



# Neural networks enforcing physical symmetries in nonlinear dynamical lattices: The case example of the Ablowitz–Ladik model

Wei Zhu<sup>a,\*</sup>, Wesley Khademi<sup>b,c</sup>, Efstathios G. Charalampidis<sup>d</sup>, Panayotis G. Kevrekidis<sup>a</sup>

<sup>a</sup> Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, MA 01003-4515, USA

<sup>b</sup> Computer Science and Software Engineering Department, California Polytechnic State University, San Luis Obispo, CA 93407-0403, USA

<sup>c</sup> School of Electrical Engineering and Computer Science, Oregon State University, OR 97331-5501, USA

<sup>d</sup> Mathematics Department, California Polytechnic State University, San Luis Obispo, CA 93407-0403, USA

## ARTICLE INFO

### Article history:

Received 20 October 2021  
Received in revised form 16 February 2022  
Accepted 14 March 2022  
Available online 28 March 2022  
Communicated by D. Pelinovsky

### Keywords:

Nonlinear dynamical lattices  
Discrete integrable systems  
Spatio-temporal parity symmetries  
Spatial and temporal localization  
Group equivariant Neural Networks

## ABSTRACT

In this work we introduce symmetry-preserving, physics-informed neural networks (S-PINNs) motivated by symmetries that are ubiquitous to solutions of nonlinear dynamical lattices. Although the use of PINNs have recently attracted much attention in data-driven discovery of solutions chiefly to partial differential equations, we demonstrate that they fail at enforcing important physical laws including symmetries of solutions and conservation laws. Through the correlation of parity symmetries in both space and time of solutions to differential equations with their group equivariant representation, we construct group-equivariant NNs which respect spatio-temporal parity symmetry. Moreover, we adapt the proposed architecture to enforce different types of periodicity (or localization) of solutions to nonlinear dynamical lattices. We do so by applying S-PINNs to the completely integrable Ablowitz–Ladik model, and performing numerical experiments with a special focus on waveforms that are related to rogue structures. These include the Kuznetsov–Ma soliton, and Akhmediev breather as well as the Peregrine soliton. Our numerical results demonstrate the superiority and robustness of the proposed architecture over standard PINNs.

© 2022 Elsevier B.V. All rights reserved.

## 1. Introduction

After their original measurement in the North Sea [1–4], rogue waves which appear out of nowhere and disappear without a trace [5] have become a topic of extensive study. Indeed, in recent years, their study has expanded towards a variety of other fields, such as most notably water tank experiments [6–9] and nonlinear optical systems [10–13]. Further efforts have been considered in the realms of plasma physics [14–17] and also Bose–Einstein condensates in atomic physics [18]. These efforts have been captured in various reviews [19,20], as well as books [21,22] on this rapidly developing subject.

At the same time, in recent years, there have been numerous computational developments that can be impactful toward the analysis and numerical exploration of rogue waves. More concretely, with the advance of computing resources and algorithmic innovation in machine learning, data-driven solvers for partial differential equations (PDEs) based on deep neural networks (DNNs) have become a burgeoning domain in applied

and computational mathematics [23–40]. The core idea of these methods is to represent the PDE solution using a neural network (NN), whose parameters are trained via (stochastic) gradient descent (GD) of some variational loss function associated with the PDE under consideration. Among these methods, the physics-informed neural networks (PINNs) [31–33,40] have garnered much attention from the scientific computing community because of their flexibility and gridless nature. Consider, for instance, a PDE of the form

$$\begin{cases} \mathcal{N}\psi = f, & \text{in } \Omega, \\ \mathcal{B}\psi = g, & \text{on } \partial\Omega, \end{cases} \quad (1)$$

where  $\psi$  is an unknown function on  $\Omega$ ,  $\mathcal{N}$  is a (potentially nonlinear) differential operator, and  $\mathcal{B}$  is an operator associated with a specific boundary condition. A PINN for Eq. (1) is a feed-forward NN ansatz  $\psi(\mathbf{x}; \theta)$  that approximates the solution of Eq. (1), where  $\theta$  is the collection of all trainable weights. The optimal  $\theta$  is obtained by solving via GD the following empirical least square minimization of the PDE residuals in the strong form

$$\min_{\theta} \frac{1}{N_f} \sum_{i=1}^{N_f} |\mathcal{N}\psi(\mathbf{x}_i; \theta) - f(\mathbf{x}_i)|^2 + \frac{1}{N_g} \sum_{i=1}^{N_g} |\mathcal{B}\psi(\mathbf{y}_i; \theta) - g(\mathbf{y}_i)|^2, \quad (2)$$

\* Corresponding author.

E-mail addresses: [zhu@math.umass.edu](mailto:zhu@math.umass.edu) (W. Zhu), [khademiw@oregonstate.edu](mailto:khademiw@oregonstate.edu) (W. Khademi), [echarala@calpoly.edu](mailto:echarala@calpoly.edu) (E.G. Charalampidis), [kevrekid@umass.edu](mailto:kevrekid@umass.edu) (P.G. Kevrekidis).

where  $(\mathbf{x}_i)_{i=1}^{N_f}$  and  $(\mathbf{y}_i)_{i=1}^{N_g}$  are discrete random samples drawn from the uniform distributions on  $\Omega$  and  $\partial\Omega$ , respectively. For certain types of linear PDEs, when the sample size and network width approach infinity, convergence analysis of PINNs has been established based on a priori and a posteriori error estimates for the residual minimization [cf. Eq. (2)] in Sobolev spaces [41,42] and Barron-type spaces [43].

Very recently, there has been an ever expanding literature that attempts to bridge these two fields. Indeed, PINNs have been used in order to identify not only solitonic, but also breather, as well as (different-order) rogue wave solutions of the nonlinear Schrödinger (NLS) equation. The NLS is undoubtedly the most prototypical nonlinear model [44,45] that features these types of coherent waveforms and, at the same time, operates as an envelope wave description of a wide range of water-wave, as well as optical, atomic and plasma-wave systems. Hence, it is rather naturally the tool of choice to exemplify such numerical methods, especially because its integrable structure facilitates the analytical availability of such solutions. Indeed, the attempt to use PINNs in integrable systems has been expanding [46], including the consideration of explicit conservation laws in the loss functions, as was done, e.g., recently in [47]. Further studies have simply considered different (but rather similar) dispersive PDEs, including the defocusing NLS [48], a higher-order NLS [49], the derivative NLS equation [50,51], or the Chen–Lee–Liu equation [52], among others.

Our aim in the present work is to present a modified formulation of the PINN approach in comparison to the above works. Most of the above efforts utilize a loss function based on matching the equation of motion and the identification of special solutions (often known via integrability). Only one of these works incorporates explicitly (to our knowledge) in this dispersive setting the presence of conservation laws [47] and even in that case, nonconservative solutions are only penalized in the loss function, and conservation laws are thus not exactly enforced. Moreover, these conservation laws need to be known in advance based on PDE analysis or other similar methods. Our aim here is to incorporate *generic* symmetries of the PDE, such as parity or time-reversal symmetry, but in a way built-into the NN considerations, i.e., through the construction of NN layers with guaranteed spatio-temporal parity symmetry towards representing the solution. On the one hand, the notions of parity and time-reversal (the so-called  $\mathcal{PT}$ -symmetry) has been a topic of wide relevance in its own right in a wide range of systems, summarized, e.g., in [53,54]. On the other hand, more generally, our motivation is to incorporate symmetries that may be of broad relevance to Hamiltonian systems and, indeed, beyond (as these symmetries are neither sufficient, nor necessary for the system to be Hamiltonian). We note that time-reversal symmetry has also been leveraged in the prior work [55] to build ODE networks for simulating time-reversal complex dynamics, but such symmetry is again not *exactly* enforced in this work as the discrepancy between forward and backward dynamics is only penalized in a loss function. In a sense, our effort to construct PINNs incorporating group equivariance, the hereafter referred to as S-PINNs (with S standing for symmetry) draws parallels to the attempts to formulate symplectic neural networks, such as the SympNets of [56] (see also earlier related attempts cited therein). We believe that such an attempt will be of broad relevance to PDE systems bearing symmetries and we hope that it will be more widely used accordingly in the future elsewhere.

Our tool of choice, partly to distinguish ourselves from earlier studies, and partly to focus on the important and wide literature of nonlinear dynamical lattices [57], is a spatially *discrete* (integrable) system that also bears such rogue wave solutions: the so-called Ablowitz–Ladik lattice [45]. The relevant rogue waveforms

have been identified, e.g., in the work of [58], and have been used, e.g., for a systematic study of the stability of time-periodic (so-called Kuznetsov–Ma) and rogue (Peregrine) waveforms in the recent work of [59]. The spatio-temporal localization of the relevant rogue-wave patterns renders them an especially appealing (and challenging) testbed for the consideration of S-PINNs, in comparison to standard single-soliton structures. Moreover, the increasing interest in rogue waves within experiments in water waves, nonlinear optics and elsewhere also adds to their recent and widespread appeal [19,20].

The structure of our presentation of the model and of the application of S-PINNs on it will proceed as follows. In Section 2, we offer the background of the model and of the fundamental nonlinear wave solutions of interest. We complement this with some basic notions associated with PINNs and the equivariant NNs of interest herein. Then, in Section 3, we formulate theoretically the S-PINNs proposed, explaining how to construct the respective layers and how to represent the solutions of interest. Then, in Section 4, we present a series of numerical examples, illustrating the performance of the method and its superiority over conventional PINNs. Finally, in Section 5, we summarize our findings and present some conclusions, as well as some suggestions for future work.

## 2. Background

### 2.1. The model and its theoretical setup

We start by discussing the model and its solutions that we will subsequently employ for our numerical experiments. In particular, the model that we consider in this work is the (discrete) completely integrable Ablowitz–Ladik (AL) model [60,61] given by

$$i\dot{\psi}_n + (\psi_{n+1} - 2\psi_n + \psi_{n-1}) + (\psi_{n+1} + \psi_{n-1})|\psi_n|^2 = 0, \quad (3)$$

where  $\psi_n := \psi_n(t) : \mathbb{Z} \times \mathbb{R} \rightarrow \mathbb{C}$  corresponds to the complex wavefunction of the  $n$ th lattice site ( $n \in \mathbb{Z}$ ) at time  $t \in \mathbb{R}$ ,  $i = \sqrt{-1}$ , and the overdot stands for time differentiation. While this model has not been (as of yet) realized in a physical experiment, it is well-known to serve as the comparison tool of choice for comparing/contrasting with the behavior of the standard non-integrable discrete NLS (so-called DNLS) model involving a local cubic nonlinearity  $|\psi_n|^2\psi_n$  [62] instead of the nonlinear term  $(\psi_{n+1} + \psi_{n-1})|\psi_n|^2$  in Eq. (3). Indeed, it has been utilized for developing perturbative calculations for the solitonic solutions [63], their stability features [64] and recently even for rogue (and related) waveforms [59]. We will focus on bi-periodic solutions of Eq. (3), that is, solutions bearing two frequencies,  $2q^2$  and  $\omega$  (or  $\tilde{\tau}$ ) to be explained in what follows. The first frequency,  $2q^2$ , corresponds to the overall background phase,  $e^{2iq^2t}$ , of the plane wave on top of which the solutions of Eq. (3) will be evolving. We thus introduce the separation of variables ansatz:

$$\psi_n = \psi_n e^{2iq^2t}, \quad (4)$$

where  $\psi_n = \psi_n(t) : \mathbb{Z} \times \mathbb{R} \rightarrow \mathbb{C}$  is the soliton on top of the background  $e^{2iq^2t}$ . Upon inserting Eq. (4) into Eq. (3), we have

$$i\dot{\psi}_n + (\psi_{n+1} - 2\psi_n + \psi_{n-1}) + (\psi_{n+1} + \psi_{n-1})|\psi_n|^2 - 2q^2\psi_n = 0, \quad (5)$$

where  $q$  fixes the background amplitude of the solution, as indicated above. Hereafter, we set  $q \equiv 1/\sqrt{2}$  for convenience.

The complete integrability of the AL model [cf. Eq. (5) or, equivalently, Eq. (3)] is tantamount to the existence of an infinite number of conserved quantities and the presence of an underlying Lax pair formulation [60,61]. The solutions of interest have

**Table 1**  
Notations introduced for the AL model in Section 2.1.

Notation	Description	Reference
$\Psi_n = \psi_n(t)$	Complex wavefunction of the $n$ th lattice site at time $t$	Eq. (3)
$2q^2$	Frequency of the background phase of the plane wave	Eq. (4)
$\psi_n = \psi_n(t)$	Soliton solution on top of the background	Eq. (5)
$\omega$	Temporal frequency of the KM soliton	Eq. (6)
$\theta, G, r$	Parameters determined by $\omega$ in the KM soliton	Eq. (6)
$\tilde{r}$	Spatial frequency of the Akhmediev breather	Eq. (7)
$\tilde{\omega}, \tilde{\theta}, \tilde{G}$	Parameters determined by $\tilde{r}$ in the Akhmediev breather	Eq. (7)

been independently derived via direct techniques [58], and via the Inverse Scattering Transform (IST) [65]. We will primarily focus on three solutions that differ in terms of their localization. On the one hand, the discrete temporally periodic Kuznetsov–Ma soliton is given by

$$\psi(n, t) := \psi_n(t) = \frac{1}{\sqrt{2}} \frac{\cos(\omega t + i\theta) + G \cosh(rn)}{\cos(\omega t) + G \cosh(rn)}, \quad (6)$$

where  $\omega$  is the second frequency (related to the period  $T = 2\pi/\omega$ ) in addition to the background  $2q^2$  [cf. Eq. (4)]; the parameters  $\theta, r, G$  are determined by  $\omega$  through  $\theta = -\operatorname{arcsinh}(\omega)$ ,  $r = \operatorname{arccosh}([2 + \cosh(\theta)]/3)$  and  $G = -\omega/(\sqrt{3} \sinh(r))$ . On the other hand, a spatially periodic, yet discrete and modulationally unstable solution (being reminiscent of the Akhmediev breather [66] of the Nonlinear Schrödinger (NLS) equation at the continuum limit [45]) is given by

$$\psi(n, t) := \psi_n(t) = \frac{1}{\sqrt{2}} \frac{\cos(\tilde{r}n) + \tilde{G} \cosh(\tilde{\omega}t + i\tilde{\theta})}{\cos(\tilde{r}n) + \tilde{G} \cosh(\tilde{\omega}t)}, \quad (7)$$

where  $\tilde{r}$  corresponds to the second (spatial) frequency, and the parameters  $\tilde{\omega}, \tilde{\theta}, \tilde{G}$  depend on  $\tilde{r}$  through  $\tilde{\omega} = -\sqrt{1 - [3 \cos \tilde{r} - 2]^2}$ ,  $\tilde{\theta} = \arccos[1 - 3(1 - \cos \tilde{r})]$ , and  $\tilde{G} = \pm 3 \sin \tilde{r} / \sin \tilde{\theta}$ .

