2022

# Discovering Governing Equations in Discrete Systems Using PINNs

Sheikh Saqlain
*University of Massachusetts Amherst*

Wei Zhu
*University of Massachusetts Amherst*

Efstathios G. Charalampidis
*California Polytechnic State University, San Luis Obispo*

Panayotis G. Kevrekidis
*University of Massachusetts Amherst*, kevrekid@math.umass.edu

# Discovering Governing Equations in Discrete Systems Using PINNs

Sheikh Saqlain[1], Wei Zhu[1], Efstathios G. Charalampidis[2], and Panayotis G. Kevrekidis[1]

[1]Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, MA 01003-4515, USA
[2]Mathematics Department, California Polytechnic State University, San Luis Obispo, CA 93407-0403, USA

**Abstract**

Sparse identification of nonlinear dynamical systems is a topic of continuously increasing significance in the dynamical systems community. Here we explore it at the level of lattice nonlinear dynamical systems of many degrees of freedom. We illustrate the ability of a suitable adaptation of Physics-Informed Neural Networks (PINNs) to solve the inverse problem of parameter identification in such discrete, high-dimensional systems inspired by physical applications. The methodology is illustrated in a diverse array of examples including real-field ones ($\phi^4$ and sine-Gordon), as well as complex-field (discrete nonlinear Schrödinger equation) and going beyond Hamiltonian to dissipative cases (the discrete complex Ginzburg-Landau equation). Both the successes, as well as some limitations of the method are discussed along the way.

## 1 Introduction

While the study of nonlinear partial differential equations (PDEs) captures the lion's share of modeling efforts in physical, chemical and biological problems, the study of nonlinear dynamical lattices has received growing attention over the last decades [1, 2, 3, 4]. To some degree, this interest stems from the consideration of discretization methods for simulating PDEs. However, arguably, the most appealing aspect of such lattice problems is that they naturally emerge as the suitable model in systems where there is a degree of "granularity"/lattice structure. This may stem from waveguides and their arrays in nonlinear optics [5], or in experiments of micromechanical oscillator arrays [6] as well as lattice nonlinear electrical circuits [7]. It may arise in material science systems [8, 9], in antiferromagnetic [10], or more generally anharmonic [11, 12] crystals, in superconducting settings of Josephson-junction ladders [13, 14] or in biological models of DNA base pairs [15, 16]. Such lattice models may also be effective ones, emulating the periodic variation of optical lattices in atomic condensates [17].

On the other hand, over the past few years there has been an explosion of interest in the use of data-driven techniques towards the study of physical phenomena and the development, as well as identification of relevant models [18]. Among the most dominant methodologies in that regard for

the solution of both inverse and forward problems in PDEs have been methodologies such as PINNs (Physics-Informed Neural Networks) [19], and the subsequent extension of DeepXDE [20], as well as the SINDY (sparse identification of nonlinear systems) method of [21], sparse optimization in [22], meta-learning [23], and neural operators [24]. There have been numerous variations and extensions of these approaches in a wide range of problems (a small subset of which, e.g., contain [25, 26, 27, 28, 29]). Yet, it can be argued that these approaches have been, by and large, limited to continuum PDE problems and the emergent aspect of nonlinear dynamical lattices has been somewhat overlooked.

Here, we build on the earlier work of some of the present authors [30], which was aiming to build in the neural networks more of the physical structure of the underlying problem (in that case through symmetries, while other authors have also enforced, e.g., the symplectic structure of potential underlying systems [31]). Our emphasis in the present work is to adapt methods of the above type, most notably PINNs, to nonlinear dynamical lattices. In particular, we will select a sequence of progressively more complex yet physically relevant examples and seek to leverage the above computational methodology, albeit now in an inherently discrete, high-dimensional setting. By high-dimensional here, we refer to the number of degrees of freedom (and not the spatial dimension of the problem). Our aim will be to solve the inverse problem of the identification of linear and nonlinear coefficients of the models building progressively from simpler to more complex. We will start from a real $\phi^4$ discrete nonlinear Klein-Gordon system [32] and subsequently move to the complex variant of the model, namely the discrete nonlinear Schrödinger (DNLS) model [33]. We will subsequently explore an example bearing a different type of complexity where the nonlinearity is not a pure power law, but rather a sinusoidal one in the form of the Frenkel-Kontorova [34] or discrete sine-Gordon [35] nonlinearity. This will serve to showcase some of the challenges and limitations of the approach. Finally, we will extend considerations beyond the Hamiltonian class of examples to the discrete variant [36, 37] of the Ginzburg-Landau equation [38], a topic that continues to be of wide interest in its own right as evidenced in the recent review of [39].

Our presentation will be structured as follows. In Section 2, we will provide some of the mathematical background of the problem, both at the level of the dynamical models under consideration (Sec. 2.1) and as concerns the PINN approach (Sec. 2.2). Then, upon explaining how to adapt the discovery of the governing equation to nonlinear dynamical lattices in Sec. 3, we present our numerical experiments in Sec. 4. Finally, in Sec. 5, we summarize our findings and present a number of possibilities for future studies.

# 2 Background

## 2.1 Discrete nonlinear lattices

In this work, we consider a variety of 1D discrete nonlinear lattices consisting of $N$ nodes. In all the cases that we will focus on hereafter, $u_n(t)$ (which can be real or complex, depending on the model) will correspond to a dynamical variable with $n = 1, \ldots, N$. We start our presentation of the models by considering first the discrete $\phi^4$ model [32]

$$\ddot{u}_n = C(u_{n+1} + u_{n-1} - 2u_n) + 2(u_n - u_n^3), \quad u_n \in \mathbb{R}, \tag{1}$$

where the overdot stands for the temporal derivative of $u_n$, and $C = 1/h^2 (> 0)$ effectively represents the coupling constant with $h$ representing the lattice spacing between adjacent nodes. It should be further noted that neighboring sites in Eq. (1) are coupled due to the presence of the $(u_{n+1} + u_{n-1} - 2u_n)$ discrete Laplacian term therein, and the strength of the coupling is dictated by the magnitude of $C$. That is, a large value of $C$, i.e., $C \gg 1$ or equivalently $h \ll 1$ signifies that Eq. (1) is close to the continuum $\phi^4$ limit, whereas a small value of $C$ will result in a highly discrete system. Moreover, this coupling term (involving $C$), that will be ubiquitous in all of the models that we consider herein, emanates from the discretization of the Laplacian operator in 1D by using a centered, second-order accurate finite difference scheme. This provides a vein along which the model can interpolate between the so-called anti-continuum limit [1] of $C = 0$ and the continuum limit of the respective PDE. Eq. 1 is a model, variants of which have been useful towards understanding solitary wave dynamics in a simpler, real lattice nonlinear system [40, 41]. We thus use it as a preamble to studying the complex DNLS variant of the model.

Indeed, we subsequently focus on the well-known yet physically relevant, discrete nonlinear Schrödinger (DNLS) equation [33] with a focusing (cubic) nonlinearity:

$$i\dot{u}_n = -C(u_{n+1} + u_{n-1} - 2u_n) - |u_n|^2 u_n, \quad u_n \in \mathbb{C}. \tag{2}$$

Here, we allow the relevant field representing, e.g., the envelope of the electric field along an optical waveguide array [5] or the quantum-mechanical wavefunction along the nodes of a deep optical lattice [17], to be complex.

Another intriguing example consists of the discrete sine-Gordon (DsG) [35], also known in dislocation theory as Frenkel-Kontorova model [34]:

$$\ddot{u}_n = C(u_{n+1} + u_{n-1} - 2u_n) - \sin(u_n), \quad u_n \in \mathbb{R}. \tag{3}$$

This model, similarly to Eq. (1) admits kink-like solutions. However, it also has a key distinguishing feature from the former. Namely, it bears a transcendental nonlinear function, one that cannot be expressed as a simple power law. Indeed, the difficulty to represent such a simple pendulum (unless further, e.g., Hamiltonian structure of the problem is built in the sparse identification approach) has been previously documented, e.g., as concerns SINDY in [42].

