Machine Learning of Nonlinear Waves: Data-Driven Methods for Computer-Assisted Discovery of Equations, Symmetries, Conservation Laws, and Integrability

Jimmie Adriazola

School of Mathematical and Statistical Sciences, Arizona State University

E-mail: jimmie.adriazola@asu.edu

Panayotis G. Kevrekidis

Department of Mathematics and Statistics, University of Massachusetts, Amherst

E-mail: kevrekid@umass.edu

Vassilis Koukouloyannis

Department of Mathematics, University of the Aegean, Karlovassi, Greece

E-mail: vkouk@aegean.gr

Wei Zhu

School of Mathematics, Georgia Institute of Technology

E-mail: weizhu@gatech.edu

August 2025

Abstract. The purpose of this article is to provide a perspective —admittedly, a rather subjective one— of recent developments at the interface of machine learning/data-driven methods and nonlinear wave studies. We review some recent pillars of the rapidly evolving landscape of scientific machine learning, including deep learning, data-driven equation discovery, and operator learning, among others. We then showcase these methods in applications ranging from learning lattice dynamical models and reduced order modeling of effective dynamics to discovery of conservation laws and potential identification of integrability of ODE and PDE models. Our intention is to make clear that these machine learning methods are complementary to the preexisting powerful tools of the nonlinear waves community, and should be integrated into this toolkit to augment and enable mathematical discoveries and computational capabilities in the age of data.

Keywords: Nonlinear Waves, Machine Learning, Data-Driven Methods, Neural Networks, Hamiltonian Dynamical Systems, Symbolic Learning, Conservation Laws, Integrability, Variational Methods, Moment Methods.

1. Introduction

Nonlinear wave systems lie at the heart of many of the most profound and important problems in physics. From shallow water waves and tsunamis, to optical solitons in fibers, to matter waves in Bose–Einstein condensates (BECs), these systems exhibit rich and intricate dynamics including solitons, dispersive shocks, wave turbulence, and modulational instabilities [2, 79, 249, 222, 46]. These phenomena typically emerge from the delicate interplay between nonlinearity and dispersion (and often, in practice, from the balance of non-conservative contributions therein such as damping and forcing). For decades, the nonlinear waves community has led the way towards developing powerful analytical tools and numerical methods to understand these features, often leveraging deep insights provided by integrability, conservation laws, and spectral decompositions [238, 2].

And yet, we are now in the age of data; this is an era in which massive simulations, high-fidelity experiments, and large-scale sensor networks are producing data at unprecedented scales [38]. The traditional tools of nonlinear wave theory, powerful as they may be, were not designed with this data-rich regime in mind. At the same time, the rise of machine learning (ML) offers a complementary paradigm of learning patterns, dynamics, and even governing equations directly from data [39, 206, 195, 118]. We emphasize that ML is not a replacement for the theoretical insights of nonlinear wave physics. Indeed, an aim of this review is to advocate that it can enable us to extend what we know, accelerate what we can simulate, and reveal what we have yet to understand. In that sense, it constitutes an invitation to the Nonlinear Waves community to (further) explore and integrate this new toolkit, and to the Scientific Machine Learning (SciML) community to (further) adapt and purpose these methods towards the remarkable challenges of the field of Nonlinear Waves.

Recent years have seen an explosion of interest in leveraging ML methods to address fundamental challenges in wave physics [165, 256]. Supervised learning approaches including recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and attention-based architectures have demonstrated success in forecasting complex wave dynamics [145]. For example, data-driven models trained on time series have been used to predict rogue waves, phase modulations, or dispersive spreading with accuracy that rivals traditional solvers, particularly on short time horizons [147]. Moreover, unsupervised learning techniques have shown value in uncovering low-dimensional structures within high-dimensional wave fields [139, 223]. Principal component analysis (PCA), dynamic mode decomposition (DMD), and nonlinear autoencoders can identify dominant modes, suitable reduced order models, and enable the study of coherent structures and attractors in wave turbulence and modulationally unstable media [36, 223]. Similarly in spirit to traditional phase space methods that defined an entire era of nonlinear science, these data-driven methods offer an intuitive geometric picture in which to view dynamics that may defy analytical simplification [221].

Meanwhile, physics-informed approaches such as physics-informed neural networks

(PINNs) offer a compelling hybrid framework [233, 45]. These approaches are relevant not only to forward problems—especially in high-dimensional settings—but are perhaps even more impactful in solving inverse problems. In the latter, the data inform the identification of parameters in the physically inspired model terms. By encoding PDEs directly into the architecture or loss function of a neural network, they bridge the gap between data and physics. This allows for the solution of forward and inverse problems, even in regimes with limited or corrupted data [233], on equations such as the nonlinear Schrödinger (NLS), Korteweg–de Vries (KdV), and sine-Gordon (SG) models [45]. Moreover, the true power of PINNs is evident in their use as surrogate models in inverse design problems where forward solvers are extremely costly to compute when evaluating a quantity of interest [96].

When boundary or initial input data are not fixed during training, operator learning techniques such as deep operator networks (DeepONets) and Fourier neural operators (FNOs) replace the use of PINNs [153, 132]. PINNs are trained to learn functions that simultaneously satisfy the PDE operator as well as specified boundary/initial data. On the other hand, operator learning is the numerical construction of a mapping between a class of boundary/initial data and solutions in infinite-dimensional spaces [132]. Neural operators have recently demonstrated impressive capabilities in simulating nonlinear wave phenomena. For example, MITONet, an autoregressive neural operator, emulates two-dimensional shallow-water dynamics under varying boundary and bathymetry conditions, allowing for rapid and accurate forecasting of tide-driven coastal flows [202]. Other studies apply neural operators for long-time integration of canonical wave equations, for example, Lei et al. use a recurrent integration scheme with neural operators to accurately simulate the Korteweg-de Vries, sine-Gordon and Klein-Gordon equations over extended time horizons [140]. The latter task naturally poses significant challenges in connection to the preservation of the different conserved quantities and the fundamental (e.g., symplectic) structural features of such integrable, as well as near-integrable systems.

In experimental settings, neural operators have also been used to recover and interpret the emergent behavior of KdV-Burgers in nonlinear droplet systems, effectively extracting underlying Green's functions directly from the data [86]. Spherical FNOs have been designed to predict the evolution of shallow-water waves in spherical domains such as the Earth's surface, allowing time-history forecasts of wave patterns on planetary scales [141]. Neural operators have also found novel applications in soliton identification. In particular, Zhang, et al., develop a method that learns a mapping between initial and final states of quasi-one-dimensional BEC systems, recovering ground states and solitonic dynamics from data while respecting the Gross–Pitaevskii equation [250].

Other particularly exciting directions include the automated discovery of governing equations from data [40, 47]. Techniques such as sparse regression and symbolic neural models are being used to learn the form of evolution equations from observed dynamics [40, 63]. These tools offer an attractive possibility, namely that we may reverse the traditional modeling pipeline, where we infer laws from observations rather

than the other way around [47]. Importantly, as will be discussed in detail below, such techniques and tailored variants thereof have also been used in order to discover conservation laws [150, 149, 258] and eventually the potential full integrablity [134, 70, 7] of the models at hand.

This presents, in our view, not a "passing trend", but rather a real opportunity to address fundamental open problems and to pave new avenues of exploration. ML methods can act as surrogate models for rapid exploration [235], as interpretable embeddings for discovering new structures, and as inverse tools for designing experiments or reconstructing models or parameters thereof [85]. They may enable insights into the solution of difficult problems such as the discovery of conservation laws, of Lax pairs [7], of suitable low-dimensional reductions [38, 31] or, e.g., of modulation equations when these are unknown or seemingly too complex to extract (e.g., in higher dimensions) [79]. Of course, challenges remain. Generalization to unseen data, interpretability, extrapolation, and physical consistency, among them, are precisely the kinds of foundational problems that have emerged [17] and which will continue to challenge the nonlinear wave and machine learning communities in years to come.

We believe that the future of nonlinear wave science will rest not only on analytical rigor and numerical precision, but also on insights derived from data-driven methods, hybrid models, and intelligent inference. We anticipate that the next generation of nonlinear wave scientists may benefit from integrating a deeper understanding of this diverse toolbox and its potential. Toward this vision, we aspire to summarize (in, admittedly, a way bearing a strong imprint of our personal taste) some of the recent advances in ML methods and the substantial promise they hold for applications in nonlinear wave science.

We begin, in Section 2, by reviewing the recent pillars of scientific machine learning. Section 3 discusses some of the recent advances based on PINNs, while Section 4 focuses on reduced-order modeling, spearheaded by SINDy [40] and related approaches. Section 5 addresses the learning of structural properties of models from data (e.g., conservation laws, Hamiltonian structure), and Section 6 is dedicated to the discovery of integrability and its associated features. Finally, in Section 7, we summarize our findings and offer an outlook on potential future studies.

2. Recent Pillars of Scientific Machine Learning

2.1. Deep Learning

2.1.1. Neural Network Architectures. Modern deep learning is grounded in a diverse family of neural network architectures, each designed to capture particular structures in data. We aim to briefly summarize some of these structures, and trust that the interested reader can learn about them in detail from the listed citations. The most basic network architecture is perhaps the *fully connected feedforward network* (also called multilayer perceptron, MLP), which consists of layers of linear transformations followed by

nonlinear activation functions. These models are universal approximators [67, 20, 246] and remain widely used for tabular data and as baseline models in scientific ML [93, 28]. Their main strengths are flexibility and simplicity, but they often require large amounts of data and struggle with high-dimensional structured inputs.

Convolutional neural networks (CNNs) introduce weight sharing and local connectivity to exploit spatial structure. Originally developed for image recognition [137], CNNs are now widely applied to problems with translation invariance and local correlation structure, including fluid dynamics and PDE surrogate modeling [100, 225]. Their strengths lie in parameter efficiency and strong inductive bias for spatial data, though they may fail on data without clear grid-like structure [34, 58, 255].

Recurrent neural networks (RNNs) are designed for sequential data, processing inputs recursively while maintaining a hidden state. They have been widely applied in time-series analysis and natural language processing [208]. However, standard RNNs often suffer from vanishing or exploding gradients during training [183]. To overcome this, architectures such as the *long short-term memory* (LSTM) [106] and *gated recurrent unit* (GRU) [55] were introduced, enabling networks to capture long-range dependencies more effectively. While LSTMs and GRUs remain influential, their sequential nature limits parallelization, and future work in this direction is warranted, should they find further application in large scale nonlinear wave modeling [219].

In recent years, *transformer architectures* [231] have displaced recurrent networks in many domains. By relying on self-attention mechanisms [15] rather than recurrence, transformers scale more efficiently and excel at modeling long-range dependencies. Their strengths include parallelizable training and superior performance across language, vision, and multimodal learning, though their large parameter counts and training costs are potential drawbacks.

Beyond these canonical classes, specialized architectures continue to expand the landscape. Graph neural networks (GNNs) generalize convolutions to non-Euclidean data [34], making them powerful tools for modeling physical systems defined on meshes or networks. Autoencoders and variational autoencoders (VAEs) [125] provide unsupervised representation learning, while generative adversarial networks (GANs) [95] offer powerful generative models. Each architecture introduces inductive biases and trade-offs, and the choice depends strongly on the problem domain.

We emphasize the strengths and weaknesses of the major families of neural network architectures. MLPs serve as general-purpose approximators, CNNs excel at spatially structured data, RNNs (and their variants LSTMs and GRUs) are well-suited for sequential data, transformers capture long-range dependencies, and GNNs effectively model relational structures. Together, these architectures form the foundation of modern scientific machine learning applications, some of which we highlight in this review. Naturally, it is impossible to cover every domain where these architectures arise, and we refer the reader to the broader literature for further exploration.

2.1.2. Learning Functions through Physics-Informed Neural Networks In recent years, a novel class of NN-based techniques, the so-called *Physics-Informed Neural Networks* (PINNs), has emerged, designed to solve (forward problem) and discover (inverse problem) differential equations. They were introduced in [195], although they had been announced earlier in two arXiv preprints [193, 194]. The key idea is to embed physical laws, expressed through partial differential equations (PDEs), directly into the loss function used to train the networks. This approach enables the learning of complex physical phenomena from sparse or noisy data, providing a compelling alternative to traditional numerical solvers and facilitating the systematic discovery of physical models from data.

Traditional methods for solving PDEs, such as finite difference, finite volume, and finite element methods, can be computationally expensive, especially in high dimensions, and require extensive domain-specific knowledge. They also heavily rely on mathematically well-posed formulations. Conversely, PINNs offer mesh-free solutions that are generalizable and (potentially) scalable, opening new avenues in engineering, physics, and biomedical sciences. Importantly, also, we note in passing that they heavily rely on optimization methods that may provide answers (potentially pertaining to local extrema etc.) in the case of incomplete/non-well-posed data. It does not elude us that this feature formulates some novel deep mathematical analysis questions in its own right.

More concretely, in PINNs, given a PDE of the form:

$$\mathcal{N}[u(x,t)] = 0, \quad x \in \Omega,\ t \in [0,T],$$

where $\mathcal{N}$ is a (usually nonlinear) differential operator and $u(x,t)$ is the unknown solution, the neural network is trained to minimize a composite loss function consisting of the sum of two parts: a self-supervised one and a supervised one.

$$\mathcal{L} = \mathcal{L}_{ss} + \mathcal{L}_s$$

For the self-supervised part of the loss function, which contains the physics originating information, we consider a set of N_{ss} points (x_i^{ss}, t_i^{ss}) in the $\Omega \times [0,T]$ domain and can be for example of the form

$$\mathcal{L}_{ss} = \frac{1}{N_{ss}} \sum_{i=1}^{N_{ss}} \mathcal{N}[u(x_i^{ss}, t_i^{ss})]^2,$$

while for the supervised part we consider N_s points (x_j^s, t_j^s) in the boundary of the $\Omega \times [0,T]$ domain containing the information taken by the known initial and boundary conditions as well as in a possible number of points where the solution could be known inside the $\Omega \times [0,T]$ domain. For instance, for Dirichlet boundary conditions or, in general, when the solution is known at the collocation points (x_j^s, t_j^s), this part may take the form

$$\mathcal{L}_s = \frac{1}{N_s} \sum_{j=1}^{N_s} [u(x_j^s, t_j^s) - u_j]^2.$$

Differentiations, when required, are typically carried out by automatic differentiation (AD) [21] which is a key enabler of PINNs. By computing derivatives analytically through the computational graph of the neural network, PINNs avoid potential numerical issues associated with finite difference approximations. Libraries such as TensorFlow [1] and PyTorch [184] make AD straightforward and efficient.

For the implementation of the PINN, a fully connected feedforward neural network is typically used. The input layer receives spatial and temporal coordinates (x, t), while the output layer predicts the physical quantity of interest $u(x, t)$. The network parameters are updated via backpropagation using stochastic gradient descent or variants like Adam or L-BFGS (see, e.g., [94]). DeepXDE [155] is a powerful python package for PINNs which includes and automates many of the previously mentioned methods.

One of the most powerful features of PINNs is their ability to solve inverse problems, such as identifying unknown parameters of a PDE under consideration or initial conditions from sparse observations [195, 155]. In these works, we assume a prior knowledge or guess of the functional form of the PDE. On the other hand, in [192] a configuration of two NNs is used. The first is used for the automatic differentiations needed which are realized through TensorFlow while the second provides the approximation of the discovered PDE. This treatment has the advantage that it doesn't need any prior assumption on the form of the PDE but has the disadvantage that the NN approximation is a "black box" one and does not provide any closed form equation.

The applicability of PINNs is extremely wide since they can be used in every system described by PDEs. A traditional class of examples involves **Fluid Dynamics** where they have been used to solve Navier-Stokes equations in fluid flow problems [195, 196]. For instance, the reconstruction of velocity and pressure fields from sparse measurements around an obstacle showcases their potential in real-time simulation and flow control. Another field of application is this of **Solid Mechanics** and in particular in elasticity and plasticity modeling [259]. For example, PINNs can model stress-strain relationships under dynamic loading without relying on dense meshes or predefined constitutive laws. In **heat conduction** problems, PINNs can predict temperature distributions over time with limited boundary condition data, offering faster alternatives to traditional solvers [167], while they are also used in **Biomedical Applications** including blood flow modeling, cardiac electrophysiology, and tumor growth simulations [126]. Their ability to integrate prior knowledge with sparse patient data makes them suitable for personalized medicine.

PINNs offer several benefits over conventional numerical solvers: First of all, there is no need for mesh-discretization of the integration domain, which simplifies implementation and improves flexibility. This can become especially valuable in suitably high dimensional examples [195, 217]. The method can also easily generalize from sparse or noisy datasets by leveraging physical laws. The ability of PINNs to estimate model parameters and unknown sources directly is also invaluable. Finally, they can easily be

applied to a variety of linear and nonlinear PDEs across different scientific fields.

Despite the important advantages of PINNs there are also some limitations that have to be taken under consideration. Training a PINN can be slow and occasionally unstable. The loss landscapes are often highly non-convex, thus rendering questionable the potential result of convergence and occasionally needing numerous realizations to ensure a meaningful result. Moreover, gradients from physics-based losses can vanish or explode, particularly in complex domains. PINNs often struggle with stiff PDEs and multi-scale problems due to the difficulty in capturing disparate scales with a single network. Adaptive sampling and domain decomposition techniques have been proposed to mitigate this [109]. The computational cost can also be a serious caveat. Although PINNs avoid meshing, their computational cost can be high due to the need for evaluating derivatives via AD and performing global optimization over large parameter spaces. Finally, when the physics-based constraints are not sufficiently strong or the data is too sparse, PINNs may overfit or fail to generalize, especially in high-dimensional problems.

To address some of the above challenges, several variants of PINNs have been proposed, including but not been restricted to the following: **Adaptive PINNs** which modify the sampling of collocation points to focus on areas with higher residuals [166], **XPINNs** in which the domain is partitioned and the corresponding sub-networks are trained independently to improve scalability [108]. In the case of **Fourier Features PINNs (FF-PINNs)**, Fourier embeddings are used to capture fine-scale features and improve convergence [210], while **Bayesian PINNs (B-PINNs)** incorporate uncertainty quantification using Bayesian methods or dropout-based inference [243].

2.1.3. Deep Operator Learning. In traditional supervised learning, neural networks approximate functions, mapping inputs to outputs drawn from finite-dimensional spaces (e.g., $\mathbb{R}^n \to \mathbb{R}^m$). In contrast, operator learning seeks to approximate mappings between infinite-dimensional function spaces, such as learning the solution operator that maps an initial condition or forcing term of a PDE to its solution. This distinction is crucial.

While function learning captures pointwise relationships, operator learning captures entire transformations of functions. Unlike traditional neural networks, neural operators are resolution-invariant, that is, once trained, they can accept new functions a evaluated at the same sensor points, regardless of the original function's discretization. This makes operator learning especially suitable for scientific machine learning tasks where the underlying objects of interest are PDE solution operators rather than isolated function values.

In this section, we briefly discuss two very successful approaches to operator learning. The first approach we discuss involves Deep Operator Networks (DeepONets), introduced by Lu, Jin, and Karniadakis [153], which are neural architectures designed to learn nonlinear operators between Banach spaces. The key innovation of DeepONets is the separation of the input function and the evaluation location through two coordinated neural networks; one is called the *branch network* and the other the *trunk network*.

To begin, let us state a more precise mathematical setting for the purposes of operator learning. Let $\mathcal{G} : \mathcal{A} \to \mathcal{B}$ be a nonlinear operator mapping between Banach spaces, where $\mathcal{A}$ is a function space that can be sampled from (e.g., $C(\Omega)$) and $\mathcal{B}$ is typically a space of scalar-valued functions on a domain Y. The goal is to learn the mapping $a \mapsto \mathcal{G}(a)(y)$ for any $a \in \mathcal{A}$ and any $y \in Y$. To construct a data-driven approximation of $\mathcal{G}$, we assume access to a finite collection of pairs $\{a_j, \mathcal{G}(a_j)(y_i)\}$, where a_j are input functions sampled from $\mathcal{A}$ and $\{y_i\}$ are locations at which the output is evaluated.

The DeepONet is defined as a neural network that approximates the operator $\mathcal{G}$ via the ansatz

$$\mathcal{G}_\theta(a)(y) := \sum_{k=1}^{p} b_k(a(x_1), \ldots, a(x_m))\, t_k(y), \tag{1}$$

where $a(x_1), \ldots, a(x_m)$ are the evaluations of the input function a at fixed sensor locations $\{x_i\}_{i=1}^m \subset D$, $b_k : \mathbb{R}^m \to \mathbb{R}$ is the output of the *branch network*, $t_k : Y \to \mathbb{R}$ is the output of the *trunk network*, evaluated at y, and p is the number of basis functions in the network, i.e., the dimension of the internal representation. The branch and trunk networks are typically standard feedforward neural networks, parameterized jointly by θ. The architecture enforces a bilinear structure in the final layer.

A foundational result in [153] shows that DeepONets are universal approximators of continuous nonlinear operators:

Theorem 1 (Universal Approximation of Operators)**.** Let $K \subset C(D)$ be a compact set, and let $\mathcal{G} : K \to \mathbb{R}$ be a continuous operator. Then, there exists a DeepONet $\mathcal{G}_\theta$ of the form above such that

$$\sup_{a \in K} |\mathcal{G}_\theta(a)(y) - \mathcal{G}(a)(y)| < \varepsilon$$

for any $\varepsilon > 0$ and fixed $y \in Y$.

For a more precisely stated version of this result, please see Theorem 5 in [153]. Of course, this result relies on the classical universal approximation theorem for neural networks and relies on many of the assumptions therein. Nevertheless, the intention here is to communicate that the DeepONet ansatz is theoretically capable of representing arbitrary continuous operators.

Concerning training, the model is trained on a dataset $\{a^{(j)}, y^{(j)}, \mathcal{G}(a^{(j)})(y^{(j)})\}_{j=1}^N$ to minimize a loss function, typically the empirical L^2 loss:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{j=1}^{N} \left| \mathcal{G}_\theta(a^{(j)})(y^{(j)}) - \mathcal{G}(a^{(j)})(y^{(j)}) \right|^2. \tag{2}$$

Optimization is performed using standard techniques such as stochastic gradient descent. Key hyperparameters and choices in the DeepONet design include the number of sensor points m used to discretize the input function, the number of basis functions

p, determining the dimensionality of the internal representation, and the network architecture including the depth of branch/trunk subnetworks.

DeepONets have been applied to a wide variety of problems, including parametric PDE solvers [152, 143], surrogate modeling [234, 146], inverse problems [110, 24], and control [154, 189]. Their theoretical grounding and empirical success have positioned them as a foundational architecture for data-driven operator learning [131].

Fourier Neural Operators (FNOs), introduced by Li et al. [144], are a class of neural architectures designed to learn mappings between function spaces by lifting functions into Fourier space, performing global convolutions, and mapping back to physical space. Just like DeepONets, FNOs are especially suited for learning solution operators of parametric PDEs, offering resolution-invariant and efficient surrogates for complex physical models.

In contrast to DeepONets, which approximate operators through a branch–trunk decomposition, the FNO realizes $\mathcal{G}$ as the composition of spectral integral operators. At layer l, the network represents the state as a vector-valued function $v^{(l)} : D \to \mathbb{R}^d$, where for each spatial location $x \in D$, the vector $v^{(l)}(x)$ encodes d latent features of the input. This representation can be interpreted as a graph signal on the discretization of D, with nodes given by spatial locations and features attached to each node. An FNO layer then performs message passing in the Fourier domain, updating the features according to

$$v^{(l+1)}(x) \;=\; \sigma\Big(W v^{(l)}(x) \;+\; \mathcal{F}^{-1}\Big(R \cdot \mathcal{F}(v^{(l)})\Big)(x)\Big)\,, \tag{3}$$

where W is a pointwise linear transformation shared across x, $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier and inverse Fourier transforms applied channel-wise, R is a learnable Fourier-domain filter (typically diagonal in frequency), and σ is a nonlinear activation. Stacking L such layers defines the operator approximation $\mathcal{G}_\theta$. In this sense, the FNO may be viewed as a graph neural network with globally coupled message passing implemented efficiently through spectral convolution.

For concreteness, in one spatial dimension with input $v \in \mathbb{R}^{n_x \times d}$, a Fourier layer can be described in the following steps:

(i) Compute the Fourier transform $\widehat{v} = \mathcal{F}(v)$.

(ii) Retain the lowest K Fourier modes (spectral truncation).

(iii) Apply learnable weights $\widehat{R}_k$ mode-wise: $\widehat{v}_k \mapsto \widehat{R}_k \widehat{v}_k$ for $|k| \le K$.

(iv) Set higher modes to zero: $\widehat{v}_k = 0$ for $|k| > K$.

(v) Transform back to physical space with the inverse FFT.

This spectral filtering amounts to a parameter-efficient, resolution-agnostic convolution with a global receptive field.

