t}$ and $\overline{\Delta x_t}$ means to take a further average on different perturbations $\Delta\phi(x)$ under a fixed ϵ . For the second example, since the BEC always spreads diffusely inside the box with density modulations, we hence replace $\overline{\Delta x_t}$ by the visibility of density, i.e.,

$$\overline{V_t} = \frac{|\psi_{\text{pred}}(x, t)|_{\text{max}}^2 - |\psi_{\text{pred}}(x, t)|_{\text{min}}^2}{\rho_{\text{avg}}}, \quad (22)$$

with $|\psi_{\text{pred}}(x, t)|_{\text{max, min}}^2$ being the maximal (minimal) density and $\rho_{\text{avg}} = N/2R$ denoting the averaged density.

Figures 4(a) and 4(b) display the stability calculations of the two examples using the CNO, respectively. To be more specific, the left panels in Figs. 4(a1) and 4(b1) show $\overline{x_0}$ (dashed lines) and $\overline{x_T}$ (dotted lines) of the third soliton ϕ_3 (circles) and the fifth soliton ϕ_5 (diamonds) as functions of the perturbation strength ϵ ; the right panel in Fig. 4(a2) shows the envelop width $\overline{\Delta x_0}$ (dashed lines) and $\overline{\Delta x_T}$ (dotted lines) of the first example (homogeneous GP equation); Fig. 4(b2) shows the density visibility $\overline{V_0}$ (dashed lines) and $\overline{V_T}$ (dotted

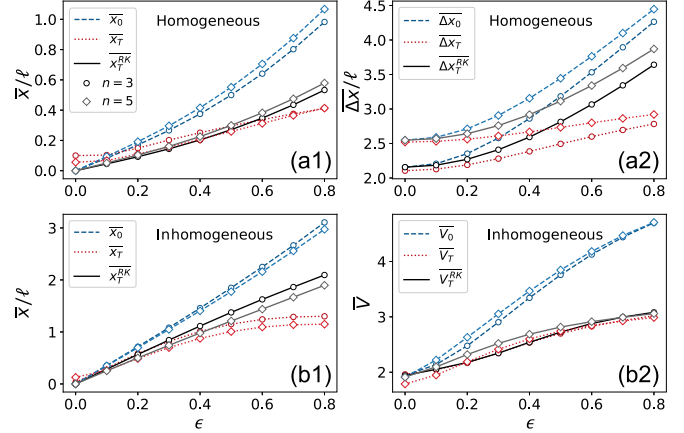


FIG. 4. Stability analysis. Panels (a) and (b) correspond to the cases of the homogeneous and inhomogeneous GP equations. Panels (a1) and (b1) show the center of mass \overline{x} as a function of ϵ , where the dashed lines denote \overline{x} of the initial state $\psi(x, 0)$; the dotted lines and the solid lines indicate \overline{x} of the final state $\psi(x, T)$ using the CNO and the RK method, respectively. Panel (a2) shows the variation of the envelop width $\overline{\Delta x}$. Panel (b2) shows the variation of the density visibility \overline{V} . In each panel, circles and diamonds correspond to the results of the third and the fifth solitons, respectively.

lines) of the second example (inhomogeneous GP equation). In each panel, the black solid line ($\overline{x_T^{\text{RK}}}$, $\overline{\Delta x_T^{\text{RK}}}$, or $\overline{V_T^{\text{RK}}}$) is plotted using the traditional RK finite-difference method which uses the fourth-order RK formula to deal with the time and the finite difference for the spatial discretization. The RK results serve as the ground truth of the evolution. One can clearly observe in Fig. 4 that the CNO predictions are in qualitative agreement with those obtained by the RK method, especially in the regime of small ϵ . Furthermore, for either \overline{x} or $\overline{\Delta x}$, the response at time $t = T$ is always smaller than that at the initial time $t = 0$, which indicates the solitons found out by the CNO are stable. Particularly in the linear response regime $\epsilon \ll 1$, the responses are roughly linear in ϵ , whereas, for ϵ far away from the linear response regime, both examples exhibit apparent nonlinear behaviors.