Finally, the other fundamental model of interest in the present work is the discrete, complex Ginzburg-Landau (DCGL) equation:

$$\dot{u}_n = (1+i)C(u_{n+1} + u_{n-1} - 2u_n) - (1-i)|u_n|^2 u_n + u_n, \quad u_n \in \mathbb{C}, \tag{4}$$

with a cubic nonlinearity [36]. The DCGL can be considered as a (dissipative) perturbation of the DNLS [cf. Eq. (2)]. Such settings are of interest in the same contexts as DNLS when dissipative perturbations are present, as, e.g., in experimental studies such as that of [43] in optics or the one of [44] in atomic Bose-Einstein condensates.

A recap of the principal models of interest can be found in the Table 1. The table contains not only the mathematical form of each of the models but also the initial conditions (ICs) that are used therein in order to perform the model training (cf. Section 4.1). We conclude this section by mentioning the boundary conditions (BCs) that we will employ for all the above models. In

| Model | Equation | IC($u$) |
|---|---|---|
| DNLS | $i\dot{u}_n = -C(u_{n+1} + u_{n-1} - 2u_n) - \lvert u_n\rvert^2 u_n$ | $e^{-x_n^2}$ |
| DCGL | $\dot{u}_n = (1+i)C(u_{n+1} + u_{n-1} - 2u_n) - (1-i)\lvert u_n\rvert^2 u_n + u_n$ | $\tanh x_n \exp\left(i \ln\left(\cosh x_n\right)\right)$ |
| Discrete $\phi^4$ | $\ddot{u}_n = C(u_{n+1} + u_{n-1} - 2u_n) + 2(u_n - u_n^3)$ | $\tanh\left(\frac{x_n}{\sqrt{1-v^2}}\right)$ |
| DsG | $\ddot{u}_n = C(u_{n+1} + u_{n-1} - 2u_n) - \sin\left(u_n\right)$ | $4\arctan\left(\exp\left(\frac{x_n}{\sqrt{1-v^2}}\right)\right)$ |

Table 1: Discrete nonlinear lattices that are considered in this work together with the respective initial conditions. Note that $x_n$ are grid points with $n = 1, \ldots, N$, taken uniformly from the interval $\left[-\frac{N}{2\sqrt{C}}, \frac{N}{2\sqrt{C}}\right]$.

particular, we impose free BCs at both ends of the lattice, i.e., $u_0 = u_1$ and $u_{N+1} = u_N$. These BCs can be thought of as the discrete analogues of zero Neumann BCs in the continuum limit, and emanate through the discretization of the latter through first-order accurate, forward and backward finite difference formulas, respectively. Having discussed about the models of interest herein, we now turn into a brief overview of Physics-Informed Neural Networks (PINNs).

### 2.2 Physics-Informed Neural Networks

Since their introduction by Raissi et al. [19], PINNs have garnered growing attention from the scientific machine learning community due to their flexible and gridless design in data-driven modeling of forward and inverse problems. Consider, for instance, the following parametrized PDE:

$$\begin{cases} u_t = \mathcal{N}(u; \lambda), & \boldsymbol{x} \in \Omega, t \in [0, T], \\ u(\boldsymbol{x}, 0) = g(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ \mathcal{B}u(\boldsymbol{x}, t) = h(\boldsymbol{x}, t), & \boldsymbol{x} \in \partial\Omega, t \in [0, T], \end{cases} \tag{5}$$

where $u(\boldsymbol{x}, t)$ is the unknown, $\mathcal{N}(\cdot; \lambda)$ is a (spatial) nonlinear differential operator parametrized by $\lambda$, and $\mathcal{B}$ is an operator associated with a specific BC.

In *forward problems*, i.e., when the model parameter $\lambda$ is fixed and given, one aims to derive the (numerical) solution $u(\boldsymbol{x}, t)$ of Eq. (5) with the specified initial and boundary conditions. A PINN for Eq. (5) in this setting is a neural network ansatz $\hat{u}(\boldsymbol{x}, t; \boldsymbol{\theta})$ that serves as a surrogate of the solution $u(\boldsymbol{x}, t)$, where $\boldsymbol{\theta}$ is the collection of all trainable parameters of the neural network, e.g., weights and biases of a fully-connected feed-forward PINN. The optimal solution $\hat{u}(\boldsymbol{x}, t; \boldsymbol{\theta}^*)$ is searched such that the constraints imposed by the PDE and the initial/boundary conditions are (approximately) satisfied. More specifically, let $\mathcal{T}_\mathcal{N} \subset \Omega \times [0, T]$, $\mathcal{T}_g \subset \Omega \times \{t = 0\}$ and $\mathcal{T}_h \subset \partial\Omega \times [0, T]$ be three finite collections of scattered "training" points sampled from their corresponding regions. The discrepancy between $\hat{u}(\boldsymbol{x}, t; \boldsymbol{\theta})$ and the constraints in Eq. (5) is measured through the following loss function $\mathcal{L}(\boldsymbol{\theta}; \mathcal{T}_\mathcal{N}, \mathcal{T}_g, \mathcal{T}_h)$ defined as a weighted sum of the discrete $l^2$ norms of the residuals for the PDE and the initial/boundary conditions:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{T}_\mathcal{N}, \mathcal{T}_g, \mathcal{T}_h) \coloneqq w_\mathcal{N}\mathcal{L}_\mathcal{N}(\boldsymbol{\theta}; \mathcal{T}_\mathcal{N}) + w_g\mathcal{L}_g(\boldsymbol{\theta}; \mathcal{T}_g) + w_h\mathcal{L}_h(\boldsymbol{\theta}; \mathcal{T}_h), \tag{6}$$

where

$$\mathcal{L}_{\mathcal{N}}(\boldsymbol{\theta};\mathcal{T}_{\mathcal{N}}) = \frac{1}{|\mathcal{T}_{\mathcal{N}}|}\sum_{(\boldsymbol{x},t)\in\mathcal{T}_{\mathcal{N}}} |\hat{u}_t(\boldsymbol{x},t;\boldsymbol{\theta}) - \mathcal{N}(\hat{u};\lambda)(\boldsymbol{x},t;\boldsymbol{\theta})|^2\,, \tag{7}$$

$$\mathcal{L}_{g}(\boldsymbol{\theta};\mathcal{T}_{g}) = \frac{1}{|\mathcal{T}_{g}|}\sum_{(\boldsymbol{x},0)\in\mathcal{T}_{g}} |\hat{u}(\boldsymbol{x},0;\boldsymbol{\theta}) - g(\boldsymbol{x})|^2\,, \tag{8}$$

$$\mathcal{L}_{h}(\boldsymbol{\theta};\mathcal{T}_{h}) = \frac{1}{|\mathcal{T}_{h}|}\sum_{(\boldsymbol{x},t)\in\mathcal{T}_{h}} |\mathcal{B}\hat{u}(\boldsymbol{x},t;\boldsymbol{\theta}) - h(\boldsymbol{x},t)|^2\,, \tag{9}$$

$|\mathcal{T}_{\mathcal{N}}|, |\mathcal{T}_g|, |\mathcal{T}_h|$ are the cardinalities of the sets $\mathcal{T}_{\mathcal{N}}, \mathcal{T}_g, \mathcal{T}_h$, and $w_{\mathcal{N}}, w_g, w_h > 0$ are the weights. The differential operators in the loss function $\mathcal{L}(\boldsymbol{\theta};\mathcal{T}_{\mathcal{N}},\mathcal{T}_g,\mathcal{T}_h)$ are obtained through automatic differentiation [45], and $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta};\mathcal{T}_{\mathcal{N}},\mathcal{T}_g,\mathcal{T}_h)$ is typically solved by gradient-based optimization methods (such as ADAM [46] or L-BFGS [47]).