Finally, alongside the KM and Akhmediev breathers, a doubly localized solution, that is, a solution which is localized in both space and time, exists, and it is the discrete analogue of the so-called Peregrine soliton in the form:

$$\psi(n, t) := \psi_n(t) = \frac{1}{\sqrt{2}} \left[ 1 - \frac{6(1 + 2it)}{1 + 2n^2 + 6t^2} \right]. \quad (8)$$

It should be noted that the Peregrine soliton of Eq. (8) can be obtained from the KM structure [cf. Eq. (6)] in the limiting case of  $T \rightarrow \infty$ . We summarize in Table 1 the notations introduced in this section with the corresponding reference equations.

We conclude this section by highlighting an important property of all the above solutions, which itself will be the building block for constructing S-PINNs in the following. In particular, the KM, Akhmediev, and Peregrine waves obey the spatio-temporal parity symmetry

$$\psi(n, -t) = \overline{\psi(n, t)}, \quad \psi(-n, t) = \psi(n, t). \quad (9)$$

Indeed, the latter is a property of the AL model which respects parity and time-reversal. Hereafter, we will seek to adapt NNs so as to respect these key spatio-temporal symmetries of the model.

## 2.2. The need for group equivariance and equivariant neural networks

As indicated in the introduction, over the past few years, PINNs have become a prototypical tool of choice for the leveraging of the substantial advances in the realm of NNs for the study of physically inspired (chiefly) linear and nonlinear PDE problems. Nevertheless, and despite their name, PINNs are known to fail at enforcing important physical laws such as symmetries and conservation laws in the data-driven solutions [42]. This is especially the case when the numbers of the collocation points  $N_f$  and  $N_g$

used for training in Eq. (2) are relatively small, which is inevitable when solving high-dimensional PDEs. For example, Fig. 1 displays the exact and PINN predicted KM solution [cf. Eq. (6)] of the AL model trained with  $N_f = 1,000$  collocation points; in particular, the bottom two panels present the spatial distribution of the amplitude  $|\psi_n(t = -0.67)|$  and its temporal evolution  $|\psi_0(t)|$ . It is evident that when  $N_f$  is small, PINN fails to learn a solution obeying time-periodicity and spatio-temporal parity symmetry specified by Eq. (9). It is thus important to incorporate such physical symmetry into the network models such that data-driven PDE solvers can become more generalizable, especially in the small-data regime.

Group equivariance is a symmetry property for a mapping  $f: \mathcal{X} \rightarrow \mathcal{Y}$  to commute with the group actions on the domain  $\mathcal{X}$  and codomain  $\mathcal{Y}$ . More specifically, let  $G$  be a group, and  $T_g^{\mathcal{X}}$  and  $T_g^{\mathcal{Y}}$ , respectively, be group actions on  $\mathcal{X}$  and  $\mathcal{Y}$ . A function  $f: \mathcal{X} \rightarrow \mathcal{Y}$  is said to be  $G$ -equivariant if

$$f(T_g^{\mathcal{X}}x) = T_g^{\mathcal{Y}}(f(x)), \quad \forall g \in G, x \in \mathcal{X}. \quad (10)$$

In the context of the AL model [cf. Eq. (5)], the spatio-temporal parity symmetry of the solutions specified by Eq. (9) can be equivalently described as  $\psi(\cdot, \cdot): \mathbb{Z} \times \mathbb{R} \rightarrow \mathbb{C} \cong \mathbb{R}^2$  being equivariant to the group  $G = \mathbb{Z}_2 \times \mathbb{Z}_2$ , where  $\mathbb{Z}_2 = \{0, 1\}$  is the cyclic group of order 2. More specifically, we have  $\mathcal{X} = \mathbb{Z} \times \mathbb{R}$ ,  $\mathcal{Y} = \mathbb{R}^2$ ,

$$T_g^{\mathcal{X}}(n, t)^T = ((-1)^{g_1}n, (-1)^{g_2}t)^T, \quad T_g^{\mathcal{Y}}(u, v)^T = (u, (-1)^{g_2}v)^T, \quad (11)$$

for all  $g = (g_1, g_2) \in G$ ,  $(n, t)^T \in \mathbb{Z} \times \mathbb{R}$ ,  $(u, v)^T \in \mathbb{R}^2$ , and the solution  $\psi: \mathcal{X} \rightarrow \mathcal{Y}$  is  $G$ -equivariant:

$$\psi \circ T_g^{\mathcal{X}} = T_g^{\mathcal{Y}} \circ \psi, \quad \forall g \in G. \quad (12)$$

Group-equivariant NNs are a special class of DNN architectures that are guaranteed to represent only maps satisfying the group equivariance specified by Eq. (10). Inspired originally by computer vision applications to incorporate into DNN models the discrete translation and rotation symmetries through group convolutions [67], the method has been generalized in [68–70] and applied to 2D rotations  $SO(2)$  [71–78], 3D rotations  $SO(3)$  [79–85], rescaling [86–90], and permutation [91–95]. In the next section, we explain how to construct symmetry-preserving physics-informed neural networks (S-PINNs) for the AL model that are guaranteed to respect *simultaneously* the spatio-temporal parity symmetry and space/time-periodicity using group-equivariant NNs.

## 3. Symmetry-preserving physics-informed neural networks (S-PINNs)

### 3.1. Spatio-temporal parity symmetry

We first explain how to construct an  $L$ -layer NN  $\Phi(n, t; \theta)$ , where  $\theta$  is the collection of trainable parameters, with guaranteed spatio-temporal parity symmetry of Eq. (9) to represent the solution  $\psi(n, t) := \psi_n(t)$  of the AL model [cf. Eq. (5)]. We consider the input and output vector spaces  $\mathcal{F}_0 = \mathbb{R} \times \mathbb{R} \supset \mathbb{Z} \times \mathbb{R}$  and  $\mathcal{F}_L = \mathbb{R}^2 \cong \mathbb{C}$ , and the  $G = \mathbb{Z}_2 \times \mathbb{Z}_2$ -actions  $T_g^{\mathcal{F}_0}$  and  $T_g^{\mathcal{F}_L}$  defined on  $\mathcal{F}_0$  and  $\mathcal{F}_L$ , respectively, corresponding to the spatio-temporal parity symmetry [cf. Eq. (9)–(11)]

$$T_g^{\mathcal{F}_0}(n, t)^T = ((-1)^{g_1}n, (-1)^{g_2}t)^T, \quad T_g^{\mathcal{F}_L}(u, v)^T = (u, (-1)^{g_2}v)^T. \quad (13)$$

Next, we need to specify a sequence of hidden feature spaces  $(\mathcal{F}_l)_{l=1}^{L-1}$  and (nonlinear) mappings  $(\Phi_l: \mathcal{F}_{l-1} \rightarrow \mathcal{F}_l)_{l=1}^L$  between

consecutive layers such that their composition  $\Phi = \Phi_L \circ \dots \circ \Phi_1 : \mathcal{F}_0 \rightarrow \mathcal{F}_L$  is  $G$ -equivariant:

$$T_g^{\mathcal{F}_0} \circ \Phi_L \circ \dots \circ \Phi_1 = \Phi_L \circ \dots \circ \Phi_1 \circ T_g^{\mathcal{F}_L}, \quad \forall \mathbf{g} \in G. \quad (14)$$

Since the composition of equivariant maps is still equivariant, it suffices for Eq. (14) to hold if we equip the feature spaces  $(\mathcal{F}_l)_{l=1}^{L-1}$  with group actions  $(T_g^{\mathcal{F}_l})_{l=1}^{L-1}$  and require each  $\Phi_l$  between consecutive layers to be equivariant:

$$T_g^{\mathcal{F}_l} \circ \Phi_l = \Phi_l \circ T_g^{\mathcal{F}_{l-1}}, \quad \forall \mathbf{g} \in G, \quad \forall 1 \leq l \leq L. \quad (15)$$

In this paper, we consider the hidden feature spaces  $\mathcal{F}_l = (\mathbb{R}^{D_l})^G = \{f : G \rightarrow \mathbb{R}^{D_l}\}$  for all  $l \in \{1, \dots, L-1\}$ , and equip  $\mathcal{F}_l$  with the group action  $T_g^{\mathcal{F}_l}$  defined as

$$T_g^{\mathcal{F}_l} f(\tilde{\mathbf{g}}) = f(\tilde{\mathbf{g}} - \mathbf{g}), \quad \forall f \in \mathcal{F}_l, \quad \forall \mathbf{g}, \tilde{\mathbf{g}} \in G, \quad \forall 1 \leq l \leq L-1. \quad (16)$$

**Remark 1.** We note that  $T_g^{\mathcal{F}_l}$  in Eq. (16) corresponds to the regular representation of  $G$  on the vector space  $\mathcal{F}_l = (\mathbb{R}^{D_l})^G$  [96]. In principle, one could also consider (combinations of) irreducible representations of  $G$  on the feature spaces, but empirical study suggests such model typically yields inferior performance due to its less expressive nature [70,71,97].

### 3.1.1. Equivariant linear maps

With the group actions  $T_g^{\mathcal{F}_l}$  on  $\mathcal{F}_l$  defined as in Eqs. (13) and (16), we first identify the necessary and sufficient condition for a linear map  $\tilde{\Phi}_l \in \text{Hom}(\mathcal{F}_{l-1}, \mathcal{F}_l)$  to obey the  $G$ -equivariance of Eq. (15).

**Theorem 1.** Let  $\tilde{\Phi}_l \in \text{Hom}(\mathcal{F}_{l-1}, \mathcal{F}_l)$  be a linear map between  $\mathcal{F}_{l-1}$  and  $\mathcal{F}_l$ ,  $1 \leq l \leq L$ . Then  $\tilde{\Phi}_l$  is equivariant under  $T_g^{\mathcal{F}_l}$  of Eqs. (13) and (16), i.e., Eq. (15) holds for all  $l$ , if and only if

- When  $l = 1$ : there exists a matrix  $\mathbf{W}^{(1)} = [\mathbf{W}_1^{(1)}, \mathbf{W}_2^{(1)}] \in \mathbb{R}^{D_1 \times 2}$  such that,  $\forall \mathbf{g} = (g_1, g_2) \in G$ ,

$$[\tilde{\Phi}_1(n, t)](\mathbf{g}) = [(-1)^{g_1} \mathbf{W}_1^{(1)}, (-1)^{g_2} \mathbf{W}_2^{(1)}] \begin{bmatrix} n \\ t \end{bmatrix}. \quad (17)$$

- When  $1 < l < L$ : there exists a matrix-valued function  $\mathbf{W}^{(l)} : G \rightarrow \mathbb{R}^{D_l \times D_{l-1}}$  such that,  $\forall \mathbf{g} \in G, \quad \forall f \in \mathcal{F}_{l-1}$ ,

$$[\tilde{\Phi}_l f](\mathbf{g}) = \sum_{\mathbf{g}' \in G} \mathbf{W}^{(l)}(\mathbf{g} - \mathbf{g}') f(\mathbf{g}'). \quad (18)$$

We note that Eq. (18) is the group convolution proposed in [67].

- When  $l = L$ : there exists a matrix  $\mathbf{W}^{(L)} = [\mathbf{W}_1^{(L)}, \mathbf{W}_2^{(L)}] \in \mathbb{R}^{D_L \times 2}$  such that,  $\forall f \in \mathcal{F}_{L-1}$ ,

$$\tilde{\Phi}_L f = [\mathbf{W}_1^{(L)T} \sum_{\mathbf{g} \in G} f(\mathbf{g}), \mathbf{W}_2^{(L)T} \sum_{\mathbf{g} \in G} (-1)^{g_2} f(\mathbf{g})]^T. \quad (19)$$

**Theorem 1** can be viewed as a special case of the result in [68] where the feature space  $\mathcal{F}_l$  is a degenerate fiber bundle with base space  $B \cong \{0\}$  and the canonical fiber  $\mathcal{F}_l$ . We provide an easier proof in the Appendix A for completeness of the paper. We note that parameter sharing within the group  $G$  is the main reason of achieving equivariant  $\tilde{\Phi}_l \in \text{Hom}(\mathcal{F}_{l-1}, \mathcal{F}_l)$ ; indeed, for  $1 < l < L$ , the dimension of the space of trainable weights [cf. Eq. (18)] is

$$\dim(\{\mathbf{W}^{(l)} : G \rightarrow \mathbb{R}^{D_l \times D_{l-1}}\}) = |G| \cdot (D_l D_{l-1}) = 4D_l D_{l-1}, \quad (20)$$

whereas the dimension of all linear maps  $\text{Hom}((\mathcal{F}_{l-1}, \mathcal{F}_l))$  is

$$\dim[\text{Hom}((\mathcal{F}_{l-1}, \mathcal{F}_l))] = \dim(\mathcal{F}_{l-1}) \cdot \dim(\mathcal{F}_l) \quad (21)$$

$$= \dim[(\mathbb{R}^{D_l})^G] \cdot \dim[(\mathbb{R}^{D_{l-1}})^G] \quad (22)$$

$$= |G| D_l \cdot |G| D_{l-1} \quad (23)$$

$$= 16D_l D_{l-1} \quad (24)$$

### 3.1.2. Equivariant affine maps and nonlinearity

In practice, affine maps  $\Phi_l \in \text{Aff}(\mathcal{F}_{l-1}, \mathcal{F}_l)$  instead of linear maps  $\tilde{\Phi}_l \in \text{Hom}(\mathcal{F}_{l-1}, \mathcal{F}_l)$  are typically used in a DNN model. This amounts to adding the biases  $b^{(l)} \in \mathbb{R}^{D_l}$  to  $\tilde{\Phi}_l$  defined in Eqs. (17), (18), (19):

$$\begin{cases} [\Phi_1(n, t)](\mathbf{g}) = [\tilde{\Phi}_1(n, t)](\mathbf{g}) + b^{(1)}, \\ [\Phi_l f](\mathbf{g}) = [\tilde{\Phi}_l f](\mathbf{g}) + b^{(l)}, & 1 < l < L \\ \Phi_L = \tilde{\Phi}_L. \end{cases} \quad (25)$$

Note that in order to maintain equivariance (15), the biases  $b^{(l)} \in \mathbb{R}^{D_l}$ ,  $1 \leq l < L$  cannot depend on the group element  $\mathbf{g} \in G$ , and there is no  $b^{(L)} \in \mathbb{R}^{D_L}$  in the last layer.

Finally, we need to specify the equivariant nonlinearity  $\sigma : \mathcal{F}_l \rightarrow \mathcal{F}_l$ ,  $1 < l < L$ . Note that the group actions given by Eq. (16) on the hidden layers can be viewed as permutations on  $G$ , and hence pointwise nonlinearity is equivariant as it commutes with permutation. More specifically,  $\forall f \in \mathcal{F}_l$ ,  $1 < l < L$ ,

$$[\sigma f](\mathbf{g}) := (\sigma[f_1(\mathbf{g})], \dots, \sigma[f_{D_l}(\mathbf{g})])^T, \quad (26)$$

where  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  can be any smooth nonlinear function, which we choose as  $\sigma = \tanh$  throughout this work. The  $L$ -layer NN  $\Phi(n, t; \theta)$  with guaranteed spatio-temporal parity symmetry is thus defined as

$$\Phi(\cdot; \theta) := \Phi_L \circ \sigma \circ \Phi_{L-1} \cdots \circ \sigma \circ \Phi_1, \quad (27)$$

where  $\Phi_l$  and  $\sigma$  are defined in Eqs. (25) and (26), and

$$\theta = \{\mathbf{W}^{(1)}, b^{(1)}, \mathbf{W}^{(l)}(\mathbf{g}), b^{(l)}, \mathbf{W}^{(L)} \mid \mathbf{g} \in G, 1 < l < L\} \quad (28)$$

is the collection of all trainable parameters. We note that the symmetry-enforcing NN architecture of Eqs. (27) and (28) can be implemented as standard feed-forward NNs after lexicographically ordering the group  $G$  and subsequently identifying the hidden feature space  $\mathcal{F}_l = (\mathbb{R}^{D_l})^G$  with  $\mathbb{R}^{4D_l}$ , and we defer the technical implementation details to Appendix B.