Finally, let us compare the running time of CNO and traditional PDE solvers. Two traditional algorithms, explicit RK (same as before) and explicit Euler's methods, are considered here. Practically, we evolve the perturbed ϕ_3 state (with fixed $\epsilon = 0.1$) from $t = 0$ to $t = T = 1/\omega$, repeat this process 100 times, and then take the average of the total running time. Figures 5(a) and 5(b) respectively show the running time of various methods in the homogeneous and inhomogeneous examples, where the horizontal axis is a time-space ratio [54]

$$r = \frac{\delta t}{\delta x^2}, \quad (23)$$

with δt and δx being the minimal steps in time and space, respectively. r can be seen as the generalized Courant number [56], being closely related to the stability of algorithms. Generally, explicit algorithms would become unstable as r increases. We remind that the stability here refers to the stability condition of a PDE solver, rather than the stability of the soliton solutions mentioned above. Algorithms that do not satisfy the stability condition would generate exponentially

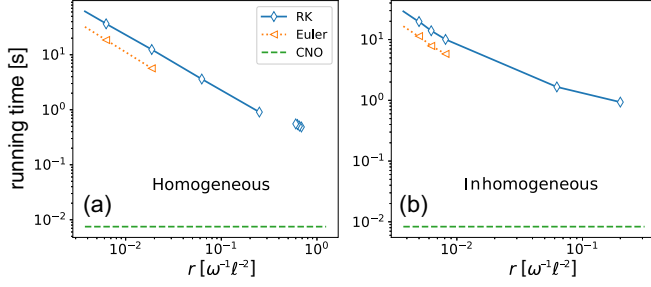


FIG. 5. Running time as a function of the time-space ratio r on the examples of (a) homogeneous and (b) inhomogeneous GP equations. In each panel, dashed lines, triangles, and diamonds indicate the running time of CNO, explicit Euler, and explicit RK methods, respectively. For the later two methods, the connected dotted and solid lines label out the regions where algorithms are stable.

large numerical errors. Our calculation shows that, in Fig. 5, the solid line (RK) and the dotted line (Euler) connected areas are the stable areas, and the unconnected areas on the right are unstable. In our calculations, $\delta x = 0.04\ell$ is fixed. To reduce r means to reduce δt , and hence the total running time behaves $\propto 1/\delta t$, i.e., a straight line with a slope of -1 in the log-log plots.

It is clearly shown in Fig. 5 that, for a given stable δt (or r), the computational time of CNO is significantly less time than that of RK and Euler's methods, and is also δt independent. This is consistent with our complexity analysis in Sec. II B. Specifically, we chose $l = 4$, which is thus a δt -independent constant being much smaller than $t_s = T/\delta t$. On the other hand, for certain stable δt , Euler always consumes less time than RK since Euler's method performs fewer calculations within one δt iteration.

V. SUMMARY AND DISCUSSION

We proposed a data-based approach to search for the solitary solutions of dynamical PDEs. By extending the FNO architecture to the complex field, we developed an architecture called the CNO which can learn the complex mapping between the initial and final states. A combination of the CNO and an energy-restricted search algorithm allows us to identify solitary waves within a limited space of states. Our approach was demonstrated on one-dimensional GP equations with both homogeneous and inhomogeneous nonlinearities, and the resulting solitons exhibited a high overlap with those derived by Newton's method. We also showed the application of the CNO to the stability analysis of solitons. In terms of computational complexity, CNO-based calculations were more efficient than traditional PDE solvers.

There are a number of follow-up questions. Extending our algorithm to higher-dimensional systems is not as straightforward as one might expect. In the above 1D examples, we used a dataset of size 10^4 to ensure that the FNO can capture the initial-final state mapping G . In higher dimensions, much more data points are needed, which places a heavy burden

on data generation. As a result, randomly generating data in higher dimensions is no longer feasible. In addition, the training process for larger FNO models is also slow and time consuming. One possible solution is to adopt semi-supervised learning algorithms, such as active learning [57–59], which can serve as an efficient data acquisition strategy to optimize the data generation and learning efficiency. Furthermore, although symmetries are useful in reducing the search space, as illustrated in the examples, this does not mean that all symmetries are exploitable. In the case of the GP equation with spin-orbit coupling [60–62] or the Klein-Gordon equation with negative mass [63,64], spontaneous symmetry breaking would lead to phase transitions, such that the solitary states exhibit lower symmetry than the Lagrangian. In such a case, the broken symmetry cannot be used as a constraint. In turn, this motivates us to think about how to use the CNO or some other machine-learning algorithms to identify phase transitions. We expect this work, as well as these questions, to prompt more studies in the fields of machine learning and many-body quantum physics.