In *inverse problems*, the parameter $\lambda$ in Eq. (5) is not known, and the objective is to infer the unknown parameter $\lambda$ from some extra measurement of the system in addition to the PDE and the initial/boundary conditions. For instance, let $\mathcal{T}_f \subset \Omega \times [0,T]$, and assume that the values of the solution $u(\boldsymbol{x},t)$ are known for $(\boldsymbol{x},t) \in \mathcal{T}_f$:

$$u(\boldsymbol{x},t) = f(\boldsymbol{x},t), \quad \forall (x,t) \in \mathcal{T}_f \subset \Omega \times [0,T]. \tag{10}$$

In this setting, the loss function $\mathcal{L}(\boldsymbol{\theta},\lambda;\mathcal{T}_{\mathcal{N}},\mathcal{T}_g,\mathcal{T}_h,\mathcal{T}_f)$ will have an extra term corresponding to the additional information of the system given by Eq. (10):

$$\mathcal{L}(\boldsymbol{\theta},\lambda;\mathcal{T}_{\mathcal{N}},\mathcal{T}_g,\mathcal{T}_h,\mathcal{T}_f) \coloneqq w_{\mathcal{N}}\mathcal{L}_{\mathcal{N}}(\boldsymbol{\theta},\lambda;\mathcal{T}_{\mathcal{N}}) + w_g\mathcal{L}_g(\boldsymbol{\theta},\lambda;\mathcal{T}_g) + w_h\mathcal{L}_h(\boldsymbol{\theta},\lambda;\mathcal{T}_h) + w_f\mathcal{L}_f(\boldsymbol{\theta},\lambda;\mathcal{T}_f), \tag{11}$$

where

$$\mathcal{L}_{f}(\boldsymbol{\theta},\lambda;\mathcal{T}_{f}) = \frac{1}{|\mathcal{T}_{f}|}\sum_{(\boldsymbol{x},t)\in\mathcal{T}_{f}} |\hat{u}(\boldsymbol{x},t;\boldsymbol{\theta}) - f(\boldsymbol{x},t)|^2\,. \tag{12}$$

Another notable change from Eq. (6) to Eq. (11) is that the unknown $\lambda$ also becomes the trainable parameter of the model, and it is jointly searched with $\boldsymbol{\theta}$ by minimizing Eq. (11).

## 3 Discovering governing equations in discrete systems

The problem of interest to us herein concerns the data-driven discovery of governing equations, and in particular, the ones corresponding to the nonlinear dynamical lattices discussed in Section 2.1. To do so, consider a 1D lattice of $N$ nodes with a dynamical variable $u_n(t) \in \mathbb{R}$ (or $\mathbb{C}$) associated to each node $n = 1, 2, \ldots, N$. Assume that the evolution of $\boldsymbol{u} = (u_1, u_2, \ldots, u_N)$ is governed by the following nonlinear dynamics

$$\dot{\boldsymbol{u}} = (\dot{u}_1, \dot{u}_2, \ldots, \dot{u}_N) = \mathcal{N}(u_1, \ldots, u_N), \tag{13}$$

where $\mathcal{N} : \mathbb{R}^N \to \mathbb{R}^N$ (or $\mathbb{C}^N \to \mathbb{C}^N$) is an operator comprised of inter-site couplings between nearest neighbors, but the explicit form of $\mathcal{N}(u_1, \ldots, u_N)$ is unknown. Our objective is to learn the governing equation $\mathcal{N}$ from sparse (temporal) observations of the nonlinear dynamics $\boldsymbol{u}(t) = \boldsymbol{f}(t)$ at times $t \in \mathcal{T}_{\boldsymbol{f}} \subset [0, T]$, where $T > 0$ is the terminal time of the system.

The differences between our setting and the PDE inverse problem explained in Section 2.2 are two-fold. First, even though the inter-site couplings in $\mathcal{N}$ between nearest neighbors can sometimes be viewed as finite differences, bearing resemblance to their continuous counterparts of (spatial) differential operators (cf. Section 2.1), the system described by Eq. (13) is intrinsically discrete. Second, unlike the parametric nonlinear operator $\mathcal{N}(u; \lambda)$ in Eq. (5), whose explicit dependence on $\lambda$ is given, the governing equation [cf. Eq. (13)] is generally unknown, except for the prior knowledge that the right-hand-side $\mathcal{N}(u_1, \ldots, u_N)$ involves only a shift-invariant coupling to nearby sites.

We thus make the following modifications to the PINN model of Eq. (11). For systems with real dynamical variables $\boldsymbol{u}(t) \in \mathbb{R}^N$, the PINN $\hat{\boldsymbol{u}} : \mathbb{R} \to \mathbb{R}^N$ takes only time $t$ as the input, which is mapped through an $L$-layer fully-connected neural network to the output corresponding to an $N$-dimensional vector $\hat{\boldsymbol{u}}(t) = (\hat{u}_1(t), \hat{u}_2(t), \ldots, \hat{u}_N(t)) \in \mathbb{R}^N$. Since the form of $\mathcal{N}(u_1, \ldots, u_N)$ is not known, we build an overcomplete library $\mathrm{Lib} = \{D_\alpha\}_{\alpha \in A}$ of shift-invariant discrete spatial operators modeling the linear inter-site couplings between nearest neighbors, as well as different types of nonlinear contributions. For instance, one dictionary element that is included in many of our numerical experiments is the discrete Laplacian

$$(D_2 \boldsymbol{u})_n = u_{n-1} - 2u_n + u_{n+1}. \tag{14}$$

The unknown operator $\mathcal{N} : \mathbb{R}^N \to \mathbb{R}^N$ is then modeled as a linear combination $\mathcal{N} = \sum_{\alpha \in A} \lambda_\alpha D_\alpha$ of elements $D_\alpha$ in the library, and the expansion coefficients $\boldsymbol{\lambda} = (\lambda_\alpha)_{\alpha \in A}$ are learned by minimizing the loss function

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_{\mathcal{N}}, \mathcal{T}_{\boldsymbol{f}}) \coloneqq w_{\mathcal{N}} \mathcal{L}_{\mathcal{N}}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_{\mathcal{N}}) + w_{\boldsymbol{f}} \mathcal{L}_{\boldsymbol{f}}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_{\boldsymbol{f}}), \tag{15}$$

where

$$\mathcal{L}_{\mathcal{N}}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_{\mathcal{N}}) = \frac{1}{|\mathcal{T}_{\mathcal{N}}|} \sum_{t \in \mathcal{T}_{\mathcal{N}}} \left| \dot{\hat{\boldsymbol{u}}}(t; \boldsymbol{\theta}) - \sum_{\alpha \in A} \lambda_\alpha D_\alpha \hat{\boldsymbol{u}}(t; \boldsymbol{\theta}) \right|^2, \tag{16}$$

$$\mathcal{L}_{\boldsymbol{f}}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_{\boldsymbol{f}}) = \frac{1}{|\mathcal{T}_{\boldsymbol{f}}|} \sum_{t \in \mathcal{T}_{\boldsymbol{f}}} \left| \hat{\boldsymbol{u}}(t; \boldsymbol{\theta}) - \boldsymbol{f}(t; \boldsymbol{\theta}) \right|^2, \tag{17}$$

and $\mathcal{T}_{\mathcal{N}}$, $\mathcal{T}_{\boldsymbol{f}}$, respectively, are subsets of $[0, T]$ corresponding to the training collocation points at which the ODE residual and the discrepancy between $\hat{\boldsymbol{u}}$ and the observed $\boldsymbol{f}$ are minimized. Nevertheless, it should be noted that although the notation $D_2$ prompts one to think of derivative operators, the relevant symbolism of $D_\alpha$ more generally concerns elements of the nonlinear operator, some of which will, by necessity, reflect the nonlinearity of the model (so they should be generally thought of as nonlinear operators).