To summarize in this simple setting, consider a one-dimensional input $a \in \mathbb{R}^{n_x \times d_a}$ defined on n_x spatial grid points with d_a input channels. The FNO architecture consists of three stages:

(i) **Lifting layer:** apply a pointwise linear map $P : \mathbb{R}^{d_a} \to \mathbb{R}^d$ at each grid point. Concretely, for every x_i, the input vector $a(x_i) \in \mathbb{R}^{d_a}$ is mapped to a latent feature vector $v^{(0)}(x_i) = P\,a(x_i) \in \mathbb{R}^d$. This produces the initial feature representation $v^{(0)} \in \mathbb{R}^{n_x \times d}$.

(ii) **Fourier layers:** propagate the latent features through L stacked spectral layers of the form in Equation (3), which combine pointwise linear updates with spectral convolutions.

(iii) **Projection layer:** apply a final pointwise linear map $Q : \mathbb{R}^d \to \mathbb{R}^{d_u}$ at each grid point to map the latent representation back to the physical output. For each x_i, $u(x_i) = Q\,v^{(L)}(x_i) \in \mathbb{R}^{d_u}$, yielding the network prediction $u \in \mathbb{R}^{n_x \times d_u}$.

Just as with DeepONet, learning a FNO, that is, learning the the lifting, Fourier, and project layers, can be realized by solving an empirical risk minimization problem that uses Equation (2) as the loss function.

A notable property of FNOs is *resolution invariance*: since the Fourier transform is naturally defined in function space, a model trained on one discretization can be evaluated on finer or coarser meshes without retraining. Computationally, FNOs scale as $O(n_x \log n_x)$ due to the FFT, with only $K \ll n_x$ Fourier modes retained. Compared to convolutional neural networks, FNOs achieve global receptive fields at every layer and demonstrate faster convergence on many PDE-based tasks.

Fourier Neural Operators (FNOs) have been applied across a broad range of scientific machine learning tasks. They were first introduced as resolution-invariant solvers for parametric elliptic and time-dependent PDEs such as Burgers, Darcy flow, and the Navier–Stokes equations [144], and have since been extended to modeling spatiotemporal turbulence [12] and large-scale climate and weather dynamics, including urban microclimate simulation and ocean circulation prediction [186, 56]. Beyond forward modeling, FNOs have also been employed for faster-than-real-time PDE inference and control in fluid dynamics [200], as well as in PDE-constrained optimization tasks such as optimal boundary control for nonlinear optics [163]. Finally, stochastic adaptations of FNOs have been developed for surrogate modeling and uncertainty quantification in geophysical and stochastic systems [56]. Given their versatility in PDE solving, dynamical modeling, control, and stochastic inference, FNOs, and operator learning more broadly, are poised to become central tools in advancing the scientific machine learning of nonlinear waves.

2.2. Sparse and Symbolic Regression

Modeling complex dynamical systems is a fundamental challenge across various scientific and engineering disciplines. Traditional approaches often require detailed first-principles knowledge or extensive empirical data. Yet, this is not always possible since, e.g., we may be unaware of the entirety of mechanisms underlying these procedures. Sparse Identification of Nonlinear Dynamics (SINDy) is a data-driven methodology that seeks

to identify governing equations directly from time-series data by exploiting the inherent sparsity in physical system descriptions.

While machine learning and black-box models such as neural networks have gained popularity in system identification, they often lack interpretability. The *SINDy* framework, introduced in the seminal work of [39], offers an appealing alternative. It leverages sparse regression techniques to identify the fewest terms necessary to describe the system's dynamics, leading to interpretable and often physically meaningful models. SINDy is rooted in the assumption that most dynamical systems can be described by a small number of active terms out of a large possible/plausible function library. By combining time-series data and sparse regression, the method identifies these active terms, producing parsimonious models with potentially valuable physical interpretations.

The SINDy framework works by first considering a continuous-time dynamical system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \tag{4}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{f}$ represents the unknown nonlinear dynamics. The key assumption is that $\mathbf{f}$ can be approximated as a sparse linear combination of functions from a predefined library, $\Theta(\mathbf{x}) = [\theta_1(\mathbf{x}), \ldots, \theta_p(\mathbf{x})]$, where the basis functions $\theta_i(\mathbf{x})$ may consist, e.g., of polynomials or trigonometric functions:

$$\mathbf{f}(\mathbf{x})^T \approx \Theta(\mathbf{x}^T)\Xi, \tag{5}$$

where $\Theta(\mathbf{x}^T) \in \mathbb{R}^{1\times p}$ is the library evaluated on the data, and $\Xi \in \mathbb{R}^{p\times n}$ contains the sparse coefficients. In order to calculate Ξ, we first construct the matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, by considering a set of m snapshots $\mathbf{x}_i = \mathbf{x}(t_i)$ of the state vector,

$$\mathbf{X} = \begin{bmatrix} | & | & | & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_m \\ | & | & | & | \end{bmatrix}^T$$

and the corresponding matrix $\dot{\mathbf{X}}$

$$\dot{\mathbf{X}} = \begin{bmatrix} | & | & | & | \\ \dot{\mathbf{x}}_1 & \dot{\mathbf{x}}_2 & \ldots & \dot{\mathbf{x}}_m \\ | & | & | & | \end{bmatrix}^T \approx \frac{d\mathbf{X}}{dt},$$

which contains the corresponding time-derivatives. These can be computed using finite differences or smoothed estimates.

Then, the key element of the method is to construct an appropriate library of candidate nonlinear functions evaluated on the data. A typical such matrix $\Theta(\mathbf{X}) \in \mathbb{R}^{m\times p}$ contains basic functions like polynomials or trigonometric functions:

$$\Theta(\mathbf{X}) = \left[\mathbf{1}, \mathbf{X}, \mathbf{X}^2, \sin(\mathbf{X}), \cos(\mathbf{X}) \ldots\right].$$

Here, higher polynomials are denoted as $\mathbf{X}^2$, $\mathbf{X}^3$, where $\mathbf{X}^2$ denotes the quadratic nonlinearities in the state x:

$$\mathbf{X}^2 = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}$$

Note though, that the choice of the vocabulary elements is free and depends on the imagination of the researcher, or/and their understanding of the physical/chemical/biological processes involved. The only objective is the best and more interpretable description of the actual system within the reach of the selected library.

Since we are looking for a matrix $\mathbf{X}$ for which it is $\dot{\mathbf{X}} \simeq \Theta(\mathbf{X})\Xi$, one could assume that it would be sufficient to solve a minimization problem of the form

$$\boldsymbol{\Xi} = \underset{\boldsymbol{\Xi}'}{\operatorname{argmin}} \|\dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}'\|_F^2. \tag{6}$$

But the heart of the SINDy method lies in assuming that this coefficient matrix should also be sparse. This way, the resulting dynamics will be, if not more physically relevant, for sure more interpretable. In order to perform this sparse regression one can consider the minimization problem:

$$\boldsymbol{\Xi} = \underset{\boldsymbol{\Xi}'}{\operatorname{argmin}} \left(\|\dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}'\|_F^2 + \lambda \|\boldsymbol{\Xi}'\|_1 \right) \tag{7}$$

instead of (6). The addition of the 1-norm penalizes the small entries in the $\boldsymbol{\Xi}$ matrix, while λ is a sparcity parameter which controls the strength of the penalization, namely the largest the value of λ the more coefficients become zero in $\boldsymbol{\Xi}$. This problem is solved through the LASSO method [103]. Although LASSO is a very well known and popular method, for sparse results in minimization problems it ends up that it can be computationally inefficient for SINDy since it can often produce very small but non zero coefficients [252]. To circumvent this issue the Sequentially Thresholded Least Squares (STLSQ) method was introduced and it has also been used in original SINDy paper [39].

In STLSQ, the original least squares problem (6) for $\boldsymbol{\Xi}$ is solved as $\boldsymbol{\Xi} = \boldsymbol{\Theta}^{\dagger}\dot{\mathbf{X}}$. Then, all coefficients below the value of the sparsity coefficient (which now plays the role of a threshold) are zeroed out. After that we refit the model using only the remaining terms. This process iterates until convergence, producing a sparse and interpretable dynamical model. The STLSQ is straightforward, efficient and produces sparse results but it also bears shortcomings.

One of the alternatives which has been proposed in order to to address some practical issues of LASSO and STLSQ is the Sparse Relaxed Regularized Regression (or SR3, for short) algorithm [252, 50], which is also used in PySINDy [71, 117], a Python package for SINDy including many of the features mentioned here. There, by

introducing an alternative function and an extra term in (7), the algorithm manages to (i) better handle outliers and corrupt data within noisy sensor measurements, (ii) to consider the parametric dependencies in candidate library functions, and (iii) impose physical constraints of the problem.

A straightforward variant of SINDy involves using it in order to discover **discrete-time dynamical systems** (see, e.g., [41]) of the form

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n). \tag{8}$$

This kind of systems could originate from time series exploring a naturally discrete phenomenon or the periodic measurement of a continuous procedure. The corresponding minimization problem would be:

$$\boldsymbol{\Xi} = \underset{\boldsymbol{\Xi}'}{\operatorname{argmin}} \left(\|\mathbf{X}_{n+1} - \boldsymbol{\Theta}(\mathbf{X_n})\boldsymbol{\Xi}'\|_F^2 \right)$$

where $\mathbf{X}_n$ is the matrix of the specific now (discrete-time) snapshots of the system and $\mathbf{X}_{n+1}$ its corresponding images through (8). For the sparse identification, all of the minimization techniques mentioned in the continuous case can be used.

The second obvious variant was to also consider the discovery of **PDEs**. This was done also in the original paper [39] as well in [207, 214]. In this case the methodology is the same but the size of the matrix of the candidate dictionary increases significantly. For example, in [214] where a PDE of the second order (in space) with second degree nonlinearities of the form $u_t = F(u, u_x, u_{xx})$ is considered, the library matrix should be of the form

$$\boldsymbol{\Theta} = \left[\mathbf{1}, \mathbf{U}, \mathbf{U}^2, \mathbf{U}_x, \mathbf{U}_x^2, \mathbf{U}\mathbf{U}_x, \mathbf{U}_{xx}, \mathbf{U}_{xx}^2, \mathbf{U}\mathbf{U}_{xx}, \mathbf{U}_x\mathbf{U}_{xx} \right],$$

where $\mathbf{U}$ is the spatially discretized version of u. One can easily understand that if one would consider higher derivatives or nonlinearities the library expands rapidly, and the identification of the sparse dynamics becomes quickly significantly harder. One can also appreciate the relevant additional complications further in the case where the field u is no longer scalar.

There are also several important extensions of the SINDy method. First of all **SINDYc** which extends SINDy to systems with control inputs of the form

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}),$$

where $\mathbf{u}(t)$ is the control input. The library is expanded to include terms involving both $\mathbf{x}$ as well as $\mathbf{u}$.

On the other hand, **Implicit SINDy** [162] has been introduced to handle systems with algebraic constraints or implicit dynamics:

$$\mathbf{g}(\mathbf{x}, \dot{\mathbf{x}}) = 0.$$

The corresponding minimization problem is

$$\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\Xi} = \mathbf{0} \tag{9}$$

where the library matrix $\boldsymbol{\Theta}$ is generalized to include functions of x and $\dot{x}$. However, this approach requires solving for a sparse matrix $\boldsymbol{\Xi}$ in the null space of $\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})$ which leads to highly ill-conditioned computations for noisy data. This causes the whole procedure to commonly relax to the trivial solution. In order to stabilize the relevant procedure, the **SINDy-PI** method [113] has been proposed which also includes a parallel realization. The method relies on the idea that if even a single term of the dynamics is known which corresponds to a column $\theta_j(\mathbf{x}, \dot{\mathbf{x}}) \in \boldsymbol{\Theta}(\mathbf{x}, \dot{\mathbf{x}})$, it is possible to rewrite (9) as

$$\theta_j(\mathbf{X}, \dot{\mathbf{X}}) = \boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}} | \theta_j(\mathbf{X}, \dot{\mathbf{X}}))\xi_j,$$

where $\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}} | \theta_j(\mathbf{X}, \dot{\mathbf{X}}))$ is the library $\boldsymbol{\Theta}(\mathbf{X}, \dot{\mathbf{X}})\boldsymbol{\Xi}$ with the column θ_j removed. Now, the problem is not implicit anymore and can be solved as a usual sparse minimization problem avoiding the relaxation to zero. **Weak Form SINDy** [168, 169] has been proposed to deal with the problem of noise sensitivity of the method. Since numerical differentiation is very sensitive to noisy measurements, in these works the corresponding derivatives are replaced with integrals that are more robust in that sense.

The method has already been applied to several disciplines including but not restricted to Fluid Mechanics, in modeling wake dynamics and vortex shedding [39], Epidemiology, in deriving models for infectious disease spread [48], Control Systems, in learning dynamics for model predictive control (MPC) [114], Biological Networks, in reconstructing gene and neural systems from expression data [124], and Chemical Reactions systems like the Belousov–Zhabotinsky reaction [113].

Despite its strengths, SINDy faces notable challenges. *Noise Sensitivity*: Derivative estimation is sensitive to noise. Approaches such as total variation regularization, Gaussian processes, and weak formulations help mitigate this issue. *Library Selection*: Selecting an expressive yet efficient function library remains critical and often requires domain expertise. Symbolic regression and neural networks are emerging to automate this. *Scalability and High Dimensions*: For high-dimensional systems such as PDEs, SINDy requires dimensionality reduction techniques like Proper Orthogonal Decomposition (POD) or manifold learning.

There are numerous promising future research directions. *Neural-SINDy Hybrids*: this would involve integrating SINDy with neural networks for structure discovery while retaining interpretability. *Bayesian SINDy*: this emerging approach accounts for uncertainty in model structure and parameters. Finally, another challenge is that of *Real-Time Systems*, where it can contribute to improving performance for real-time system identification and control.

Concluding, SINDy provides a powerful framework for discovering interpretable models of nonlinear systems from data. By leveraging sparsity, it uncovers governing equations that offer both predictive accuracy and potentially physical insight. With

continued development—particularly in noise robustness, scalability, and integration with deep learning, SINDy continues to bear significant potential as a pillar of data-driven scientific discovery. Having now presented some of these emergent pillars, we turn to a number of concrete, recent applications thereof in the field of Nonlinear Waves.

3. Applications of Physics-Informed Neural Networks

3.1. PINNs for Learning Nonlinear Dynamics on Lattices

One of the recent examples of the application of PINNs has been in the consideration and potential discovery of 1D nonlinear dynamical lattices consisting of a finite number (N) of nodes in the work of [212]. Such lattices have been one of the workhorses of nonlinear wave theory, given their broad relevance to a number of settings [13, 83, 87, 120]. Among the many relevant applications, we mention atomic Bose-Einstein condensates in optical lattices [174], optical waveguide arrays [138], micromechanical oscillator arrays [213] nonlinear electrical circuits [199], engineered granular (metamaterial) crystals [220, 57], antiferromagnetic crystals [80], and superconducting settings of Josephson-junction ladders [25, 229].

The relevant models that were considered for the (real or complex, $n = 1, \ldots, N$ node) field $u_n(t)$ included, e.g., the discrete ϕ^4 model [64]

$$\ddot{u}_n = C(u_{n+1} + u_{n-1} - 2u_n) + 2(u_n - u_n^3), \quad u_n \in \mathbb{R}, \tag{10}$$

where an overdot stands for the temporal derivative of u_n, and $C = 1/h^2 (> 0)$ is the coupling constant (and h the lattice spacing) between adjacent nodes. Also relevant to this type of study was the discrete sine-Gordon (DsG) [66], often also referred to as the Frenkel-Kontorova model [32]:

$$\ddot{u}_n = C(u_{n+1} + u_{n-1} - 2u_n) - \sin(u_n), \quad u_n \in \mathbb{R}. \tag{11}$$

This model similarly to the discrete ϕ^4 showcased the existence of kink (and antikink) solutions.

Another prototypical model that has been found to be physically relevant consisted of the discrete nonlinear Schrödinger (DNLS) equation [122], in particular with a focusing nonlinearity:

$$i\dot{u}_n = -C(u_{n+1} + u_{n-1} - 2u_n) - |u_n|^2 u_n, \quad u_n \in \mathbb{C}. \tag{12}$$

Finally, one more key model of broad interest, including the important feature of breaking the Hamiltonian character was the discrete, complex Ginzburg-Landau (DCGL) equation:

$$\dot{u}_n = (1+i)C(u_{n+1} + u_{n-1} - 2u_n) - (1-i)|u_n|^2 u_n + u_n, \quad u_n \in \mathbb{C}, \tag{13}$$

with a cubic nonlinearity.

In the work of [212], a number of relevant modifications were introduced in connection to the regular PINNs. In particular, for the case of real dynamical variables $\boldsymbol{u}(t) \in \mathbb{R}^N$, the PINN $\hat{\boldsymbol{u}} : \mathbb{R} \to \mathbb{R}^N$ only involved time t as the input. This was mapped through an L-layer fully-connected neural network to the output forming 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 specific right hand side of the nonlinear model $\mathcal{N}(u_1, \ldots, u_N)$ was not known, an overcomplete library $\mathrm{Lib} = \{D_\alpha\}_{\alpha \in A}$ of shift-invariant discrete spatial operators was utilized. The latter was selected so as to incorporate both the linear couplings between nearest neighbors, and the different forms of nonlinear contributions.

The unknown operator $\mathcal{N} : \mathbb{R}^N \to \mathbb{R}^N$ is then considered to be a linear combination $\mathcal{N} = \sum_{\alpha \in A} \lambda_\alpha D_\alpha$ of elements D_α in the library, and the scope of the relevant effort was to identify the expansion coefficients $\boldsymbol{\lambda} = (\lambda_\alpha)_{\alpha \in A}$, i.e., to solve the associated inverse problem by minimizing the loss function

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_\mathcal{N}, \mathcal{T}_{\boldsymbol{f}}) \coloneqq w_\mathcal{N} \mathcal{L}_\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_\mathcal{N}) + w_{\boldsymbol{f}} \mathcal{L}_{\boldsymbol{f}}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_{\boldsymbol{f}}). \tag{14}$$

Here

$$\mathcal{L}_\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\lambda}; \mathcal{T}_\mathcal{N}) = \frac{1}{|\mathcal{T}_\mathcal{N}|} \sum_{t \in \mathcal{T}_\mathcal{N}} \left| \dot{\hat{\boldsymbol{u}}}(t; \boldsymbol{\theta}) - \sum_{\alpha \in A} \lambda_\alpha D_\alpha \hat{\boldsymbol{u}}(t; \boldsymbol{\theta}) \right|^2, \tag{15}$$

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

with $\mathcal{T}_\mathcal{N}$, $\mathcal{T}_{\boldsymbol{f}}$, respectively, being subsets of $[0, T]$ representing the training collocation points. It is, in particular, at these points that the ODE residual and the discrepancy between $\hat{\boldsymbol{u}}$ and the observed $\boldsymbol{f}$ was minimized.

The initial value problem was solved with a suitable integrator, e.g., 4th order Runge-Kutta for the case of the ϕ^4 model that we will detail below. As relevant initial conditions, a traveling kink soliton was used for the ϕ^4 data generation. The DeepXDE library of [155] was used and appropriately modified in order to consider a first order system (e.g., for positions and velocities in the ϕ^4 case), with t as the relevant input variable. The neural network used in all the experiments considered only interior nodes in the relevant loss function. Moreover, the associated architecture involved fully-connected networks of three hidden layers with 40 neurons each, and utilizing a tanh activation function.

In what follows, we report the prototypical results from the ϕ^4 case considered in [212], which were representative of the relevant findings also for the DsG, as well as for the DNLS and the DCGL models reported in the same publication. In the case of Fig. 1(a), a library set of terms including

$$\mathrm{Lib}^{(1)} = \Big\{ \left(u_{n+1} + u_{n-1} - 2u_n\right), \left(u_{n+1} - u_{n-1}\right)/2, u_n, u_n^3 \Big\}, \tag{17}$$

was utilized, inspired by the continuum variant of the ϕ^4 model. It can be clearly seen that the PINN learns the correct coefficients, leading the existing coefficients (including the nonlinearity and the second difference one) to acquire their expected values, while the "discrete derivative" term is accurately recognized as featuring a vanishing prefactor.

Extending this perspective in Figs. 1(b)-(d), a more "inherently discrete" approach to the relevant problem was followed and an expanded library function choice was made. The relevant libraries were of the form.

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

$$\mathrm{Lib}^{(3)} = \left\{\mathrm{Lib}^{(2)}, u_{n+2}, u_{n-2}\right\}, \tag{19}$$

and

$$\mathrm{Lib}^{(4)} = \left\{\mathrm{Lib}^{(2)}, u_{n+1}^2 u_n, u_{n-1}^2 u_n, u_{n+1} u_{n-1} u_n, u_{n+1}^2 u_{n-1}, u_{n-1}^2 u_{n+1}, u_n^2 u_{n+1}, u_n^2 u_{n-1}, u_{n+1}^3, u_{n-1}^3\right\}. \tag{20}$$

Among these, $\mathrm{Lib}^{(2)}$ was deemed to be the simplest example encompassing the principal ingredients of the model at hand. $\mathrm{Lib}^{(3)}$ constituted an extension of $\mathrm{Lib}^{(2)}$ encompassing the next-nearest neighbors, namely, $u_{n\pm2}$ are appended therein. Finally, $\mathrm{Lib}^{(4)}$ further expanded on the possibilities of the cubic nonlinear term including several options towards the cubic nonlinearity of the model.

In all the cases depicted in Figs. 1(b)-(d), the nonlinear dynamical lattice model was accurately "discovered". This entails both all the enclosed terms being identified correctly as "participating" in the model and with the right prefactors, as well as "extraneous" terms reaching a converged vanishing prefactor, thus revealing their absence from the model dynamics. That being said, features such as the role of symmetries (that will be further explored in what follows through Structure-Preserving PINNs) was found to be of importance here. More specifically, in the case where the libraries contained quadratic or quartic terms, data augmentation and the fact that $-u$ is a solution if u is a solution were used in order for the model to converge to the appropriate coefficients.

3.2. Graph Neural Networks Learn Short and Long Range Interactions on Lattices

While for many of the tasks of interest, using "vanilla" feedforward neural networks will suffice, for specific tasks, such as the discovery of complex interaction patterns between elements (particles) in lattice systems, it has been argued that more specialized neural networks, such as, e.g., graph neural networks, may offer superior performance [92, 89]. Indeed, it has been argued therein that since data that exhibits relationships between elements can be modeled as a graph, graph neural networks are a rather natural choice. Here, elements are named graph nodes, and the relationships between them are naturally represented by edges. Accordingly, the graph can serve as a systematic representation

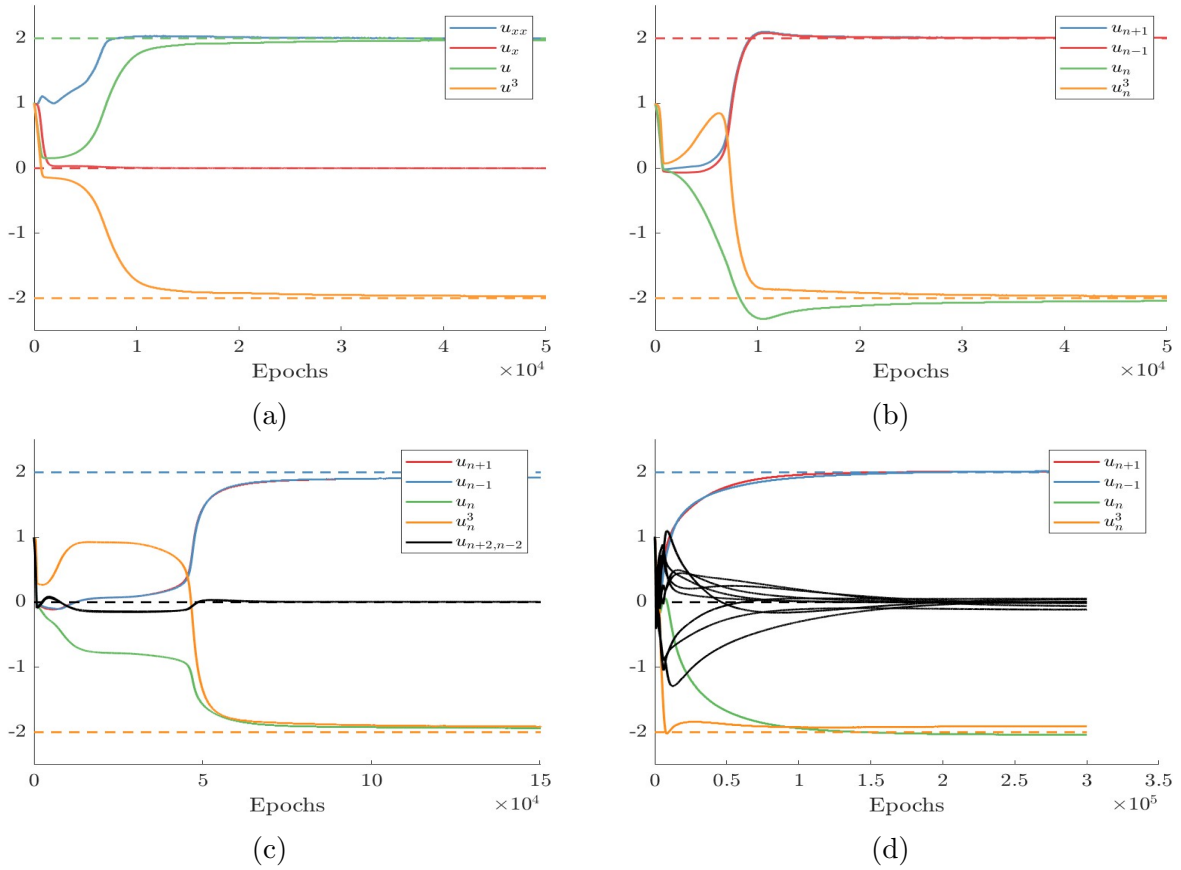


Figure 1: Discrete ϕ^4 model numerical results [cf. Eq. 10] for a coupling of $C = 2$, adapted from Ref. [212]. The library $\text{Lib}^{(1)}$ of Eq. (17) was considered in panel (a) with the solid blue, red, green and yellow lines corresponding to the discrete representation of the second and first derivative, as well as u, and u^3, respectively. The numerical results obtained by using the library $\text{Lib}^{(2)}$ [cf. Eq. (18) 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. Panels (c) and (d) utilized the libraries of Eqs. (19) and (20), respectively (with the same notational conventions). The solid black lines therein correspond to (c) the terms $u_{n\pm2}$, and (d) to all the other cubic terms. It is relevant to highlight that the dashed lines represent the reference values for the coefficients.

of lattice (nonlinear) dynamical systems. In that vein, Graph Neural Networks (GNNs) are rapidly emerging as a new field for studying nonlinear dynamical lattices.