### 3.2. Space/time-periodicity

Apart from the spatio-temporal parity symmetry, breather solutions for the AL model, such as those given by Eqs. (6) and (7), are also known to be periodic in time and space, respectively. We detail next how to modify the architecture explained in Section 3.1 such that the solution is guaranteed to respect simultaneously the space/time-periodicity and spatio-temporal parity symmetry. Without loss of generality, we assume the solution is time-periodic with angular frequency  $\omega$ , e.g., the KM soliton of Eq. (6), as building a space-periodic solution on the other hand, is similar after reversing the role of space and time.

To ensure time-periodicity, we transform, in the first layer, the input  $(n, t)^T \in \mathcal{F}_0 = \mathbb{R}^2$  to  $(n, \cos \omega t, \sin \omega t)^T \in \mathbb{R}^3$ . This corresponds to changing the first hidden feature space from  $\mathcal{F}_1 = (\mathbb{R}^{D_1})^G$  to  $\tilde{\mathcal{F}}_1 = \mathbb{R}^3$  and setting the first layer (nonlinear) operation  $\Phi_1 : \mathbb{R}^2 \rightarrow \tilde{\mathcal{F}}_1$  as  $\Phi_1(n, t) = (n, \cos \omega t, \sin \omega t)^T$ . One can readily check that  $\Phi_1 : \mathcal{F}_0 \rightarrow \tilde{\mathcal{F}}_1$  is equivariant after equipping  $\tilde{\mathcal{F}}_1$  with the  $G$ -action

$$T_g^{\tilde{\mathcal{F}}_1}(x, c, s)^T = ((-1)^{g_1} x, c, (-1)^{g_2} s), \quad \forall \mathbf{g} = (g_1, g_2) \in G. \quad (29)$$

Therefore, we only need to further modify the second layer operation  $\Phi_2 : \tilde{\mathcal{F}}_1 \rightarrow \mathcal{F}_2$  such that it is equivariant under Eqs. (16) and (29). Similar to Theorem 1, we have

**Theorem 2.** Let  $\tilde{\Phi}_2 \in \text{Hom}(\tilde{\mathcal{F}}_1, \mathcal{F}_2)$  be a linear map between  $\tilde{\mathcal{F}}_1$  and  $\mathcal{F}_2$ . Then  $\tilde{\Phi}_2$  is equivariant under Eqs. (16) and (29) if and only if there exists a matrix  $\mathbf{W}^{(2)} = [\mathbf{W}_1^{(2)}, \mathbf{W}_2^{(2)}, \mathbf{W}_3^{(2)}] \in \mathbb{R}^{D_2 \times 3}$  such that, for all  $\mathbf{g} = (g_1, g_2) \in G$ ,

$$[\tilde{\Phi}_2(n, c, s)](\mathbf{g}) = [(-1)^{g_1} \mathbf{W}_1^{(2)}, \mathbf{W}_2^{(2)}, (-1)^{g_2} \mathbf{W}_3^{(2)}] \begin{bmatrix} n \\ c \\ s \end{bmatrix}. \quad (30)$$



We omit the proof of [Theorem 2](#) as it is nearly identical to the first case of [Theorem 1](#). Similar to Eq. (25), a (trainable  $G$ -independent) bias vector  $b^{(2)} \in \mathbb{R}^{D_2}$  can be added to  $\tilde{\Phi}_2$  such that  $[\Phi_2(n, c, s)](\mathbf{g}) = [\tilde{\Phi}_2(n, c, s)](\mathbf{g}) + b^{(2)}$  becomes an equivariant affine map between  $\tilde{\mathcal{F}}_1$  and  $\mathcal{F}_2$ .

Before ending this section, we would like to note that the symmetry-preserving methodologies developed herein on the discrete Ablowitz–Ladik model can be easily extended to continuous Hamiltonian nonlinear systems such as the Nonlinear Schrödinger (NLS) equation [44,45]. Indeed, the  $\mathcal{PT}$ -symmetry (defined on  $\mathbb{R} \times \mathbb{R}$  for NLS instead of  $\mathbb{Z} \times \mathbb{R}$  for the AL system) and the space/time-periodicity of the solutions of NLS can be modeled similarly using equivariant NNs based on [Theorems 1](#) and [2](#). Our considerations can naturally be extended beyond discreteness, and possibly beyond the Hamiltonian limit, provided suitable symmetries (such as parity) may exist.

#### 4. Numerical results

We demonstrate, in this section, the superior performance of the S-PINNs over standard PINNs after enforcing the physical symmetries detailed in Section 3.

##### 4.1. Data-driven breather solutions of the AL model

We first consider the initial-boundary value problem (IBVP) for  $\psi_n(t)$  on the truncated domain  $\Omega_T = \Omega \times [-T, T]$ , where  $\Omega = \{-N, \dots, N\}$  is a finite 1D lattice,  $N = 50$ , and  $T = 5$ :

$$\begin{cases} \mathcal{N}\psi(n, t) = 0, & (n, t) \in \Omega_T, \\ \psi_n(0) = \psi_n^*(0), & n \in \Omega, \\ \psi_{-N}(t) = \psi_N^*(t), \quad \psi_N(t) = \psi_{-N}^*(t), & t \in [-T, T], \end{cases} \quad (31)$$

where

$$\mathcal{N}\psi = i\dot{\psi}_n + (\psi_{n+1} - 2\psi_n + \psi_{n-1}) + (\psi_{n+1} + \psi_{n-1})|\psi_n|^2, \quad (32)$$

and  $\psi_n^*(t)$  is a (known) analytic solution of the AL model, such as those of Eqs. (6), (7), and (8). It is important to clarify once again here for the reader that we use the overbar to denote complex conjugation, while the asterisk notation is used to denote an analytically available waveform. We use the architecture explained in Section 3.1 for S-PINN, and modify the first and second layer mappings according to Section 3.2 if the solution is further known to be periodic in space or time. The  $L$ -layer PINN and S-PINN, both denoted as  $\Phi(n, t; \theta)$  with two output neurons corresponding to the real and imaginary parts of the solution  $\psi$  of Eq. (31), are learned by minimizing the following mean squared error (MSE)

$$\text{MSE}(\theta) = \text{MSE}_0 + \text{MSE}_b + \text{MSE}_f, \quad (33)$$

with  $\text{MSE}_0$ ,  $\text{MSE}_b$ ,  $\text{MSE}_f$  defined as

$$\begin{cases} \text{MSE}_0 = \frac{1}{2N+1} \sum_{n=-N}^N |\Phi(n, 0; \theta) - \psi_n^*(0)|^2, \\ \text{MSE}_b = \frac{1}{N_t} \sum_{j=1}^{N_t} |\Phi(-N, t_j; \theta) - \psi_{-N}^*(t_j)|^2 \\ \quad + |\Phi(N, t_j; \theta) - \psi_N^*(t_j)|^2, \\ \text{MSE}_f = \frac{1}{(2N-1)N_t} \sum_{j=1}^{N_t} \sum_{n=1}^{N-1} |\mathcal{N}\Phi(n, t_j; \theta)|^2, \end{cases} \quad (34)$$

where  $\{t_j\}_{j=1}^{N_t}$  are randomly sampled from a uniform distribution on  $[-T, T]$ ,  $\mathcal{N}\Phi(n, t; \theta)$  is defined in Eq. (32) by replacing  $\psi_n(t)$  with  $\Phi(n, t; \theta)$ , and the time derivative is obtained via automatic differentiation [98]. The models are trained with 30K iterations

**Table 2**

Comparison of numerical accuracy in obtaining data-driven solutions of the AL model, measured in relative  $L_2$  error against the analytic solutions of Eqs. (6), (7), and (8). The mean and standard deviation of the error after three independent random trials are displayed. The number  $N_t$  measures the number of collocation points used for training the models [cf. Eq. (34)].

Error in learning the KM soliton solution of Eq. (6)				
Models	$N_t = 10$	$N_t = 20$	$N_t = 30$	$N_t = 40$
PINN	$(1.57 \pm 0.15)\text{e-}1$	$(7.32 \pm 3.17)\text{e-}2$	$(6.39 \pm 2.53)\text{e-}2$	$(3.87 \pm 0.90)\text{e-}2$
S-PINN	$(4.26 \pm 1.67)\text{e-}2$	$(2.30 \pm 0.79)\text{e-}3$	$(3.36 \pm 2.11)\text{e-}4$	$(2.03 \pm 1.35)\text{e-}4$
Error in learning the Akhmediev breather solution of Eq. (7)				
Models	$N_t = 10$	$N_t = 20$	$N_t = 30$	$N_t = 40$
PINN	$(7.62 \pm 1.42)\text{e-}1$	$(6.13 \pm 1.97)\text{e-}1$	$(4.25 \pm 1.46)\text{e-}1$	$(2.99 \pm 1.51)\text{e-}1$
S-PINN	$(8.02 \pm 1.25)\text{e-}2$	$(9.25 \pm 1.41)\text{e-}3$	$(2.51 \pm 2.14)\text{e-}3$	$(1.06 \pm 0.37)\text{e-}3$
Error in learning the Peregrine soliton solution of Eq. (8)				
Models	$N_t = 10$	$N_t = 15$	$N_t = 20$	$N_t = 25$
PINN	$(3.65 \pm 2.15)\text{e-}1$	$(5.59 \pm 3.50)\text{e-}1$	$(1.69 \pm 1.04)\text{e-}1$	$(5.89 \pm 4.76)\text{e-}2$
S-PINN	$(3.81 \pm 2.80)\text{e-}2$	$(1.57 \pm 1.01)\text{e-}2$	$(7.32 \pm 5.56)\text{e-}3$	$(1.09 \pm 0.25)\text{e-}3$

**Table 3**

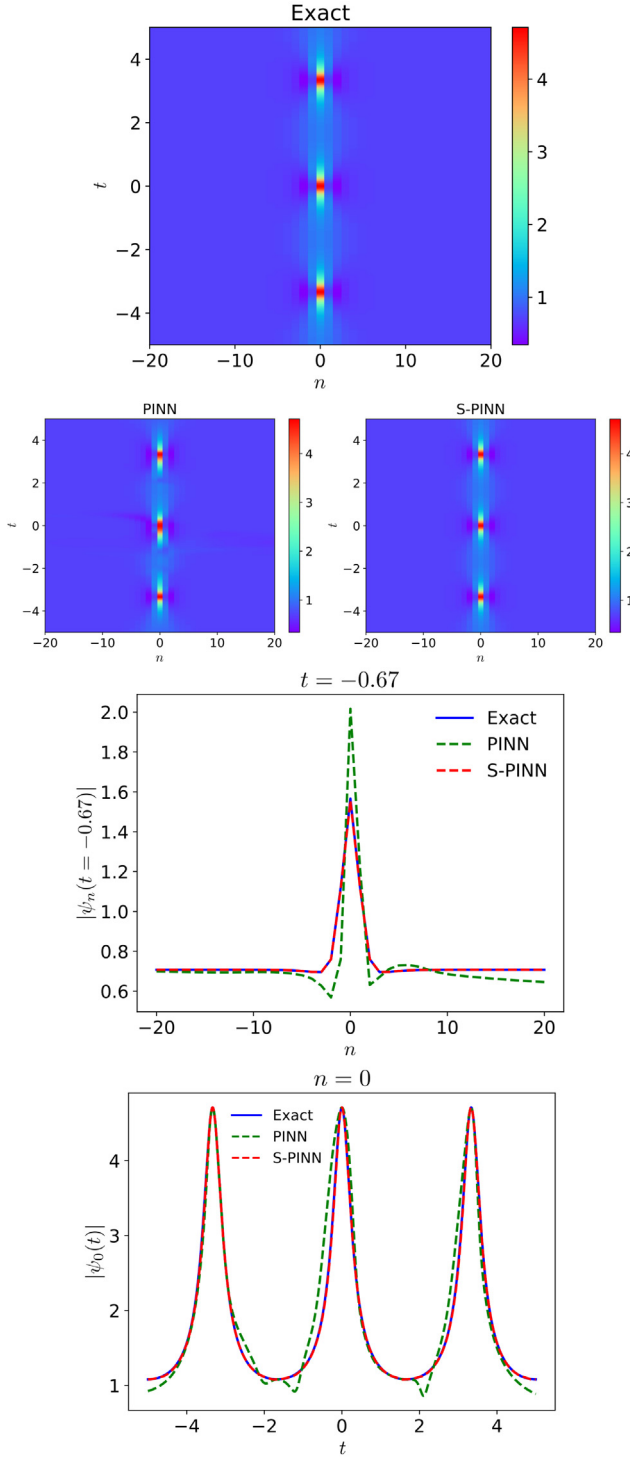
The effect of the network width  $D$  and depth  $L$  on the performance of the models. The mean and standard deviation of the relative error after three independent random trials are displayed.