In concluding this section, it is worth pointing out that when the dynamical variables are complex, i.e., $\boldsymbol{u}(t) \in \mathbb{C}^N$, they can be decomposed into real and imaginary parts, i.e., $\boldsymbol{u}^{(\mathrm{R})}$

and $\boldsymbol{u}^{(\mathrm{I})}$, respectively, thus rendering the dynamical variable $\boldsymbol{u}$ to be a mapping of the form of $\boldsymbol{u} : [0, T] \to \mathbb{R}^{2N}$. This way, the PINN $\hat{u} : \mathbb{R} \to \mathbb{R}^{2N}$ is mapped again through a fully-connected neural network to an output $2N$-dimensional vector now being itself of the form of $\hat{\boldsymbol{u}}(t) = (\hat{u}_1^{(\mathrm{R})}(t), \hat{u}_2^{(\mathrm{R})}(t), \ldots, \hat{u}_N^{(\mathrm{R})}(t), \hat{u}_1^{(\mathrm{I})}(t), \hat{u}_2^{(\mathrm{I})}(t), \ldots, \hat{u}_N^{(\mathrm{I})}(t)) \in \mathbb{R}^{2N}$. Having set up the stage of our computations, we are now ready to turn to the details of our numerical experiments.

# 4 Numerical Experiments

In all the numerical experiments that we discuss below, we consider the models summarized in Table 1. Moreover, we will use $N$, i.e., the number of lattice sites, to be in the range 20 to 31. In our experiments we find that changing the size of the lattice does not seem to change the results in any dramatic way. On the other hand, our experiments suggest that learning is faster when more lattice sites are involved in the dynamics as compared to when the dynamics is local to only a few sites. We made this observation while trying different initial conditions for the $\phi^4$ model where ICs that led to dynamical behavior involving a larger number of lattice nodes led to faster learning and convergence.

## 4.1 Data generation

At first, we solve the initial-value problems (IVPs) consisting of the models of Table 1 and the specific ICs in order to obtain spatio-temporal data that will be used for training our PINNs. To do so, we employ temporal integration. In particular, we use a fourth-order Runge-Kutta (RK4) method for the discrete $\phi^4$ and DsG examples, and an implicit backward difference scheme [48] for the DNLS and DCGL examples. The ICs for our data generation (see, Table 1) are inspired by the exact solutions to the continuous analogues of our models, although they do not always constitute ones such. For example, and as per the discrete $\phi^4$ and DsG cases, we use a traveling kink solution of the respective continuum cases. On the other hand, we use a Gaussian pulse in lieu of a bright soliton for the DNLS in order to observe an example of dynamics not necessarily proximal to a solitonic equilibrium. On the other hand, in the DCGL case, we use a form of the so-called Nozaki-Bekki holes [38].

As such, we solve the respective IVPs from $t = 0$ to $t = 10$ with a time step-size of $dt = 10^{-3}$. We then extract 50 samples from the simulated data at equal time intervals ($\Delta t = 0.2$), and use these samples to train our neural network.

## 4.2 Neural Network setup

We conducted all of our experiments, presented here and otherwise, using the DeepXDE [20] library. Our neural networks take as input only time ($t$) and for first-order systems output $u_n(t)$, $\forall n \in 1, \ldots, N$, while for second-order systems output $u_n(t)$, $\forall n \in 1, \ldots, N$, and $v_n = \dot{u}_n(t)$, $\forall n \in 1, \ldots, N$. Furthermore, as it was already mentioned in Section 3, in the cases where we have complex data (such as in the DNLS and DCGL cases), we split the data into real and imaginary parts and learn the two simultaneously, i.e., our neural network outputs: $u_n^{\mathrm{R}}(t)$ and $u_n^{\mathrm{I}}(t)$, $\forall n \in 1, \ldots, N$.

In the residual losses we construct, we only consider the interior nodes (i.e., nodes having both nearest-neighbors). This eliminates the need to know the BCs that govern the underlying data. Furthermore, and as per the second-order systems, we consider residual losses on both displacements and velocities $(u_n, v_n)$, as discussed above.

Lastly, the neural network architecture used in all the experiments involves fully-connected networks consisting of three hidden layers with 40 neurons each, and each layer uses the tanh activation function.

### 4.3 Results and Discussion

We start our results presentation by the arguably simplest among our selected models, namely the (real) discrete $\phi^4$ model given by Eq. (1). We consider first the following library of terms

$$\text{Lib}^{(1)} = \Big\{ \left(u_{n+1} + u_{n-1} - 2u_n\right), \left(u_{n+1} - u_{n-1}\right)/2, u_n, u_n^3 \Big\}, \tag{18}$$

which contains the discrete Laplacian operator as well as the finite difference representation of the first derivative (i.e., the second element in Eq. (18) corresponding to a centered, second-order accurate finite difference operator for $u_x$) alongside linear and cubic terms in $u_n$. One can argue that this is a library inspired by the continuum analogue of the model and its respective (potential) derivative term inclusions. Our respective results for $C = 2$ are depicted in Fig. 1(a) for this case where the solid red, blue, green and yellow lines correspond to the discrete representation of $u_{xx}$, $u_x$, $u$, and $u^3$, respectively. Indeed, the PINN learns the correct coefficients, and most importantly, it learns that there is no $u_x$ (namely, its discrete version) present in the governing equation for our data.

In the experiments that are shown in Figs. 1(b)-(d), we are taking a more "inherently discrete" approach to the relevant problem. More specifically, motivated by the discreteness coupling to near-neighbors (rather than to combinations prompting towards derivatives), we disaggregate the (discrete) operator $(u_{n+1} + u_{n-1} - 2u_n)$ into its constituents. Namely, instead of trying to learn the particular form of the inter-component difference, we learn the dependence of the governing equation on each of the sets of nearest neighbors. In particular, the panels (b), (c), and (d) in the figure consider respectively the following libraries:

$$\text{Lib}^{(2)} = \Big\{ u_{n+1}, u_{n-1}, u_n, u_n^3 \Big\}, \tag{19}$$

$$\text{Lib}^{(3)} = \Big\{ \text{Lib}^{(2)}, u_{n+2}, u_{n-2} \Big\}, \tag{20}$$

and

$$\text{Lib}^{(4)} = \Big\{ \text{Lib}^{(2)}, u_{n+1}^2 u_n, u_{n-1}^2 u_n, u_{n+1} u_{n-1} u_n, u_{n+1}^2 u_{n-1}, u_{n-1}^2 u_{n+1}, u_n^2 u_{n+1}, u_n^2 u_{n-1}, u_{n+1}^3, u_{n-1}^3 \Big\}. \tag{21}$$

$\text{Lib}^{(2)}$ is the simplest example containing the main ingredients of the original model. Hence, one would like to check whether the methodology can "disentangle" the role of these ingredients from other similar contributions to both the linear and nonlinear terms. With that in mind, $\text{Lib}^{(3)}$ is

essentially an augmentation of $\text{Lib}^{(2)}$ whence the next-nearest neighbors, i.e., $u_{n\pm2}$ are appended therein. Moreover, $\text{Lib}^{(4)}$ contains several possibilities for the cubic nonlinearity of the model. Indeed, in the latter case all possible combinations involving nearest neighbor contributions to a cubic nonlinearity have been incorporated; see, e.g., Ref. [49] where also the complex variant of such terms is discussed in the context of the DNLS model.

In Fig. 1(b), we observe that our gradient optimization method converges to the correct coefficients, thus constructing the correct nonvanishing prefactors for each of the relevant term contributions. Next, in the numerical experiments presented in Figs. 1(c)-(d), we consider the libraries of Eqs. (20) and (21), respectively, and try to find the dependence on next-to-next neighbors of each node. Here, we expect that solely the relevant "ingredient" terms will be selected, while the prefactor of extraneous contributions will converge to zero. However, it is important to note, as a limitation of the method, that for libraries that contained even-ordered terms (in particular, quadratic and quartic terms in our experiments), the model had difficulty learning the correct coefficients, and only by using data augmentation, were we able to get the model to learn the correct coefficients. More specifically, we accomplished data augmentation by using the fact that if $u$ is a solution to our system, so is $-u$, i.e., leveraging the relevant invariance of the model under this parity transformation of the field. This is, in line, with the earlier work of [30], where we have leveraged the symmetries of the model to enhance the network's ability to solve the inverse problem. We thus made the model learn both solutions simultaneously, and as expected, the model learned that the governing equations do not have any even order terms in them. In that sense, we have ensured (results not shown here for brevity) that, using both $u$ and $-u$, even-ordered terms in the library do not alter the findings presented in Fig. 1.