The paradigms explored initially in [92] and subsequently in [89] concerned many degrees of freedom and complex interactions. It was in such settings where GNNs were found to exhibit a proficient ability to accurately identify key information around nodes, ultimately leading to an improved accuracy of the model discovery. This was inspired also by the outstanding performance of GNNs in some classic systems, e.g., in the works of [211, 27]. Indeed, lattice systems such as gravitationally interacting celestial bodies, can be formulated on the basis of fully connected graphs, since they involve interactions

between all particles. This may allow the capturing of even long-range interactions [136].

The approach of [92, 89] involved the representation of relationships in the form of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1 \cdots v_N\}$ being the set of nodes and $\mathcal{E} = \{e_{ij}\}$ the set of (directed) edges between nodes. The method then consisted of two parts, with the first one adjusting the weight of the edge through trajectory data to extract the underlying interaction between particles, while the second part utilized the learned interactions towards the more accurate and effective trajectory prediction. Moreover, the learning was designed to take place in two components, a potential energy learning part (V-net) and a kinetic energy learning part (T-net). Each of these involved a K-layer graph neural network updating nodes (for both the T- and V-nets) and edges (only for the V-net).

Then, assuming that the Lagrangian of the model could be written as the difference of the kinetic energy T_θ minus the potential energy V_θ, the relevant loss function for the first of the two parts indicated above (the structural learning) can be written as:

$$\mathcal{L} = \mathcal{L}_{pred} + \gamma \mathcal{L}_{GL}, \tag{21}$$

where

$$\mathcal{L}_{pred} = \left\| \frac{\partial T_\theta}{\partial \mathbf{p}} - \frac{d\mathbf{q}}{dt} \right\|_2 + \left\| -\frac{\partial V_\theta}{\partial \mathbf{q}} - \frac{d\mathbf{p}}{dt} \right\|_2. \tag{22}$$

Also, here the the graph learning loss $\mathcal{L}_{GL}$ was used in the form:

$$\mathcal{L}_{GL} = \|\alpha\|_F^2, \tag{23}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The choice of the hyperparameter $\gamma \in \mathbb{R}$ balances the two terms. A sparse representation of the graph is sought, to the degree possible, through the graph learning loss. Here, $\alpha = \{\alpha_{i,j}\}$ represents a parameter matrix initialized by a neural network acting on each edge and adaptively labeling the strength of the edges through the training process. It is practically the key to learning the graph structure in the α-SGHN (i.e., separable graph Hamiltonian network) model [89]. After the update has been completed, there is a prediction stage, where the predicted trajectory is obtained from the network as: $(\hat{\mathbf{q}}^t, \hat{\mathbf{p}}^t)$, through integration according to

$$(\hat{\mathbf{q}}^t, \hat{\mathbf{p}}^t) = (\mathbf{q}^0, \mathbf{p}^0) + \int_{t_0}^{t} \alpha\text{-SGHN} dt, \tag{24}$$

with suitable initial values $(\mathbf{q}^0, \mathbf{p}^0)$. Here the loss is defined based on the accuracy of the prediction of the trajectory, i.e., $\mathcal{L} = \mathcal{L}_{pred}$.

The relevant methodology was explored in multiple test examples in [89]. These included the Frenkel-Kontorova lattice (bearing one conserved quantity), with the Hamiltonian:

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + \frac{(q_{i+1} - q_i)^2}{2} + 1 - \cos(q_i) \right), \tag{25}$$

the rotator lattice (which has two conserved quantities) with Hamiltonian:

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + \frac{(q_{i+1} - q_i)^2}{2} + 1 - \cos(q_{i+1} - q_i) \right), \tag{26}$$

and the Toda lattice (which is integrable and bears as many conserved quantities as the number of degrees of freedom) with Hamiltonian:

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + \exp(q_i - q_{i+1}) \right). \tag{27}$$

The evolution over time of the different conserved quantities of these models in the case of the evolution prediction from the α-SHGN model, in comparison with the standard multilayer perceptron (denoted as MLP) and the so-called Hamiltonian neural network (denoted as HNN) [98] is shown in Fig. 2. While there may still be some room for improvement in higher order conserved quantities, generically the α-SHGN model was found to do a consistently better job in associated conservation laws.

3.3. Structure-Preserving PINNs

As explained in Section 2.1.2, PINNs aim to integrate physical principles into the learning process by minimizing losses derived from the underlying differential equations. While this approach has shown success across a range of problems, standard PINNs often fail to capture deeper structural properties of the system, such as symmetries and conservation laws, which are essential for accurately modeling nonlinear dynamics. These limitations are particularly pronounced in nonlinear wave phenomena, where spatio-temporal symmetries, periodicity, and localization critically influence the solution dynamics.

To address these challenges, recent work has focused on structure-preserving variants of PINNs that explicitly encode such invariants into the architecture. A notable example is the development of structure-preserving PINNs (S-PINNs) [257], which are motivated by the special structures intrinsic to solutions of nonlinear dynamical lattices. Consider, for instance, the completely integrable Ablowitz–Ladik (AL) model [4, 3, 122], given by

$$\mathrm{i}\dot{\Psi}_n + (\Psi_{n+1} - 2\Psi_n + \Psi_{n+1}) + (\Psi_{n+1} + \Psi_{n-1}) |\Psi_n|^2 = 0, \tag{28}$$

where $\Psi_n(t) : \mathbb{Z} \times \mathbb{R} \to \mathbb{C}$ denotes the complex wavefunction at lattice site $n \in \mathbb{Z}$ and time $t \in \mathbb{R}$, and $\mathrm{i} = \sqrt{-1}$. Using the ansatz

$$\Psi_n = \psi_n e^{2\mathrm{i}q^2 t}, \tag{29}$$

with background amplitude q^2, Eq. (28) transforms into

$$\mathrm{i}\dot{\psi}_n + (\psi_{n+1} - 2\psi_n + \psi_{n+1}) + (\psi_{n+1} + \psi_{n-1}) |\psi_n|^2 - 2q^2\psi_n = 0. \tag{30}$$

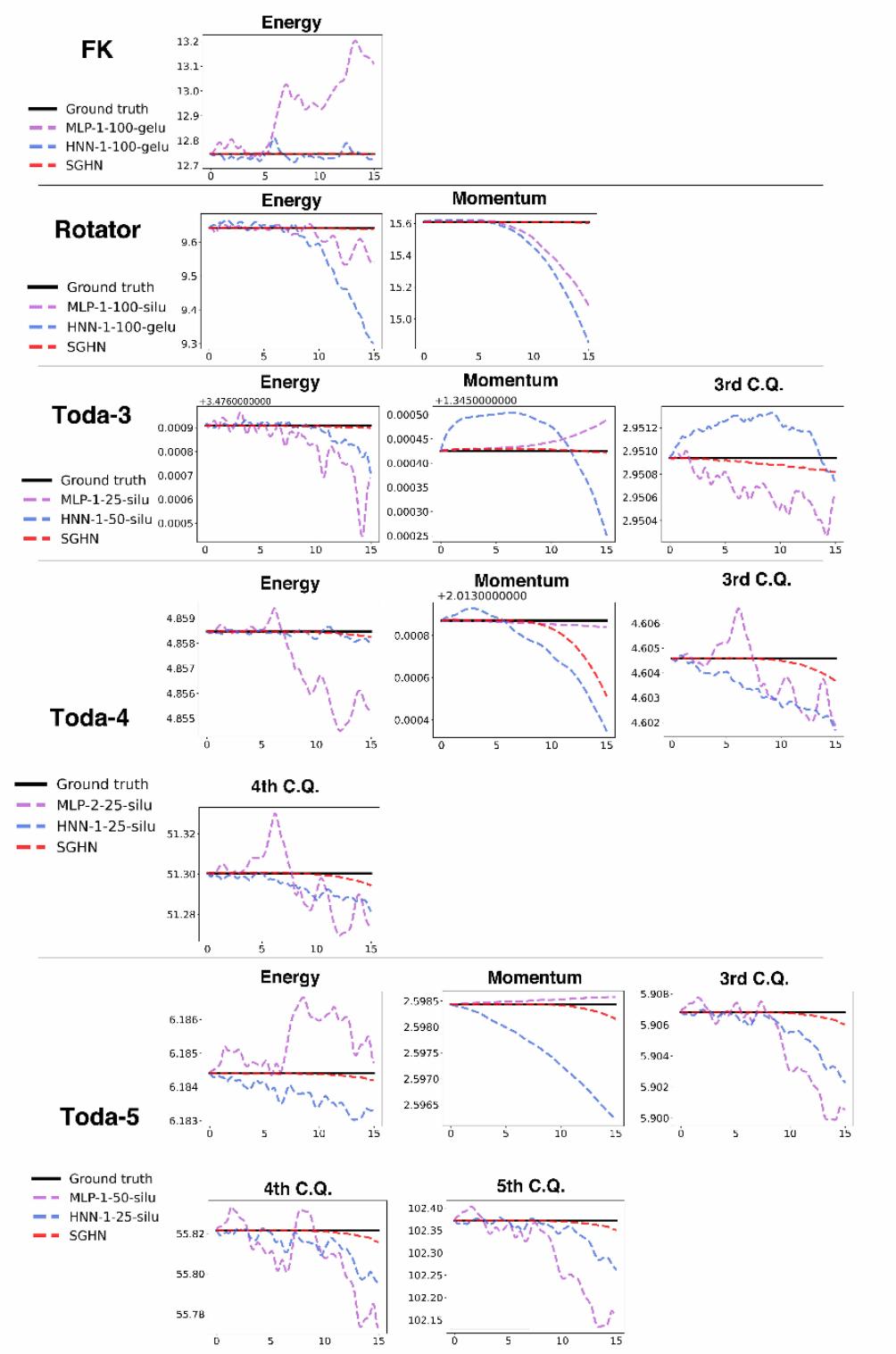


Figure 2: This figure, adapted from [89], illustrates evolution over time of the average true conservation values and the average predicted values of 20 samples for the different networks. C.Q. represents conserved quantity.

Hereafter we set $q \equiv 1/\sqrt{2}$ for simplicity.

Among the exact solutions of Eq.(30), one notable example is the Kuznetsov–Ma (KM) soliton [8, 188], a discrete solution that is temporally periodic and spatially localized:

$$\psi(n,t) := \psi_n(t) = \frac{1}{\sqrt{2}} \frac{\cos(\omega t + \mathrm{i}\theta) + G\cosh(rn)}{\cos(\omega t) + G\cosh(rn)}, \tag{31}$$

where the period is $T = 2\pi/\omega$, and the parameters θ, r, and G are determined by ω through

$$\theta = -\operatorname{arcsinh}(\omega), \quad r = \operatorname{arccosh}\left(\frac{2+\cosh(\theta)}{3}\right), \quad G = -\frac{\omega}{\sqrt{3}\sinh(r)}. \tag{32}$$

Besides temporal periodicity, a key feature of the KM soliton is its spatio-temporal parity symmetry:

$$\psi(n,-t) = \overline{\psi(n,t)}, \quad \psi(-n,t) = \psi(n,t). \tag{33}$$

which reflects the underlying parity and time-reversal invariance of the AL model itself.

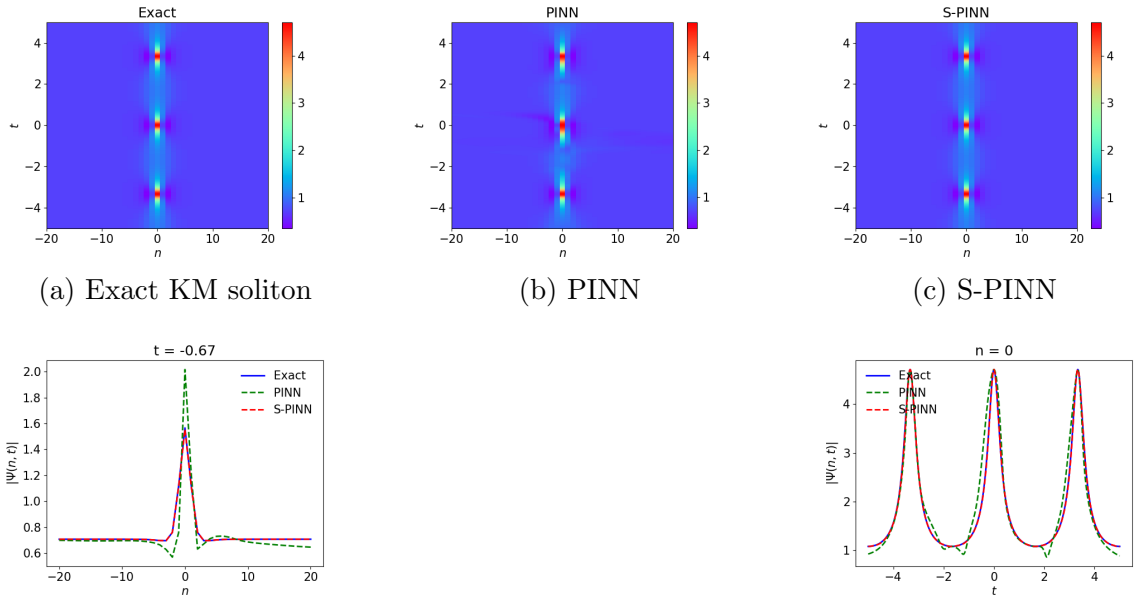
Zhu et al. [257] proposed a principled approach to encode such physical structures directly into the neural network architecture. By enforcing spatio-temporal parity symmetry (33) and temporal periodicity within the network design, they introduced a structure-preserving PINN (S-PINN) that significantly outperformed conventional PINNs. As illustrated in Figure 3, S-PINNs produce markedly more accurate reconstructions of the KM soliton across both spatial and temporal domains. Standard PINNs, by contrast, fail to respect the symmetry and periodicity of the underlying solution, leading to substantial inaccuracies. This highlights the critical role of embedding structural priors when modeling nonlinear wave phenomena with neural networks.

3.4. PINNs for Scale- and Translational-Invariant Wave Solution Identification: the Burgers Equation Case Example

One of the aspects of the usefulness of PINNs that we also briefly touch upon concerns their ability to dynamically evolve equations, including potentially along a group orbit (e.g., of translation or of self-similar rescaling), leading to the identification of potential stationary —or, for that matter, dynamical— states in such frames. Here, we provide a prototypical example along this vein, from the recent work of [119], concerning the self-similar and moving evolution of a wave in the well-known Burgers equation, inspired from the earlier example of [204].

Consider the one-dimensional Burgers equation, also considered in the realm of PINNs in [218]:

$$\partial_t u = \nu \partial^2_{xx} u - u\partial_x u \equiv \mathcal{D}_x(u). \tag{34}$$



(a) Exact KM soliton (b) PINN (c) S-PINN

(d) $\psi(n,t)$ for $t=-0.67$ (e) $\psi(n,t)$ for $n=0$

Figure 3: Numerical results, adapted from [257], for the KM soliton using PINN and S-PINN. Top panels show the spatio-temporal evolution of the amplitude $|\psi(n,t)|$ for the exact solution (a), PINN (b), and S-PINN (c). Bottom panels show the spatial profile at $t=-0.67$ and the temporal evolution at $n=0$. Solid blue lines indicate the exact solution, while dashed green and red lines correspond to PINN and S-PINN, respectively. Standard PINN fails to capture the time-periodicity and spatio-temporal parity symmetry (33), whereas S-PINN accurately preserves both.

The value of viscosity for this example has been set to $\nu = 0.025$. Using scaling arguments starting from an ansatz of the form

$$u(x,t) = Bw\left(\frac{x-c}{A}\right) \equiv Bw(y), \tag{35}$$

one infers that $B = 1/A$.

With this scaling, one obtains:

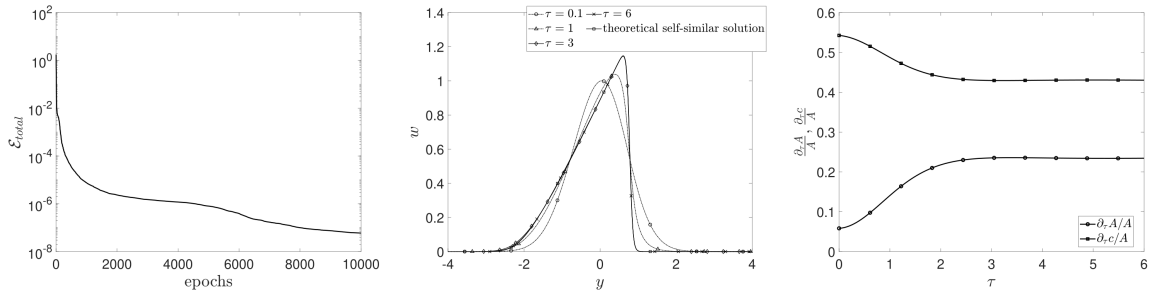
$$\mathcal{D}_x\left(\frac{1}{A}w\left(\frac{x-c}{A}\right)\right) = A^{-3}\mathcal{D}_y(w). \tag{36}$$

This, in turn, inspires the dynamic scaling choice of the form:

$$u(x,t) = \frac{1}{A(\tau)}w\left(\frac{x-c(\tau)}{A(\tau)}, \tau(t)\right), \tag{37}$$

incorporating a rescaling of time $\tau = \tau(t)$, which leads the original Burgers equation to be reshaped as:

$$\partial_\tau w = \nu\partial^2_{yy}w - w\partial_y w + \frac{\partial_\tau A}{A}(w + y\partial_y w) + \frac{\partial_\tau c}{A}\partial_y w, \quad \text{with } y = \frac{x-c}{A}. \tag{38}$$



(a) (b) (c)

Figure 4: PINN prediction of the self-similar dynamics for the Burgers equation adapted from the work of [119]. (a) Convergence of the total loss $\mathcal{E}_{loss}$ during training for the rescaled/co-moving Burgers equation. (d) Snapshots of rescaled PINN predicted solution w at different τ values. The converged self-similar solution w (at $\tau \approx 3$) and the analytically predicted self-similar solution [239] are practically coincident. (c) Evolution of the scaling rates $\partial_\tau A/A$ and $\partial_\tau c/A$ over rescaled time τ. For sufficiently long τ values ($\tau > 3$), we can observe the convergence of the rescaled solution to a stationary, self-similar profile.

The two τ-dependent unknown quantities (as they need to be to ensure the factoring out of the symmetries), $\partial_\tau A/A$ and $\partial_\tau c/A$ are determined by imposing two template constraints, as described in the work of [119, 204]. Each of these conditions eliminates one of the corresponding symmetries/degeneracies, enabling the identification of a unique solution upon application of the PINN methodology. The conditions arise, e.g., from minimizing

$$E \equiv \int_{y_{min}}^{y_{max}} \left(w - \frac{1}{A} T\left(\frac{y-c}{A} \right) \right)^2 \mathrm{d}y, \tag{39}$$

evaluated at $A = 1$ and $c = 0$ with $T(y)$ being a chosen template function. This results in the constraints:

$$\int_{y_{min}}^{y_{max}} (w - T)\,(T + y\partial_y T)\,\mathrm{d}y = 0 \quad \text{and} \quad \int_{y_{min}}^{y_{max}} (w - T)\,\partial_y T \mathrm{d}y = 0. \tag{40}$$

The specific $T(y)$ chosen in [119] was $T(y) = \exp(-y^2)$.

The above theoretical formulation of equation (38) was solved using PINNs in the domain $y \in [-6, 6]$ with zero-flux boundary conditions at both ends in [119]. A similar proposal for the solution of fluid mechanical models —such as variants of the Euler equation— bearing (asymptotic) self-similarity was made earlier in the work of [236]. The initial condition is $w(y, 0) = \exp(-y^2)$. For this implementation, the neural network $\mathcal{N}_w(y, \tau)$ consisted of three hidden layers with 20 neurons each, with a hyperbolic tangent activation function. An auxiliary network $\mathcal{N}_p(\tau)$ was used in order to predict $\partial_\tau A/A$ and $\partial_\tau c/A$ and contained a single hidden layer with 4 neurons. The neural network $\mathcal{N}_w(y, \tau)$ was trained using about 180,000 collocation points in the $y - \tau$ domain. Training was

performed using the L-BFGS optimizer and the corresponding convergence of the total loss $\mathcal{E}_{total}$ is shown in Figure 4. Upon the spatio-temporal evolution, the initial data was found to steepen into a sharp front, eventually leading to a stationary state in the renormalized frame of variables (y, τ); see also [204] and Figure 4a. A comparison of the resulting self-similar solution at $\tau = 6$ with the analytical self-similar solution of the Burgers equation [239]:

$$u(x, t_0) = \sqrt{\frac{\nu}{\pi t^*}} \frac{\left[\exp\left(A^*/\left(2\nu\right)\right) - 1\right] \exp\left[-\left(x - c^*\right)^2 / \left(4\nu t^*\right)\right]}{1 + \left[\exp\left(A^*/\left(2\nu\right)\right) - 1\right] / 2 \cdot \operatorname{erfc}\left(\left(x - c^*\right) / \sqrt{4\nu t^*}\right)}, \tag{41}$$

for a suitable choice of A^*, c^* and t^*, revealed the accuracy of the obtained result. as shown in Figure 4b. In turn, the evolution of $\partial_\tau A/A$ and $\partial_\tau c/A$ over rescaled time τ is shown in Figure 4c, converging to the asymptotic values of the relevant quantities.

3.5. Lax Pair Informed Neural Networks

The recent work of [190] shows how Lax pairs can be used to assist in the training of neural networks to solve integrable systems. A Lax pair of operators, denoted by $(L(t), P(t))$ and depending on time t, consists of operators acting on some fixed Hilbert space that satisfy the *Lax equation*

$$\frac{dL}{dt} = [P, L] := PL - LP. \tag{42}$$

This solvability condition amounts to a so-called integrable nonlinear partial (or lattice) differential equation. For example, it is easy to verify that the KdV equation

$$u_t + 6uu_x + u_{xxx} = 0 \tag{43}$$

has the Lax Pair

$$L = -\partial_x^2 + u, \quad P = 4\partial_x^3 - 3(u\partial_x + \partial_x u). \tag{44}$$

It is well-known that for operators satisfying the Lax equation, one finds the spectral operator $L(t)$ evolves by a similarity transformation:

$$L(t) = Q(t)L(0)Q(t)^{-1}, \quad \dot{Q} = PQ.$$

Therefore, the *spectrum* of $L(t)$ is preserved for all t. This is the algebraic hallmark of integrability, that is, the invariants of motion are the spectral invariants of L (e.g., in finite dimensions the traces $\operatorname{tr}(L^n)$).

In the PDE setting, L and P are typically *differential operators* acting on a spectral function $\psi(x, t)$ through the overdetermined system

$$L(u)\psi = \lambda\psi, \quad \psi_t = P(u)\psi, \tag{45}$$

where λ is the so-called spectral parameter and $u(x, t)$ is the physical field corresponding to original PDE dynamics. The compatibility condition $\psi_{xt} = \psi_{tx}$ yields Equation (42) and hence the nonlinear PDE for u.