Error in learning the KM solution of Eq. (6)				
L	D = 40 (PINN)	D = 80 (PINN)	D = 40 (S-PINN)	D = 80 (S-PINN)
4	$(7.22 \pm 3.44)\text{e-}2$	$(7.38 \pm 5.03)\text{e-}2$	$(3.08 \pm 0.92)\text{e-}3$	$(1.15 \pm 0.15)\text{e-}3$
6	$(2.77 \pm 0.02)\text{e-}2$	$(8.81 \pm 1.83)\text{e-}2$	$(3.21 \pm 0.56)\text{e-}4$	$(4.09 \pm 1.40)\text{e-}4$
8	$(4.65 \pm 1.71)\text{e-}2$	$(8.01 \pm 2.69)\text{e-}2$	$(4.31 \pm 4.01)\text{e-}4$	$(6.69 \pm 3.07)\text{e-}3$
Error in learning the Akhmediev breather solution of Eq. (7)				
L	D = 40 (PINN)	D = 80 (PINN)	D = 40 (S-PINN)	D = 80 (S-PINN)
4	$(3.30 \pm 2.04)\text{e-}2$	$(1.71 \pm 0.90)\text{e-}2$	$(2.84 \pm 0.93)\text{e-}3$	$(2.05 \pm 0.10)\text{e-}3$
6	$(4.15 \pm 1.07)\text{e-}3$	$(1.33 \pm 1.14)\text{e-}2$	$(8.65 \pm 1.40)\text{e-}4$	$(7.08 \pm 3.75)\text{e-}4$
8	$(4.86 \pm 1.48)\text{e-}3$	$(6.44 \pm 5.71)\text{e-}2$	$(7.44 \pm 3.97)\text{e-}4$	$(1.74 \pm 0.77)\text{e-}3$
Error in learning the Peregrine soliton solution of Eq. (8)				
L	D = 40 (PINN)	D = 80 (PINN)	D = 40 (S-PINN)	D = 80 (S-PINN)
4	$(7.34 \pm 0.38)\text{e-}3$	$(6.92 \pm 1.52)\text{e-}3$	$(3.15 \pm 0.74)\text{e-}3$	$(2.80 \pm 1.06)\text{e-}3$
6	$(1.27 \pm 0.13)\text{e-}3$	$(1.56 \pm 0.20)\text{e-}3$	$(7.61 \pm 0.87)\text{e-}4$	$(1.32 \pm 0.54)\text{e-}3$
8	$(1.03 \pm 0.19)\text{e-}3$	$(1.06 \pm 0.15)\text{e-}3$	$(4.57 \pm 2.07)\text{e-}4$	$(5.16 \pm 3.21)\text{e-}4$

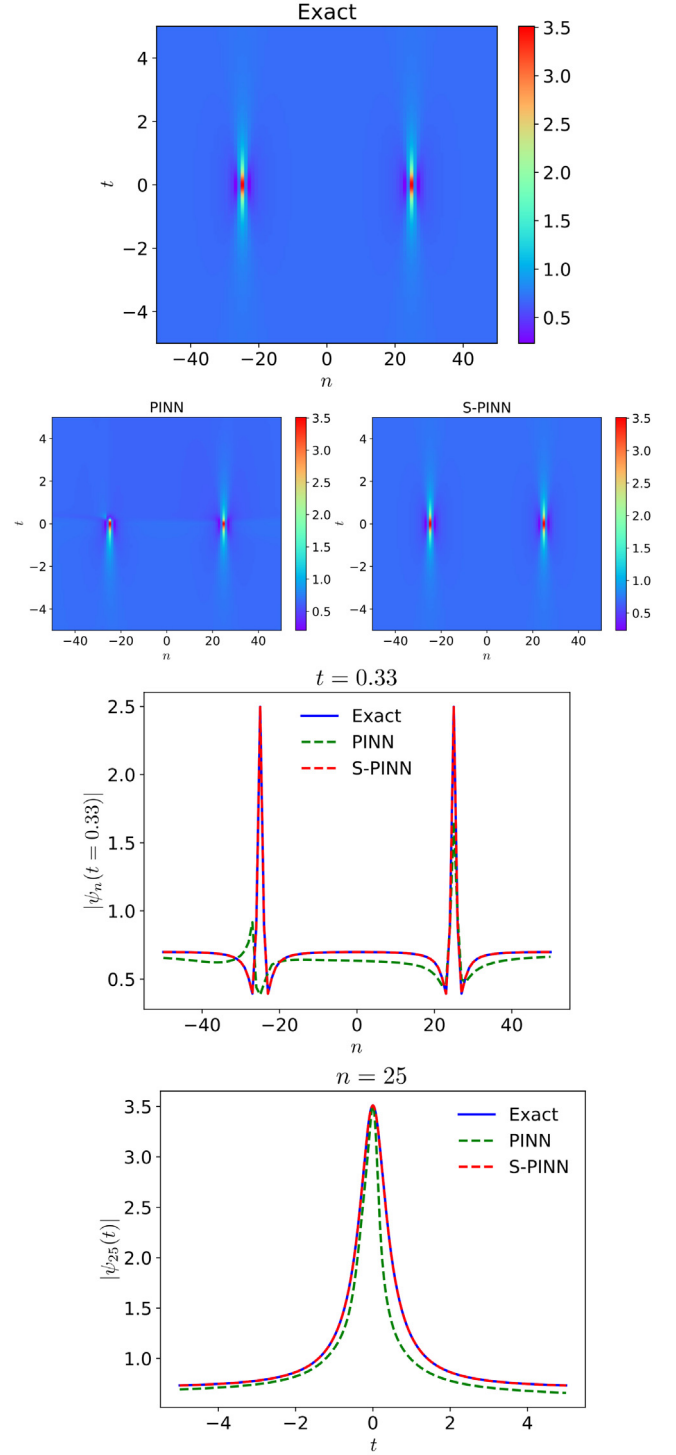
of ADAM [99], and 15K iterations of L-BFGS [100] to ensure convergence. We set the network depth  $L = 6$ , and the width  $(D_l)_{1 \leq l \leq L}$  of the hidden layers is set to  $D = 100$ .

[Table 2](#) displays the relative discrete  $L_2$  error of the data-driven solutions (against the known analytic solutions of Eqs. (6), (7), and (8)) on a grid of size  $101 \times 3001$  sampling the computation domain  $\Omega_T$ . We report the mean and standard deviation of the error after three independent random trials; during each trial, the S-PINN and the regular PINN use the same set of collocation points for training. It is evident that, after enforcing physical symmetry, S-PINN is able to learn solutions with typically around one order of magnitude more accuracy compared to those of the regular PINN, especially when the number of collocation points (measured by  $N_t$ , the number of random time steps used in calculating the MSE [cf. Eq. (34)]) is small during training. Since S-PINN uses the same loss function, Eq. (34), as the regular PINN, the error reduction in [Table 2](#) results only from the fact that physical structures of the solution have been correctly enforced by S-PINN. Indeed, [Figs. 1, 2, and 3](#) provide a visual illustration of the learned solutions, and one can readily verify that physical symmetries of the solutions are *indeed* enforced by S-PINNs, but not regular PINNs.

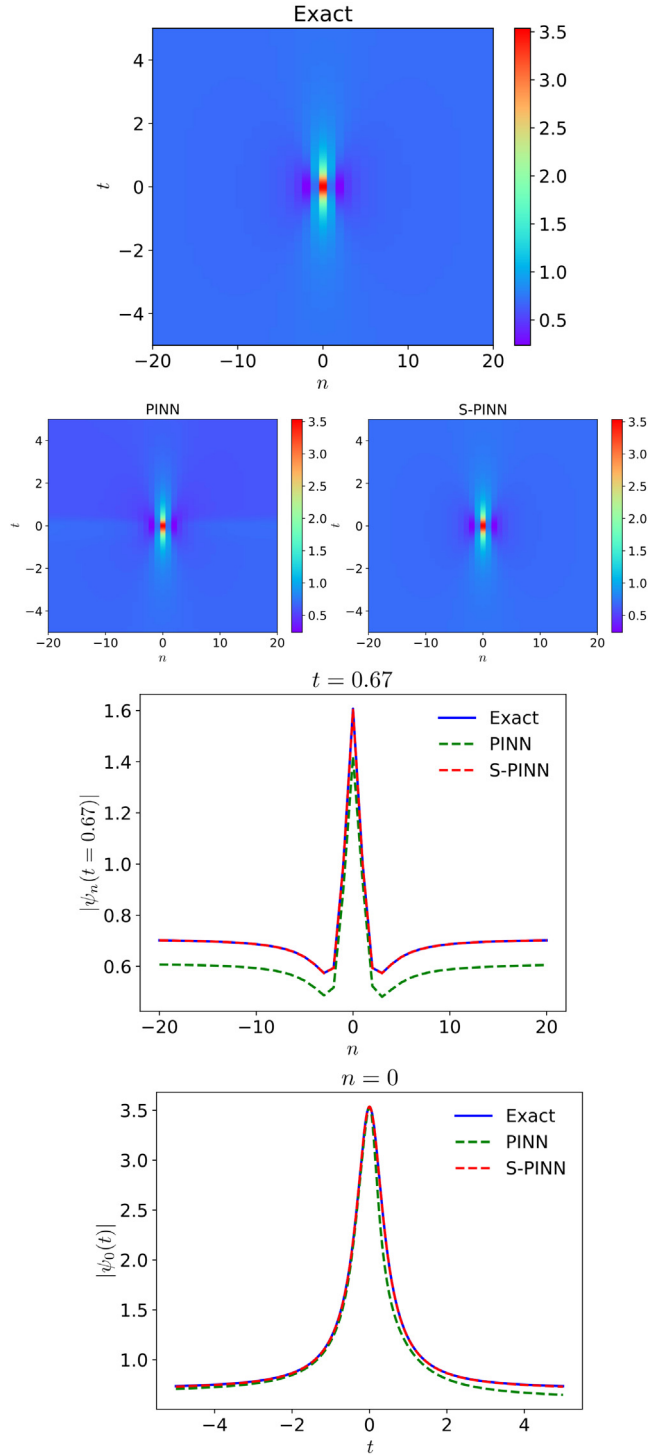
We also provide a systematic study on the effect of the network width  $D$  and depth  $L$  on the performance of the models. The number of training collocation points is fixed to be small throughout the experiments by setting  $N_t = 30$  for the KM and Akhmediev states, and  $N_t = 20$  for the Peregrine soliton. [Table 3](#) displays the mean and standard deviation of the relative error after three independent trials. It can be seen that S-PINNs consistently outperform regular PINNs by typically around an order of magnitude in different settings; nevertheless it is interesting to observe that in [Table 3](#), this advantage is lower in the case of the Peregrine soliton; the reason is that spatial or temporal



**Fig. 1.** Numerical results for the KM soliton obtained by using PINN and S-PINN with training collocation points sampled from the entire computation domain  $\Omega_T$ . The top three panels depict the spatio-temporal evolution of the amplitude  $|\psi(n, t)|$  for the exact KM soliton solution (top panel), and data-driven solutions obtained by PINN (middle left panel) and S-PINN (middle right panel). The bottom two panels present the spatial distribution of the amplitude  $|\psi_n(t = -0.67)|$  (at  $t = -0.67$ , i.e., at a time before the solution attains its maximum amplitude at  $t = 0$ ), and its temporal evolution  $|\psi_0(t)|$  at  $n = 0$ . The solid blue lines in these panels highlight the exact solution whereas the dashed green and red lines correspond to the data-driven solutions obtained by PINN and S-PINN, respectively. It is evident that the regular PINN fails to learn a solution obeying time-periodicity and spatio-temporal parity symmetry specified by Eq. (9), whereas S-PINN successfully captures both symmetries (and hence is producing significantly more accurate solutions) after enforcing them explicitly in the network architecture.



**Fig. 2.** Same as Fig. 1 but for the Akhmediev breather (i.e., the spatially periodic solution to the AL model) with training collocation points sampled from the entire computation domain  $\Omega_T$ . Similarly to Fig. 1, the top three panels depict the spatio-temporal evolution of the amplitude  $|\psi(n, t)|$  for the exact Akhmediev breather (top panel), and data-driven solutions obtained by PINN (middle left panel) and S-PINN (middle right panel). The bottom two panels present the spatial distribution of the amplitude  $|\psi_n(t = 0.33)|$  (i.e., at  $t = 0.33$  after the one the solution attains its maximum amplitude at  $t = 0$ ), and its temporal evolution  $|\psi_{25}(t)|$  (i.e., at the  $n = 25$ th site). The coloring and styles of the lines in these panels are the same as the ones used in Fig. 1 (see also the legends in these panels). Again, the superiority of the use of S-PINN over (regular) PINN in capturing the correct profiles is clearly evident.



**Fig. 3.** Same as Fig. 1 but for the doubly localized, Peregrine soliton with training collocation points sampled from the entire computation domain  $\Omega_T$ . The format of the panels is the same as the one of Figs. 1 and 2. Note that the second to last panel depicts the spatial distribution of the amplitude  $|\psi_n(t=0.67)|$  (i.e., at  $t=0.67$  after the solution attains its maximum amplitude at  $t=0$ ) whereas the last panel, the temporal evolution of  $|\psi_0(t)|$  at  $n=0$ . The use of S-PINN captures again the correct profiles as is evident in the bottom two panels therein.

periodicity is no longer available as an additional constraint for S-PINN to enforce in the case of the Peregrine soliton. Deeper networks typically learn solutions with higher precision, but the improvement in accuracy plateaus when  $L$  and  $D$  are sufficiently large. In fact, both PINN and S-PINN tend to slightly overfit,

**Table 4**

Accuracy of the data-driven solutions of the AL model *extrapolated* beyond the convex hull of the training samples, measured in relative  $L_2$  error against the analytic solutions of Eqs. (6), (7), and (8). The mean and standard deviation of the error after three independent random trials are displayed. The number  $N_t$  measures the number of collocation points used for training the models.

Error in learning the KM solution of Eq. (6)				
Models	$N_t = 10$	$N_t = 20$	$N_t = 30$	$N_t = 40$
PINN	$(1.88 \pm 0.04)e-0$	$(1.96 \pm 0.11)e-0$	$(1.80 \pm 0.03)e-0$	$(1.74 \pm 0.03)e-0$
S-PINN	$(3.12 \pm 0.13)e-2$	$(1.60 \pm 1.01)e-2$	$(9.96 \pm 6.73)e-3$	$(6.16 \pm 2.34)e-3$
Error in learning the Akhmediev breather solution of Eq. (7) (extrapolation)				
Models	$N_t = 10$	$N_t = 20$	$N_t = 30$	$N_t = 40$
PINN	$(6.32 \pm 0.73)e-1$	$(5.76 \pm 0.67)e-1$	$(6.30 \pm 0.92)e-1$	$(5.33 \pm 0.97)e-1$
S-PINN	$(1.68 \pm 0.72)e-1$	$(3.25 \pm 1.22)e-3$	$(4.18 \pm 0.86)e-3$	$(2.37 \pm 1.09)e-3$
Error in learning the Peregrine soliton solution of Eq. (8)				
Models	$N_t = 10$	$N_t = 15$	$N_t = 20$	$N_t = 25$
PINN	$(1.62 \pm 0.11)e-0$	$(1.86 \pm 0.16)e-0$	$(1.77 \pm 0.19)e-0$	$(1.67 \pm 0.06)e-0$
S-PINN	$(3.37 \pm 3.12)e-2$	$(1.03 \pm 0.21)e-2$	$(7.15 \pm 2.37)e-3$	$(6.43 \pm 4.78)e-3$

**Table 5**

Accuracy of PINN and S-PINN in learning a particular solution of the AL model with non-decaying far-field oscillations [59]. The upper and, respectively, lower half of the table display the relative  $L_2$  error of the learned solutions when training collocation points are sampled from the entire domain  $\Omega_T$  and partial domain  $\hat{\Omega}_T$ .