Our next example involved the complex variant of the model, namely the DNLS [cf. Eq. 2], which enables considerable additional richness in terms of the available nonlinear terms (see, e.g., Eq. (16.11) in [49]). It should be noted again that the PINN models consider real coefficients and thus we split our coefficients herein and state variables into real and imaginary parts. As a result, we construct separate losses for the real and imaginary parts, and set up the PINN to learn the respective coefficients simultaneously. Indeed, the panels (a) and (b) in Fig. 2 summarize our results herein for $C = 2$ (a setting more proximal to the continuum limit) and $C = 1/2$ (i.e., a rather discrete case), respectively. For this numerical experiment, we consider a library of the form:

$$\dot{u}_n = \alpha_1 u_{n+1} + \alpha_2 u_{n-1} + \alpha_3 u_n + \alpha_4 |u_n|^2 u_n, \;\; \alpha_k = a_k + i b_k \in \mathbb{C}, \;\; a_k, b_k \in \mathbb{R}, \;\; k = \{1, 2, 3, 4\}, \tag{22}$$

for both cases (i.e., Figs. 2(a)-(b)). It can be discerned from both panels of Fig. 2 that the PINN learns purely imaginary coefficients (i.e., $b_k$) as expected (see the red, blue, green, and yellow lines therein). That is, in this case, the scheme detects the effectively conservative nature of the model, since real coefficients would be tantamount to gain terms. Consequently, here the real coefficients $a_k$ denoted by solid black lines converge to zero. For convenience, in both panels we include the correct values of the coefficients for comparison.

We further performed numerical experiments on the DNLS with other libraries, motivated by the general cubic nonlinearity form presented in [49] and found that the PINN is capable of learning

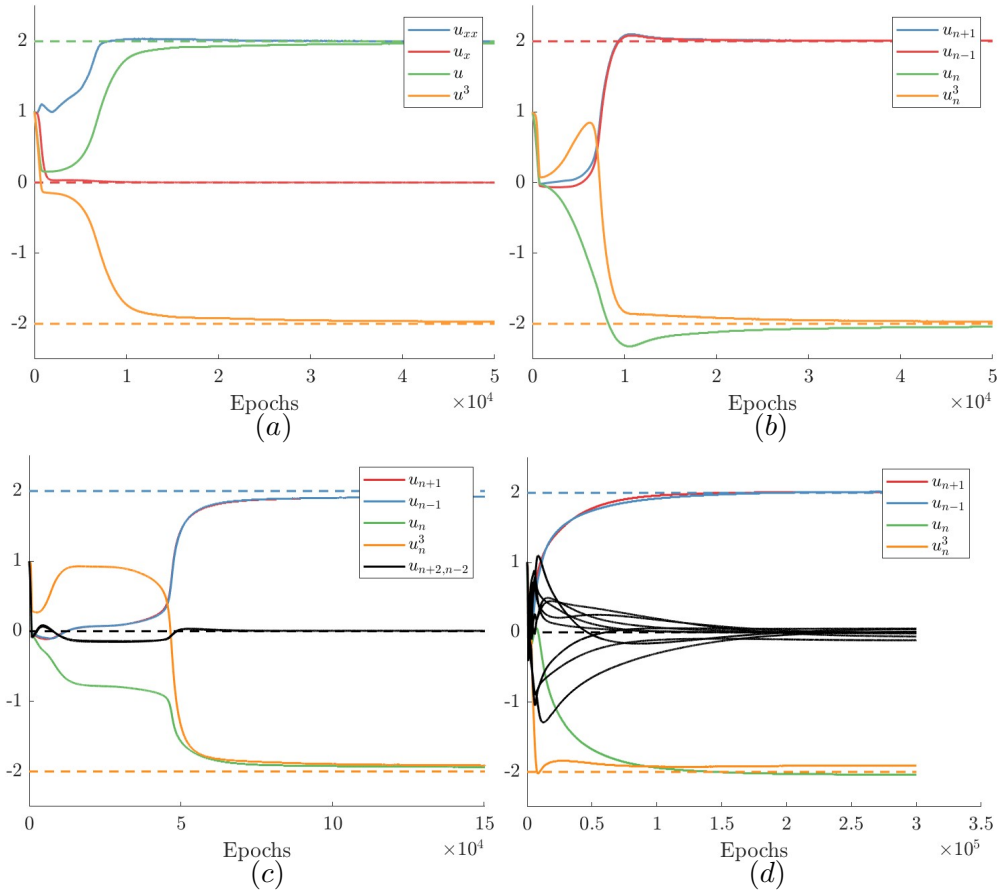


Figure 1: Numerical results for the discrete $\phi^4$ model [cf. Eq. 1] with $C = 2$. In panel (a), the library $\text{Lib}^{(1)}$ of Eq. (18) was considered where the solid blue, red, green and yellow lines correspond to the discrete representation of $u_{xx}$, $u_x$, $u$, and $u^3$, respectively. The numerical results obtained by using the library $\text{Lib}^{(2)}$ [cf. Eq. (19) are presented in panel (b) where solid blue, red, green, and yellow depict the $u_{n+1}$, $u_{n-1}$, $u_n$, and $u_n^3$, respectively. The same line-coloring-to-terms correspondence is used in panels (c) and (d) utilizing the libraries of Eqs. (20) and (21), respectively. The solid black lines therein correspond to (c) the terms $u_{n\pm2}$, and (d) to all the other cubic terms. In all the panels, the dashed lines serve as reference values for the actual values of the coefficients.

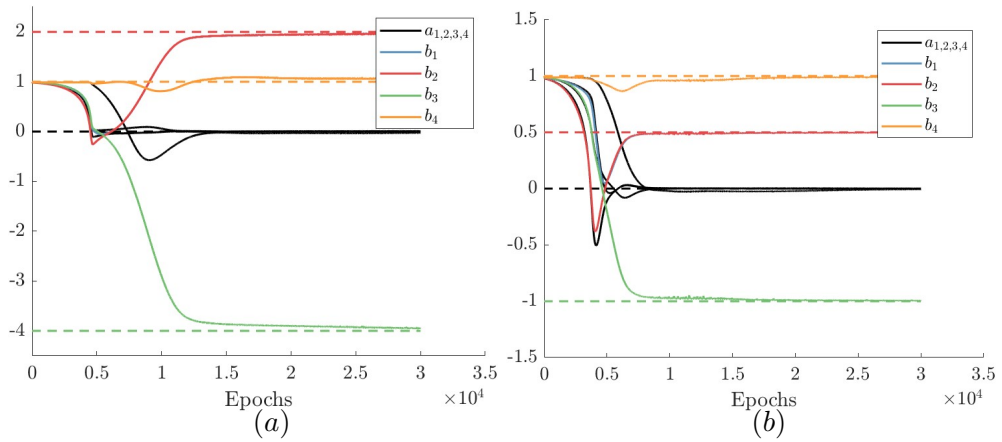


Figure 2: Numerical results for the DNLS [cf. Eq. 2] with (a) $C = 2$ and (b) $C = 1/2$ (see, also Eq. (22)). The dashed lines are reference values corresponding to the actual values for the coefficients (see text). The solid lines (see the legends of each panel) with colors other than black correspond to the imaginary parts of the trained coefficients $b_1$ (blue), $b_2$ (red), $b_3$ (green), and $b_4$ (orange). The solid black lines depict the real parts of the trained coefficients ($a_k$).