Another particularly useful viewpoint, when $x \in \mathbb{R}$, comes from rewriting the Lax pair in terms of *auxiliary linear problems* for the wavefunction $\Phi(x,t)$:

$$\Phi_x = U(u,\lambda)\,\Phi, \quad \Phi_t = V(u,\lambda)\,\Phi, \tag{46}$$

where U and V are matrix- (or operator-) valued functions of the field u and the spectral parameter λ. In this formulation, U plays the role of a spatial connection and V a temporal connection on a trivial vector bundle over the (x,t)-plane. The requirement that these two linear equations be compatible, $\Phi_{xt} = \Phi_{tx}$, is equivalent to the *flatness* of the associated connection

$$U_t - V_x + [U,V] = 0. \tag{47}$$

Equation (47) is known as the *zero curvature condition* or *Zakharov–Shabat equation*. It encodes the original nonlinear PDE in the vanishing of the curvature of a connection depending on the spectral parameter, thereby tying integrability to the machinery of gauge theory.

With this context in mind, a Lax Pair Informed Neural Network (LPNN) can be built depending on the formulation of Lax integrability used. More specifically, in this approach one seeks to *embed* the Lax pair structure into a scientific machine learning model for $u(x,t)$. Just as with PINNs and all of the previous frameworks presented in this section, one regresses in the sense of L^2, alongside boundary and inital data, to learn the trainable parameters that furnish the original PDE variable. The key difference here is that LPNNs are structure preserving in that they also learn the spectral function ψ. This is the essence of what [190] calls LPNN version 1 (LPNN-v1). LPNN version 2 (LPNN-v2) considers, in tandem with version 1, an additional loss term that incorporates the zero-curvature condition, either given by the Zakharov-Shabat or auxiliary linear problem depending on the context.

For a concrete example, consider the Korteweg-de Vries equation which has a spectral function ψ that satisfies

$$\psi_{xx} = (\lambda - u)\psi, \qquad \psi_t = u_x\psi - (4\lambda + 2u)\psi_{xxx}.$$

Since the KdV equation can be derived from the compatibility condition $\psi_{xxt} - \psi_{txx} = 0$, this compatibility condition can be reformulated into a standard loss function for LPNN-v1. Thus, with appropriate boundary and intitial data, the basic idea of LPNN-v1 here is, for a fixed λ, to parametrize $u(x,t)$ and $\psi(x,t)$ with a feedforward fully connected neural network and to regress until the compatibility condition $\psi_{xxt} - \psi_{txx} = 0$ and KdV Equation (96) are approximately satisfied.

As a benchmark, the work of [190] compares LPNN-v1 with a vanilla PINNs approach; see [190] for the computational details. More specifically, it is reported that the LPNN-v1 error, in the sense of L^2, is about twice that of PINNs, yet the training time is about four times less. The work of [190] similarly goes on to study the Camassa-Holm, and the Kadomtsev-Petviashvili, a two-dimensional generalization of the KdV equation, reporting, more or less, similar results.

LPNN-v2 seems to show more promise in that it consistently outperforms PINNs in the resulting accuracy. Perhaps most noteworthy is their result for the modified KdV (mKdV) equation

$$u_t + 6u^2 u_x + u_{xxx} = 0.$$

Here, the auxiliary linear problem involves the following matrices

$$U = \begin{bmatrix} \lambda & u \\ -u & -\lambda \end{bmatrix}, \qquad V = \begin{bmatrix} -4\lambda^3 - 2u^2\lambda & -4u\lambda^2 - 2u_x\lambda - 2u^3 - u_{xx} \\ 4u\lambda^2 - 2u_x\lambda + 2u^3 + u_{xx} & 4\lambda^3 + 2u^2\lambda \end{bmatrix}.$$

LPNN-v2 simply parametrizes the complex two by one vector Φ, along with u, using a neural network and trains until the compatibility condition $\Phi_{xt} - \Phi_{tx} = 0$ is approximately satisfied. The work of [190] reports that LPNN-v2 achieves an accuracy eight times that of PINNs. They further go on to adapt LPNN-v2 to the Sine-Gordon, nonlinear Schrödinger, and Short-Pulse equations, in all cases outperforming PINNs in terms of accuracy.

We comment that LPNNs, as used by [190] are an example of function learning. Instead of learning the function satisfying a PDE $\partial_t u = \mathcal{F}(u)$ and associated spectral functions directly, one may adapt to learn *operator-valued maps*

$$u \mapsto L_\theta(u), \quad u \mapsto P_\varphi(u),$$

with trainable parameters θ, φ, constrained so that the evolution predicted by the network satisfies

$$\frac{d}{dt} L_\theta(u(t)) \approx [P_\varphi(u(t)),\, L_\theta(u(t))]. \tag{48}$$

In an LPNN discretization, L_θ could be represented by a symmetric banded matrix with main diagonal given by u, and P_φ by a skew-symmetric matrix encoding the learned differential stencil. However the discretization is performed, learning L_θ and P_φ can be facilitated by either a DeepONet or FNO as discussed in Section 2.1.3.

Besides outperforming PINNs in terms of accuracy, there are a few advantages to consider going forward with deep learning integrable and nearly-integrable dynamics using LPNNs.

- *Structure preservation:* Spectral invariants are (approximately) preserved by construction, improving long-term stability of learned dynamics.
- *Integrability bias:* By constraining to a (possibly learned) Lax form, the network embeds a strong prior aligned with integrable or near-integrable physics.
- *Interpretability:* After learning L_θ and P_θ there may be opportunities to recover hidden symmetries, conserved quantities, or other approximate integrable structures.

4. Discovering Dynamical Systems and Reduced Order Models from Dispersive PDE

4.1. Use of SINDy to Develop Moment Equation Reductions Stemming from Nonlinear Dispersive PDEs

Often in the context of nonlinear wave equations, it is possible to infer effective information about the system's dynamics (e.g., its center of mass, its variance and the associated wavefunction width, the curtosis etc.) through the consideration of the so-called moments [191, 90]. Indeed, in these (and related) works, the technique of moment methods was demonstrated as being particularly useful in exploring the dynamics of NLS models and their effective dynamics including the examination of collapse type phenomena and their potential prevention. Such methods were not only used in themes from optics to atomic Bose-Einstein condensates [121], but they were also extended to models of Fisher-KPP type with potential applications to the dynamics of brain tumors [22].

Given that these moment equations are effective ODEs that may *or may not* close at the level of analytical considerations, a natural use of data-driven methods is to potentially use, e.g., the SINDy approach, or other ones such as Neural ODE [52], in order to identify the relevant ODEs in either one of these cases, and to compare the results of training data with possible testing evolution data to examine the potential discovery of known —or, more excitingly unknown— such moment relations. Such approaches were recently considered in the work of [244]. Here, we will consider some of the basic examples of the latter work and comment on the extensions and further possibilities (as well as on more recent work) along this vein.

Our relevant starting point for the purposes of this discussion will be a prototypical NLS model with a parabolic trap (of wide relevance to atomic BECs [187]) in the form of:

$$iu_t = -\frac{1}{2}u_{xx} + \frac{1}{2}x^2u + g\left(|u|^2, t\right)u, \tag{49}$$

where $g\left(|u|^2, t\right)$ denotes the nonlinearity. It is well known from the theory of [191] that relevant moment quantities in such a setting are of the form:

$$I_k(t) = \int_{\mathbb{R}} x^k |u(x,t)|^2 dx, \tag{50}$$

$$V_k(t) = 2^{k-1}i \int_{\mathbb{R}} x^k \left(u(x,t)\frac{\partial \bar{u}(x,t)}{\partial x} - \bar{u}(x,t)\frac{\partial u(x,t)}{\partial x} \right) dx, \tag{51}$$

$$K(t) = \frac{1}{2}\int_{\mathbb{R}} \left| \frac{\partial u(x,t)}{\partial x} \right|^2 dx, \tag{52}$$

$$J(t) = \int_{\mathbb{R}} G\left(\rho(x,t), t\right) dx = \int_{\mathbb{R}} G\left(|u(x,t)|^2, t\right) dx. \tag{53}$$

Here $\bar{u}$ denotes the complex conjugate of u, while $\rho(x,t) = |u(x,t)|^2$, and also $G = G(\rho, t)$ is a function such that $\frac{\partial G}{\partial \rho}(\rho, t) = g(\rho, t)$. The definitions of (50)-(53) can be intuitively

interpreted. For instance, the first moment $I_1(t)$ represents the center of mass of the (unnormalized) probability density $\rho = |u|^2$. I_k represent higher moments associated with this density distribution such as its variance, while V_k reflects the moments associated with the momentum density (the latter is the quantity in the corresponding parenthesis in the right hand side of the V_k definition). K is drawn from the kinetic part of the Schrödinger problem energy, while J from the nonlinear part of the corresponding energy.

We now focus on a couple of concrete (simple) examples of associated moment dynamics. For instance, the moments I_1 and V_0 satisfy

$$\begin{cases} \dfrac{dI_1}{dt} = V_0, \\ \dfrac{dV_0}{dt} = -I_1. \end{cases} \tag{54}$$

This is an intuitive analytical finding suggesting that the center of mass of the density distribution behaves as a harmonic oscillator inside a parabolic trap. Interestingly, this happens irrespectively of the nonlinearity $g(\rho, t)$.

In the work of [244], the authors constructed data matrices $\mathbf{X}^{(0)} = [\mathbf{I}_1^{(0)}, \mathbf{V}_0^{(0)}] \in \mathbb{R}^{N\times 2}$ and $\mathbf{X}^{(1)} = [\mathbf{I}_1^{(1)}, \mathbf{V}_0^{(1)}] \in \mathbb{R}^{N\times 2}$ from numerically solving the PDE (49) with two-distinct suitably localized initial conditions in the form of Gaussian and sech-like pulses. Enlarging the relevant dataset using $\mathbf{X} = [\mathbf{X}^{(0)\top}, \mathbf{X}^{(1)\top}]^\top \in \mathbb{R}^{2N\times 2}$ to avoid overfitting, it was found that SINDy predicts the correct dynamics that match Eq. (54), even when deploying a large library $\mathbf{\Theta}_{\deg\leq n}(\mathbf{x})$ for n up to 16, i.e., it yields:

$$\begin{cases} \dfrac{dI_1}{dt} = 1.000 V_0, \\ \dfrac{dV_0}{dt} = -1.000 I_1, \end{cases} \tag{55}$$

where the coefficients are rounded to three decimal places. This suggested the potential usefulness of the concatenation of data stemming from different time series.

Another example considered in the work of [244] involved the case where the nonlinearity $g(\rho, t) = g(\rho)$ is time-independent and given by $g(\rho) = g_0\rho^2$, where $g_0 \in \mathbb{R}$ is a constant. Interestingly, in this case, the dynamics of the moments I_2, V_1, K and J is not closed, yet it becomes closed under a suitable coordinate transformation, namely $E = K + J$:

$$\begin{cases} \dfrac{dI_2}{dt} = V_1, \\ \dfrac{dV_1}{dt} = 4E - 2I_2, \\ \dfrac{dE}{dt} = -\dfrac{1}{2} V_1, \end{cases} \tag{56}$$

In this case too, a diverse set of localized single and multihump initial data was evolved and the corresponding data was concatenated for the selected moments $\mathbf{x} = [I_2, V_1, K, J]$,

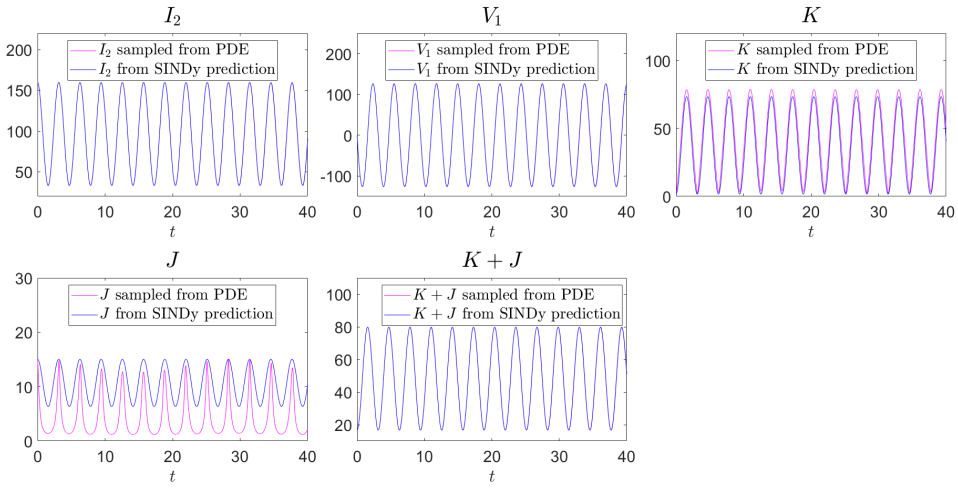


Figure 5: [Adapted from [244]] Comparison of the moment evolutions of $[I_2, V_1, K, J, K+J]$ between SINDy and the ground truth. The training occurs for SINDy only on the selected moments $\mathbf{x} = [I_2, V_1, K, J]$, where a closure does not exist, using a linear library $\mathbf{\Theta}_{\text{deg}=1}(\mathbf{x})$. Interestingly, SINDy captures the correct dynamics for $E = K + J$, although not so the individual ones of K and J.

to examine the outcome of SINDy in this setting. When trying a polynomial *linear* library to discover the equations for these moments, the following system was obtained:

$$\begin{cases} \dfrac{dI_2}{dt} = 1.000V_1, \\ \dfrac{dV_1}{dt} = -2.000I_2 + 4.000K + 3.998J, \\ \dfrac{dK}{dt} = -0.569V_1, \\ \dfrac{dJ}{dt} = 0.069V_1. \end{cases} \tag{57}$$

When simulating this system and comparing it with the ground truth, it was found that it does not capture the correct dynamics for K and J for which there is no closed system; see, in particular, Fig. 5.

However, when the ODE for K was added to that for J, the resulting system is found to be:

$$\begin{cases} \dfrac{dI_2}{dt} = 1.000V_1, \\ \dfrac{dV_1}{dt} = -2.000I_2 + 4.000K + 3.998J, \\ \dfrac{d\,(K+J)}{dt} = -0.500V_1. \end{cases} \tag{58}$$

which matches well the relevant ground truth. This is also suitably reflected in the last panel of Fig. 5, where the two relevant time series are added, capturing very satisfactorily the ground truth result.

Some key additional findings of the above work were that the potential use of quadratic libraries could be problematic for the above case which only closes for the quantity $K + J$ (but not individually for K or/and J). In particular, in such a setting overfitting issues arose, leading to ODE models which are proximal but not identical to the theoretically expected ones. This type of issue also arose in earlier studies such as those of [49] and especially [16]. Indeed, in such settings where the selected library may be "richer" than the terms anticipated, it often turns out to be the case that the resulting "discovered" model will not be the one theoretically expected. This is because the "wealth" of available nonlinear dependent variable combinations within the libraries may yield better data approximations. However, the key question of how well such models generalize is, in principle, unclear beyond the relevant training data.

Another vein that was pursued in this context is that of *identification* of a coordinate transformation in the form of $\widetilde{\mathbf{y}} = \widetilde{\mathbf{A}}^\top \mathbf{x}$, where the transformation matrix $\mathbf{A} \in \mathbb{R}^{4\times 3}$ (and is not unique). The set of $\mathbf{A}$ satisfying the constraint $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_{3\times 3}$ is called the *Stiefel manifold* [74, 6]. Thus, in [244], a concurrent minimization seeking to identify $\mathbf{A}$, along with identifying the optimal, sparse set of resulting differential equations was pursued and was found to accurately obtain the dynamics of $[I_2, V_1, K, J]$.

Finally, the method was also sought to be used in scenarios where no closed-form moment equations were known to exist. As a concrete example of this kind, the non-autonomous case of the time-depdendent nonlinearity was utilized

$$g(\rho, t) = (\sin(t) + 2)\,|\rho|^2\,, \tag{59}$$

for different initial conditions and the dynamics of the moments $\mathbf{x} = [I_2, V_1, E]$, where $E = K + J$, was monitored for all the relevant cases accordingly. The result was found to be quite encouraging both for cases of simpler, bounded, oscillatory dynamics, as well as for ones of apparent unstable growth. What was seen in the relevant results of [244] was that the proximity between the ground truth and the SINDy-based outcomes extended well past the training time interval of $T = 20$ and the matching between the two was qualitatively (and even semi-quantitatively) very good, even up to times about 5 times as large.

4.2. Use of SINDy to Discover Soliton Effective Particle Dynamics

In the above section, we saw how to leverage sparse methods for the identification of nonlinear dynamics via ODEs in the context of moments of the original distribution in nonlinear Schrödinger type models. Another vein of derivation of effective ODEs that has been extremely popular in the physical literature has been the extraction of ordinary differential equations for the features of solitary waves (and their interactions) via the so-called variational approximation [160]. In this class of methods, one approximates the profile of the wave field via a solitary wave or a collection of solitary waves, potentially adding them —notably, in the case of bright solitary waves— (or, e.g., multiplying them in the case of dark solitons).

Typically, this effective solitary wave "manifold" is characterized by the position of the wave center (and the corresponding canonically conjugate variable of the velocity), but additionally other parameters come into play including the amplitude and the width of the wave, or possibly factors involving its phase, such as the so-called chirp [160, 46]. The relevant Ansatz (i.e., attempted wave description) is inserted typically in the Lagrangian of the model and then Euler-Lagrange equations are extracted for the solitary wave parameters. In a Hamiltonian system, these come in pairs of conjugate variables and essentially represent a nonlinear, wave-dynamics motivated, low-dimensional projection of the infinite-dimensional PDE dynamics on the "soliton manifold". Naturally, also, such representations fail to capture radiation features. The latter can result, e.g., from the potential non-integrability of the models and, possibly, from the interaction between solitary waves or between one of them and the (potentially) spatially heterogeneous landscape. As long as the dynamics does not produce considerable radiation (and the interactions are not dramatic enough to drastically change the character of the solitary waves), this approximation can provide a meaningful and interpretable characterization of the soliton dynamics and interactions.

It is also worthwhile to add here that although the variational method is inherently Hamiltonian in the vast majority of its considerations, non-conservative generalizations thereof can be formulated based, e.g., on the work of [88]. Indeed, relevant applications to nonlinear wave systems have been considered in [123] and have also been used for the examination of the temporal dynamics of cavity solitons in the experimentally relevant example of [203].

It is in the above spirit of the variational approximation that the recent work of [245] is seeking to identify the sparse dynamics of solitary waves for both the case of dark and that of bright solitary waves. We examine here, for proof-of-principle purposes, the case of the dark solitons which is structurally somewhat simpler due to the adequacy of a description merely involving the positions and associated velocities of the solitary waves. Indeed, it is known from classical results on the variational approximation of dark solitons [127] that they feature an exponential tail-tail interaction in one spatial dimension. It is similarly known from the central work of [43] (and its generalization for arbitrary, slowly-varying potentials in [129]) that dark (and grey) solitary waves are subject to a restoring force which amounts to one half of the gradient of the external confining potential in which they move.

The combination of the above two effects led to the characterization of the motion of dark solitons and of their interactions in the context of parabolic traps of the form

$$V_{\mathrm{MT}}(x) = \frac{1}{2}\Omega^2 x^2. \tag{60}$$

relevant to atomic Bose-Einstein condensates [187]. Then, for a sequence of dark solitary waves inside the trap, variational approximations have been brought to bear in the work of [59] to give rise to the effective model:

$$\ddot{\xi}_i = 8\exp\left(2\left(\xi_{i-1}-\xi_i\right)\right) - 8\exp\left(2\left(\xi_i-\xi_{i+1}\right)\right) - \frac{\Omega^2}{2}\xi_i. \tag{61}$$

In the work of [245], this type of variational prediction was sought to be verified firstly for a single dark solitary wave in a parabolic trap and then for solitary waves both inside, as well as without a trap and even for four such solitary waves (again, with and without a trap). We briefly survey some representative results from this effort. In a parabolic trap of frequency $\Omega = 0.025$, using the time series of the positions and velocities of the dark soliton extracted from the PDE simulation, SINDy was used to predict the relevant motion from a library using monomials (up to 10th order). The relevant result was:

$$\ddot{\xi} = -0.00031287\xi. \tag{62}$$

which is very proximal to the theoretically predicted prefactor of the linear term (which should be $\Omega^2/2$ in accordance to [43, 129]). Here, SINDy with control via the so-called FROLS (i.e., Forward Regression Orthogonal Least Squares) optimizer which is a greedy method, has been used to extract the relevant result.

On the other hand, for the case of two dark solitons in the presence of the trap, additionally, an exponential term associated with the soliton interaction, as well as monomials up to 10th order were used in the library and the relevant SINDy prediction was:

$$\begin{aligned} \ddot{\xi}_1 &= -7.757\exp\left(2\left(\xi_1-\xi_2\right)\right) - 0.00031635\xi_1, \\ \ddot{\xi}_2 &= 7.803\exp\left(2\left(\xi_1-\xi_2\right)\right) - 0.00031674\xi_2. \end{aligned} \tag{63}$$

The same approach was also attempted for the case of 4 solitary waves in which case the SINDy prediction was found to be:

$$\begin{aligned} \ddot{\xi}_1 &= -7.950\exp\left(2\left(\xi_1-\xi_2\right)\right) - 0.00031272\xi_1, \\ \ddot{\xi}_2 &= 8.420\exp\left(2\left(\xi_1-\xi_2\right)\right) - 7.506\exp\left(2\left(\xi_2-\xi_3\right)\right), \\ \ddot{\xi}_3 &= 7.522\exp\left(2\left(\xi_2-\xi_3\right)\right) - 8.406\exp\left(2\left(\xi_3-\xi_4\right)\right), \\ \ddot{\xi}_4 &= 7.900\exp\left(2\left(\xi_3-\xi_4\right)\right) - 0.00031206\xi_4. \end{aligned} \tag{64}$$

In all of these cases, SINDy can be seen to be quite successful towards retrieving the dark soliton dynamics. A pictorial representation of such outcomes in the context of two and four dark solitary waves without a trap is given in Fig. 6.

Nevertheless, it is interesting to point out that while the data-driven variational approximation yields results quite proximal to the expected ones, there are some deviations. These stem for instance, from the fact that the force prefactors pairwise between the solitons are not found to be equal and opposite. Here, a hard-wired implementation of Newton's 3rd law (regarding action and equal-opposite reaction) is relevant towards a more physically meaningful result, as was subsequently shown in [245]. Moreover, one can see that a greedy method may select to weigh more on the exponential dynamics rather than introduce a parabolic trap term in the intermediate solitary waves such as ξ_2 and ξ_3. There are numerous such features worth considering. Perhaps, even more notably, in the study of [245], it was found to be considerably more difficult to capture the dynamics of bright solitons, especially in light of the numerous degrees of freedom (potentially 4 or 6) that each solitary wave might bear.

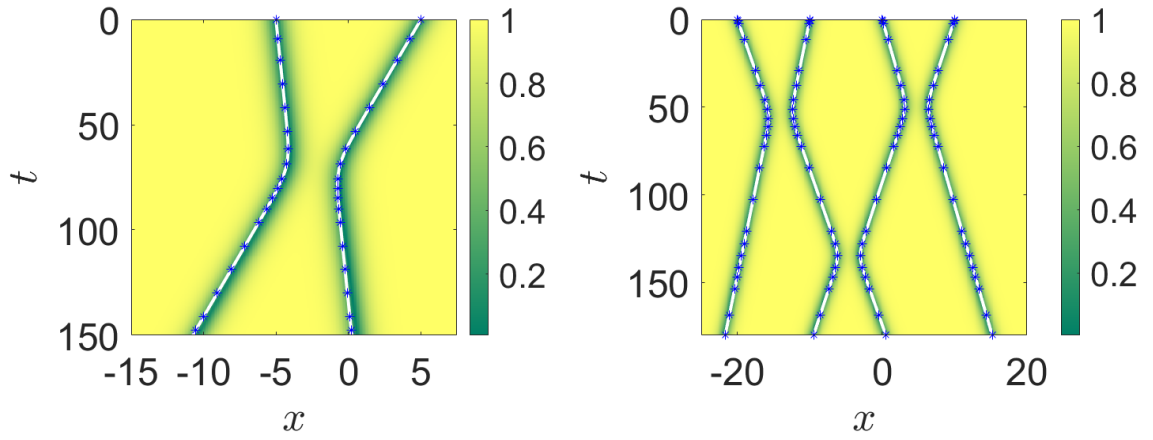


Figure 6: The spatio-temporal dark-soliton interaction dynamics in the absence of a parabolic trap: two dark solitary waves can be discerned on the left, and four on the right, interacting once before they indefinitely separate. The underlying contour map represents the PDE, while white solid curves and blue asterisks refer, respectively, to the theoretical prediction of the soliton positions based on the variational approximation (for $\Omega = 0$) and the SINDy prediction, respectively.

This suggests the perhaps anticipated feature that if the regression within the "soliton manifold" is performed at the level of a well-physically-informed and sufficiently low dimensional manifold, there is a very good chance of extracting meaningful, interpretable results. For high dimensional manifolds, potentially encompassing multiple features at different scales (e.g., the individual soliton motion and their exponential or modulated exponential tail-tail interaction), it becomes considerably harder to accurately retrieve, and more importantly discover in cases where they are unknown, the appropriate solitary wave dynamical and interaction equations.