ACKNOWLEDGMENTS

This work was done when the first author was visiting Microsoft Research. L.C. would like to thank H.-b. Luo for the helpful discussion. L.C. acknowledges support from the National Natural Science Foundation of China (Grants No. 12174236 and No. 12147215) and from the fund for the Shanxi 1331 Project.

APPENDIX: COMPLIMENTS FOR THE ENERGY-RESTRICTED SEARCH

In Figs. 2(c) and 3(c), we observed the energy error ΔE exhibiting a decreasing tendency as n increases. This phenomenon can be attributed to the fact that the energy constraint [the last term in Eq. (7)] becomes increasingly significant as $E_{\psi(x,0)}$ approaches the bound E_{\max} , rendering the searching algorithm less likely to be trapped in a local minimum, as was mentioned in the paragraph above Eq. (18). In the following, we provide a mathematical proof of this statement.

Proposition 1. In the optimization problem (P): $\min_{\psi} \Delta F$ subject to $E_{\psi} \leq E_{\max}$, for any n ($E_{\phi_n} \leq E_{\max}$), there is $E_{\psi_n^*}$ (or ΔE_n) monotonically increasing about E_{\max} , where ψ_n^* is the optimal solution to the problem (P).

Proof. For any n , take E_{\max}^1 (P1) and E_{\max}^2 (P2), satisfying $E_{\phi_n} \leq E_{\max}^2 < E_{\max}^1$. Denote the feasible region of problems (P1) and (P2) being s_1 and s_2 , respectively, then $s_2 \subset s_1$.

For the problem (P1), if $\psi_{n,1}^* \in s_2$, then $\psi_{n,1}^* = \psi_{n,2}^*$. In this case, $E_{\psi_{n,1}^*} = E_{\psi_{n,2}^*}$. Otherwise if $\psi_{n,1}^* \in s_1 \setminus s_2$, and $\psi_{n,2}^* \in s_2$, then $E_{\psi_{n,2}^*} \leq E_{\max}^2 < E_{\psi_{n,1}^*}$.

In summary, when $E_{\phi_n} \leq E_{\max}^2 < E_{\max}^1$, there is $E_{\psi_{n,2}^*} \leq E_{\psi_{n,1}^*}$ ($\Delta E_n^2 \leq \Delta E_n^1$). That is, $E_{\psi_n^*}$ (ΔE_n) monotonically increases with respect to E_{\max} . ■