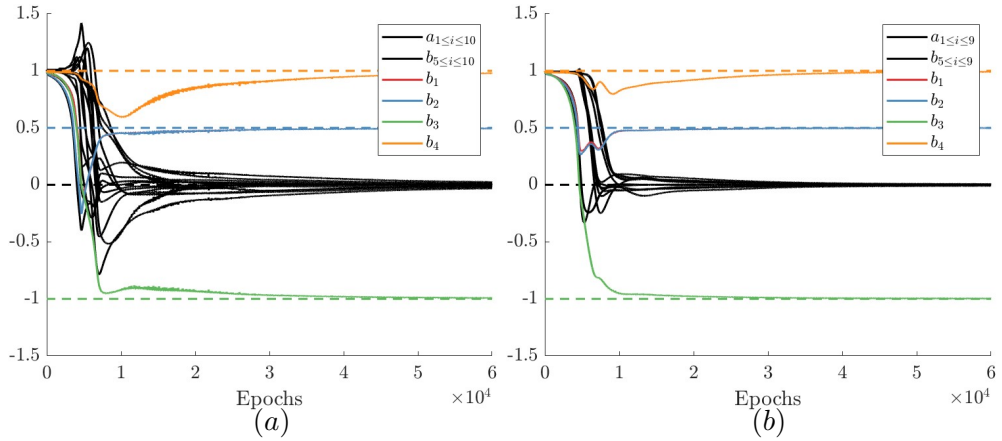


Figure 3: Numerical results for the DNLS [cf. Eq. 2] both with $C = 1/2$, and by using the library of (a) Eq. (23) and (b) Eq. (24). Similar to Fig. 2, the dashed lines are reference values corresponding to the actual values for the coefficients. For the line coloring, see the legends of each panel.

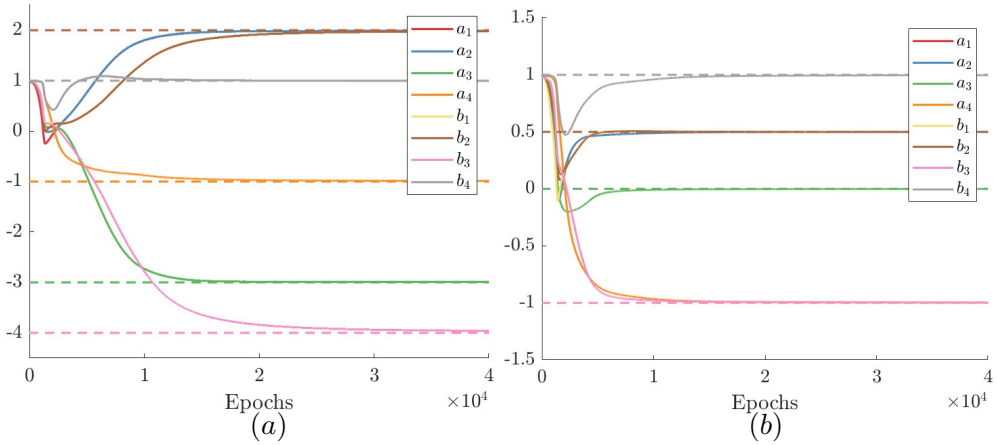


Figure 4: Numerical results for the DCGL [cf. Eq. 4] with (a) $C = 2$ and (b) $C = 1/2$ (see, also Eq. (22)). Same as before, the dashed lines are reference values corresponding to the actual values for the coefficients. Note that $\mathrm{Re}(\alpha_i) = a_i$ and $\mathrm{Im}(\alpha_i) = b_i$ (see also the legend for the coloring-to-coefficients mapping).

the coefficients of the DNLS correctly. Indicatively, we demonstrate in Figs. 3(a)-(b) two cases with $C = 1/2$ that, respectively, consider the following libraries:

$$\dot{u}_n = \alpha_1 u_{n+1} + \alpha_2 u_{n-1} + \alpha_3 u_n + \alpha_4 |u_n|^2 u_n + \alpha_5 |u_n|^2 + \alpha_6 u_n^2 + \alpha_7 (u_n^*)^2 + \alpha_8 \frac{u_{n+1} + u_{n-1}}{2} u_n + \alpha_9 |u_n|^2 (u_{n+1} + u_{n-1}) + \alpha_{10} u_n (|u_{n+1}|^2 + |u_{n-1}|^2), \quad \alpha_k = a_k + i b_k \in \mathbb{C}, \quad k = \{1, \ldots, 10\}, \tag{23}$$

and

$$\dot{u}_n = \alpha_1 u_{n+1} + \alpha_2 u_{n-1} + \alpha_3 u_n + \alpha_4 |u_n|^2 u_n + \alpha_5 |u_n|^2 + \alpha_6 |u_{n+1}|^2 + \alpha_7 |u_{n-1}|^2 + \alpha_8 |u_{n+1}|^2 u_{n+1} + \alpha_9 |u_{n-1}|^2 u_{n-1}, \quad \alpha_k = a_k + i b_k \in \mathbb{C}, \quad k = \{1, \ldots, 9\}, \tag{24}$$

where both $a_k$ and $b_k$ are real as before. Notice that once again here, similarly to the $\phi^4$ case discussed above, we have included terms that are quadratic in nature, and, similarly to the $\phi^4$ case we needed to use data augmentation in order to retrieve the correct coefficients in the presence of such quadratic terms. Despite the generality of the above libraries containing themselves different quadratic and cubic nonlinearities, the PINN model learned correctly the (purely imaginary) coefficients, as this can be discerned from panels (a) and (b) of Fig. 3. We mention in passing that we tried other values for the coupling constant $C$ as well as other libraries (alongside the ones presented so far), and found that in all the cases considered the PINN discovered the correct coefficients.

Having discussed the DNLS, we turn our focus to the DCGL model [cf. Eq. (4)]. Our motivation in doing so was to investigate whether PINNs can learn complex coefficients, when ones such are relevant for the (general) libraries considered herein. At first, we consider the prototypical library of Eq. (22), where we expect to discover the complex prefactor of both the discrete Laplacian term, including the equal (complex) coefficients of the $u_{n\pm1}$ terms, as well as that of the linear term

$\propto u_n$. Finally, the PINN method is able to capture equally accurately not only the above linear terms, but also the complex $(-(1-i)))$ term of the cubic nonlinearity. As is clearly shown in Fig. 4, all the relevant terms are accurately identified, while the prefactors of additional, irrelevant terms in the library converge to vanishing values beyond a suitably large number of Epochs. We found this true for a variety of libraries, including ones with even ordered terms and next to next neighbors. We even performed experiments using the libraries of Eq 23 Eq 24 and found that our models learned the correct coefficients.

Finally, we choose the discrete sine-Gordon (DsG) model of Eq. (3) as an intriguing example due to the fact that it contains the $\sin(u_n)$ term. The latter can be expanded in Taylor series, yet it cannot be fully approximated by means of a power-law library. It is presumably for this reason that relevant attempts at the inverse problem of the pendulum [42] or the double pendulum [50] involve libraries containing trigonometric terms (rather than purely power-law ones). This key difference of the present model from the earlier ones inspired us to use sine series-based libraries, polynomial libraries and mixed libraries in order to explore what the PINN model would learn in such a case and what the limitations of each case example may be. For all the numerical experiments that we discuss here, we picked $C = 1/2$, and the respective results are shown in Fig. 5. In particular, Fig. 5(a) considers the library (once again, effectively, building in the $u_n \to -u_n$ invariance of the model):

$$\ddot{u}_n = \alpha_1 u_{n+1} + \alpha_2 u_{n-1} + \alpha_3 u_n + \alpha_4 \sin(u_n) + \alpha_5 \sin(2u_n) + \alpha_6 \sin(3u_n) + \alpha_7 \sin(4u_n) + \alpha_8 \sin(5u_n). \tag{25}$$

Here, the PINN model is able to learn the correct sine term (notice that the solid black lines in the figure correspond to $\alpha_k = 0,\ k = 5, 6, 7, 8$, upon convergence). Next, the results presented in Fig. 5(b) explore the case of a power-law-based library of functions. Indeed, in this case, we consider a library that contains three terms of the Taylor series expansion of the sine function as:

$$\ddot{u}_n = \alpha_1 u_{n+1} + \alpha_2 u_{n-1} + \alpha_3 u_n + \alpha_4 u_n^3 + \alpha_5 u_n^5. \tag{26}$$

It can be discerned from panel (b) of the figure that the model tries to learn a (truncated) Taylor series expansion of the sine-term. However, we should mention that the model seems to be very sensitive when it comes to polynomial libraries while simultaneously, the number of terms in the library seems to have a considerable impact on what the model learns (even when all the terms are odd powers). Indeed, in this case, for instance, the yellow curve associated with coefficient $a_4$ converges to a finite value which is clearly distinct from, e.g., the theoretical Taylor-function prediction of $1/5$! We can thus observe the relevant limitation of the approach in that a polynomial-based library is unable to fully capture the effects of a sinusoidal nonlinearity.