Relative error in learning the numerical solution Fig. 7				
Models	$N_t = 10$	$N_t = 15$	$N_t = 20$	$N_t = 25$
PINN	$(4.39 \pm 0.05)e-1$	$(4.19 \pm 0.22)e-1$	$(4.17 \pm 0.17)e-1$	$(4.00 \pm 2.69)e-1$
S-PINN	$(4.20 \pm 0.34)e-3$	$(1.26 \pm 0.12)e-3$	$(2.79 \pm 0.58)e-2$	$(5.57 \pm 0.56)e-2$
Relative error in learning the numerical solution Fig. 7 (extrapolation)				
Models	$N_t = 10$	$N_t = 15$	$N_t = 20$	$N_t = 25$
PINN	$(2.03 \pm 0.39)e-0$	$(2.20 \pm 0.11)e-0$	$(3.08 \pm 0.69)e-0$	$(2.14 \pm 0.22)e-0$
S-PINN	$(2.99 \pm 1.09)e-3$	$(1.03 \pm 0.29)e-3$	$(1.78 \pm 0.56)e-3$	$(1.18 \pm 0.26)e-3$

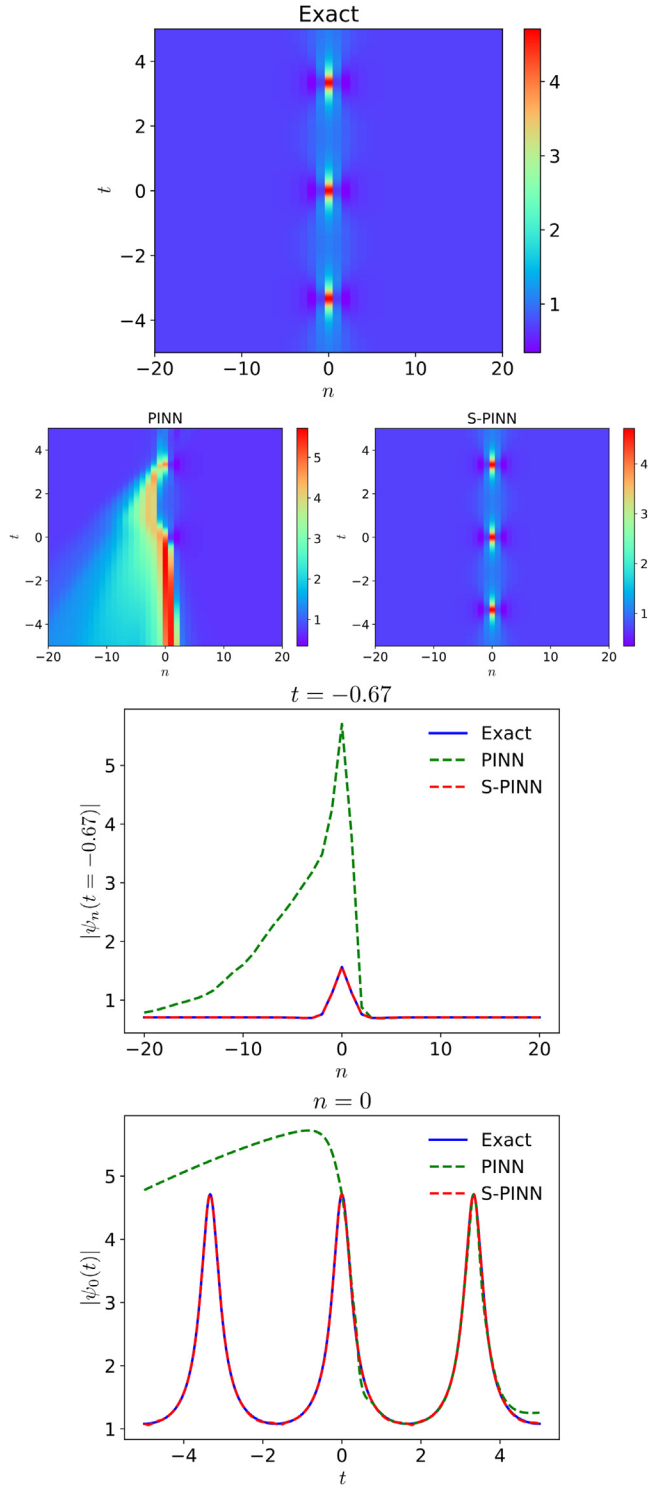
i.e., learning a solution with larger *test* error, in the small-data regime when the width and depth reach  $D=80$  and  $L=8$ .

#### 4.2. Solution extrapolation

We next examine the accuracy of the learned solutions beyond the domain from which the training collocation points are sampled. More specifically, we modify the MSE given by Eq. (34) for training by including in the sum only collocation points from the first quadrant  $\hat{\Omega}_T = \{0, \dots, N\} \times [0, T]$  of the computation domain  $\Omega_T$ , and then we calculate the error of the learned solution on the *entire* domain  $\Omega_T$ , extrapolating beyond the convex hull of the training samples. Table 4 displays the relative  $L_2$  error of the solutions after three independent trials. The accuracy of the extrapolated solutions obtained by the regular PINN stays low as the number of training samples increases, while in comparison S-PINN achieves multiple orders of magnitude more accurate solutions. Figs. 4, 5, and 6 provide a visual illustration on the difference between the solutions learned by regular PINNs and S-PINNs: even though regular PINNs can produce reasonable solutions on the sampling domain  $\hat{\Omega}_T$ , the solutions outside  $\hat{\Omega}_T$  become non-meaningful. S-PINNs, on the other hand, can achieve accurate solutions far beyond the sampling domain after enforcing physical symmetries in the learning process.

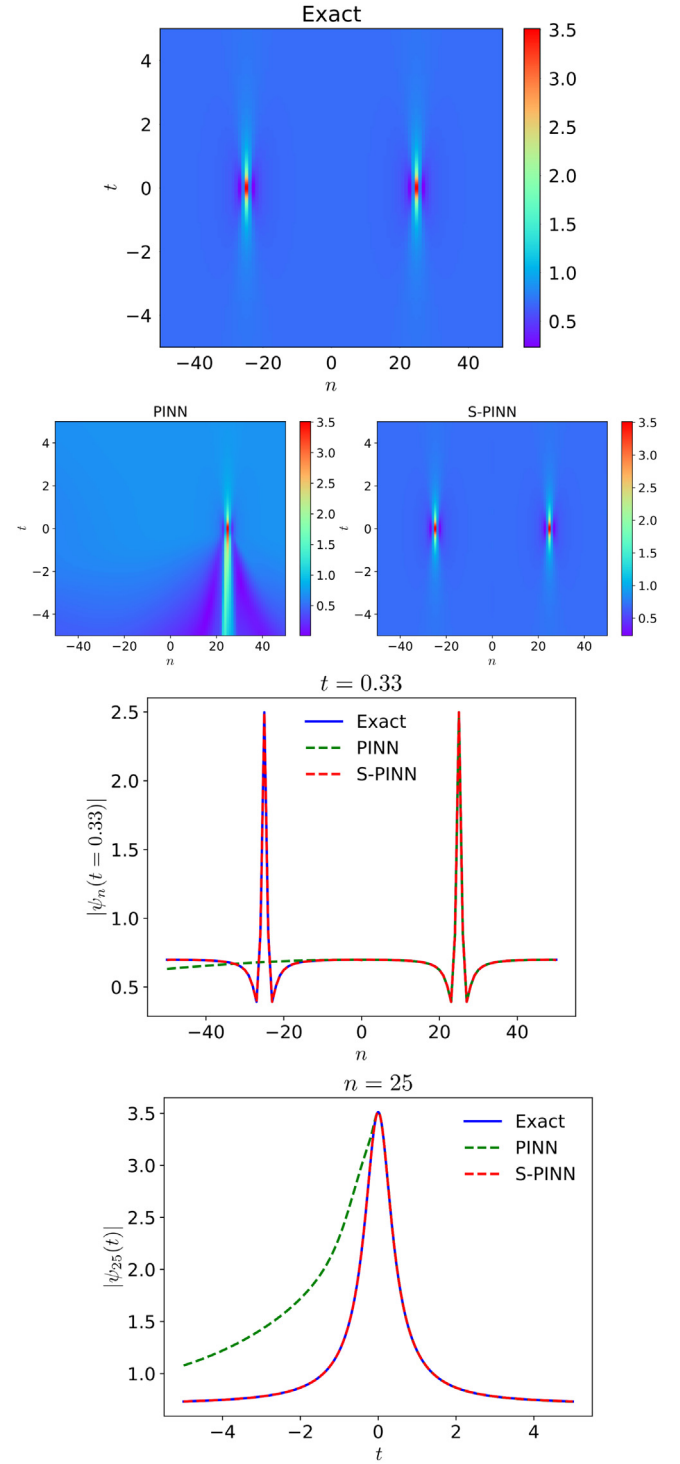
#### 4.3. KM breathers with an oscillatory background: S-PINN vs PINN

Alongside the KM soliton solution of Eq. (6), recently in [59], a time-periodic solution that features small, yet non-decaying far-field oscillations was obtained (through numerical continuation and fixed-point methods) for the AL model. We will use this (numerically exact) solution to demonstrate a case in which both PINN and S-PINN present difficulty in learning solutions with such background oscillatory patterns. To that end, Figs. 7 and 8 summarize our results for both PINN and S-PINN. Moreover, Table 5 shows the error of the solutions learned with comparing models



**Fig. 4.** Same as Fig. 1 but the training collocation points are sampled only from the first quadrant  $\Omega_T$  of the computation domain [cf. Section 4.2], and the extrapolated solutions are shown on the entire domain  $\Omega_T$ . The format of the panels is the same as those of Fig. 1. Although PINN fails in this case, the use of S-PINN demonstrates its robust performance in capturing the correct behavior of the KM soliton.

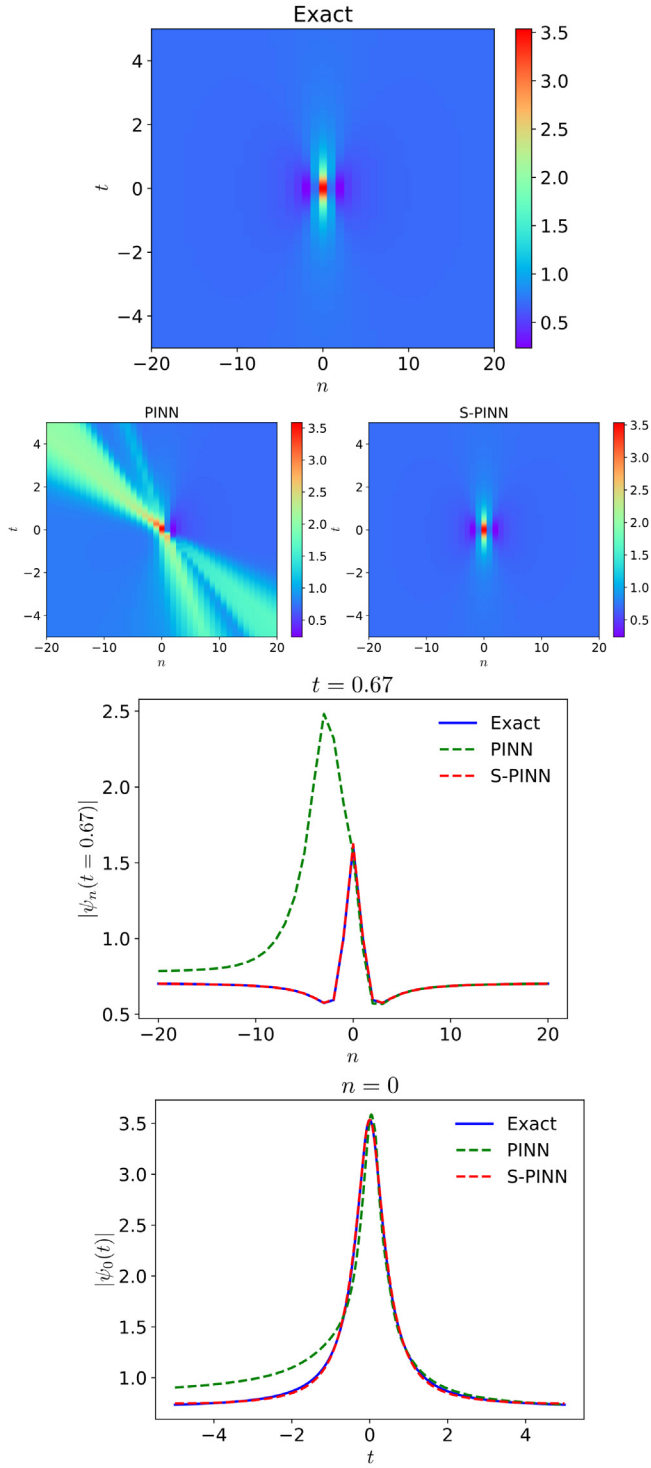
when the training collocation points are sampled from either the entire domain  $\Omega_T$  or the partial domain  $\tilde{\Omega}_T$ , i.e., extrapolated solutions. It can be seen that S-PINN still (significantly) outperforms PINN upon enforcing the physical symmetries discussed above.



**Fig. 5.** Same as Fig. 2 but the training collocation points are sampled only from the first quadrant  $\Omega_T$  of the computation domain [cf. Section 4.2]. Again, the format of the panels is the same as the one of Fig. 2. It is clearly evident that the use of S-PINN correctly constructs the Akhmediev breather in this case too.

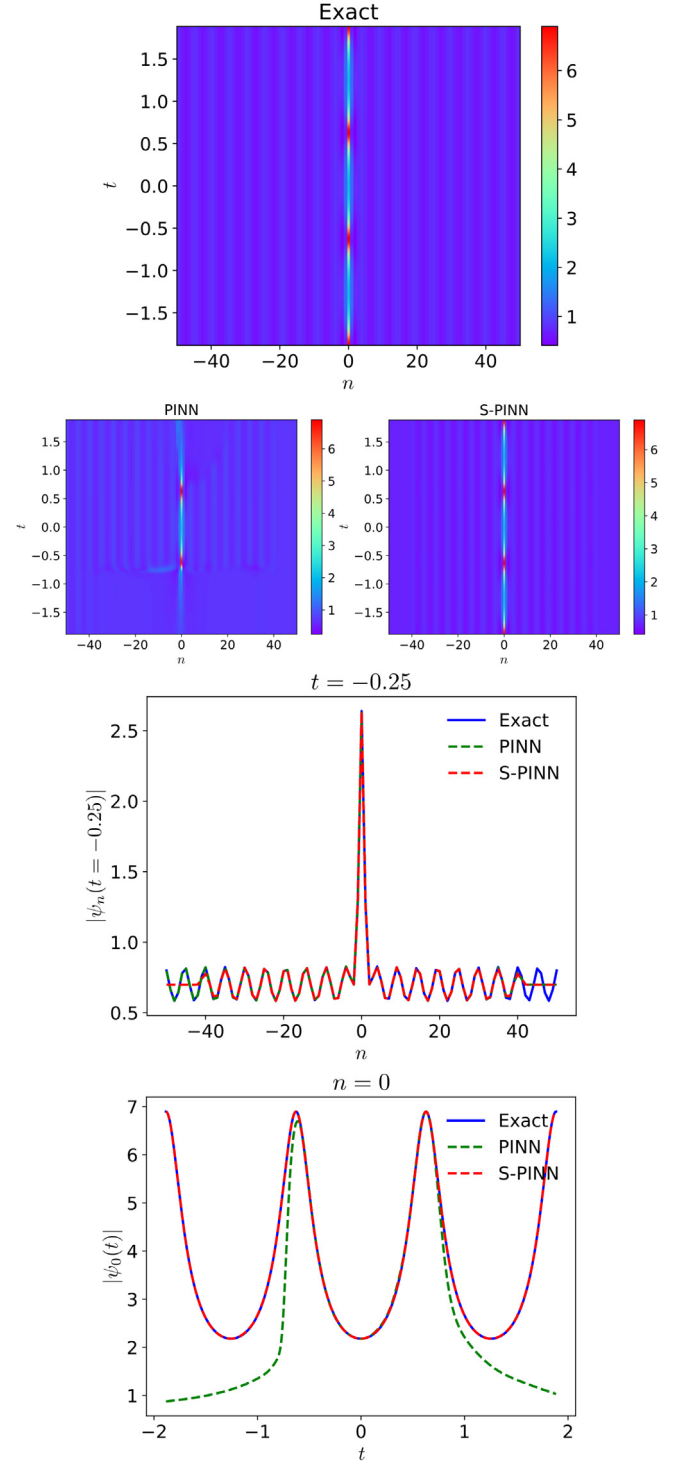
However, the performance of S-PINN starts to deteriorate as  $N_t$  exceeds 15. This can also be observed in Figs. 7 and 8, where the S-PINN has difficulty in capturing the background oscillatory patterns. A closer look at the learning curves in Fig. 9 reveals that the reason is that the training loss decays much more slowly as the number of training samples increases. In fact, the loss is still slowly decaying when we terminate the optimizers. One