5. Learning Structure from Data

5.1. Learning Hamiltonians from Data

Data-driven modeling of dynamical systems has progressed rapidly over the past decade, yielding methods that infer the governing equations directly from time-series measurements. As discussed above, Sparse Identification of Nonlinear Dynamics (SINDy) recovers parsimonious ordinary or partial differential equations by selecting a few basis functions that best fit observed time-derivatives [40, 206]. Symbolic-regression approaches such as Eureqa [216] likewise search the space of closed-form expressions. Deep-learning techniques broadened the toolkit: PDE-Net learns convolutional filters that approximate unknown spatial operators [151]; Koopman autoencoders embed nonlinear flows in linearly evolving latent coordinates [157]; Neural Ordinary Differential Equations (Neural ODE) estimate continuous-time dynamics with adjoint

backpropagation [52]; and Physics-Informed Neural Networks (PINNs) encode differential-equation residuals as soft constraints during training [195]. While these approaches can capture complex dynamics when ample data are available, they generally do not enforce energy conservation or symplectic structure, which are crucial for accurately modeling Hamiltonian systems. This limitation has motivated the incorporation of Hamiltonian structure as a physical prior, enhancing model generalization, particularly in low-data regimes.

Consider a *Hamiltonian system* of d degrees of freedom,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \quad \mathbf{f}(\mathbf{x}) = J(\mathbf{x})\nabla H(\mathbf{x}), \quad \forall \mathbf{x} \in D \subset \mathbb{R}^{2d}, \tag{65}$$

where $H : D \to \mathbb{R}$ is the Hamiltonian, and $J(\mathbf{x}) \in \mathbb{R}^{2d\times 2d}$ is antisymmetric. In particular, under canonical coordinates $\mathbf{x} = (\mathbf{q}, \mathbf{p})$, with positions $\mathbf{q}$ and momenta $\mathbf{p}$, the dynamics simplifies to:

$$J(\mathbf{x}) \equiv \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}, \quad \text{and} \quad \begin{cases} \dfrac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \nabla_{\mathbf{p}} H, \\ \dfrac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\nabla_{\mathbf{q}} H. \end{cases} \tag{66}$$

Instead of directly approximating the vector field $\mathbf{f}(\mathbf{x})$ in Eq. (65), Hamiltonian Neural Networks (HNNs) [98, 23] leverage prior knowledge of Hamiltonian structure and aim to learn the Hamiltonian $H(\mathbf{x})$ itself. The corresponding vector field $\mathbf{f}(\mathbf{x}) = J(\mathbf{x})\nabla H(\mathbf{x})$ can then be obtained via automatic differentiation. More specifically, consider a dataset of uniformly sampled trajectories in canonical coordinates:

$$\mathcal{D} = \left\{ \mathbf{x}^{(i,j)} = \left(\mathbf{q}^{(i,j)}, \mathbf{p}^{(i,j)}\right) : 1 \le i \le I,\ 1 \le j \le J \right\}, \tag{67}$$

with a small sampling interval Δt. The time derivatives can be approximated via finite differences:

$$\dot{\mathcal{D}} = \left\{ \dot{\mathbf{x}}^{(i,j)} = \left(\dot{\mathbf{q}}^{(i,j)}, \dot{\mathbf{p}}^{(i,j)}\right) : 1 \le i \le I,\ 1 \le j \le J \right\}. \tag{68}$$

Greydanus et al. [98] parameterize the Hamiltonian as a neural network $H_{\boldsymbol{\theta}}(\mathbf{q}, \mathbf{p})$, trained to minimize the mean squared error between the symplectic gradient $(\nabla_{\mathbf{p}} H_{\boldsymbol{\theta}}, -\nabla_{\mathbf{q}} H_{\boldsymbol{\theta}})$ and the time derivatives $(\dot{\mathbf{q}}, \dot{\mathbf{p}})$ over the dataset $\mathcal{D}$:

$$\mathcal{L}_{\text{HNN}} = \frac{1}{IJ} \sum_{i,j} \left\| \nabla_{\mathbf{p}} H_{\boldsymbol{\theta}} \left(\mathbf{q}^{(i,j)}, \mathbf{p}^{(i,j)}\right) - \dot{\mathbf{q}}^{(i,j)} \right\|^2 + \left\| \nabla_{\mathbf{q}} H_{\boldsymbol{\theta}} \left(\mathbf{q}^{(i,j)}, \mathbf{p}^{(i,j)}\right) + \dot{\mathbf{p}}^{(i,j)} \right\|^2. \tag{69}$$

However, accurate estimation of the time derivatives in (68) requires trajectories sampled at small time intervals Δt. In the presence of observation noise, finite difference approximations of time derivatives become extremely sensitive, rendering pointwise l^2 loss (69) between the vector field and the symplectic gradient impractical in realistic settings. Consequently, many follow-up works instead use an ODE solver (based

on the symplectic gradient of the parameterized Hamiltonian) to generate predicted trajectories, and define the loss based on the mismatch between predicted and observed (ground-truth) trajectories. To train the network, gradients can be computed either by backpropagating through the numerical integrator or via constant-memory adjoint methods as in Neural ODEs [52]. Early approaches employed generic integrators such as the Runge–Kutta method (RK4) [253], while subsequent works explored symplectic integrators to better preserve the phase space structure. These include leapfrog [54], symplectic Euler and implicit midpoint rule [69], variational integrators [209], and higher-order symplectic schemes [72, 241].

Beyond incremental refinements to HNNs, a parallel line of work seeks to bypass explicit Hamiltonian discovery and numerical time stepping altogether. Instead, it aims to approximate the time-t flow map $\mathbf{x} \mapsto \phi_t(\mathbf{x})$, which transports an initial state $\mathbf{x}$ to its position after time t. For any Hamiltonian system described by (66), this flow is symplectic—meaning it preserves the canonical two-form—and therefore obeys

$$(\nabla_{\mathbf{x}}\phi_t)^\top J \nabla_{\mathbf{x}}\phi_t = J \tag{70}$$

Leveraging this geometric constraint, Jin et al. [112] constructed "SympNets," deep networks whose layers are composed of analytically symplectic building blocks, and proved that such architectures can uniformly approximate any smooth symplectic map. Chen and Tao [51] pursued a complementary route: they parameterized a generating function whose gradient defines ϕ_t implicitly, ensuring that (70) holds by construction. Empirically, both approaches achieve long-horizon predictions whose global error grows only linearly with t, in stark contrast to the exponential error accumulation observed when one first learns the Hamiltonian and then integrates it with ODE solvers.

Other studies extend Hamiltonian learning by addressing generalized systems under noncanonical coordinates (65) [62, 111]; employing (variational) autoencoders that embed high-dimensional observations into a latent symplectic manifold governed by a learned Hamiltonian [228, 54, 209]; integrating symplectic constraints into message-passing layers through Hamiltonian graph neural networks [211]; and developing adaptive HNNs that, once trained on trajectories from only a few bifurcation-parameter values, accurately predict dynamics at unseen parameters [102].

5.2. Methods Using Symplectic Transformations and the Discovery of Action-Angle Variables

A cornerstone of integrable Hamiltonian dynamics is the existence of action–angle coordinates $(\mathbf{I}, \boldsymbol{\varphi})$, in which the Hamiltonian depends only on the conserved actions $\mathbf{I}$, and the angle variables evolve linearly in time,

$$\dot{\boldsymbol{\varphi}} = \omega(\mathbf{I}), \quad \text{and} \quad \dot{\mathbf{I}} = 0, \tag{71}$$

as guaranteed by the Liouville–Arnold theorem [10].

However, constructing explicit action–angle transformations from canonical coordinates $(\mathbf{q}, \mathbf{p})$ is notoriously challenging for all but a few classical models. Recognizing this gap, Bondesan and Lamacraft [30] proposed a neural-network-based method to learn such transformations directly from trajectory data by enforcing symplectic structure in the mapping.

Their framework is built around a *symplectic normalizing flow*, parameterized as a sequence of symplectic invertible layers that ensure the learned map

$$T : (\mathbf{q}, \mathbf{p}) \mapsto (\hat{\mathbf{q}}, \hat{\mathbf{p}}) \equiv (\mathbf{I}, \boldsymbol{\varphi}) \tag{72}$$

is a canonical symplectic embedding. Each layer preserves the Hamiltonian structure, enabling efficient learning of integrable transformations.

They demonstrated this approach on three paradigmatic integrable systems: the Kepler problem, Neumann model, and Calogero–Moser system. The training objective encourages trajectories in $(\mathbf{I}, \boldsymbol{\varphi})$-space to map to simple circular motion (constant actions, linear phase), reflecting the canonical action–angle flow. Once trained, the network maps observed $(\mathbf{q}, \mathbf{p})$-trajectories into linear dynamics on tori, effectively identifying the action–angle coordinates.

Building on this idea, Daigavane et al. [68] propose a related framework that leverages symplectic neural networks [112] to learn action–angle representations from data. Their approach focuses on the complete dynamics of integrable systems, building a simulator in action–angle space that captures both the conserved quantities and the linear evolution of angle variables.

5.3. Learning Conserved Quantities Using Deep Learning

In recent years, data-driven approaches for discovering conservation laws and assessing integrability have advanced rapidly, fueled by machine learning techniques. Broadly, existing algorithms fall into two classes: those that incorporate explicit knowledge of the governing differential equations (DEs) and those that operate solely based on observed trajectories.

For DE-informed methods, recent contributions include the approaches described in [148, 149, 258]. Liu et al. [148] proposed a regularized loss function to train a family of conservation laws simultaneously, with a penalty term that encourages pointwise orthogonality of their gradients to promote functional independence. Building on this idea, Liu et al. [149] further combined sparse regression with a prescribed dictionary of basis functions to enhance interpretability. The neural deflation framework [258] adopts an iterative perspective: each newly identified integral is learned with a deflated loss that enforces functional independence with respect to those already discovered, yielding complete families of functionally independent conservation laws across a range of test problems. It is worth noting that unlike the orthogonality-based penalty in [148], which is sufficient but not necessary for functional independence, the deflation approach enforces independence directly and is provably consistent in the infinite-sample limit:

a function is a new conservation law *if and only if* the deflated loss achieves zero, a guarantee not provided by the earlier method.

Trajectory-based methods address the more practical scenario in which the governing equations are unavailable. Siamese neural networks [237] extract a single invariant by comparing pairs of states, whereas the approach of Ha and Jeong [101] leverages grouped data sampled from level sets to learn multiple conservation laws. Model-agnostic techniques such as the method of Arora et al. [11] incorporate prior information about the number of invariants, and manifold-learning formulations [150, 156] aim to uncover the geometry of invariant manifolds directly in state space.

While each of these methodologies has its own strengths and limitations, we now focus on the neural deflation method [258], which has consistently demonstrated the ability to recover complete sets of functionally independent, Poisson-commuting conservation laws in Hamiltonian systems.

Consider the d-dimensional Hamiltonian system (65). Let F and G be two smooth functions defined on the phase space D. Their *Poisson bracket*, $\{F, G\} : D \to \mathbb{R}$, is given by

$$\{F, G\}(\mathbf{x}) \coloneqq \nabla F(\mathbf{x})^T J(\mathbf{x}) \nabla G(\mathbf{x}), \quad \forall \mathbf{x} \in D. \tag{73}$$

A continuously differentiable function $I : D \to \mathbb{R}$ is called a *conservation law* of system (65) if it remains constant along system trajectories. That is, for any solution $\mathbf{x}(t)$ to Eq. (65), we have

$$I(\mathbf{x}(t)) \equiv I(\mathbf{x}(0)), \quad \forall t \geq 0. \tag{74}$$

It is straightforward to verify that I is a conservation law if and only if its Poisson bracket with the Hamiltonian H vanishes on D, i.e.,

$$\{I, H\}(\mathbf{x}) = \nabla I(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in D \tag{75}$$

A collection $\{I_k : D \to \mathbb{R}\}_{k=1}^K$ of K conservation laws of system (65) is called *functionally independent* if their gradients $\{\nabla I_k(\mathbf{x})\}_{k=1}^K$ are linearly independent vectors in $\mathbb{R}^{2d}$ for almost every $\mathbf{x} \in D$. Intuitively, this means that none of the conserved quantities can be expressed as a (nonlinear) function of the others. Moreover, these integrals are said to be *in involution* or *Poisson commuting* if their pairwise Poisson brackets vanish, i.e., $\{I_j, I_k\} = 0, \forall j \neq k$. For a Hamiltonian system with d degrees of freedom, there can be at most d functionally independent conservation laws in involution. When such a collection exists, the system is said to be *completely integrable* in the sense of Liouville [10].

The neural deflation framework [258] offers a principled, data-driven strategy for identifying a maximal set of functionally independent, Poisson-commuting conservation laws when the underlying dynamics is explicitly known. The central idea is to learn each conserved quantity in sequence, modifying the loss function at every step to enforce independence and Poisson-commutativity with those already identified. This strategy

draws motivation from the idea of "deflation" in numerical PDEs, where previously found solutions are actively avoided to uncover new ones [81].

The process begins by randomly sampling a training set $\mathcal{T}$ and a validation set $\mathcal{V}$ from the phase space $D \subset \mathbb{R}^{2d}$. Each candidate conservation law is represented by a neural network $I_k(\mathbf{x};\boldsymbol{\theta}_k)$ with parameters $\boldsymbol{\theta}_k$. The first such network, I_1, is trained by minimizing the following loss:

$$\mathcal{L}_1(\boldsymbol{\theta}_1;\mathcal{T}) \coloneqq \frac{1}{|\mathcal{T}|}\sum_{\mathbf{x}\in\mathcal{T}}\left|\widehat{\mathbf{f}}(\mathbf{x})\cdot\widehat{\nabla I_1}(\mathbf{x};\boldsymbol{\theta}_1)\right|^2, \tag{76}$$

where the hat notation, $\widehat{\mathbf{f}}$, here denotes the l^2-normalized version of $\mathbf{f}$. This normalization ensures the training process does not converge to trivial constant solutions and makes the loss independent of scaling.

To learn additional conserved quantities, the method proceeds inductively. Suppose $K-1$ such laws $\{I_k\}_{k=1}^{K-1}$ have been obtained. Then the K-th candidate $I_K(\mathbf{x};\boldsymbol{\theta}_K)$ is learned by minimizing a new loss that combines three objectives: (i) it must be conserved under the dynamics, (ii) it must Poisson-commute with each previously learned law, and (iii) it must be functionally independent from the identified laws. The combined loss is given by:

$$\mathcal{L}_K(\boldsymbol{\theta}_K;\mathcal{T}) \coloneqq \frac{1}{|\mathcal{T}|}\sum_{\mathbf{x}\in\mathcal{T}}\frac{\overbrace{\ell_{\text{conserv}}[\boldsymbol{\theta}_K;\mathbf{x}]}^{\text{conservation loss}}+\overbrace{\sum_{k=1}^{K-1}\ell_{\text{inv}}[\boldsymbol{\theta}_k^*,\boldsymbol{\theta}_K;\mathbf{x}]}^{\text{involution loss}}}{\underbrace{K\left|\ell_{\text{ind}}[\boldsymbol{\theta}_K|\boldsymbol{\theta}_1^*,\ldots,\boldsymbol{\theta}_{K-1}^*;\mathbf{x}]\right|^{\alpha}}_{\text{independent loss}}}, \tag{77}$$

where each term plays a specific role:

- The *conservation loss* enforces the defining property of conserved quantities:

$$\ell_{\text{conserv}}[\boldsymbol{\theta}_K;\mathbf{x}] \coloneqq \frac{1}{|\mathcal{T}|}\sum_{\mathbf{x}\in\mathcal{T}}\left|\widehat{\mathbf{f}}(\mathbf{x})\cdot\widehat{\nabla I_K}(\mathbf{x};\boldsymbol{\theta}_K)\right|^2. \tag{78}$$

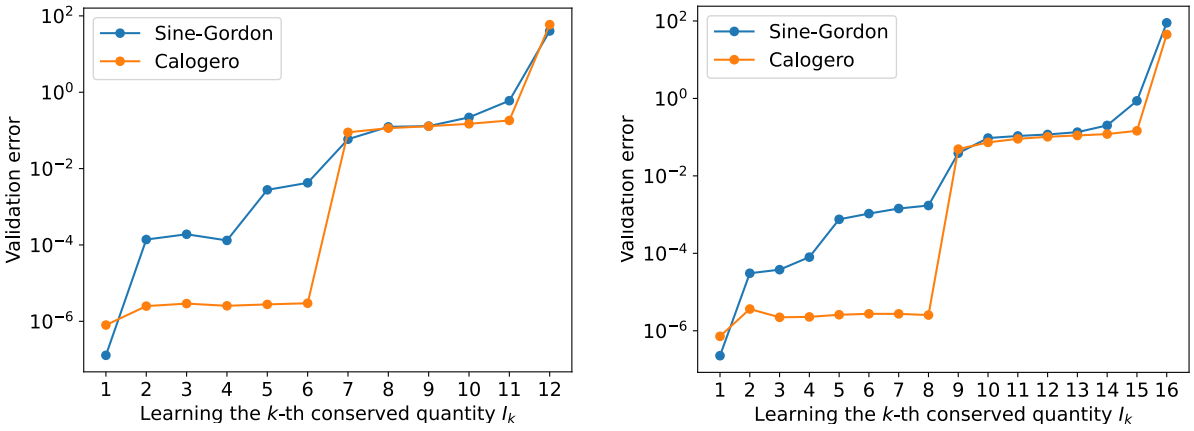
- The *involution loss* penalizes nonzero Poisson brackets with earlier conservation laws:

$$\ell_{\text{inv}}[\boldsymbol{\theta}_k^*,\boldsymbol{\theta}_K;\mathbf{x}] \coloneqq \left|\{I_k(\cdot;\boldsymbol{\theta}_k^*),I_K(\cdot;\boldsymbol{\theta}_K)\}(\mathbf{x})\right|^2 \tag{79}$$

- The *independent loss* in the denominator is given by:

$$\ell_{\text{ind}}[\boldsymbol{\theta}_k^*|\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_{K-1};\mathbf{x}] \coloneqq \left\|\text{Proj}_{\text{span}\{\widehat{\nabla I_k}(\mathbf{x};\boldsymbol{\theta}_k^*)\}_{k\in[K-1]}^{\perp}}\widehat{\nabla I_K}(\mathbf{x};\boldsymbol{\theta}_K)\right\|^2, \tag{80}$$

  where $\text{Proj}_{\text{span}\{\widehat{\nabla I_k}(\mathbf{x};\boldsymbol{\theta}_k^*)\}_{k\in[K-1]}^{\perp}}\widehat{\nabla I_K}(\mathbf{x};\boldsymbol{\theta}_K)$ denotes the projection of $\widehat{\nabla I_K}(\mathbf{x};\boldsymbol{\theta}_K)$ onto the orthogonal complement of the subspace spanned by $\{\widehat{\nabla I_k}(\mathbf{x};\boldsymbol{\theta}_k^*)\}_{k\in[K-1]}$ in $\mathbb{R}^{2d}$. This term introduces a singularity that penalizes any lack of independence between $\widehat{\nabla I_K}(\mathbf{x};\boldsymbol{\theta}_K)$ and the previously learned $\{\widehat{\nabla I_k}(\mathbf{x};\boldsymbol{\theta}_k^*)\}_{k\in[K-1]}$. The hyperparameter $\alpha>0$ controls how strongly this constraint is enforced.



(a) Number of lattice sites $N = 6$ (b) Number of lattice sites $N = 8$

Figure 7: [Adapted from [258]] Validation losses for the learned conservation laws in the *non-integrable* discrete sine-Gordon system and the *integrable* Calogero-Moser system, each with $d = N$ degrees of freedom.

This formulation can be shown to be consistent in the limit of infinite data. Specifically, if the earlier conservation laws exactly capture a true set of independent, commuting integrals, and if empirical averages are replaced by expectations under an absolutely continuous probability distribution, then the minimum of $\mathcal{L}_K$ is zero if and only if I_K extends the set to K such laws. A rigorous proof is given in [258].

The algorithm continues adding new conservation laws until training one more leads to a significant rise in the loss on the validation set, indicating that no further independent commuting conserved quantities can be found. At that point, the process terminates with a maximal set of $d_0 = K - 1$ such laws.

Figure 7 illustrates the neural deflation method applied to two nonlinear dynamical lattices of N identical nodes ($d = N$ degrees of freedom): the *non-integrable* discrete sine-Gordon (DsG) system [65] and the *integrable* Calogero-Moser (CM) system [175, 44]. For the CM system, a sharp increase in validation loss at $k = d + 1 = N + 1$ indicates that the algorithm accurately detects integrability and recovers a *complete* set of independent conserved quantities in involution. In contrast, for the DsG system, the loss always jumps at $k = 2$, consistent with the presence of only *one* conserved quantity (the Hamiltonian), independent of lattice size. Similar patterns arise in other systems, such as the Toda lattice [226] and Fermi–Pasta–Ulam-Tsingou chains [82, 87].

These results demonstrate that the neural deflation method not only provides a principled and effective framework for identifying complete sets of functionally independent, Poisson-commuting conservation laws when the governing equations are known, but also serves as a foundation for data-driven integrability detection. By combining it with Hamiltonian neural networks [98], recent extensions [53] enable direct application to trajectory data, thereby broadening its applicability to settings where only observational data are available.

5.4. Learning Hamiltonian Structure in Dissipative Systems

As discussed extensively throughout this section, a central objective in the data-driven discovery of dynamical systems is to uncover an underlying *Hamiltonian structure* directly from observations. Yet, many real-world systems are not Hamiltonian due to, for example, the presence of dissipative or time-dependent forces. To this end, there exist several geometric formalisms that extend Hamiltonian ideas to broader classes of systems. For example, the *port-Hamiltonian* framework generalizes Hamiltonian dynamics to open systems with inputs and outputs, making it widely used in control theory [230, 73]. The *metriplectic* and *GENERIC* formalisms extend Hamiltonian structure to include dissipative processes while maintaining thermodynamic consistency [173, 99, 178]. Other extensions include Lie–Poisson systems [164] and multi-symplectic formulations for PDEs [33]. Of course, a comprehensive treatment of all such frameworks is beyond the scope of this work. For this reason, we focus on GENERIC-type structures, which encompass both reversible (Hamiltonian) and irreversible (dissipative) dynamics in a unified, physically interpretable way.

To begin, we first discuss the general nature of a *metriplectic* system. Metriplecticity provides a geometric framework for dissipative dynamics by combining a Poisson bracket $\{\cdot,\cdot\}$, encoding the reversible part, with a symmetric, positive semi-definite bracket $(\cdot,\cdot)$, encoding the irreversible (dissipative) part. For a state variable z in the phase space and functionals F and G, these brackets satisfy $\{F,G\} = -\{G,F\}$ and $(F,G) = (G,F) \geq 0$. Given an energy functional E and an entropy functional S, the metriplectic evolution of any observable $F(z)$ is governed by

$$\frac{dF}{dt} = \{F,E\} + (F,S), \tag{81}$$

with the degeneracy conditions $\{S,E\} = 0$ and $(E,F) = 0$ for all F, ensuring that E is conserved and S is non-decreasing.

The *GENERIC* (**G**eneral **E**quation for **N**on-**E**quilibrium **R**eversible–**I**rreversible **C**oupling) formalism generalizes this structure. Let z denote the state variables, $E(z)$ the total energy, and $S(z)$ the total entropy. The evolution equation takes the form

$$\dot{z} = L(z)\frac{\delta E}{\delta z} + M(z)\frac{\delta S}{\delta z}, \tag{82}$$

where $L(z)$ is an antisymmetric operator defining the Poisson bracket, $M(z)$ is a symmetric, positive semi-definite operator defining the dissipative bracket, and the state z lives in some abstract Banach space Ω. The degeneracy conditions are now

$$L(z)\frac{\delta S}{\delta z} = 0, \quad M(z)\frac{\delta E}{\delta z} = 0 \tag{83}$$

which recover the conservation of energy and the monotonicity of entropy, reducing Equation (82) to Equation (81) in the metriplectic case. GENERIC provides a unifying geometric framework for coupled reversible–irreversible dynamics, ensuring

compatibility with the first and second laws of thermodynamics for both finite- and infinite-dimensional systems.

Work by Zhang, et al. [251], introduces *GENERIC Formalism Informed Neural Networks* (GFINNs). GFINNs are physics-informed architectures designed to learn dynamical systems while preserving the exact structure of the GENERIC formalism. Let us consider the case where L and M are known, and the task is to learn the functionals E and S. It may be tempting to directly parametrize the gradients of E and S, as they appear in Equation (82). This approach loses the direct recovery of the functionals since not every vector function is the gradient of a scalar function. Thus, GFINNs introduce fully connected feedforward neural networks to model the energy and entropy functional, computing their gradients via automatic differentiation.