We conclude our series of experiments by discussing Fig. 5(c) which embodies the library:

$$\ddot{u}_n = \alpha_1 u_{n+1} + \alpha_2 u_{n-1} + \alpha_3 u_n + \alpha_4 u_n^3 + \alpha_5 u_n^2 + \alpha_6 u_n^5 + \alpha_7 \sin(u_n) + \alpha_8 \sin(2u_n), \tag{27}$$

i.e., a setting containing both polynomial and sinusoidal terms. Our numerical results suggest that there is some sort of a competition between the polynomial and sine terms in trying to describe the nonlinearity of the model. It should be noted however, that when trying to learn the coefficients for this model, the choice of ICs may have a significant impact, especially in cases of this sort

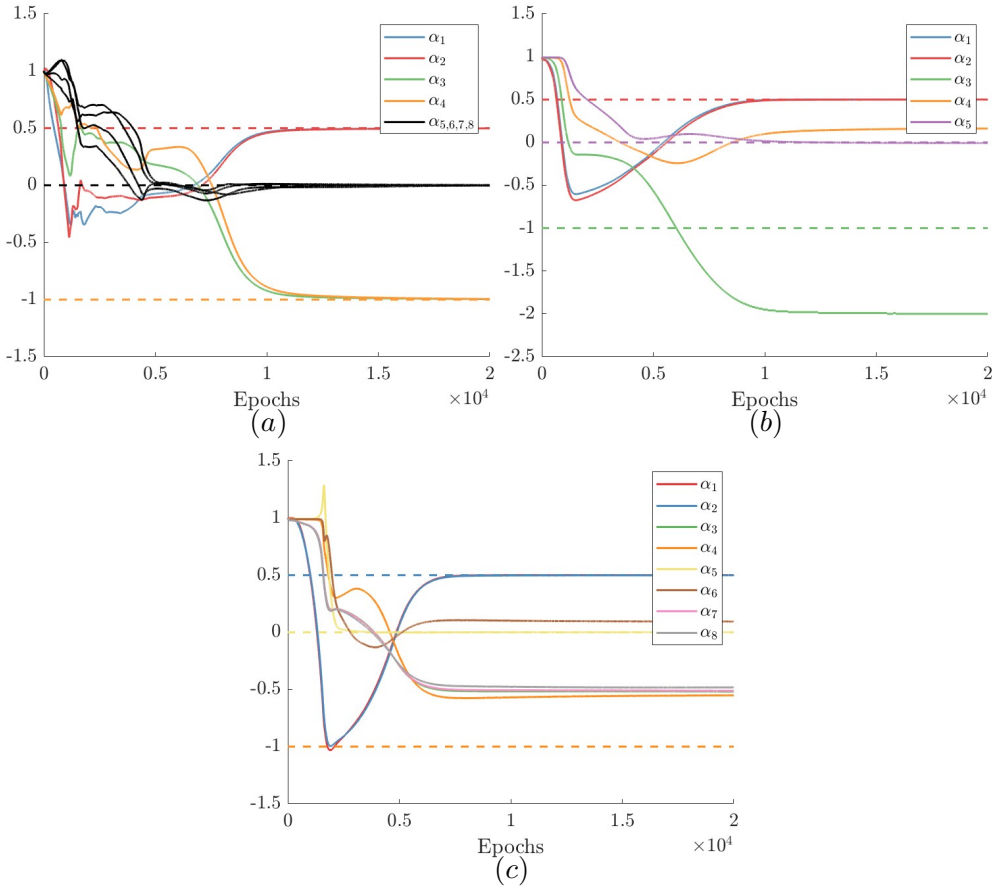


Figure 5: Numerical results for the DsG model [cf. Eq. 3] with $C = 1/2$. In panel (a), the library of Eq. (25) was considered whereas panel (b) utilized the library of Eq. (26). The numerical results while using the library of Eq. (27) are depicted in panel (c). The legends in each of these panels offer the line-coloring-to-terms correspondence, and the dashed lines serve as reference values for the actual values of the coefficients.

with different competing terms contributing at the same order. We briefly report that using the exact solution of the continuous sine-Gordon seemed to work well for some libraries, and using the exact solution of the continuous $\phi^4$ seemed to work well with other libraries. In particular, using the exact solution of the continuous sine-Gordon seems to work better for libraries with sine terms in them while the exact solution of the continuous $\phi^4$ seems to work better for libraries with polynomial terms corresponding to the Taylor expansion of the sin nonlinearity.

## 5 Conclusions and Future Challenges

In the present work, we have explored the methodology of Physics-Informed Neural Networks (PINNs) and how PINNs perform when attempting to solve the inverse problem in the context of nonlinear dynamical lattices with many degrees of freedom. We argued herein that, in addition to

the relevant problem for PDEs, such lattice models are of particular interest in their own right for various physical contexts ranging from optics to atomic physics to materials science. Hence, a detailed understanding of the solution of the inverse problem of coefficient identification is of particular relevance in this context as well. Indeed, we envision the rather mature experimental observation and data acquisition techniques in such settings to be of value in the near future, not only for machine-learning-based classification tasks, as, e.g., has recently been realized in [51], but also for data-driven modeling efforts.

We started with a simpler real system case example in the form of the $\phi^4$ model. Here, we were able to identify the coefficients, although to avoid the possibility of quadratic terms, a relevant limitation concerned the use of dynamics both for $u$ and $-u$ to establish the invariance of the model under such a transformation. Both in the real case of the $\phi^4$ and in its complex analogue of the discrete nonlinear Schrödinger lattice, we considered a wide variety of nonlinear terms. We thus confirmed that the additional nonlinearities bear prefactors that eventually (for sufficiently many epochs) tend to vanishing values, thereby establishing the models of interest. We did not restrict our considerations to purely real (or purely imaginary) coefficients, but rather extended them to models with complex ones such as the discrete complex Ginzburg-Landau equation. Finally, we considered cases beyond the setting of purely power-law nonlinearities, such as the sine-Gordon lattice. Here, too, we explored some of the limitations of the PINNs, such as their inability to capture the fully sinusoidal effects with power-law-based libraries, but also the potential sensitivity that the concurrent presence of trigonometric and power-law terms may lead to.

Naturally, the field of such inverse (and forward) problems in the realm of nonlinear dynamical lattices is still at a particularly early stage, and further studies are certainly warranted. Among the numerous points meriting further exploration, we note the case example of nonlinearities beyond nearest-neighbors (and indeed the case of longer-range kernels). Moreover, here we have restricted considerations to (1+1)-dimensional problems, yet the examination of higher dimensional settings is of particular interest in its own right. Beyond these examples, a progressively deeper understanding of the limitations of the PINN (or SINDY) type approaches and of how further inclusion of the model structure (conservation laws, symmetries, symplectic nature etc.) of the underlying system may facilitate convergence are, in our view, questions of importance for further studies. Such efforts are presently in progress and will be reported in future publications.

## References


[1] S. Aubry. Discrete breathers: Localization and transfer of energy in discrete Hamiltonian nonlinear systems. *Physica D*, 216:1, 2006.

[2] S. Flach and A.V. Gorbach. Discrete breathers - Advances in theory and applications. *Phys. Rep.*, 467(1):1 – 116, 2008.

[3] G. Gallavotti. *The Fermi–Pasta–Ulam Problem: A Status Report.* Springer-Verlag, Berlin, Germany, 2008.