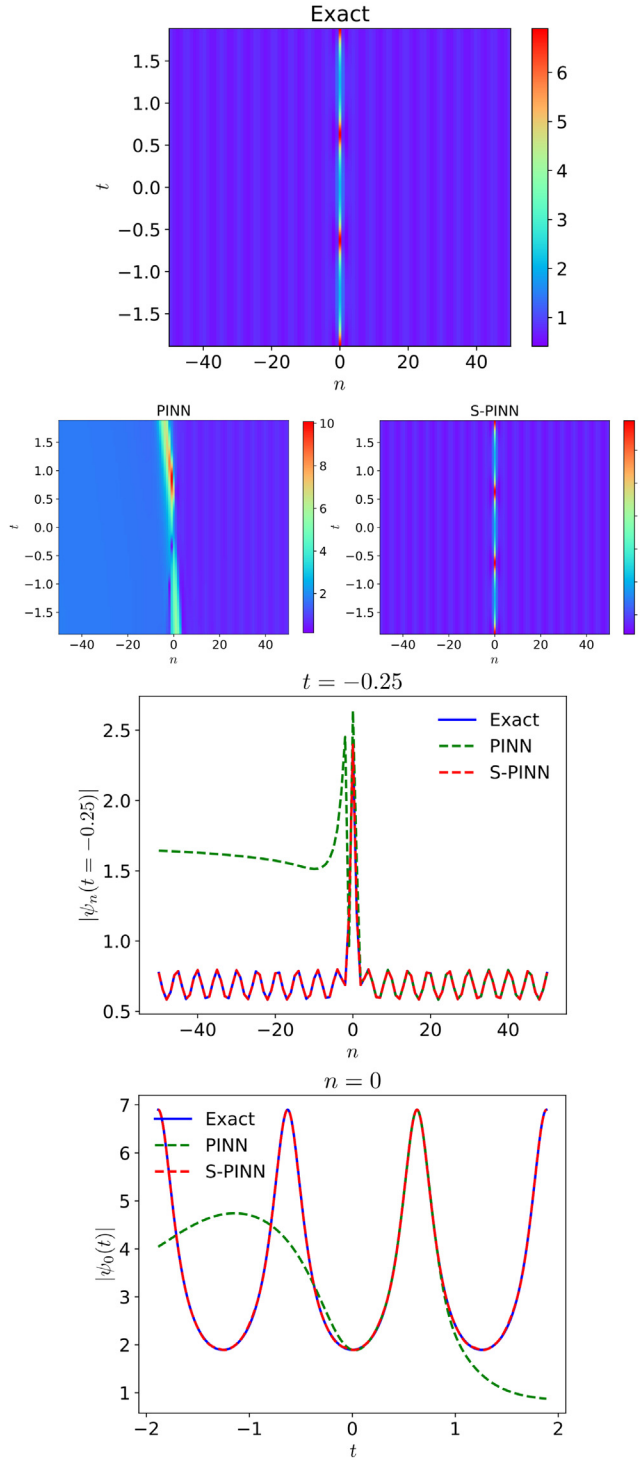


**Fig. 6.** Same as Fig. 3 but the training collocation points are sampled from only the first quadrant  $\Omega_T$  of the computation domain [cf. Section 4.2]. The format of the panels is the same as in Fig. 3. This case demonstrates once again the superiority of S-PINN over (regular) PINN in capturing the Peregrine soliton in such a sampling scenario.

remedy for this issue may be to build  $2N + 1$  networks (one for each discrete spatial location) with only time dependence, but enforcing physical symmetries on such models needs to be formulated differently. This is a topic that is worthwhile of further study, but since the corresponding architecture is fundamentally different, this will be deferred to future work.



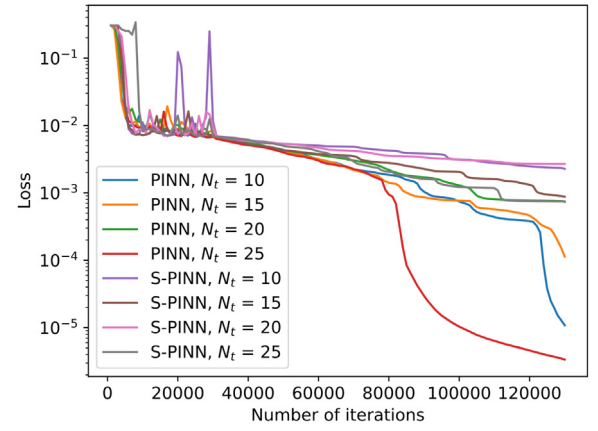
**Fig. 7.** Numerical results on the full space for the KM breather (sitting atop of an oscillatory background) to the AL model which itself was first reported in [59] (see, Fig. 3(c) therein). The top three panels depict the spatio-temporal evolution of the amplitude  $|\psi_n(t)|$  for the (numerically) exact KM breather (top panel), and data-driven solutions obtained by PINN (middle left panel) and S-PINN (middle right panel) over 3 periods. The bottom two panels present the spatial distribution of the amplitude  $|\psi_n(t=-0.25)|$  at  $t=-0.25$  and its temporal evolution at  $n=0$   $|\psi_0(t)|$ . Note how the regular PINN fails in this case (see the bottom panel) although we also report the disparity between the (numerically) exact solution and data-driven KM breather using S-PINN close to the right boundary (see the next to last panel).



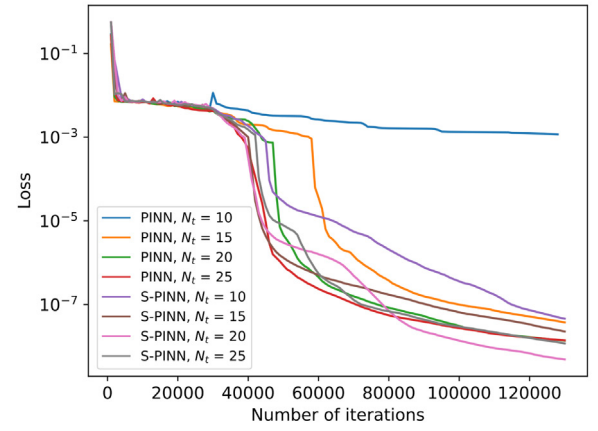
**Fig. 8.** Same as Fig. 7 but the training collocation points are sampled only from the first quadrant  $\bar{\Omega}_T$  of the computation domain [cf. Section 4.2]. The format of all panels is the same as the one of Fig. 7. We note that the disparity that was shown in the bottom panel of Fig. 7 disappears in this case, thus rendering S-PINN to be quite robust in constructing data-driven solutions solely based on numerical data.

## 5. Conclusions and future work

In the present work we have revisited the topic of PINNs that has been extensively considered recently in the context of dispersive nonlinear media and, particularly, their rogue wave solutions.



(a) Loss history with training points sampled from the entire domain  $\Omega_T$ .



(b) Loss history with training points sampled from the first quadrant  $\bar{\Omega}_T$ .

**Fig. 9.** (Training) loss history for PINN and S-PINN in learning KM breathers with an oscillatory background [cf. Section 4.3]. Panel (a): training collocation points are sampled from the entire domain  $\Omega_T$ . The training loss of S-PINN is still slowly decreasing as the optimizer terminates, which explains the (nonintuitive) deteriorating accuracy of S-PINN when the number of training samples  $N_t$  increases [cf. Table 5]. Panel (b): training collocation points are sampled from the first quadrant  $\bar{\Omega}_T$  of the computation domain. Both models have fast decaying training loss, but only S-PINN learns a generalizable solution beyond the convex hull of the training samples after enforcing physical symmetry [cf. Fig. 8].

We have opted to introduce here two elements of novelty. One of them is the consideration of a nonlinear dynamical lattice model in the form of the important integrable paradigm of the Ablowitz–Ladik system. More important from the methodological point of view is the incorporation of the underlying model symmetries, such as parity and time-reversal. In that vein, the formulation of equivariant neural networks provided a natural avenue for extending standard PINNs to the herein proposed S-PINNs, where S stands for symmetry. This extension was systematically shown to be superior to regular PINNs by typically one or in some cases more orders of magnitude for different solutions within our model of choice.

Nevertheless, we could identify (recently obtained numerically) case examples where both methodologies present limitations. Such nanopteronic solutions constitute natural possibilities for developing extensions of the present work, although it should be noted that S-PINNs outperform regular PINNs in this case too. Of course, we remain astutely aware of the fact that in the underlying model considered, in addition to parity and time-reversal symmetries, there exist additional symmetries, indeed infinitely

many of them. Hence, the incorporation of corresponding constraints, especially ones related to physical symmetries (e.g., U(1) invariance associated with mass conservation etc.) may be of particular further interest towards S-PINN extensions. Nevertheless, our motivation herein also stemmed from the broad relevance of these symmetries (parity and time-reversal) in discrete and continuum systems alike.

### CRedit authorship contribution statement

**Wei Zhu:** Software, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. **Wesley Khademi:** Software, Investigation, Visualization. **Efstathios G. Charalampidis:** Investigation, Methodology, Writing – review & editing. **Panayotis G. Kevrekidis:** Conceptualization, Methodology, Writing – review & editing, Supervision.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

WZ was partially supported by the US National Science Foundation under Grant Nos. DMS-2052525 and DMS-2140982. This material is also based upon work supported by the US National Science Foundation under Grant Nos. DMS-1809074 and PHY-2110030 (P.G.K.).

### Appendix A. Proof of Theorem 1

**Proof.** The sufficiency of Eqs. (17), (18), and (19) is easy to verify, and we only prove them also being necessary to achieve equivariance [cf. Eq. (15)]. To simplify notation, we are dropping the layer index ( $l$ ) in  $\tilde{\mathbf{W}}^{(l)}$ .

- When  $l = 1$ : any  $\tilde{\Phi}_1 \in \text{Hom}(\mathcal{F}_0, \mathcal{F}_1)$  is of the form

$$[\tilde{\Phi}_1(n, t)](\mathbf{g}) = \tilde{\mathbf{W}}(\mathbf{g}) \begin{bmatrix} n \\ t \end{bmatrix} = [\tilde{\mathbf{W}}_1(\mathbf{g}), \tilde{\mathbf{W}}_2(\mathbf{g})] \begin{bmatrix} n \\ t \end{bmatrix}, \quad (\text{A.1})$$

where  $\tilde{\mathbf{W}}(\mathbf{g}) = [\tilde{\mathbf{W}}_1(\mathbf{g}), \tilde{\mathbf{W}}_2(\mathbf{g})] \in \mathbb{R}^{D_1 \times 2}$ ,  $\forall \mathbf{g} = (g_1, g_2) \in G$ . We thus have, for any  $\tilde{\mathbf{g}} = (\tilde{g}_1, \tilde{g}_2) \in G$ ,

$$T_{\tilde{\mathbf{g}}}^{\mathcal{F}_1}[\tilde{\Phi}_1(n, t)](\mathbf{g}) = [\tilde{\mathbf{W}}_1(\mathbf{g} - \tilde{\mathbf{g}}), \tilde{\mathbf{W}}_2(\mathbf{g} - \tilde{\mathbf{g}})] \begin{bmatrix} n \\ t \end{bmatrix}, \quad (\text{A.2})$$

$$[\tilde{\Phi}_1 T_{\tilde{\mathbf{g}}}^{\mathcal{F}_0}(n, t)](\mathbf{g}) = [\tilde{\mathbf{W}}_1(\mathbf{g}), \tilde{\mathbf{W}}_2(\mathbf{g})] \begin{bmatrix} (-1)^{\tilde{g}_1} n \\ (-1)^{\tilde{g}_2} t \end{bmatrix}. \quad (\text{A.3})$$

Setting  $\tilde{\mathbf{g}} = \mathbf{g}$ , we see that the necessary condition for equivariance described by Eq. (15) to hold when  $l = 1$  is

$$[\tilde{\mathbf{W}}_1(\mathbf{g}), \tilde{\mathbf{W}}_2(\mathbf{g})] = [(-1)^{\tilde{g}_1} \tilde{\mathbf{W}}_1(\mathbf{0}), (-1)^{\tilde{g}_2} \tilde{\mathbf{W}}_2(\mathbf{0})], \quad (\text{A.4})$$

i.e., Eq. (17) holds for  $[\mathbf{W}_1, \mathbf{W}_2] = [\tilde{\mathbf{W}}_1(\mathbf{0}), \tilde{\mathbf{W}}_2(\mathbf{0})]$ .

- When  $1 < l < L$ : any  $\tilde{\Phi}_l \in \text{Hom}(\mathcal{F}_{l-1}, \mathcal{F}_l)$  is of the form

$$[\tilde{\Phi}_l f](\mathbf{g}) = \sum_{\mathbf{g}' \in G} \tilde{\mathbf{W}}(\mathbf{g}, \mathbf{g}') f(\mathbf{g}'), \quad \forall f \in \mathcal{F}_{l-1}, \quad (\text{A.5})$$

where  $\tilde{\mathbf{W}}(\mathbf{g}, \mathbf{g}') \in \mathbb{R}^{D_l \times D_{l-1}}$ . We thus have, for any  $\tilde{\mathbf{g}} \in G$ ,

$$\begin{aligned} T_{\tilde{\mathbf{g}}}^{\mathcal{F}_l}[\tilde{\Phi}_l f](\mathbf{g}) &= [\tilde{\Phi}_l f](\mathbf{g} - \tilde{\mathbf{g}}) = \sum_{\mathbf{g}' \in G} \tilde{\mathbf{W}}(\mathbf{g} - \tilde{\mathbf{g}}, \mathbf{g}') f(\mathbf{g}'), \\ [\tilde{\Phi}_l T_{\tilde{\mathbf{g}}}^{\mathcal{F}_{l-1}} f](\mathbf{g}) &= \sum_{\mathbf{g}' \in G} \tilde{\mathbf{W}}(\mathbf{g}, \mathbf{g}') f(\mathbf{g}' - \tilde{\mathbf{g}}). \end{aligned} \quad (\text{A.6})$$

Setting  $\tilde{\mathbf{g}} = \mathbf{g}$ , we have  $\tilde{\mathbf{W}}(\mathbf{g}, \mathbf{g}') = \tilde{\mathbf{W}}(\mathbf{g} - \mathbf{g}', \mathbf{0})$ , which proves Eq. (18) after setting  $\tilde{\mathbf{W}}(\mathbf{g}) = \tilde{\mathbf{W}}(\mathbf{g}, \mathbf{0})$ .