Having parametrized the functionals directly, care must be taken to satisfy the degeneracy conditions. A key to this is to recognize the multiplicative nature of gradients of $l-$layer neural networks. If one lets $f(x;\theta):\mathbb{R}^d\to\mathbb{R}$ denote an l-layer neural network and $g:\mathbb{R}^d\to\mathbb{R}^d$ be any differentiable function, the gradient of $(f\circ g)(x) := f(g(x))$ with respect to x is given by

$$\nabla_x f(g(x);\theta) = (Jg(x))^\top \left(W^l D_{l-1}\cdots W^2 D_1 W^1\right)^\top = (Jg(x))^\top \nabla_x f\circ g(x). \tag{84}$$

Here, D_j is a diagonal matrix whose (i,i)-entry is $\phi'\left(z_i^j(g(x))\right)$, ϕ denoting a nonlinear activation function, for $1\le j<l$ and $1\le i\le n_j$, and $\mathrm{J}g(x)$ is the Jacobian of g at x. The product form on the right-hand side of Equation (84) is what Zhang *et al.* refer to as the multiplicative structure of the neural network gradient.

Now, without loss of generality, let us only consider the pair (L,M), as what follows will also apply to the pair (M,S). Very loosely speaking and for sake of brevity, the trick GFINNs introduces is to use a tailored projection-like transformation $\mathcal{P}_L$ in the very first layer of the neural network. To define the transformation, one first assumes that the nullspace of $L(z)$ is constant over all possible state variables z. Define the matrices $\tilde{Q}_L(\boldsymbol{z}) = \left[q_L^1(\boldsymbol{z}),\ldots,q_L^{\tilde{n}_L}(\boldsymbol{z})\right]$, where each q_L^j is a vector function in the largest, hence invariant, subspace of the nullspace of L, and $F_L(\boldsymbol{z}) = \left[F_L^1(\boldsymbol{z}),\ldots,F_L^{\hat{n}_L}(\boldsymbol{z})\right]^\top$ assuming that there exist F^j such that

$$\operatorname{span}\left\{\nabla F_L^j(\boldsymbol{z}) : j=1,\ldots,\hat{n}_L\right\}\bigoplus \operatorname{null}_{\text{invariant}} L(\boldsymbol{z}) = \operatorname{null} L(\boldsymbol{z}).$$

The projection operator is then defined as

$$\mathcal{P}_L(\boldsymbol{z}) = \left[\tilde{Q}_L^\top(\boldsymbol{z})\boldsymbol{z}; F_L(\boldsymbol{z})\right]\in\mathbb{R}^{n_L}$$

Note that the first $\tilde{n}_L$ components of $\mathcal{P}_L(\boldsymbol{z})$ are the orthogonal projection coefficients of $\boldsymbol{z}$ onto $\operatorname{null}_{\text{invariant}}\ L(\boldsymbol{z})$ and the remaining components that complete the null space of L.

For neural networks $f(z;\theta):\mathbb{R}^d\to\mathbb{R}$, if one defines

$$E_{\text{NN}}\left(\boldsymbol{z};\theta_L\right) = f\left(\mathcal{P}_L(\boldsymbol{z});\theta_L\right) \tag{85}$$

it then follows from the multiplicative structure of Equation (84) and the construction of the projection transformation that any E_{NN} of the form (85) satisfies $L(\boldsymbol{z})\nabla E(\boldsymbol{z}) = \mathbf{0}$ for all $\boldsymbol{z}$ in the state space Ω. With this in hand, Zhang *et al.* show that a universal approximation result that exactly preserves the degeneracy condition is readily available; see Theorem 1 in [251]. Theorem 2 in [251], and surrounding text discuss the extension of a similar universal approximation result in the context of unknown matrix-valued functions $L(z)$ and $M(z)$.

We emphasize that the power of GFINNs, given the surrounding assumptions in the construction of the projection operators $\mathcal{P}_L$ and $\mathcal{P}_M$, is that the full learned dynamics take the form of Equation (82) with all GENERIC properties preserved to machine precision. Therefore, because the architecture encodes the GENERIC constraints intrinsically, GFINNs act as universal approximators for dynamics compatible with the GENERIC structure, while ensuring that predictions remain physically consistent. Zhang *et al.* show that GFINNs are successful in modeling both deterministic and stochastic systems, such as heat–volume exchange in coupled gas containers, thermo-elastic pendulums, and underdamped Langevin dynamics, consistently outperforming unconstrained neural networks in both predictive accuracy and adherence to thermodynamic laws.

GFINNs have also been pushed beyond finite-dimensional benchmarks toward the regime of field equations. Weak-form GFINNs (WGFINNs) recast the GENERIC constraints in a variational setting to target PDE residuals directly. Related metriplectic/Onsager-inspired models encode the same reversible–irreversible split for continuum mechanics and fluid systems, and pseudo-Hamiltonian neural networks adapt that decomposition to nonlinear dispersive–dissipative PDEs, offering a template for wave dynamics. Complementary structure-preserving lines include discretization-aware architectures (e.g., FINN adapted to shallow-water equations for bathymetry inference) and neural-operator approaches tailored to Helmholtz/elastic waves, underscoring that GENERIC-style priors play well with PDEs in strongly nonlinear, wave-dominated settings [251, 180, 247, 105, 75, 107, 260].

Another approach that has not been covered explicitly here is approximations via pseudo-Hamiltonian neural networks (PHNNs). PHNNs offer a compelling approach, originally developed for ordinary differential equations but later adapted to the PDE setting. PHNNs partition the learned representation into distinct components (internal conservative dynamics, dissipative effects, and external forces) each modeled by separate neural sub-networks. This modular architecture improves interpretability and robustness, particularly when environmental influences change or are removed. Eidnes and Lye [75] have successfully extended this PHNN framework to partial differential equations (PDEs), demonstrating, just as with GFINNs, that PHNNs outperform standard, monolithic neural models, such as vanilla PINNs, while maintaining the physics provided by the GENERIC formalism. We now turn to the widely studied and fundamental in its own right question of identifying (in our case, via data-driven techniques) integrability in nonlinear ordinary, as well as partial differential equations.

6. Learning Integrability of Differential Equations

6.1. Summary of Recent Community Efforts

Given a dynamical system with state x and equations of motion $\dot{x} = f(x)$, the goal here is to *identify integrability features* directly from either numerical or symbolic data. One approach is to try to learn a *Lax pair* (L, P) in mechanics or a *Lax connection* (U, V) in field theory. Recall from Section 3.5, for a Lax pair one requires $\frac{dL}{dt} = [P, L]$ so that spectral invariants $\operatorname{tr} L^k$ are conserved; for field theories recall the *zero–curvature* (flatness) condition

$$U_t - V_x + [U, V] = 0$$

holds for the relevant and compatible dynamics.

In addition to the zero–curvature formulation, integrability can also be encoded at the level of the underlying Poisson structure. In the so–called r–matrix formulation one prescribes a Lie–Poisson bracket for the Lax operator,

$$\{L_1, L_2\} = [r_{12}, L_1] - [r_{21}, L_2],$$

where the subscripts denote tensor placement: $L_1 = L \otimes I$ acts on the first slot, $L_2 = I \otimes L$ on the second, and r_{12} (respectively r_{21}) acts on both slots of $V \otimes V$ (with the tensor factors ordered as $1, 2$ or $2, 1$). In components, this compact tensorial notation encodes all Poisson brackets between the entries of L. The condition guarantees that the spectral invariants of L are in involution and hence generate commuting flows. Importantly, once a candidate L is identified, the associated r–matrix can in principle also be learned directly from data, providing access not just to the dynamics but also to the Hamiltonian structure of the system.

Pioneering work by Krippendorf et al. [134], considers the deep learning of Lax pairs. This approach is unsupervised, physics-constrained (through the equation of motion), and returns *analytic* candidates for Lax operators (and r–matrices) that can be verified a posteriori. Depending on the mechanical context, a type of soft constraint on the Lax/flatness residual is enforced

$$\mathcal{L}_{\text{Lax}} = \big\|\partial_t L - [P, L]\big\|^2 \quad \text{or} \quad \mathcal{L}_{\text{flat}} = \big\|U_t - V_x + [U, V]\big\|^2.$$

Time derivatives are easily accessed by the chain rule of the equation of motion, e.g.,

$$\dot{L} = \frac{\partial L}{\partial x}\,\dot{x} = \frac{\partial L}{\partial x}\,f(x).$$

To align with the expected behavior of Lax operators, Krippendorf, et al., write each learned spectral operator entry as $z_k(x)$, enforcing that $\dot{z}_k$ is linear in a small set of elementary time derivatives and proportional to at least one such element, which yields auxiliary losses

$$\mathcal{L}_t = \sum_k \min_{\{c_{kj}\}} \Big\|\dot{z}_k - \sum_j c_{kj}\,\xi_j\Big\|^2, \qquad \mathcal{L}_M = \sum_k \min_{\{d_{kj}\}} \Big\|[L, M]_k - \sum_j d_{kj}\,\xi_j\Big\|^2,$$

with ξ_j chosen from the components of $f(x)$ and their linear combinations. To avoid degenerate solutions $L \equiv 0$, a type of *mode-collapse* penalty (see terms with subscript MC in what follows) encourages nontrivial entries, and the total objective for pairs or connections is

$$\mathcal{L}_{\text{pair}} = a_1\mathcal{L}_{\text{Lax}} + a_2\mathcal{L}_t + a_3\mathcal{L}_M + a_4\mathcal{L}_{\text{MC}}, \qquad \mathcal{L}_{\text{conn}} = a_1\mathcal{L}_{\text{flat}} + a_2\mathcal{L}_t + a_3\mathcal{L}_M + a_4\mathcal{L}_{\text{MC}}.$$

With this loss function in hand, training is performed using typical deep learning machinery. Once L is known, the classical r–matrix can be fit by minimizing

$$\mathcal{L}_r = \big\|[L_1, L_2] - [r_{12}, L_1] + [r_{21}, L_2]\big\|^2,$$

where, again, tensor indices denote action on separate copies of phase space.

We summarize the results of Krippendorf, et al., here:

- **Harmonic oscillator:** The method recovers a valid 2×2 Lax pair with $\operatorname{tr} L^2 \propto H$.
- **KdV:** A 2×2 operator connection (A_x, A_t) with quadratic/derivative terms matches the known KdV Lax form up to benign gauge shifts. Here the authors needed to enforce constraints in the connection (e.g., the existence of off-diagonal elements) and then got approximations up to small constant terms of the KdV model.
- **Heisenberg ferromagnet:** Using an SU(2)-equivariant ansatz and a suitably restricted form of A_x, the algorithm reproduces the standard Lax connection.
- **Sine–Gordon and principal chiral model:** The zero–curvature condition is satisfied if and only if the respective equations of motion hold, again upon suitable choice of both A_t and A_x (e.g., with the latter depending only on the first spatial derivative of the field).

The work of [134] also finds that for perturbed systems (e.g., a 2D coupled oscillator or a perturbed Heisenberg model), training losses *adapt* and decrease over time for integrable perturbations (the network can retune the Lax structure), but plateau at larger values for non-integrable ones, furnishing a practical diagnostic.

The more recent work of de Koster and Wahls [70] takes a different approach to learning integrability of differential equations, using a more traditional data-driven approach. Since many nonlinear wave PDEs can be cast in Lax form and solved by the inverse scattering method (ISM), one would hope to be able to leverage the spectral operator L, which induces a nonlinear Fourier transform (NFT), in practical applications. When only data are available and the governing equation is unknown (or only approximately Lax-integrable), the question is whether one can *identify* a Lax structure from data, sufficient to enable analysis via NFT.

The paradigm on which [70] focuses concerns the celebrated Ablowitz-Kaup-Newell-Segur (AKNS) class, whose spectral operator has the canonical 2×2 form

$$L = \begin{pmatrix} i\partial_x & -i\,q(u) \\ i\,r(q) & -i\partial_x \end{pmatrix}, \qquad P = \begin{pmatrix} P_{11}(u) & P_{12}(u) \\ P_{21}(u) & -P_{11}(u) \end{pmatrix},$$

with potentials $q(u)$ and $r(q)$ that distinguish members of the AKNS hierarchy (e.g., NLSE, mKdV, sine–Gordon) and relatives via simple choices such as $r = \pm q$, $r = \pm q^*$, or $r = -1$. Together with the linearized dispersion relation, the choice of (q, r) *completely determines* an integrable PDE within the AKNS family, hence also $A(u)$.

AKNS systems admit an infinite sequence of conserved integrals $C_k[q, r]$ obtained recursively from the spectral problem. The first few are explicit functionals of q, r and their derivatives; for example

$$C_1 = \int q\, r\, dx, \quad C_2 = \int (r\, q_x - r_x\, q)\, dx, \quad C_3 = \int \left(q^2 r^2 + q_x r_x\right) dx,$$

with higher C_k mixing higher derivatives and nonlinear combinations. These C_k serve as *invariants* that should remain (approximately) constant along the evolution for a Lax-integrable model. This is the key feature of AKNS that de Koster and Wahls exploit in [70]. Naturally, of course, this raises the question about how these C_i quantities will be known without expert insight into the model. In the work of [70], they are assumed to be a priori "given" and their constancy practically formulates the loss function of choice.

Given just two snapshots $u(\cdot, t_0)$ and $u(\cdot, t_1)$ (more can be used), the method of [70] searches over parameters c_d in a parameterized library for $q(u)$:

$$q(u) = \sum_{d=1}^{D} c_d g_d(u), \text{ with } D = 5,$$
$$G = \left\{g_1 = u, g_2 = u_x, g_3 = u_{xx}, g_4 = u^2, g_5 = u u_x\right\},$$

which also implies a library once a functional form of $r(q)$ is specified. For each candidate, the method computes the first few conserved quantities $C_k[q, r]$ at t_0 and t_1, measures their variation $\Delta C_k = C_k(t_1) - C_k(t_0)$, and selects the (q, r) that *minimizes* a weighted norm of $\{\Delta C_k\}_k$. More precisely, de Koster and Wahls solve the following optimization problem to identify r and q within the AKNS family

$$\left(r^{(\mathrm{ID})}, c^{(\mathrm{ID})}\right) = \underset{r \in \{-1, \pm q, \pm q^*\}}{\operatorname{argmin}} \underset{c \in \mathcal{C}}{\operatorname{argmin}} \left[w(c) E(c, r; u)\right], \quad c \subseteq \mathbb{R}^d \tag{86}$$

with $w(c)$ denoting the weight of coefficient c and where the function

$$E(c, r; u) = \sum_{k \in \{1,3,5\}} \left(\sum_{n=1}^{N} \frac{\sigma\left[C_k\left(t; c, r, u^{(n)}\right)\right]}{\mu\left[\left|C_k\left(t; c, r, u^{(n)}\right)\right|\right]} \right)$$

with the mean and standard deviation defined by

$$\mu\left[C_k(t)\right] = \frac{1}{M} \sum_{m=1}^{M} C_k\left(t_m\right), \qquad \sigma\left[C_k(t)\right] = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(C_k\left(t_m\right) - \mu\left[C_k(t)\right]\right)^2}.$$

Simply put, de Koster and Wahl optimize over the parameter c, and then perform a brute force search over the 5 options of r implied by Equation (86).

The paper demonstrates identification on (nearly) Lax-integrable data from mKdV, NLSE, sine–Gordon, and a transformed KdV example, and explores a viscous Burgers case with complex viscosity as a stress test. In these examples, the correct AKNS spectral operator is recovered from noisy data (on the order of $\lesssim 1\%$ of the signal amplitude in most cases).

It is important to note that the method of [70] is *class*-specific to AKNS systems and targets the *spectral operator* rather than a free-form PDE. This specialization yields robustness and interpretability: once L (and the dispersion) are determined, one obtains an integrable surrogate PDE and inherits the entire AKNS solution machinery. Compared with general Lax-pair learning via neural Änsatze optimization, this approach trades breadth for reliability and data efficiency. However, we also remind the reader of the specificity requirement of being aware of the relevant conservation laws. Clearly, this work leaves room for adaptability to wider classes of integrable and nearly integrable dynamics.

Kantamneni, et al. [116], use yet another approach to learning integrability. Their insight is based on the recognition that integrable PDEs are characterized by having infinitely many conserved quantities. Despite the fact that integrable PDEs are exceptionally rare and often discovered by serendipity, the authors of [115] propose to learn new candidate integrable PDEs by fitting coefficients in a parameterized PDE form to maximize the number of conserved quantities discovered, denoted n_{CQ}.

Consider a parametric PDE of the form

$$u_t = \mathcal{N}[u;\, \alpha],$$

with α denoting a finite set of coefficients (e.g., in differential operators or nonlinear terms). The goal is to adjust α so that the resulting PDE admits as many independent conserved quantities $C_k[u]$ as possible. The conserved quantities $C_k[u]$ are defined by satisfying

$$\frac{d}{dt} C_k[u(t)] = 0$$

under the PDE evolution. In practice, the authors search over α using backpropagation. They simultaneously identify candidate invariants C_k (via weak form or PDE-specific templates), evaluate their rate of change $\dot{C}_k$ numerically, and maximize the count of invariants with $\|\dot{C}_k\|$ below a prescribed tolerance.

This framework, called OptPDE, successfully rediscovers known integrable PDEs (such as KdV-like models) within its coefficient search space. Interestingly, it also generates *novel families* of PDEs that admit at least one non-trivial conserved quantity. An example highlighted is

$$u_t = \left(u_x + a^2\, u_{xxx}\right)^3, \tag{87}$$

for which analytical exploration reveals interesting conserved properties, though integrability is not guaranteed. Kantamneni et al. study, by hand, the case $a = 0$ of Equation (87). They show that this PDE indeed possesses an infinite number of conserved quantities.

Of course, it is important to point out that maximizing n_{CQ} is *not sufficient* to prove integrability. This merely identifies promising candidates, and the method depends critically on the choice of invariant templates and detection criteria. Nevertheless, by proposing novel PDEs with nontrivial invariants, OptPDE enables a human–AI collaboration loop where a machine hypothesizes and domain scientists verify. One can argue based on the proposed examples that this is a methodology that may yield intriguing results regarding identifying PDEs with conservation laws, which occasionally may have a bearing on integrability studies, although it is not directly (a priori) geared in that direction.

6.2. Learning ODE Integrability using Sparse Identification of Lax Operators

Sparse Identification of Lax Operators (SILO) [7] takes a more basic approach to learning integrability. Instead of using deep learning, using data-driven approaches, or trying to maximize the total number of conserved quantities, SILO is a symbolic operator learning framework. The main idea behind SILO, somewhat inspired by the SINDy methodology, is to identify sparse and interpretable representations of a Lax pair compatible with a given Hamiltonian system. To describe this approach, we first illustrate its application in identifying Lax pairs for Hamiltonian ODE systems.

Let us begin with the simple harmonic oscillator $\dot{q} = p$, $\dot{p} = -q$. We may hypothesize that the Lax pair is of the form

$$\tilde{L}(q,p) = \sum_{k=1}^{N_\xi} \xi_k \Theta_k^{(L)}(q,p), \qquad \tilde{P}(q,p) = \sum_{k=1}^{N_\zeta} \zeta_k \Theta_k^{(P)}(q,p), \tag{88}$$

where $\Theta_k^{(L)}$ and $\Theta_k^{(P)}$ are matrices that are at most linear in the canonical variables q and p. A simple count shows that the vector $\eta = [\xi\ \zeta]$ belongs to $\mathbb{R}^{24}$. Now, to build the loss function, observe by the chain rule that

$$\frac{dL}{dt} = \frac{\partial L}{\partial q}\dot{q} + \frac{\partial L}{\partial p}\dot{p} = \frac{\partial L}{\partial q}\frac{\partial H}{\partial p} - \frac{\partial L}{\partial p}\frac{\partial H}{\partial q} = \{L, H\}.$$

Given the symbolically tractable hypothesis (Equation (88)), analytically calculating the Poisson bracket $\{L, H\}$ is straightforward. Since $\frac{dL}{dt} = [L, P]$, if follows that $\{L, H\} - [L, P]$ should hold for any point in phase space $(q, p) \in \mathbb{R}^2$. Lastly, we comment that the trivial solution $L \equiv 0$ is viable, so care should be taken to penalize away from $\xi \equiv 0$.

SILO is thus formulated as the following sparse, empirical risk minimization problem

$$\min_{\eta \in \mathbb{R}^{N_\eta}} \mathcal{J}[\eta] = \min_{\eta \in \mathbb{R}^{N_\eta}} \mathbb{E}_{(q,p)\sim\rho} \left\{ \sum_{i,j} \frac{\left(\{\tilde{L}, H\} - [\tilde{L}, \tilde{P}]\right)_{i,j}^2}{\{\tilde{L}, H\}_{i,j}^2} \right\} + r\mathcal{R}(\eta), \tag{89}$$

where $r \in [0, 1)$ controls the amount of desired sparsification in the search. Since the numerator, rewarding the discovery of a Lax pair compatible with Hamiltionain H, is divided by the Poisson bracket acting on $\tilde{L}$, this formulation discourages trivial solutions where $\tilde{L} \equiv 0$. Since it is infeasible to access the continuous nature of the phase space, we restrict to a finite number of samples from a uniform distribution ρ supported on a subset of the phase space $\mathbb{R}^2$.

By using sequential thresholding, SILO correctly identifies that the following two families of Lax pairs

$$\tilde{L}_1 = \begin{pmatrix} \eta_1 q & \eta_4 p \\ \eta_6 p & \eta_7 q \end{pmatrix}, \qquad \tilde{L}_2 = \begin{pmatrix} \eta_2 p & \eta_3 q \\ \eta_5 q & \eta_8 p \end{pmatrix}, \qquad \tilde{P} = \begin{pmatrix} 0 & \eta_{22} \\ \eta_{23} & 0 \end{pmatrix},$$

with all other 18 entries of η equal to zero in both families. The recovered Lax pairs reproduce the equations of motion with seven-digit precision with an unsparsified loss evaluation on the order of 10^{-15}. We comment that this precision is difficult to achieve using neural networks. See [7] for more details on the numerical implementation.

Using SILO, high-precision identification of the integrability of the famous Henon-Heiles (HH) system is made possible. This is a system with two degrees of freedom defined by the Hamiltonian [104, 84]

$$H = \frac{1}{2}\left(p_x^2 + p_y^2\right) + \frac{1}{2}\left(Ax^2 + By^2\right) + x^2 y + \varepsilon y^3, \tag{90}$$

where the parameters A, B, and ε are arbitrary. This system is known to be integrable for three different cases of these parameters [197], and we will focus on one such case: $A = B = 1$ and $\varepsilon = 1/3$.

To adapt the optimization of (89) to the Henon-Heiles setting, we introduce another simplified symbolic hypothesis on the Lax pair; please see [7] for details. Furthermore, by using

$$\{L, H\} = \frac{\partial L}{\partial x}\frac{\partial H}{\partial p_x} + \frac{\partial L}{\partial y}\frac{\partial H}{\partial p_y} - \frac{\partial L}{\partial p_x}\frac{\partial H}{\partial x} - \frac{\partial L}{\partial p_y}\frac{\partial H}{\partial y}$$

and generalizing the sampling of the phase space to four dimensions, that is, $(x_k, p_{x_k}, y_k, p_{y_k}) \sim U([-1, 1]^4)$, SILO is ready to be deployed. Indeed, SILO once again finds a loss on the order of 10^{-15}, in the integrable case of $A = B = 1$ and $\varepsilon = 1/3$.

From this point, we further investigate how SILO responds to training on parameter sets where integrability is unknown. We show, in Figure 8, that SILO experiences a significant increase in the computed loss–by several orders of magnitude–across the parameter space. Thus, restricted to the operator hypothesis used, SILO determines where the integrability lies in the parameter space (A, ε) for fixed B.