- [1] D. J. Frantzeskakis, Dark solitons in atomic bose-einstein condensates: From theory to experiments, *J. Phys. A: Math. Theor.* **43**, 213001 (2010).
- [2] K. E. Strecker, G. B. Partridge, A. G. Truscott, and R. G. Hulet, Bright matter wave solitons in bose-einstein condensates, *New J. Phys.* **5**, 73 (2003).
- [3] Y. V. Kartashov, B. A. Malomed, and L. Torner, Solitons in nonlinear lattices, *Rev. Mod. Phys.* **83**, 247 (2011).
- [4] Y. S. Kivshar, Dark solitons in nonlinear optics, *IEEE J. Quantum Electron.* **29**, 250 (1993).
- [5] A. I. Maimistov, Solitons in nonlinear optics, *Quantum Electron.* **40**, 756 (2010).
- [6] J. M. Dudley and J. R. Taylor, Ten years of nonlinear optics in photonic crystal fibre, *Nat. Photon.* **3**, 85 (2009).
- [7] M. Kauranen and A. V. Zayats, Nonlinear plasmonics, *Nat. Photon.* **6**, 737 (2012).
- [8] E. A. Kuznetsov, A. M. Rubenchik, and V. E. Zakharov, Soliton stability in plasmas and hydrodynamics, *Phys. Rep.* **142**, 103 (1986).
- [9] M. J. Ablowitz, G. Biondini, and L. A. Ostrovsky, Optical solitons: Perspectives and applications, *Chaos* **10**, 471 (2000).
- [10] H. A. Haus and W. S. Wong, Solitons in optical communications, *Rev. Mod. Phys.* **68**, 423 (1996).
- [11] A. Hasegawa, Soliton-based optical communications: An overview, *IEEE J. Select. Topics Quantum Electron.* **6**, 1161 (2000).
- [12] Y. Wu and L. Deng, Ultraslow Optical Solitons in a Cold Four-State Medium, *Phys. Rev. Lett.* **93**, 143904 (2004).
- [13] T. Hong, Spatial Weak-Light Solitons in an Electromagnetically Induced Nonlinear Waveguide, *Phys. Rev. Lett.* **90**, 183901 (2003).
- [14] D. Xu, Z. Chen, and G. Huang, Ultraslow weak-light solitons and their storage and retrieval in a kagome-structured hollow-core photonic crystal fiber, *Opt. Express* **25**, 19094 (2017).
- [15] J. Yang, *Nonlinear Waves in Integrable and Nonintegrable Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 2010).
- [16] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method* (Society for Industrial and Applied Mathematics, Philadelphia, 2003).
- [17] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed. (Dover, Mineola, NY, 2001).
- [18] W. Bao and Q. Du, Computing the ground state solution of bose-einstein condensates by a normalized gradient flow, *SIAM J. Sci. Comput.* **25**, 1674 (2004).
- [19] J. Yang and T. I. Lakoba, Accelerated imaginary-time evolution methods for the computation of solitary waves, *Stud. Appl. Math.* **120**, 265 (2008).
- [20] V. I. Petviashvili, Equation of an extraordinary soliton, *Fiz. Plazmy* **2**, 469 (1976).
- [21] Z. H. Musslimani and J. Yang, Self-trapping of light in a two-dimensional photonic lattice, *J. Opt. Soc. Am. B* **21**, 973 (2004).
- [22] M. J. Ablowitz and Z. H. Musslimani, Spectral renormalization method for computing self-localized solutions to nonlinear systems, *Opt. Lett.* **30**, 2140 (2005).
- [23] T. I. Lakoba and J. Yang, A generalized petviashvili iteration method for scalar and vector hamiltonian equations with arbitrary form of nonlinearity, *J. Comput. Phys.* **226**, 1668 (2007).
- [24] J. Yang, Newton-conjugate-gradient methods for solitary wave computations, *J. Comput. Phys.* **228**, 7007 (2009).
- [25] H.-B. Luo, B. A. Malomed, W.-M. Liu, and L. Li, Bessel vortices in spin-orbit-coupled binary bose-einstein condensates with zeeman splitting, *Commun. Nonlinear Sci. Numer. Simul.* **115**, 106769 (2022).
- [26] M. A. Nabian and H. Meidani, A deep learning solution approach for high-dimensional random differential equations, *Probabilistic Eng. Mech.* **57**, 14 (2019).
- [27] S. Karumuri, R. Tripathy, I. Bilonis, and J. Panchal, Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks, *J. Comput. Phys.* **404**, 109120 (2020).
- [28] H. Gao, L. Sun, and J.-X. Wang, PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain, *J. Comput. Phys.* **428**, 110079 (2021).
- [29] Y. Zhu, N. Zabarav, P.-S. Koutsourelakis, and P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *J. Comput. Phys.* **394**, 56 (2019).
- [30] I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Netw.* **9**, 987 (1998).
- [31] J. Blechschmidt and O. G. Ernst, Three ways to solve partial differential equations with neural networks – a review, *GAMM-Mitteilungen* **44**, e202100006 (2021).
- [32] M. 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 (2019).
- [33] M. Raissi and G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, *J. Comput. Phys.* **357**, 125 (2018).
- [34] C. Beck, W. E, and A. Jentzen, Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations, *J. Nonlinear Sci.* **29**, 1563 (2019).
- [35] C. Beck, S. Becker, P. Grohs, N. Jaafari, and A. Jentzen, Solving the kolmogorov pde by means of deep learning, *J. Sci. Comput.* **88**, 73 (2021).
- [36] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck, An overview on deep learning-based approximation methods for partial differential equations, *Discrete Contin. Dyn. Syst. Ser. B* **28**, 3697 (2023).
- [37] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Fourier neural operator for parametric partial differential equations, [arXiv:2010.08895](https://arxiv.org/abs/2010.08895).
- [38] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, Neural operator: Learning maps between function spaces with applications to PDEs, *J. Mach. Learn. Res.* **24**, 1 (2023).
- [39] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods* (Athena Scientific, Belmont, MA, 1996).
- [40] S. Scardapane, S. Van Vaerenbergh, A. Hussain, and A. Uncini, Complex-valued neural networks with non-parametric activation functions, [arXiv:1802.08026](https://arxiv.org/abs/1802.08026).
- [41] J. Basse, L. Qian, and X. Li, A survey of complex-valued neural networks, [arXiv:2101.12249](https://arxiv.org/abs/2101.12249).
- [42] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh,