- When  $l = L$ : any  $\tilde{\Phi}_L \in \text{Hom}(\mathcal{F}_{L-1}, \mathcal{F}_L)$  is of the form

$$\tilde{\Phi}_L f = \left[ \sum_{\mathbf{g} \in G} \tilde{\mathbf{W}}_1(\mathbf{g})^T f(\mathbf{g}), \sum_{\mathbf{g} \in G} \tilde{\mathbf{W}}_2(\mathbf{g})^T f(\mathbf{g}) \right]^T, \quad (\text{A.7})$$

where  $\tilde{\mathbf{W}}_1(\mathbf{g}), \tilde{\mathbf{W}}_2(\mathbf{g}) \in \mathbb{R}^{D_{L-1}}$ ,  $\forall \mathbf{g} \in G$ . We thus have, for any  $\tilde{\mathbf{g}} = (\tilde{g}_1, \tilde{g}_2) \in G$ ,

$$T_{\tilde{\mathbf{g}}}^{\mathcal{F}_L}[\tilde{\Phi}_L f] = \begin{bmatrix} \sum_{\mathbf{g} \in G} \tilde{\mathbf{W}}_1(\mathbf{g})^T f(\mathbf{g}) \\ (-1)^{\tilde{g}_2} \sum_{\mathbf{g} \in G} \tilde{\mathbf{W}}_2(\mathbf{g})^T f(\mathbf{g}) \end{bmatrix}, \quad (\text{A.8})$$

$$\tilde{\Phi}_L [T_{\tilde{\mathbf{g}}}^{\mathcal{F}_{L-1}} f] = \begin{bmatrix} \sum_{\mathbf{g} \in G} \tilde{\mathbf{W}}_1(\mathbf{g})^T f(\mathbf{g} - \tilde{\mathbf{g}}) \\ \sum_{\mathbf{g} \in G} \tilde{\mathbf{W}}_2(\mathbf{g})^T f(\mathbf{g} - \tilde{\mathbf{g}}) \end{bmatrix}. \quad (\text{A.9})$$

In order for Eqs. (A.8) and (A.9) to be equal, we need,  $\forall \mathbf{g} \in G$ ,

$$\tilde{\mathbf{W}}_1(\mathbf{g}) = \tilde{\mathbf{W}}_1(\mathbf{0}), \quad \tilde{\mathbf{W}}_2(\mathbf{g}) = (-1)^{\tilde{g}_2} \tilde{\mathbf{W}}_2(\mathbf{0}). \quad (\text{A.10})$$

We thus have Eq. (19) after setting  $\mathbf{W}_1 = \tilde{\mathbf{W}}_1(\mathbf{0}), \mathbf{W}_2 = \tilde{\mathbf{W}}_2(\mathbf{0})$ .  $\square$

### Appendix B. Implementation

S-PINNs can be implemented as standard feed-forward NNs. Indeed, we first lexicographically order the group  $G = \mathbb{Z}_2 \times \mathbb{Z}_2$  into

$$G = \{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4\} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}. \quad (\text{B.1})$$

The hidden feature space  $\mathcal{F}_l = (\mathbb{R}^{D_l})^G$  can thus be identified with  $\mathbb{R}^{4D_l}$ . More specifically, with a slight abuse of notation, a feature  $\mathcal{F}_l \ni f^{(l)} : G \rightarrow \mathbb{R}^{D_l}$  can also be viewed as a vector  $f^{(l)}$  in  $\mathbb{R}^{4D_l}$ :

$$f^{(l)} \equiv [f^{(l)}(0, 0)^T, f^{(l)}(0, 1)^T, f^{(l)}(1, 0)^T, f^{(l)}(1, 1)^T]^T. \quad (\text{B.2})$$

Under such identification, given the input  $(n, t) \in \mathbb{Z} \times \mathbb{R}$ , its first-layer feature before nonlinearity  $f^{(1)} = \Phi_1(n, t) \in \mathcal{F}_1 \cong \mathbb{R}^{4D_1}$  is obtained via the following affine transformation

$$f^{(1)} = \tilde{\mathbf{W}}^{(1)} \begin{bmatrix} n \\ t \end{bmatrix} + \tilde{b}^{(1)}, \quad (\text{B.3})$$

where the weight matrix  $\tilde{\mathbf{W}}^{(1)} \in \mathbb{R}^{4D_1 \times 2}$  and bias vector  $\tilde{b}^{(1)} \in \mathbb{R}^{4D_1}$  are assembled from the trainable weights  $\mathbf{W}^{(1)} = [\mathbf{W}_1^{(1)}, \mathbf{W}_2^{(1)}] \in \mathbb{R}^{D_1 \times 2}$  and bias  $b^{(1)} \in \mathbb{R}^{D_1}$  based on Eqs. (17) and (25):

$$\tilde{\mathbf{W}}^{(1)} = \begin{bmatrix} \mathbf{W}_1^{(1)} & \mathbf{W}_2^{(1)} \\ \mathbf{W}_1^{(1)} & -\mathbf{W}_2^{(1)} \\ -\mathbf{W}_1^{(1)} & \mathbf{W}_2^{(1)} \\ -\mathbf{W}_1^{(1)} & -\mathbf{W}_2^{(1)} \end{bmatrix}, \quad \tilde{b}^{(1)} = \begin{bmatrix} b^{(1)} \\ b^{(1)} \\ b^{(1)} \\ b^{(1)} \end{bmatrix}; \quad (\text{B.4})$$

in particular, the trainable parameters  $\mathbf{W}^{(1)}$  and  $b^{(1)}$  are shared within the group  $G$  during the matrix assembly Eq. (B.4).

Similarly, for intermediate layers  $1 < l < L$ , the affine maps  $\Phi_l : \mathbb{R}^{4D_{l-1}} \cong \mathcal{F}_{l-1} \rightarrow \mathcal{F}_l \cong \mathbb{R}^{4D_l}$  (25) (18) are obtained from

$$\Phi_l f^{(l-1)} = \tilde{\mathbf{W}}^{(l)} f^{(l-1)} + \tilde{b}^{(l)}, \quad (\text{B.5})$$

where

$$\tilde{\mathbf{W}}^{(l)} = \begin{bmatrix} \mathbf{W}^{(l)}(0, 0) & \mathbf{W}^{(l)}(0, 1) & \mathbf{W}^{(l)}(1, 0) & \mathbf{W}^{(l)}(1, 1) \\ \mathbf{W}^{(l)}(0, 1) & \mathbf{W}^{(l)}(0, 0) & \mathbf{W}^{(l)}(1, 1) & \mathbf{W}^{(l)}(1, 0) \\ \mathbf{W}^{(l)}(1, 0) & \mathbf{W}^{(l)}(1, 1) & \mathbf{W}^{(l)}(0, 0) & \mathbf{W}^{(l)}(0, 1) \\ \mathbf{W}^{(l)}(1, 1) & \mathbf{W}^{(l)}(1, 0) & \mathbf{W}^{(l)}(0, 1) & \mathbf{W}^{(l)}(0, 0) \end{bmatrix}$$

$$\tilde{\mathbf{b}}^{(l)} = \begin{bmatrix} \mathbf{b}^{(l)} \\ \mathbf{b}^{(l)} \\ \mathbf{b}^{(l)} \\ \mathbf{b}^{(l)} \end{bmatrix} \in \mathbb{R}^{4D_l} \quad (\text{B.6})$$

are assembled from the trainable parameters  $\mathbf{W}^{(l)} = \{\mathbf{W}^{(l)}(\mathbf{g}) \in \mathbb{R}^{D_l \times D_{l-1}} : \mathbf{g} \in G\}$  and  $\mathbf{b}^{(l)} \in \mathbb{R}^{D_l}$ .

Finally, the last-layer linear map  $\Phi_L : \mathbb{R}^{4D_{L-1}} \cong \mathcal{F}_{L-1} \rightarrow \mathcal{F}_L = \mathbb{R}^2$  (25) (19) can be viewed as

$$\Phi_L f^{(L-1)} = \tilde{\mathbf{W}}^{(L)} f^{(L-1)}, \quad (\text{B.7})$$

where

$$\tilde{\mathbf{W}}^{(L)} = \begin{bmatrix} \mathbf{W}_1^{(L)T} & \mathbf{W}_1^{(L)T} & \mathbf{W}_1^{(L)T} & \mathbf{W}_1^{(L)T} \\ \mathbf{W}_2^{(L)T} & -\mathbf{W}_2^{(L)T} & \mathbf{W}_2^{(L)T} & -\mathbf{W}_2^{(L)T} \end{bmatrix} \quad (\text{B.8})$$

is derived from the trainable weight matrix  $\mathbf{W}^{(L)} = [\mathbf{W}_1^{(L)}, \mathbf{W}_2^{(L)}] \in \mathbb{R}^{D_{L-1} \times 2}$ . As mentioned in Section 3.1.2, nonlinearity  $\sigma = \tanh : \mathbb{R} \rightarrow \mathbb{R}$  (26) is applied after each affine map (except for the last one) on every entry of the feature  $f^{(l)} \in \mathbb{R}^{4D_l}$ ,  $1 < l < L$ .

## References

- [1] S. Haver, A possible freak wave event measured at the Draupner jacket January 1 1995, in: *Rogue Waves 2004 : Proceedings of A Workshop Organized By Ifremer and Held in Brest, France, 2004*, 2004, pp. 1–8, URL [http://www.ifremer.fr/web-com/stw2004/rw/fullpapers/walk\\_on\\_haver.pdf](http://www.ifremer.fr/web-com/stw2004/rw/fullpapers/walk_on_haver.pdf).
- [2] D.A. Walker, P.H. Taylor, R.E. Taylor, The shape of large surface waves on the open sea and the Draupner New Year wave, *Appl. Ocean Res.* 26 (3–4) (2004) 73–83, <http://dx.doi.org/10.1016/j.apor.2005.02.001>.
- [3] T.A. Adcock, P.H. Taylor, S. Yan, Q.W. Ma, P.A. Janssen, Did the draupner wave occur in a crossing sea? *Proc. R. Soc. A* 467 (2134) (2011) 3004–3021, <http://dx.doi.org/10.1098/rspa.2011.0049>.
- [4] N. Mori, P.C. Liu, Analysis of freak wave measurements in the Sea of Japan, *Ocean Eng.* 29 (11) (2002) 1399–1414, [http://dx.doi.org/10.1016/S0029-8018\(01\)00073-7](http://dx.doi.org/10.1016/S0029-8018(01)00073-7).
- [5] C. Kharif, E. Pelinovsky, Physical mechanisms of the rogue wave phenomenon, *Eur. J. Mech. B/Fluids* 22 (6) (2003) 603–634, <http://dx.doi.org/10.1016/j.euromechflu.2003.09.002>.
- [6] A. Chabchoub, N.P. Hoffmann, N. Akhmediev, Rogue wave observation in a water wave tank, *Phys. Rev. Lett.* 106 (20) (2011) 204502, <http://dx.doi.org/10.1103/PhysRevLett.106.204502>, URL <https://link.aps.org/doi/10.1103/PhysRevLett.106.204502>.
- [7] A. Chabchoub, N. Hoffmann, M. Onorato, N. Akhmediev, Super rogue waves: Observation of a higher-order breather in water waves, *Phys. Rev. X* 2 (1) (2012) 2–7, <http://dx.doi.org/10.1103/PhysRevX.2.011015>.
- [8] M.L. McAllister, S. Draycott, T.A. Adcock, P.H. Taylor, T.S. Van Den Bremer, Laboratory recreation of the draupner wave and the role of breaking in crossing seas, *J. Fluid Mech.* 860 (2018) 767–786, <http://dx.doi.org/10.1017/jfm.2018.886>.
- [9] G. Xu, A. Chabchoub, D.E. Pelinovsky, B. Kibler, Observation of modulation instability and rogue breathers on stationary periodic waves, *Phys. Rev. Res.* 2 (3) (2020) 33528, <http://dx.doi.org/10.1103/physrevresearch.2.033528>, URL <https://link.aps.org/doi/10.1103/PhysRevResearch.2.033528>.
- [10] D.R. Solli, C. Ropers, P. Koonath, B. Jalali, Optical rogue waves, *Nature* 450 (7172) (2007) 1054–1057, <http://dx.doi.org/10.1038/nature06402>.
- [11] J.M. Dudley, F. Dias, M. Erkintalo, G. Genty, Instabilities, breathers and rogue waves in optics, *Nature Photonics* 8 (10) (2014) 755–764, <http://dx.doi.org/10.1038/nphoton.2014.220>, URL <https://www.nature.com/articles/nphoton.2014.220>.
- [12] B. Frisquet, B. Kibler, P. Morin, F. Baronio, M. Conforti, G. Millot, S. Wabnitz, Optical dark rogue wave, *Sci. Rep.* 6 (1) (2016) 1–9, <http://dx.doi.org/10.1038/srep20785>.
- [13] A. Tikan, C. Bille, G. El, A. Tovbis, M. Bertola, T. Sylvestre, F. Gustave, S. Randoux, G. Genty, P. Suret, J.M. Dudley, Universality of the peregrine soliton in the focusing dynamics of the cubic nonlinear Schrödinger equation, *Phys. Rev. Lett.* 119 (3) (2017) 33901, <http://dx.doi.org/10.1103/PhysRevLett.119.033901>, URL <https://link.aps.org/doi/10.1103/PhysRevLett.119.033901>.
- [14] M.S. Ruderman, Freak waves in laboratory and space plasmas, *Eur. Phys. J.: Special Topics* 185 (1) (2010) 57–66, <http://dx.doi.org/10.1140/epjst/e2010-01238-7>.
- [15] R. Sabry, W.M. Moslem, P.K. Shukla, Freak waves in white dwarfs and magnetars, *Phys. Plasmas* 19 (12) (2012) <http://dx.doi.org/10.1063/1.4772058>.
- [16] A.S. Bains, B. Li, L.D. Xia, Kinetic Alfvén solitary and rogue waves in superthermal plasmas, *Phys. Plasmas* 21 (3) (2014) <http://dx.doi.org/10.1063/1.4869464>, arXiv:1403.3745.
- [17] R.E. Tolba, W.M. Moslem, N.A. El-Bedwehy, S.K. El-Labany, Evolution of rogue waves in dusty plasmas, *Phys. Plasmas* 22 (4) (2015) <http://dx.doi.org/10.1063/1.4918706>.
- [18] E.G. Charalampidis, J. Cuevas-Maraver, D.J. Frantzeskakis, P.G. Kevrekidis, Rogue waves in ultracold bosonic seas, *Romanian Rep. Phys.* 70 (1) (2018) 1–25, arXiv:1609.01798.
- [19] M. Onorato, S. Residori, U. Bortolozzo, A. Montina, F.T. Arecchi, Rogue waves and their generating mechanisms in different physical contexts, *Phys. Rep.* 528 (2) (2013) 47–89, <http://dx.doi.org/10.1016/j.physrep.2013.03.001>.
- [20] K. Dudley, G. Genty, A. Mussot, A. Chabchoub, F. Dias, Rogue waves and analogies in optics and oceanography, *Nat. Rev. Phys.* 1 (2019) 675–689, <http://dx.doi.org/10.1038/s42254-019-0100-0>.
- [21] A. R. Osborne, *Nonlinear Ocean Wave and the Inverse Scattering Transform*, Elsevier, Amsterdam, 2010, <http://dx.doi.org/10.1016/b978-012613760-6/50033-4>, URL [https://books.google.com/books?hl=ja&lr=lang\\_ja%7Clang\\_en&id=wdmsn9icd7YC&oi=fnd&pg=PP1&ots=65JPoQTPfC&sig=BNak3lCc1QZjI7gVLuTulLSK-wE](https://books.google.com/books?hl=ja&lr=lang_ja%7Clang_en&id=wdmsn9icd7YC&oi=fnd&pg=PP1&ots=65JPoQTPfC&sig=BNak3lCc1QZjI7gVLuTulLSK-wE).
- [22] E. Pelinovsky, C. Kharif, *Extreme Ocean Waves*, Springer International Publishing, Cham, 2016, pp. 1–236, <http://dx.doi.org/10.1007/978-3-319-21575-4>, URL <http://link.springer.com/10.1007/978-3-319-21575-4>.
- [23] M. Dissanayake, N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, *Commun. Numer. Methods. Eng.* 10 (3) (1994) 195–201.
- [24] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Netw.* 9 (5) (1998) 987–1000.
- [25] K. Rudd, S. Ferrari, A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks, *Neurocomputing* 155 (2015) 277–285.
- [26] G. Carleo, M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* 355 (6325) (2017) 602–606.
- [27] J. Han, A. Jentzen, E. Weinan, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci.* 115 (34) (2018) 8505–8510.
- [28] E. Weinan, J. Han, A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, *Commun. Math. Stat.* 5 (4) (2017) 349–380.
- [29] J. Berg, K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, *Neurocomputing* 317 (2018) 28–41.
- [30] Y. Khoo, J. Lu, L. Ying, Solving for high-dimensional committor functions using artificial neural networks, *Res. Math. Sci.* 6 (1) (2019) 1–13.
- [31] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [32] A.D. Jagtap, E. Kharazmi, G.E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems, *Comput. Methods Appl. Mech. Engrg.* 365 (2020) 113028.
- [33] L. Lu, X. Meng, Z. Mao, G.E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, *SIAM Rev.* 63 (1) (2021) 208–228.
- [34] J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.* 375 (2018) 1339–1364.
- [35] E. Weinan, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.* 6 (1) (2018) 1–12.
- [36] Y. Gu, H. Yang, C. Zhou, Selectnet: Self-paced learning for high-dimensional partial differential equations, *J. Comput. Phys.* 441 (2021) 110444.
- [37] J. Han, L. Zhang, E. Weinan, Solving many-electron Schrödinger equation using deep neural networks, *J. Comput. Phys.* 399 (2019) 108929.
- [38] J. Hermann, Z. Schätzle, F. Noé, Deep-neural-network solution of the electronic Schrödinger equation, *Nature Chem.* 12 (10) (2020) 891–897.
- [39] D. Pfau, J.S. Spencer, A.G. Matthews, W.M.C. Foulkes, Ab initio solution of the many-electron Schrödinger equation with deep neural networks, *Phys. Rev. Res.* 2 (3) (2020) 033429.
- [40] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440.
- [41] Y. Shin, J. Darbon, G.E. Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs, 2020, arXiv preprint arXiv:2004.01806.
- [42] Y. Shin, Z. Zhang, G.E. Karniadakis, Error estimates of residual minimization using neural networks for linear PDEs, 2020, arXiv preprint arXiv:2010.08019.