The last ODE example we study using SILO is an example from rigid body dynamics. The Euler top is a three-dimensional rigid body whose angular velocities Ω_i are governed by

$$I_i \dot{\Omega}_i = \sum_{j,k=1}^{3} \epsilon_{ijk}(I_j - I_k)\,\Omega_j \Omega_k \tag{91}$$

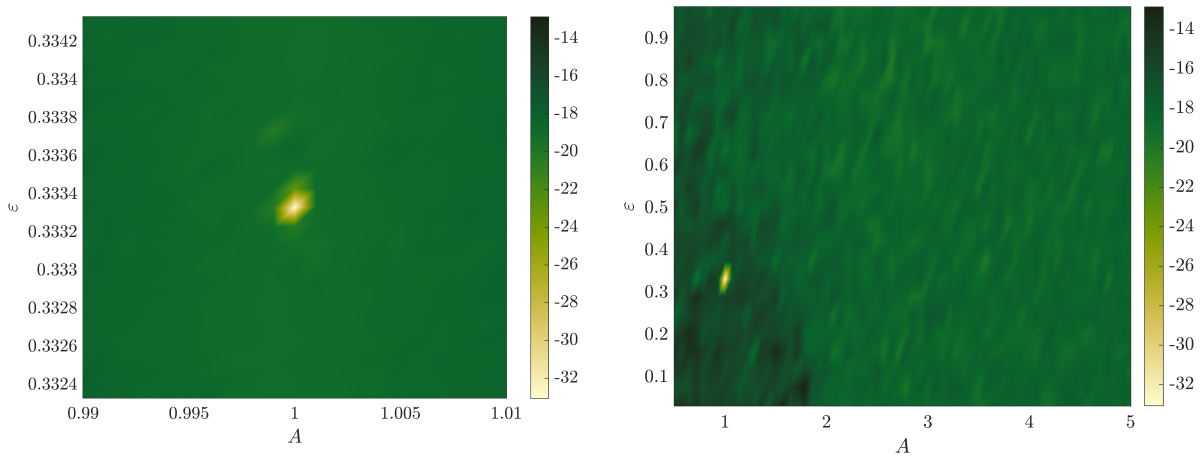


Figure 8: A broad parameter search (with B fixed to 1) for integrability detection in the HH system given by Hamiltonian (90). The optimization of the loss function, shown on a logarithmic scale, identifies a distinct position at $(A, \varepsilon) = (1, 1/3)$ where integrability is meaningfully detected, differing by several orders of magnitude from background loss values. Cubic spline interpolation of the landscape is used for visualization.

where each principal moment of intertia I_j is a constant, positive real number and ϵ_{ijk} is the Levi-Civita symbol. Lax integrability is known because (91) is compatible with the matrices

$$L = \begin{pmatrix} 0 & -M_3 & M_2 \\ M_3 & 0 & -M_1 \\ -M_2 & M_1 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{pmatrix},$$

satisfying Lax's equation $dL/dt = [L, P]$, where $M_i = I_i \Omega_i$.

Straightforward algebra shows this particular L for the Euler top does not generate the well-known three algebraically independent conserved quantities [14]. Thus, this represents a so-called fake Lax pair [201]. The trick, first found by Manakov [161], for converting this fake Lax pair into one that leads to Liouville integrability is to introduce a diagonal matrix

$$J = \frac{1}{2} \begin{pmatrix} I_2 + I_3 - I_1 & 0 & 0 \\ 0 & I_1 + I_3 - I_2 & 0 \\ 0 & 0 & I_1 + I_2 - I_3 \end{pmatrix}$$

that leaves Lax's equation invariant under

$$\frac{d}{dt}\left(L + \lambda J^2\right) = \frac{dL}{dt} = \left[L + \lambda J^2, P - \lambda J\right] = [L, P],$$

where λ is arbitrary and often referred to in the literature as the spectral parameter. This provides the remedy matrix $\hat{L} = L + \lambda J^2$ which can be shown to reproduce all three algebraically independent conserved quantities via $\frac{d}{dt}\mathrm{tr}\hat{L}^{2n} = 0$, $n = 1, 2, 3$, as long as the spectral parameter $\lambda \neq 0$.

It is useful to note that the Euler top's Hamiltonian $H = \sum_{i=1}^{3} \frac{\Omega_i^2}{2I_i}$ can be used to more compactly write the three equations of motion as $\dot{\Omega}_i = \{H, \Omega_i\}$, where curly brackets indicate once again the Poisson bracket. Knowledge of this Hamiltonian allows us to compactly represent the total derivative, in time, of any operator L satisfying Lax's equation. Using the chain rule, we observe that

$$\frac{dL}{dt} = \sum_i \frac{\partial L}{\partial \Omega_i}\dot{\Omega}_i + \frac{\partial L}{\partial I_i}\dot{I}_i = \sum_i \frac{\partial L}{\partial \Omega_i}\{H, \Omega_i\} = [L, P]$$

Once again, assuming a hypothesis pair $\left(\tilde{L}, \tilde{P}\right)$, and recognizing that Lax's equation holds pointwise in phase space, the loss function enabling statistical learning is given by

$$\mathcal{J}_{\text{loss}}(\tilde{L}, \tilde{P}) = \mathbb{E}_{\Omega\sim\rho} \sum_{j,k} \left(\sum_i \frac{\partial \tilde{L}}{\partial \Omega_i}\{H, \Omega_i\} - [\tilde{L}, \tilde{P}] \right)_{j,k}^2$$

where the expected value is taken over a uniform distribution ρ defined on the unit cube centered at the origin. Further assuming a parametrization vector η in our Lax pair hypothesis, a sparse identification of Lax pairs is found by solving the following optimization problem

$$\min_{\eta\in\mathbb{R}^N} (1-r)\mathcal{J}_{\text{loss}}[\eta] + r\mathcal{R}[\eta]$$

where $\mathcal{R}[\eta]$ and $r > 0$ in tandem promote sparse discoveries.

We have foregone a penalization to steer away from the trivial Lax pair. This turns out to be a sensitive issue that we now illustrate. Suppose that we use the simple linear hypotheses

$$\tilde{L}_{i,j} = \sum_k \xi_{i,j,k}\Omega_k, \qquad \tilde{P}_{i,j} = \sum_k \zeta_{i,j,k}\Omega_k, \tag{92}$$

and that we minimize $\mathcal{J}_{\text{loss}}$ over the 54 implied parameters $\xi_{i,j,k}$ and $\zeta_{i,j,k}$. Of course, the first difficulty we encounter is that our numerical results consistently return the (valid) trivial solution $\xi_{i,j,k} = \zeta_{i,j,k} = 0$. We have many choices to steer away from the trivial solution, including the strategy used in the context of the simple harmonic oscillator and Henon-Heiles system earlier in this section. Using that strategy, we found degenerate Lax pairs, such as for example

$$L_{\text{degen}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \Omega_1 & \Omega_2 \\ 0 & \frac{I_2(I_3-I_2)}{I_1(I_3-I_1)}\Omega_2 & -\Omega_1 \end{pmatrix}, \qquad P_{\text{degen}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{I_3-I_1}{2I_2}\Omega_3 \\ 0 & \frac{I_2-I_3}{2I_1}\Omega_3 & 0 \end{pmatrix} \tag{93}$$

which only reproduce the dynamics of two of the three expected Euler velocities.

The solution we found to learning Lax pairs for the Euler top in a principled way has two necessary ingredients. First, we found it necessary to enforce skew-symmetry

in (92) by demanding that for each i, j, and k

$$\frac{\partial \tilde{L}_{i,j}}{\partial \Omega_k} = -\frac{\partial \tilde{L}_{j,i}}{\partial \Omega_k}, \qquad \frac{\partial \tilde{P}_{i,j}}{\partial \Omega_k} = -\frac{\partial \tilde{P}_{j,i}}{\partial \Omega_k}.$$

Note that this compact notation only expresses the skew-symmetry condition when the hypothesis is linear. Also note that this choice considerably reduces the problem from a 54 parameter search to an 18 parameter one. The second ingredient is to introduce a soft penalization with the intention that each variable Ω_k remains represented so that all dynamics are reproduced by the learned Lax pair. We express this through

$$\mathcal{J}_{\text{degen}} = \prod_{k=1}^{3} \left\| \frac{\partial \tilde{L}}{\partial \Omega_k} \right\|_F^2 \left\| \frac{\partial \tilde{P}}{\partial \Omega_k} \right\|_F^2 \tag{94}$$

where $\|A\|_F^2 := \text{tr}(A^\top A)$ denotes the squared Frobenius norm. The problem is now to solve

$$\min_{\eta \in \mathbb{R}^N} \mathcal{J}_{\text{Euler}}[\eta] = \min_{\eta \in \mathbb{R}^N} (1-r)\left(\delta \mathcal{J}_{\text{degen}}^{-1}[\eta] + (1-\delta)\mathcal{J}_{\text{loss}}[\eta]\right) + r\mathcal{R}[\eta]$$

where $\delta \in (0,1)$ balances avoidance of degeneracy with model discovery through the loss $\mathcal{J}_{\text{loss}}$. We emphasize that if any factor in (94) is not considered in the design of the optimization problem, then numerical results will consistently degenerate to fake Lax pairs such as the one given by (93).

Once the skew-symmetric, non-degenerate Lax pair (L_*, P_*) is found numerically, we then search for the Manakov shift by solving the optimization problem

$$\min_{\mathcal{M}_1, \mathcal{M}_2 \in \mathbb{R}^3_{\text{diag}}} \mathbb{E}_{\Omega \sim \rho} \mathbb{E}_{\lambda \sim \Lambda} \mathcal{J}_{\text{Manakov}}(\mathcal{M}_1, \mathcal{M}_2) \tag{95}$$

where the Manakov objective is defined by

$$\mathcal{J}_{\text{Manakov}} = \sum_{j,k} \left([L_* + \lambda \mathcal{M}_1, P_* + \lambda \mathcal{M}_2] - [L_*, P_*] \right)_{j,k}^2,$$

and $\Lambda = \text{Uniform}((-1,1])$. The solution of this problem thus furnishes the Manakov shift for every arbirtrary spectral parameter $\lambda \in (0,1]$. We again report that SILO learns the expected Lax pair, with the correct conservation laws through learned Manakov shifts, to about seven digits of precision.

We see here that a drawback of SILO is the amount of user-specified information needed to build robust libraries and loss functions that avoid the degeneracy that plagues Lax's equation. Yet, if one is able to contend with these mathematical obstacles to engineer the correct SILO framework for a given Hamiltonian problem, then one can expect sharper resolution of Lax pairs and integrability of the dynamics.

6.3. Learning KdV Integrability Using SILO

We now show how a PDE system's integrability can be learned using SILO, the Korteweg-deVries (KdV) equation

$$\partial_t u - 6u\partial_x u + \partial_x^3 u = 0. \tag{96}$$

The Hamiltonian form of the KdV equation is given by [248]

$$u_t = Q\frac{\delta H}{\delta u}$$

where

$$H = \int_{-\infty}^{\infty} \left(u^3 - \frac{1}{2} u u_{xx} \right) dx := \int_{-\infty}^{\infty} h\left(u, u_{xx}\right) dx. \tag{97}$$

The KdV equation is bi-Hamiltonian, that is, two choices of Q (with corresponding choices of H) yield the equation of motion. For the choice of H used here, the relevant operator is $Q = \partial_x$.

The following Lax pair is well-known [91, 5]:

$$L = -\partial_x^2 + u, \qquad P = 4\partial_x^3 - 6u\partial_x - 3u_x. \tag{98}$$

The KdV equation can thus be viewed as the compatibility between these differential operators; that is, the equation

$$\partial_t L = [L, P]$$

reproduces Equation (96).

To yet again build a loss function for SILO, we use the chain rule. Observe that

$$\partial_t L = \frac{\partial L}{\partial u}\frac{\partial u}{\partial t} = \frac{\partial L}{\partial u} Q \frac{\delta H}{\delta u} = [L, P].$$

Thus, the expression $\frac{\partial \tilde{L}}{\partial u} Q \frac{\delta H}{\delta u}$ plays the role of the Poisson bracket in the design of the loss functions from previous sections. By direct analogy with Problem (89), our optimization problem for this Hamiltonian setting is given by

$$\min_{\eta \in \mathbb{R}^{N_\eta}} J[\eta] = \min_{\eta \in \mathbb{R}^{N_\eta}} (1-r)\mathbb{E}_{u\sim\rho} \frac{\int \left| \frac{\partial \tilde{L}}{\partial u} Q \frac{\delta H}{\delta u} u - [\tilde{L}, \tilde{P}]u \right|^2 dx}{\int \left| \frac{\partial \tilde{L}}{\partial u} Q \frac{\delta H}{\delta u} u \right|^2 dx} + r\mathcal{R}^*(\eta) \tag{99}$$

where ρ is a subset of the KdV phase space and $\mathcal{R}^*$ is a sparsification function.

In this PDE setting, there are only two additional modifications we need to make to the numerical framework from previous sections. First, we must carefully consider how we sample the phase space and how we construct the operator hypotheses. This amounts to sampling from the function space $L^p(\mathbb{R})$, $p > 1$. To this end, we construct random samples from the overcomplete basis

$$u^{rand}(x) = N \sum_j e^{-a_j(x-b_j)^2} \sum_k \frac{A_{jk}}{k^3} \sin\frac{k\pi x}{L} \tag{100}$$

where all parameters $a_j,\ b_j, A_{jk}$ are appropriately sampled from uniform distributions, L is the length of the truncated spatial domain, and the coefficient N ensures a unit norm in the space $L^1(\mathbb{R})$. In this way, we try to verify the compatibility of our operators with functions sampled from the function space $C^3(\mathbb{R}) \cap L^p(\mathbb{R})$, $p > 1$ and with equivalent

masses in the sense of $L^1(\mathbb{R})$. We ensure smoothness of the samples by the decay of the Fourier coefficients $A_{j,k}/k^3$ while also regularizing by the fact that we only take the sum over k to be finite. This smoothness is used for computational tractability in the evaluation of differential operators in the loss. We find $k = 10$ to be sufficient in the numerics.

The second modification is the operator hypothesis. In general, Lax pairs are linear differential operators (in x) with coefficients dependent on x, u, and derivatives of u. A fairly wide class of operators is given by

$$\tilde{L} = \xi_1 u + \xi_2 \partial_x + \xi_3 \partial_x^2, \quad \tilde{P} = \sum_j \sum_k \sum_m \zeta_{jkm} u^{j-1} \left(\partial_x^{k-1} u\right) \partial_x^{m-1}$$

where the indices in the sum all start at one. It may seem like the hypothesis on $\tilde{L}$ is overly restrictive. However, we must keep in mind that we seek to reproduce an evolution equation involving $\partial_t \tilde{L}$. Should any higher powers of u enter into the hypothesis of $\tilde{L}$, then the likelihood of finding an *explicit* equation of the form $u_t = F(u, u_x, u_{xx}, ...)$ decreases substantially. In this sense, we seek to preserve the semi-linear (in time) nature of the PDE of interest. To compute the spatial derivatives in the operator hypothesis, we use the Fast Fourier transform. That is, we use the formula

$$\partial_x^j u = \mathcal{F}^{-1}\left\{(ik)^j \mathcal{F}\{u\}\right\}$$

to compute the derivatives spectrally, where k denotes the grid-dependent wavenumbers.

In our training, we find that only using 20 samples is sufficient in our cross-validation. After training on these samples, we find that the loss, on average and without sparsification, is on the order of 10^{-11} when evaluated on unseen samples from Ω. We show, in Figure 9, a visual comparison between $\frac{\partial \tilde{L}}{\partial u} Q \frac{\delta H}{\delta u}$ and $[\tilde{L}, \tilde{P}]$ for four unseen sample functions. Note that despite the unit norms of the samples in $L^1(\mathbb{R})$, the generalized Poisson bracket and matrix commutators have arbitrary norms, which account for the variance of the scales across these images.

To further demonstrate the validity of our approach, we show that our framework is sensitive enough to detect the known integrability of the KdV equation, similar to how was done for the Henon-Heiles system. Consider the non-integrable perturbation $h_1 = \frac{1}{2}\left(\partial_x^2 u\right)^2$. We solve Problem (99), again without sparsification, for Hamiltonian densities $h + \varepsilon h_1$, where h is defined in Equation (97) and $\varepsilon \in [-.01, .01]$. Figure 10 shows that the loss has a nearly smooth dependence on the parameter ε with a minimum at the expected integrable point $\varepsilon = 0$. Again, the integrability point is privileged, as the loss is several orders of magnitude smaller than the nearby values of ε.

We now discuss the interpretability of building Lax pairs from solving Problem (99). Without sparsification, it is not surprising that all 39 coefficients in our operators are activated. Therefore, even with computer algebra systems such as Mathematica, we have no chance to interpret the PDE that the Lax pair is producing. Before discussing what we discover through sparsification, we make the following basic observations.

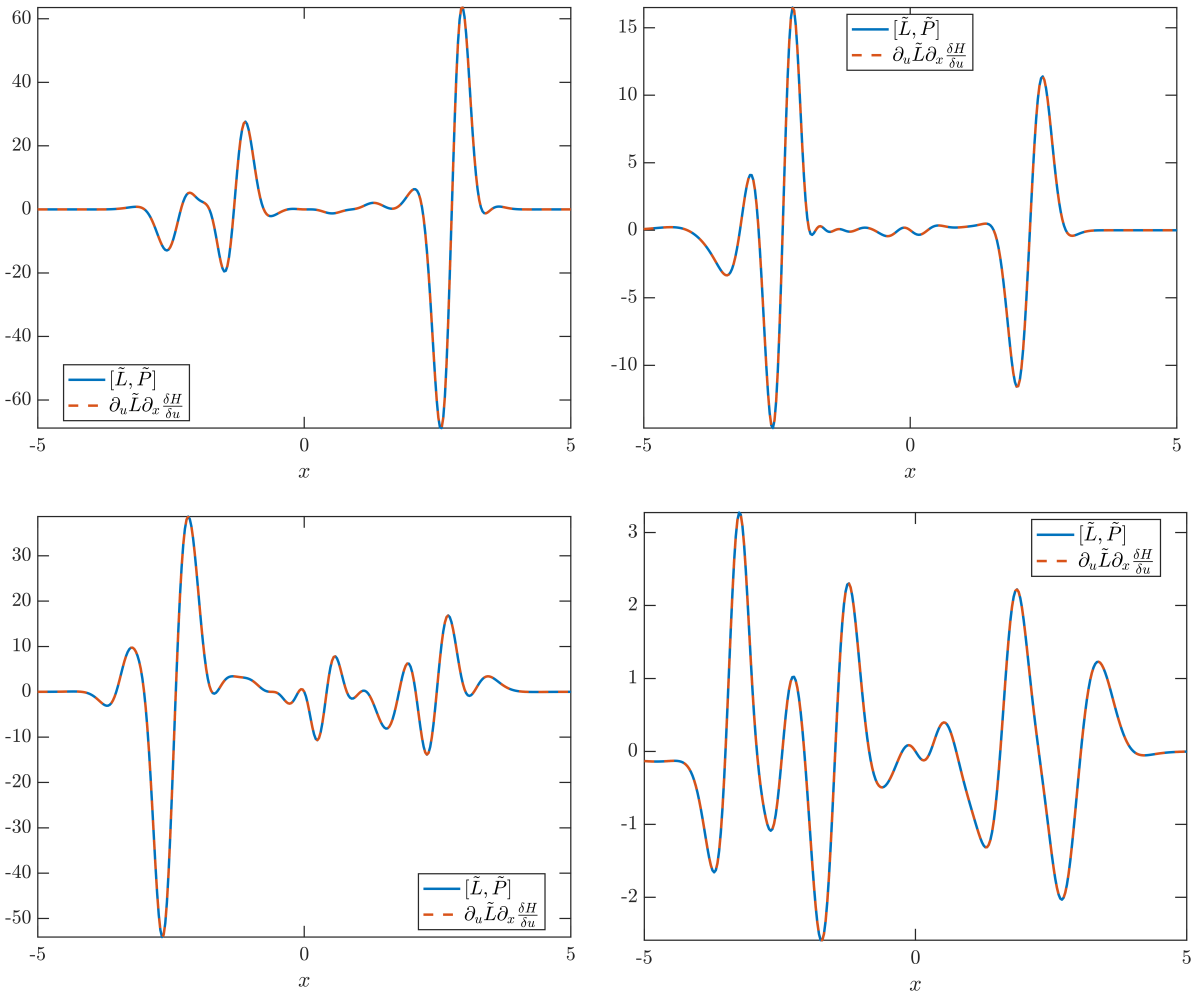


Figure 9: A numerical result of solving Problem (99) without sparsification. Visualized here is a cross-validation study displaying the generalized Poisson brackets and commutators evaluated at the optimal point η^* and on four samples from the function space Ω that were unseen during training. For all four cases, the loss is on the order of 10^{-11}.

Once again, we use the same sparsification from the finite-degree-of-freedom setting to aid us in our interpretation of our discovered Lax pairs. Alongside the expected Lax pair, which we tend to rediscover only when constraining L to be self-adjoint, SILO discovers an entirely new family of Lax pairs, communicated by the following theorem.

Theorem 2 (Existence of a new KdV Lax pair)**.** For every $u \in C^1([0,T];C^3(\mathbb{R}))$, there exists a parameter $v \in \mathbb{R}$ such that the pair of operators

$$L = \alpha u + \beta \partial_x,$$
$$P = \gamma u + \delta u^2 + \epsilon u_{xx} + \kappa \partial_x,$$

satisfying Lax's equation $\partial_t L = [L,P]$, understood as acting on the function space

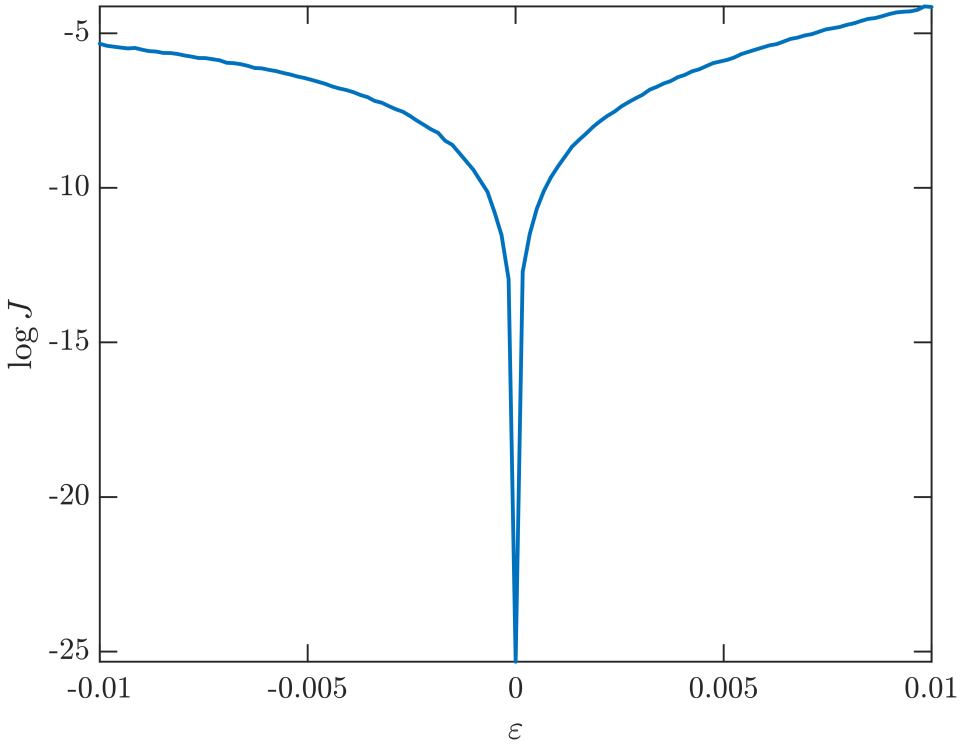


Figure 10: A perturbation study using the perturbed Hamiltonian density $h_1 = \frac{\varepsilon}{2}(\partial_x^2 u)^2$. Shown here is the numerical solution of Problem (99), without sparsification, using the full density $h+\varepsilon h_1$. We observe a near-smooth dependence on ε with a clearly discernible "special" point associated with the detection of integrability at $\varepsilon = 0$.

$C^3(\mathbb{R})$, reproduces the KdV equation

$$u_t = \frac{2\beta\delta}{\alpha} u u_x + \frac{\beta\epsilon}{\alpha} u_{\mathrm{xxx}}$$

in the co-traveling reference frame $x \to x - vt$.

A natural question raised by our discussion of Lax pairs is whether one should also consider "fake" Lax pairs, i.e., formulations that do not lead to a nontrivial IST. This is indeed a subtle point: in certain cases, such as the Euler top, it is possible to construct a Lax representation whose associated scattering data are trivial, and thus no meaningful integrable structure is obtained. The new Lax pair that we report above is indeed a fake Lax pair, which can be verified by hand when manually constructing the associated Jost functions and their dynamics. In this context, it is especially relevant for future work to consider the integration of penalizations that steer us away from the space of fake Lax pairs during training.

7. Conclusions and Outlook

In this review, we have surveyed the emerging interplay between nonlinear wave dynamics and machine learning, emphasizing how modern data-driven tools can be

strengthened by embedding analytical structure. Section 2 outlined the main pillars of scientific machine learning that form the majority of the foundation for the methods used throughout this review. Section 3 reviewed advances in PINNs and their variants, highlighting both the potential and limitations of incorporating PDE constraints directly into learning architectures. Section 4 turned to reduced order modeling, with a particular focus on SINDy [40] and related sparse regression approaches, which provide interpretable surrogates for complex dynamics. Section 5 examined methods for learning structural properties such as conservation laws, Hamiltonian and metriplectic structure, and related physical invariants. Section 6 extended this discussion to the discovery of integrability, exploring recent attempts to learn Lax pairs, integrability, and conserved-quantity hierarchies.

There are several future directions in nonlinear waves to consider, including the study of another important scenario in nonlinear waves, namely that of soliton gases. These thermodynamic ensembles of interacting solitons offer a natural bridge between integrable PDEs and statistical mechanics [76, 77, 26]. Machine learning methods may accelerate inference of effective kinetic equations, provide reduced surrogates for ensemble dynamics, and extend the scope of statistical soliton theory to non-integrable perturbations. By leveraging neural operators and generative models, it may become possible to simulate large soliton ensembles more efficiently, uncover new statistical closures, and explore thermodynamic regimes that remain inaccessible to current analytical techniques.