- Y. Bengio, and C. J. Pal, Deep complex networks, [arXiv:1705.09792](#).
- [43] N. Guberman, On complex valued convolutional neural networks, [arXiv:1602.09046](#).
- [44] M. Arjovsky, A. Shah, and Y. Bengio, Unitary evolution recurrent neural networks, [arXiv:1511.06464](#).
- [45] D. Hendrycks and K. Gimpel, Gaussian error linear units, [arXiv:1606.08415](#).
- [46] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, Improving language understanding by generative pre-training, 2018.
- [47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, [arXiv:1810.04805](#).
- [48] C. Pethick and H. Smith, *Bose-Einstein Condensation in Dilute Gases*, 2nd ed. (Cambridge University Press, Cambridge, England, 2008).
- [49] L. P. Pitaevskii and S. Stringari, *Bose-Einstein Condensation* (Oxford Science, Oxford, 2003).
- [50] J. Belmonte-Beitia, V. M. Pérez-García, V. Vekslerchik, and P. J. Torres, Lie Symmetries and Solitons in Nonlinear Systems with Spatially Inhomogeneous Nonlinearities, *Phys. Rev. Lett.* **98**, 064102 (2007).
- [51] Y. Sivan, G. Fibich, and M. I. Weinstein, Waves in Nonlinear Lattices: Ultrashort Optical Pulses and Bose-Einstein Condensates, *Phys. Rev. Lett.* **97**, 193902 (2006).
- [52] M. Olshanii, Atomic Scattering in the Presence of an External Confinement and a Gas of Impenetrable Bosons, *Phys. Rev. Lett.* **81**, 938 (1998).
- [53] T. Bergeman, M. G. Moore, and M. Olshanii, Atom-Atom Scattering under Cylindrical Harmonic Confinement: Numerical and Analytic Studies of the Confinement Induced Resonance, *Phys. Rev. Lett.* **91**, 163201 (2003).
- [54] P. O. J. Scherer, *Computational Physics: Simulation of Classical and Quantum Systems*, 2nd ed. (Springer, New York, 2013).
- [55] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](#).
- [56] R. Courant, K. Friedrichs, and H. Lewy, Über die partiellen Differenzgleichungen der mathematischen Physik, *Math. Ann.* **100**, 32 (1928).
- [57] B. Settles, *Active Learning* (Springer International Publishing, Cham, Germany, 2012).
- [58] J. Yao, Y. Wu, J. Koo, B. Yan, and H. Zhai, Active learning algorithm for computational physics, *Phys. Rev. Res.* **2**, 013287 (2020).
- [59] L. Chen, X. Liang, and H. Zhai, The bayesian committee approach for computational physics problems, [arXiv:2011.06086](#).
- [60] Y.-J. Lin, K. Jiménez-García, and I. B. Spielman, Spin-orbit-coupled bose-einstein condensates, *Nature (London)* **471**, 83 (2011).
- [61] Y. Li, L. P. Pitaevskii, and S. Stringari, Quantum Tricriticality and Phase Transitions in Spin-Orbit Coupled Bose-Einstein Condensates, *Phys. Rev. Lett.* **108**, 225301 (2012).
- [62] L. Chen, and H. Pu, Synthetic gauge field and spin-orbit coupling in ultracold atomic condensate, in *Models and Methods for Quantum Condensation and Fluids* (World Scientific, Singapore, 2023), pp. 183–202.
- [63] J. Goldstone, Field theories with superconductor solutions, *Nuovo Cim* **19**, 154 (1961).
- [64] M. E. Peskin and D. V. Schroeder, *An Introduction to Quantum Field Theory* (Addison-Wesley, Reading, MA, 1995).