[4] P.G. Kevrekidis. Non-linear waves in lattices: past, present, future. *IMA J. Appl. Math.*, 76:389–423, 2011.

[5] F. Lederer, G. I. Stegeman, D. N. Christodoulides, G. Assanto, M. Segev, and Y. Silberberg. Discrete solitons in optics. *Phys. Rep.*, 463:1, 2008.

[6] M. Sato, B. E. Hubbard, and A. J. Sievers. *Colloquium*: Nonlinear energy localization and its manipulation in micromechanical oscillator arrays. *Rev. Mod. Phys.*, 78:137, 2006.

[7] M. Remoissenet. *Waves Called Solitons*. Springer-Verlag, Berlin, 1999.

[8] Yu. Starosvetsky, K.R. Jayaprakash, M. A. Hasan, and A.F. Vakakis. *Dynamics and Acoustics of Ordered Granular Media*. World Scientific, Singapore, 2017.

[9] C. Chong and P.G. Kevrekidis. *Coherent Structures in Granular Crystals: From Experiment and Modelling to Computation and Mathematical Analysis*. Springer, New York, 2018.

[10] L. Q. English, M. Sato, and A. J. Sievers. Modulational instability of nonlinear spin waves in easy-axis antiferromagnetic chains. ii. influence of sample shape on intrinsic localized modes and dynamic spin defects. *Phys. Rev. B*, 67:024403, 2003.

[11] A. J. Sievers and S. Takeno. Intrinsic localized modes in anharmonic crystals. *Phys. Rev. Lett.*, 61:970–973, Aug 1988.

[12] J. B. Page. Asymptotic solutions for localized vibrational modes in strongly anharmonic periodic systems. *Phys. Rev. B*, 41:7835–7838, 1990.

[13] P. Binder, D. Abraimov, A. V. Ustinov, S. Flach, and Y. Zolotaryuk. Observation of breathers in Josephson ladders. *Phys. Rev. Lett.*, 84:745, 2000.

[14] E. Trías, J. J. Mazo, and T. P. Orlando. Discrete breathers in nonlinear lattices: Experimental detection in a josephson array. *Phys. Rev. Lett.*, 84:741, 2000.

[15] M. Peyrard. Nonlinear dynamics and statistical physics of DNA. *Nonlinearity*, 17:R1, 2004.

[16] S. Yomosa. Soliton excitations in deoxyribonucleic acid (DNA) double helices. *Phys. Rev. A*, 27:2120–2125, Apr 1983.

[17] O. Morsch and M. Oberthaler. Dynamics of Bose–Einstein condensates in optical lattices. *Rev. Mod. Phys.*, 78:179, 2006.

[18] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[19] 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. *Journal of Computational Physics*, 378:686–707, February 2019.

[20] L. Lu, X. Meng, Z. Mao, and G.E. Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[21] S.L. Brunton, J.L. Proctor, and J.N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, April 2016.

[22] H. Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.

[23] J. Feliu-Faba, Y. Fan, and L. Ying. Meta-learning pseudo-differential operators with deep neural networks. *Journal of Computational Physics*, 408:109309, 2020.

[24] Z. Li, N.B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[25] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

[26] E Weinan and Bing Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[27] Y. Gu, H. Yang, and C. Zhou. Selectnet: Self-paced learning for high-dimensional partial differential equations. *Journal of Computational Physics*, 441:110444, 2021.

[28] Y. Shin, Z. Zhang, and G.E. Karniadakis. Error estimates of residual minimization using neural networks for linear PDEs. *arXiv preprint arXiv:2010.08019*, 2020.

[29] T. Luo and H. Yang. Two-layer neural networks for partial differential equations: Optimization and generalization theory. *arXiv preprint arXiv:2006.15733*, 2020.

[30] W. Zhu, W. Khademi, E.G. Charalampidis, and P.G. Kevrekidis. Neural Networks Enforcing Physical Symmetries in Nonlinear Dynamical Lattices: The Case Example of the Ablowitz-Ladik Model. *Physica D: Nonlinear Phenomena*, 434:133264, June 2022.

[31] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G.E. Karniadakis. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

[32] J. Cuevas-Maraver and P. G. Kevrekidis (eds.). *A dynamical perspective on the $\phi^4$ model*. Nonlinear Systems and Complexity. Springer International Publishing, 1st edition, 2019.

[33] P.G Kevrekidis. *Discrete Nonlinear Schrödinger Equation: Mathematical Analysis, Numerical Computations and Physical Perspectives*. Springer-Verlag (Heidelberg, 2009).

[34] O.M. Braun and Y.S. Kivshar. Nonlinear dynamics of the Frenkel-Kontorova model. *Physics Reports*, 306(1-2):1–108, December 1998.

[35] J. Cuevas-Maraver, P. G. Kevrekidis, and F. L. Williams (eds.). *The sine-Gordon Model and its Applications: From Pendula and Josephson Junctions to Gravity and High-Energy Physics*. Nonlinear Systems and Complexity 10. Springer International Publishing, 1st edition, 2014.

[36] N.K. Efremidis and D.N. Christodoulides. Discrete Ginzburg-Landau solitons. *Phys. Rev. E*, 67:026606, Feb 2003.

[37] N.K. Efremidis, D.N. Christodoulides, and K. Hizanidis. Two-dimensional discrete Ginzburg-Landau solitons. *Phys. Rev. A*, 76:043839, Oct 2007.

[38] I.S. Aranson and L. Kramer. The world of the complex Ginzburg-Landau equation. *Rev. Mod. Phys.*, 74:99–143, Feb 2002.

[39] M. Salerno and F.Kh. Abdullaev. *Discrete Solitons of the Ginzburg-Landau Equation*, pages 303–317. Springer International Publishing, Cham, 2022.

[40] I. Roy, S.V. Dmitriev, P.G. Kevrekidis, and A. Saxena. Comparative study of different discretizations of the $\phi^4$ model. *Phys. Rev. E*, 76:026601, Aug 2007.

[41] I. V. Barashenkov, O. F. Oxtoby, and Dmitry E. Pelinovsky. Translationally invariant discrete kinks from one-dimensional maps. *Phys. Rev. E*, 72:035602, Sep 2005.

[42] K. Lee, N. Trask, and P. Stinis. Structure-preserving sparse identification of nonlinear dynamics for data-driven modeling. In B. Dong, Q. Li, L. Wang, and Z.-Q. J. Xu, editors, *Proceedings of Mathematical and Scientific Machine Learning*, volume 190 of *Proceedings of Machine Learning Research*, pages 65–80. PMLR, 15–17 Aug 2022.

[43] S. Xia, D. Kaltsas, D. Song, I. Komis, J. Xu, A. Szameit, H. Buljan, K.G. Makris, and Z. Chen. Nonlinear tuning of pt symmetry and non-hermitian topological states. *Science*, 372(6537):72–76, 2021.

[44] A. Müllers, B. Santra, C. Baals, J. Jiang, J. Benary, R. Labouvie, D.A. Zezyulin, V.V. Konotop, and H. Ott. Coherent perfect absorption of nonlinear matter waves. *Science Advances*, 4(8):eaat6539, 2018.

[45] A.G. Baydin, B.A. Pearlmutter, A.A Radul, and J.M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18, 2018.

[46] D.P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*, 2015.

[47] D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[48] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving ordinary differential equations I. Nonstiff Problems*. Springer-Verlag (Heidelberg, 2008).

[49] S.V. Dmitriev and A. Khare. *Exceptional Discretizations of the NLS: Exact Solutions and Conservation Laws*, pages 293–310. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[50] K. Kaheman, J.N. Kutz, and S.L. Brunton. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. R. Soc. A*, 476:20200279, 2020.

[51] Shangjie Guo, Amilson R Fritsch, Craig Greenberg, I B Spielman, and Justyna P Zwolak. Machine-learning enhanced dark soliton detection in bose–einstein condensates. *Machine Learning: Science and Technology*, 2(3):035020, jun 2021.