- [43] T. Luo, H. Yang, Two-layer neural networks for partial differential equations: Optimization and generalization theory, 2020, arXiv preprint arXiv:2006.15733.
- [44] C. Sulem, P. Sulem, The Nonlinear Schrödinger Equation: Self-Focusing and Wave Collapse, Springer-Verlag New York, 1999, <http://dx.doi.org/10.1007/b98958>, URL [https://books.google.com/books?hl=ja&lr=lang\\_ja%7Clang\\_en&id=FZ3MKMVfDQC&oi=fnd&pg=PR5&ots=P1mP41jYxZ&sig=8lgdO8z22\\_nymfQUdJgXC-dX-A](https://books.google.com/books?hl=ja&lr=lang_ja%7Clang_en&id=FZ3MKMVfDQC&oi=fnd&pg=PR5&ots=P1mP41jYxZ&sig=8lgdO8z22_nymfQUdJgXC-dX-A).
- [45] M.J. Ablowitz, B. Prinari, A.D. Trubatch, Discrete and Continuous Nonlinear Schrödinger Systems, vol. 302, Cambridge University Press, 2004.
- [46] Z. Miao, Y. Chen, Physics-informed neural network method in high-dimensional integrable systems, 2021, ArXiv E-Prints arXiv:2107.02985.
- [47] S. Lin, Y. Chen, 2012, ArXiv E-Prints arXiv:2107.01009.
- [48] L. Wang, 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 (2021) 127408, <http://dx.doi.org/10.1016/j.physleta.2021.127408>, URL <https://www.sciencedirect.com/science/article/pii/S0375960121002723>.
- [49] Y. Fang, G. Wu, Y. Wang, C.-Q. Dai, Data-driven femtosecond optical soliton excitations and parameters discovery of the high-order NLSE using the PINN, Nonlinear Dynam. 105 (2021) 603–616, <http://dx.doi.org/10.1007/s11071-021-06550-9>.
- [50] J. Pu, J. Li, Y. Chen, Solving localized wave solutions of the derivative nonlinear Schrödinger equation using an improved PINN method, Nonlinear Dynam. 105 (2021) 1723–1739.
- [51] J. Pu, W. Peng, Y. Chen, The data-driven localized wave solutions of the derivative nonlinear Schrödinger equation by using improved PINN approach, Wave Motion 107 (2021) 102823, <http://dx.doi.org/10.1016/j.wavemoti.2021.102823>, URL <https://www.sciencedirect.com/science/article/pii/S0165212521001219>.
- [52] W. Peng, J. Pu, Y. Chen, PINN deep learning for the Chen-Lee-Liu equation: Rogue wave on the periodic background, 2021, ArXiv E-Prints arXiv:2105.13027.
- [53] V.V. Konotop, J. Yang, D.A. Zezyulin, Nonlinear waves in  $\mathcal{PT}$ -symmetric systems, Rev. Modern Phys. 88 (2016) 035002, <http://dx.doi.org/10.1103/RevModPhys.88.035002>, URL <https://link.aps.org/doi/10.1103/RevModPhys.88.035002>.
- [54] D. Christodoulides, J. Yang, Parity-Time Symmetry and Its Applications, Springer Singapore, 2018, pp. 1–579, <http://dx.doi.org/10.1007/978-981-13-1247-2>, URL <https://www.springer.com/gp/book/9789811312465>.
- [55] I. Huh, E. Yang, S.J. Hwang, J. Shin, Time-reversal symmetric ODE network, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 19016–19027, URL <https://proceedings.neurips.cc/paper/2020/file/db8419f41d890df802dca330e6284952-Paper.pdf>.
- [56] P. Jin, Z. Zhang, A. Zhu, Y. Tang, G.E. Karniadakis, SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems, Neural Netw. 132 (2020) 166–179, <http://dx.doi.org/10.1016/j.neunet.2020.08.017>, URL <https://www.sciencedirect.com/science/article/pii/S0893608020303063>.
- [57] P.G. Kevrekidis, Non-linear waves in lattices: past, present, future, IMA J. Appl. Math. 76 (3) (2011) 389–423, <http://dx.doi.org/10.1093/imanat/hxr015>, arXiv:https://academic.oup.com/imanat/article-pdf/76/3/389/2257051/hxr015.pdf.
- [58] A. Ankiewicz, N. Akhmediev, J.M. Soto-Crespo, Discrete rogue waves of the Ablowitz-Ladik and Hirota equations, Phys. Rev. E 82 (2) (2010) <http://dx.doi.org/10.1103/PhysRevE.82.026602>.
- [59] J. Sullivan, E. Charalampidis, J. Cuevas-Maraver, P.G. Kevrekidis, N.I. Karachalios, Kuznetsov–Ma breather-like solutions in the Salerno model, Eur. Phys. J. Plus 135 (7) (2020) 1–12.
- [60] M.J. Ablowitz, J.F. Ladik, Nonlinear differential-difference equations and Fourier analysis, J. Math. Phys. 17 (6) (1976) 1011–1018.
- [61] M.J. Ablowitz, J.F. Ladik, Nonlinear differential-difference equations, J. Math. Phys. 16 (3) (1975) 598–603.
- [62] P. Kevrekidis, The Discrete Nonlinear Schrödinger Equation: Mathematical Analysis, Numerical Computations and Physical Perspectives, vol. 232, Springer-Verlag, 2009.
- [63] D. Cai, A.R. Bishop, N. Grønbech-Jensen, Perturbation theories of a discrete, integrable nonlinear Schrödinger equation, Phys. Rev. E 53 (1996) 4131–4136, <http://dx.doi.org/10.1103/PhysRevE.53.4131>, URL <https://link.aps.org/doi/10.1103/PhysRevE.53.4131>.
- [64] T. Kapitula, P. Kevrekidis, Stability of waves in discrete systems, Nonlinearity 14 (3) (2001) 533–566, <http://dx.doi.org/10.1088/0951-7715/14/3/306>.
- [65] B. Prinari, Discrete solitons of the focusing Ablowitz–Ladik equation with nonzero boundary conditions via inverse scattering, J. Math. Phys. 57 (8) (2016) 083510.
- [66] N.N. Akhmediev, V.M. Eleonskii, N.E. Kulagin, Exact first-order solutions of the nonlinear–Schrödinger equation, Theor. Math. Phys.; (United States) 72 (2) (1988).
- [67] T. Cohen, M. Welling, Group equivariant convolutional networks, in: International Conference on Machine Learning, PMLR, 2016, pp. 2990–2999.
- [68] T.S. Cohen, M. Geiger, M. Weiler, A general theory of equivariant CNNs on homogeneous spaces, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, URL <https://proceedings.neurips.cc/paper/2019/file/b9cfe8b6042cf759dc4c0cccb27a6737-Paper.pdf>.
- [69] R. Kondor, S. Trivedi, On the generalization of equivariance and convolution in neural networks to the action of compact groups, in: International Conference on Machine Learning, PMLR, 2018, pp. 2747–2755.
- [70] T. Cohen, M. Welling, Steerable CNNs, in: International Conference on Learning Representations, 2017.
- [71] M. Weiler, G. Cesa, General E(2)-equivariant steerable CNNs, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.
- [72] X. Cheng, Q. Qiu, R. Calderbank, G. Sapiro, RotDCF: Decomposition of convolutional filters for rotation-equivariant deep networks, in: International Conference on Learning Representations, 2019.
- [73] E. Hoogeboom, J.W. Peters, T.S. Cohen, M. Welling, HexaConv, in: International Conference on Learning Representations, 2018.
- [74] D.E. Worrall, S.J. Garbin, D. Turmukhambetov, G.J. Brostow, Harmonic networks: Deep translation and rotation equivariance, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5028–5037.
- [75] E.J. Bekkers, M. Loog, B.M. ter Haar Romeny, R. Duits, Template matching via densities on the roto-translation group, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2) (2017) 452–466.
- [76] Y. Zhou, Q. Ye, Q. Qiu, J. Jiao, Oriented response networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 519–528.
- [77] D. Marcos, M. Volpi, N. Komodakis, D. Tuia, Rotation equivariant vector field networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5048–5057.
- [78] M. Weiler, F.A. Hamprecht, M. Storath, Learning steerable filters for rotation equivariant CNNs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 849–858.
- [79] M. Weiler, M. Geiger, M. Welling, V. Boomsma, T. Cohen, 3D steerable cnns: Learning rotationally equivariant features in volumetric data, in: Advances in Neural Information Processing Systems, 2018, pp. 10381–10392.
- [80] D. Worrall, G. Brostow, Cubenet: Equivariance to 3D rotation and translation, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 567–584.
- [81] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, P. Riley, Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds, 2018, arXiv preprint arXiv:1802.08219.
- [82] T.S. Cohen, M. Geiger, J. Köhler, M. Welling, Spherical CNNs, in: International Conference on Learning Representations, 2018.
- [83] C. Esteves, C. Allen-Blanchette, A. Makadia, K. Daniilidis, Learning SO(3) equivariant representations with spherical CNNs, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 52–68.
- [84] M. Winkels, T.S. Cohen, 3D G-CNNs for pulmonary nodule detection, 2018, arXiv preprint arXiv:1804.04656.
- [85] V. Andrearczyk, J. Fageot, V. Oreiller, X. Montet, A. Depeursinge, Exploring local rotation invariance in 3D CNNs with steerable filters, in: International Conference on Medical Imaging with Deep Learning, PMLR, 2019, pp. 15–26.
- [86] A. Kanazawa, A. Sharma, D. Jacobs, Locally scale-invariant convolutional neural networks, 2014, arXiv preprint arXiv:1412.5104.
- [87] D. Marcos, B. Kellenberger, S. Loby, D. Tuia, Scale equivariance in CNNs with vector fields, 2018, arXiv preprint arXiv:1807.11783.
- [88] Y. Xu, T. Xiao, J. Zhang, K. Yang, Z. Zhang, Scale-invariant convolutional neural networks, 2014, arXiv preprint arXiv:1411.6369.
- [89] D. Worrall, M. Welling, Deep scale-spaces: Equivariance over scale, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.
- [90] I. Sosnovik, M. Szmaja, A. Smeulders, Scale-equivariant steerable networks, in: International Conference on Learning Representations, 2020.
- [91] N. Guttenberg, N. Virgo, O. Witkowski, H. Aoki, R. Kanai, Permutation-equivariant neural networks applied to dynamics prediction, 2016, arXiv preprint arXiv:1612.04530.

- [92] A. Sannai, Y. Takai, M. Cordonnier, Universal approximations of permutation invariant/equivariant functions by deep neural networks, 2019, arXiv preprint [arXiv:1903.01939](https://arxiv.org/abs/1903.01939).
- [93] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, Y.W. Teh, Set transformer: A framework for attention-based permutation-invariant neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 3744–3753.
- [94] V.G. Satorras, E. Hoogeboom, M. Welling, E (n) equivariant graph neural networks, 2021, arXiv preprint [arXiv:2102.09844](https://arxiv.org/abs/2102.09844).
- [95] N. Keriven, G. Peyré, Universal invariant and equivariant graph neural networks, Adv. Neural Inf. Process. Syst. 32 (2019) 7092–7101.
- [96] W. Fulton, J. Harris, Representation Theory: A First Course, vol. 129, Springer Science & Business Media, 2013.
- [97] F. Fuchs, D. Worrall, V. Fischer, M. Welling, SE(3)-transformers: 3D roto-translation equivariant attention networks, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 1970–1981, URL <https://proceedings.neurips.cc/paper/2020/file/15231a7ce4ba789d13b722cc5c955834-Paper.pdf>.
- [98] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018).
- [99] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR (Poster), 2015.
- [100] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1) (1989) 503–528.