Another major thread of intense recent exploration concerns the dynamics, interactions (between them, as well as with solitary waves), higher-dimensional, as well as discrete realizations of dispersive shock waves (DSWs). Here, there is a central theoretical development that still merits extensive study, namely the so-called Whitham modulation theory [238]. Despite crossing the half-century mark of numerous hallmark developments associated with this theory, the field remains extremely active to this day, in part due to the development of novel theoretical tools (such as the so-called DSW fitting [78]), and also due to the addition of numerous experimental platforms where such structures can be identified; see, e.g., the review of [79]. Nevertheless, in many cases, the equations of modulation theory remain extremely difficult to identify (especially so in non-integrable models, where, e.g., analytical waveforms for periodic waves may not be available), or in other cases can be derived, yet are extremely difficult to analyze and understand the features/structural characteristics thereof. Indeed, beyond the simpler playground of 1D continuum models, e.g., even in 1D discrete settings, or 2D continuum ones, it is fair to say that a deeper development and understanding of the theory and its implications still pose extensive and formidable challenges. It is hoped that, e.g., the techniques discussed herein and their particular relevance in especially inverse, but also direct problem solutions may be of significant assistance in this vein of research.

Reduced-order modeling within the nonlinear waves context is another key direction to consider. The recently proposed SHRED-ROM framework [227], based on recurrent decoder networks with compressive training, provides a sensor-driven route to compact

surrogates for dispersive systems and merits careful benchmarking across integrable, near-integrable, and turbulent regimes, as well as extensions to higher-dimensional wave equations. At the same time, hyperbolic and transport-dominated problems remain notoriously challenging for projection onto linear subspaces because the singular values of snapshot ensembles decay slowly—reflecting the slow Kolmogorov n-width decay of these solution manifolds [185, 97]. Recent strategies to overcome this include co-moving or transport-aware representations such as shifted POD (sPOD) [198] and its robust variants [133], registration and optimal-transport alignment of features before reduction [29], and calibration methods based on arbitrary Lagrangian–Eulerian (ALE) mappings, which allow the computational mesh to move with dominant features in order to "straighten" moving discontinuities into a lower-dimensional manifold [177].

A further avenue involves climbing the conserved-quantity hierarchy. Although recent methods can recover conserved quantities such as mass, momentum, and energy, future algorithms should aim to automatically discover higher-order invariants, thereby reconstructing larger portions of the integrable tower. Such developments would also allow for the systematic study of how integrability breaks under perturbation. Beyond fundamental insights, these advances could provide powerful diagnostic tools for distinguishing chaotic from near-integrable dynamics in high-dimensional systems. Such developments could also play a central role towards a deeper understanding of the action-angle formulations of integrable and near-integrable systems. An important open direction there concerns especially how non-integrable perturbations may form short- (or progressively longer-) networks of connections between the actions, a perspective that has been recently occasionally argued in the literature [172, 159], but has yet to be quantified more broadly.

Symbolic learning frameworks such as AI-Descartes [60] and AI-Hilbert [61] show a strong potential to complement sparse regression approaches. By combining data with background knowledge, these systems have already demonstrated the ability to rediscover canonical scientific laws. A key ingredient in their design is the integration of algebraic geometry into the learning process. Hilbert's nullstellensatz provides a theoretical foundation for certifying the validity of candidate relations, while Gröbner basis methods enable systematic elimination of variables and simplification of polynomial structures. This marriage of symbolic reasoning and computational algebra allows the algorithms not only to fit data but also to ensure that the recovered expressions satisfy fundamental consistency conditions. For nonlinear dispersive PDEs, such capabilities could be transformative, opening pathways to the automated discovery of algebraic invariants, the construction of statistical closures that respect underlying constraints, and the reduction of kinetic models into interpretable symbolic forms. In the broader landscape of nonlinear waves and machine learning, the promise of these approaches lies in their ability to bridge rigorous mathematical theory with flexible data-driven pipelines, ensuring that discovered models remain both physically grounded and algebraically consistent.

Meanwhile, operator learning continues to provide a broad frontier. FNOs and

related architectures [142, 130] have demonstrated significant success in fluid mechanics and wave propagation. Extending operator learning to dispersive and integrable systems may open new possibilities for efficient simulation and real-time control. Closely related generative models such as variational autoencoders and Boltzmann generators [125, 176] could help explore invariant measures, approximate soliton gases, and rare event statistics in nonlinear wave ensembles.

Briefly expanding upon generative models, we note that methods based on variational autoencoders and generative adversarial networks (GANs) are opening new frontiers in synthesizing realistic wave field statistics in turbulent or stochastic regimes [19], while reinforcement learning (RL) is emerging as a tool for optimal wave control in nonlinear PDEs [242]. Some of these recent advances in generative modeling have opened new avenues in fundamental wave physics, particularly for canonical systems such as water waves and BECs. For example, physics-informed diffusion models have shown promise in reconstructing high-fidelity fluid dynamics fields - including shallow-water and Navier-Stokes regimes - by enforcing equation constraints during sampling [232, 254]. Neural ODE frameworks extended with wave-equation-inspired architectures (e.g., "neural wave equation") provide continuous-time evolution learning suitable for BEC dynamics and dispersive hydrodynamics [158, 52]. Recent advances in physics-based flow matching provide a powerful framework for generating wavefield ensembles that explicitly respect the governing partial differential equations, making them highly suitable for surrogate modeling, uncertainty quantification, and accelerated simulation in wave phenomena [18]. Finally, stochastic interpolation via diffusion in function spaces, such as FunDiff, enables the generation of continuous wavefield realizations that obey conservation laws, which is critical for high-fidelity modeling in quantum and classical nonlinear wave systems [232].

It is important to acknowledge a notable omission in this review: the rapidly growing body of work on Koopman operator theory and its applications to nonlinear dynamics. The Koopman framework, which lifts nonlinear systems into an infinite-dimensional linear setting [170, 171, 42, 35], has inspired a variety of data-driven algorithms such as Dynamic Mode Decomposition (DMD) [215, 205, 135] and a wide range of extensions. These methods have already had substantial impact in fluid dynamics and coherent structure extraction, and they provide a natural language for studying spectral properties of nonlinear waves. Recent surveys [37, 128] have highlighted the breadth of Koopman approaches, covering theoretical advances, numerical algorithms, and practical applications in complex systems. Exciting developments include kernelized and extended DMD methods (EDMD) for handling nonlinear observables [240], dictionary learning strategies for adaptive Koopman embeddings [9], and deep learning–based architectures that approximate Koopman operators directly from high-dimensional data [224, 157]. Although Koopman-based approaches were not addressed in detail here, they represent a promising frontier for integration with operator learning, reduced-order modeling, and even the discovery of an integrable structure. Future directions include the design of Koopman architectures

tailored to dispersive PDEs, hybrid schemes that merge Koopman embeddings with physics-informed priors, the use of Koopman operator spectra to detect key dispersive PDE features such as conservation laws and integrability, and applications to wave turbulence and spatiotemporal chaos and beyond. It should be noted that so far only a minimal subset of publications, to our knowledge, has sought to develop Koopman-based analysis for nonlinear wave problems such as the Burgers [179] or the KdV [181, 182] and have encountered some nontrivial associated complications in their analytically driven constructions. A systematic application of EDMD and related data-driven methods in nonlinear wave systems appears to still merit considerable further study.

Taken together, these directions suggest that the integration of the rich mathematical structure of nonlinear wave systems with machine learning is still in its infancy, with much more ahead than behind. Approaches that respect conservation laws, integrability, and operator structure will remain central in guiding data-driven models so that they reflect (and shed further light on), rather than obscure, the underlying physics. Indeed, by embedding these principles, we not only safeguard interpretability and fidelity but also open new avenues for discovery, expanding the class of models and phenomena that can be meaningfully explored. The convergence of classical analysis and modern machine learning thus points toward a future in which our theoretical understanding and computational capabilities advance hand in hand. We hope that this review will encourage researchers from applied mathematics, nonlinear physics, and scientific machine learning to join in shaping this emerging landscape, using the complex, rich and ever-intriguing platform of nonlinear waves as their exploration ground of choice.

Acknowledgments

This material is based upon work supported by the U.S. National Science Foundation under the award PHY-2316622 (JA), PHY-2110030, PHY-2408988 and DMS-2204702 (P.G.K.) and DMS-2502900 (WZ) and by the Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-25-1-0079 (WZ).

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org.

[2] M. J. Ablowitz, *Nonlinear Dispersive Waves: Asymptotic Analysis and Solitons*, Cambridge University Press, 2011.

[3] M. J. Ablowitz and J. F. Ladik, *Nonlinear differential–difference equations*, Journal of Mathematical Physics, 16 (1975), pp. 598–603.

[4] ——, *Nonlinear differential–difference equations and Fourier analysis*, Journal of Mathematical Physics, 17 (1976), pp. 1011–1018.

[5] M. J. Ablowitz and H. Segur, *Solitons and the inverse scattering transform*, SIAM, 1981.

[6] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2008.

[7] J. Adriazola, W. Zhu, P. Kevrekidis, and A. Aceves, *Computer assisted discovery of integrability via silo: Sparse identification of lax operators*, arXiv preprint arXiv:2503.00645, (2025).

[8] A. Ankiewicz, N. Akhmediev, and J. M. Soto-Crespo, *Discrete rogue waves of the Ablowitz-Ladik and Hirota equations*, Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, 82 (2010).

[9] H. Arbabi and I. Mezic, *Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator*, SIAM Journal on Applied Dynamical Systems, 16 (2017), pp. 2096–2126.

[10] V. I. Arnol'd, *Mathematical methods of classical mechanics*, vol. 60, Springer Science & Business Media, 2013.

[11] S. Arora, A. Bihlo, R. Brecht, and P. Holba, *Model-agnostic machine learning of conservation laws from data*, arXiv preprint arXiv:2301.07503, (2023).

[12] M. Atif, P. Dubey, P. P. Aghor, et al., *Fourier neural operators for spatiotemporal dynamics in two-dimensional turbulence*, arXiv preprint arXiv:2409.14660, (2024).

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

[14] O. Babelon, D. Bernard, and M. Talon, *Introduction to Classical Integrable Systems*, Cambridge University Press, 2003. Discussion of Lax pairs and conserved quantities; includes examples such as the Euler top.

[15] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, International Conference on Learning Representations (ICLR), (2015).

[16] J. Bakarji, K. P. Champion, J. N. Kutz, and S. L. Brunton, *Discovering governing equations from partial measurements with deep delay autoencoders*, CoRR, abs/2201.05136 (2022).

[17] N. A. Baker, F. J. Alexander, P.-T. Bremer, A. A. Hagberg, and et al., *Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence*, arXiv preprint arXiv:1909.10799, (2019).

[18] G. Baldan, Q. Liu, A. Guardone, and N. Thuerey, *Flow matching meets pdes: A unified framework for physics-constrained generation*, arXiv preprint arXiv:2506.08604, (2025).

[19] A. Barati Farimani, M. Arjovsky, and P. J. Atzberger, *Deep generative modeling of wavefields using variational autoencoders and gans*, Journal of Computational Physics, 462 (2022), p. 111242.

[20] A. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.

[21] 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), pp. 1–43.

[22] J. Belmonte-Beitia, G. F. Calvo, and V. M. Pérez-García, *Effective particle methods for fisher–kolmogorov equations: Theory and applications to brain tumor dynamics*, Communications in Nonlinear Science and Numerical Simulation, 19 (2014), pp. 3267–3283.

[23] T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis, *On learning hamiltonian systems from data*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 29 (2019).

[24] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model reduction and*

neural networks for parametric pdes, Mathematical and Computational Research, (2021).

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

[26] G. Biondini, *Soliton gas dynamics and statistical mechanics*, Philosophical Transactions of the Royal Society A, 376 (2018), p. 20170189.

[27] S. Bishnoi, R. Bhattoo, J. Jayadeva, S. Ranu, and N. A. Krishnan, *Learning the dynamics of physical systems with Hamiltonian graph neural networks*, ICLR 2023 Workshop on Physics for Machine Learning, (2023).

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[29] J. Blickhan and B. Peherstorfer, *Learning optimal transport for model reduction of advection-dominated pdes*, Journal of Machine Learning Research, 25 (2024), pp. 1–45.

[30] R. Bondesan and A. Lamacraft, *Learning symmetries of classical integrable systems*, arXiv preprint arXiv:1906.04645, (2019).

[31] J. J. Bramburger, *Data-Driven Methods for Dynamic Systems*, vol. 201 of Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 2024.

[32] O. Braun and Y. Kivshar, *Nonlinear dynamics of the Frenkel-Kontorova model*, Physics Reports, 306 (1998), pp. 1–108.

[33] T. J. Bridges, *Multi-symplectic structures and wave propagation*, Mathematical Proceedings of the Cambridge Philosophical Society, 121 (1997), pp. 147–190.

[34] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, *Geometric deep learning: Going beyond euclidean data*, IEEE Signal Processing Magazine, 34 (2017), pp. 18–42.

[35] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, *Modern koopman theory for dynamical systems*, SIAM Review, 64 (2022), pp. 229–340.

[36] S. L. Brunton, X. Fu, J. L. Proctor, and J. N. Kutz, *Extracting spatial–temporal coherent patterns in large-scale wave turbulence simulations using dynamic mode decomposition*, Physica D: Nonlinear Phenomena, 311 (2015), pp. 106–116.

[37] S. L. Brunton, M. Korda, and I. Mezic, *Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control*, Proceedings of the National Academy of Sciences, 119 (2022), p. e2116723119.

[38] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press, (2019).

[39] 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 (2016), pp. 3932–3937.

[40] 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 (2016), pp. 3932–3937.

[41] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Sparse identification of nonlinear dynamics with control (SINDYc)*, IFAC-PapersOnLine, 49 (2016), pp. 710–715.

[42] M. Budišić, R. Mohr, and I. Mezić, *Applied koopmanisma)*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 22 (2012), p. 047510.

[43] T. Busch and J. R. Anglin, *Motion of dark solitons in trapped bose-einstein condensates*, Phys. Rev. Lett., 84 (2000), pp. 2298–2301.

[44] F. Calogero, *Solution of the one-dimensional n-body problems with quadratic and/or inversely quadratic pair potentials*, Journal of Mathematical Physics, 12 (1971), pp. 419–436.

[45] S. Cao, Q. Jin, Z. Li, and G. E. Karniadakis, *Physics-informed neural networks for high-speed flows*, Journal of Computational Physics, 447 (2021), p. 110676.

[46] R. Carretero-González, D. J. Frantzeskakis, and P. G. Kevrekidis, *Nonlinear Waves & Hamiltonian Systems: From One To Many Degrees of Freedom, From Discrete To Continuum*, Oxford University Press, Oxford, 2024.

[47] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, *Data-driven discovery of coordinates and governing equations*, Proceedings of the National Academy of Sciences, 116 (2019), pp. 22445–22451.

[48] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, *Data-driven discovery of coordinates and governing equations*, Proceedings of the National Academy of Sciences, 116 (2019), pp. 22445–22451.

[49] ——, *Data-driven discovery of coordinates and governing equations*, Proceedings of the National Academy of Sciences, 116 (2019), p. 22445–22451.

[50] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, *A unified sparse optimization framework to learn parsimonious physics-informed models from data*, IEEE Access, 8 (2020), pp. 169259–169271.

[51] R. Chen and M. Tao, *Data-driven prediction of general hamiltonian dynamics via learning exactly-symplectic maps*, in International conference on machine learning, PMLR, 2021, pp. 1717–1727.

[52] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural ordinary differential equations*, Advances in Neural Information Processing Systems, 31 (2018), pp. 6571–6583.

[53] S. Chen, P. G. Kevrekidis, H.-K. Zhang, and W. Zhu, *Data-driven discovery of conservation laws from trajectories via neural deflation*, Communications in Nonlinear Science and Numerical Simulation, 143 (2025), p. 108563.

[54] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, *Symplectic recurrent neural networks*, in International Conference on Learning Representations, 2020.

[55] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using rnn encoder–decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078, (2014).

[56] B.-J. Choi, H. S. Jin, and B. Lkhagvasuren, *Applications of the fourier neural operator in a regional ocean modeling and prediction*, Frontiers in Marine Science, 11 (2024), p. 1383997.

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

[58] T. S. Cohen and M. Welling, *Group equivariant convolutional networks*, in Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016, pp. 2990–2999.

[59] M. P. Coles, D. E. Pelinovsky, and P. G. Kevrekidis, *Excited states in the large density limit: a variational approach*, Nonlinearity, 23 (2010), p. 1753.

[60] C. Cornelio, S. Dash, V. Austel, T. R. Josephson, J. Goncalves, K. L. Clarkson, N. Megiddo, B. El Khadir, and L. Horesh, *Combining data and theory for derivable scientific discovery with ai-descartes*, Nature Communications, 14 (2023), p. 1777.

[61] R. Cory-Wright, C. Cornelio, S. Dash, B. El Khadir, and L. Horesh, *Evolving scientific discovery by unifying data and background knowledge with ai hilbert*, Nature Communications, 15 (2024), p. 5922.

[62] K. Course, T. Evans, and P. Nair, *Weak form generalized hamiltonian learning*, Advances in Neural Information Processing Systems, 33 (2020), pp. 18716–18726.

[63] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. N. Spergel, and S. Ho, *Lagrangian neural networks*, arXiv preprint arXiv:2003.04630, (2020).

[64] J. Cuevas-Maraver and P. G. K. (eds.), *A dynamical perspective on the ϕ^4 model*, Nonlinear Systems and Complexity, Springer International Publishing, 1st ed., 2019.

[65] J. Cuevas-Maraver, P. Kevrekidis, and F. Williams, *The sine-gordon model and its applications: From pendula and josephson junctions to gravity and*, High-energy Physics, 10 (2014), p. 263.

[66] 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*, vol. 10 of Nonlinear Systems and Complexity, Springer International Publishing, 2014.

[67] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Mathematics of control, signals and systems, 2 (1989), pp. 303–314.

[68] A. Daigavane, A. Kosmala, M. Cranmer, T. Smidt, and S. Ho, *Learning integrable dynamics with action-angle networks*, arXiv preprint arXiv:2211.15338, (2022).

[69] M. David and F. Méhats, *Symplectic learning for hamiltonian neural networks*, Journal of Computational Physics, 494 (2023), p. 112495.

[70] P. de Koster and S. Wahls, *Data-driven identification of the spectral operator in akns lax pairs using conserved quantities*, Wave Motion, 127 (2024), p. 103273.

[71] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, *PySINDy: A python package for the sparse identification of nonlinear dynamical systems from data*, Journal of Open Source Software, 5 (2020), p. 2104.

[72] D. DiPietro, S. Xiong, and B. Zhu, *Sparse symplectically integrated neural networks*, Advances in Neural Information Processing Systems, 33 (2020), pp. 6074–6085.

[73] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*, Springer, Berlin, Heidelberg, 2009.

[74] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM journal on Matrix Analysis and Applications, 20 (1998), pp. 303–353.

[75] S. Eidnes and K. O. Lye, *Pseudo-hamiltonian neural networks for learning partial differential equations*, Journal of Computational Physics, 500 (2024), p. 112738.

[76] G. El and A. Kamchatnov, *Kinetic equation for a dense soliton gas*, Physical Review Letters, 95 (2005), p. 204101.

[77] G. El and A. Tovbis, *Soliton gas in integrable dispersive hydrodynamics*, Physica D, 333 (2016), pp. 11–27.

[78] G. A. El, *Resolution of a shock in hyperbolic systems modified by weak dispersion*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 15 (2005).

[79] G. A. El and M. A. Hoefer, *Dispersive shock waves and modulation theory*, Physica D: Nonlinear Phenomena, 333 (2016), pp. 11–65.

[80] 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 (2003), p. 024403.

[81] P. E. Farrell, A. Birkisson, and S. W. Funke, *Deflation techniques for finding distinct solutions of nonlinear partial differential equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2026–A2045.

[82] E. Fermi, P. Pasta, S. Ulam, and M. Tsingou, *Studies of the nonlinear problems*, tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1955.

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

[84] A. P. Fordy, *The hénon-heiles system revisited*, Physica D: Nonlinear Phenomena, 52 (1991), pp. 204–210.

[85] F. Frezza, T. Righini, and N. Tedeschi, *Physics-informed neural networks for inverse problems in wave propagation*, Applied Sciences, 11 (2021), p. 8240.

[86] M. Fruchart, C. Scheibner, and V. Vitelli, *Interpreting neural operators: How nonlinear waves propagate in non-reciprocal solids*, Physical Review Letters, 133 (2024), p. 107301.

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

[88] C. R. Galley, *Classical mechanics of nonconservative systems*, Phys. Rev. Lett., 110 (2013), p. 174301.

[89] Y. Gao, R. Geng, P. Kevrekidis, H.-K. Zhang, and J. Zu, *α-separable graph hamiltonian network: A robust model for learning particle interactions in lattice systems*, Phys. Rev. E, 111 (2025), p. 015309.

[90] J. García-Ripoll and V. Pérez-García, *The moment method in general nonlinear schrodinger equations*, (1999).

[91] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura, *Method for solving the korteweg-devries equation*, Phys. Rev. Lett., 19 (1967), pp. 1095–1097.

[92] R. Geng, J. Zu, Y. Gao, and H.-K. Zhang, *Separable graph hamiltonian network: A graph deep learning model for lattice systems*, Physical Review Research, 6 (2024), p. 013176.

[93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. Standard modern textbook on deep learning.

[94] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[95] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in Advances in Neural Information Processing Systems, vol. 27, 2014.

[96] S. Goswami, Y. Zhu, S. K. Das, J. H. Panchal, and G. E. Karniadakis, *Physics-informed surrogate modeling for aerodynamic shape optimization*, Computer Methods in Applied Mechanics and Engineering, 393 (2022), p. 114718.

[97] A. Gowrachari and K. T. Carlberg, *Challenges and opportunities for reduced-order modeling of advection-dominated problems*, Annual Review of Fluid Mechanics, 57 (2025). to appear.

[98] S. Greydanus, M. Dzamba, and J. Yosinski, *Hamiltonian neural networks*, 2019.

[99] M. Grmela and H. C. Öttinger, *Dynamics and thermodynamics of complex fluids. i. development of a general formalism*, Physical Review E, 56 (1997), pp. 6620–6632.

[100] X. Guo, W. Li, and H. F. Iorio, *Convolutional neural networks for steady flow approximation*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016), pp. 481–490.

[101] S. Ha and H. Jeong, *Discovering invariants via machine learning*, Physical Review Research, 3 (2021), p. L042035.

[102] C.-D. Han, B. Glaz, M. Haile, and Y.-C. Lai, *Adaptable hamiltonian neural networks*, Physical Review Research, 3 (2021), p. 023156.

[103] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, CRC Press, Boca Raton, FL, 2015.

[104] M. Hénon and C. Heiles, *The applicability of the third integral of motion: Some numerical experiments*, The Astronomical Journal, 69 (1964), p. 73.

[105] Q. Hernández, A. Badías, F. Chinesta, and E. Cueto, *Thermodynamics-informed graph neural networks*, IEEE Transactions on Artificial Intelligence, 5 (2024), pp. 967–976.

[106] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Computation, 9 (1997), pp. 1735–1780.

[107] C. C. Horuz, M. Karlbauer, T. Praditia, S. Oladyshkin, W. Nowak, and S. Otte, *Inferring underwater topography with finn*, 2024.

[108] A. D. Jagtap and G. E. Karniadakis, *Extended physics-informed neural networks (xpinns): A generalized space–time domain decomposition based deep learning framework for nonlinear partial differential equations*, Communications in Computational Physics, 28 (2020), pp. 2002–2041.

[109] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, *Adaptive activation functions accelerate convergence in deep and physics-informed neural networks*, Journal of Computational Physics, 404 (2020), p. 109136.

[110] P. Jin, X. Meng, and G. E. Karniadakis, *Inverse problems with physics-informed deeponets*, Proceedings of the AAAI Conference on Artificial Intelligence, 36 (2022), pp. 6713–6721.

[111] P. Jin, Z. Zhang, I. G. Kevrekidis, and G. E. Karniadakis, *Learning poisson systems and trajectories of autonomous systems via poisson neural networks*, IEEE Transactions on Neural Networks and Learning Systems, 34 (2022), pp. 8271–8283.

[112] 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 (2020), pp. 166–179.

[113] K. Kaheman, J. N. Kutz, and S. L. Brunton, *Sindy-pi: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 476 (2020), p. 20200279.

[114] E. Kaiser, J. N. Kutz, and S. L. Brunton, *Sparse identification of nonlinear dynamics for model predictive control in the low-data limit*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474 (2018), p. 20180335.

[115] S. Kantamneni, Z. Liu, and M. Tegmark, *Optpde: Discovering novel integrable systems via ai-human collaboration*, arXiv preprint arXiv:2405.04484, (2024).

[116] S. Kantamneni, Z. Liu, and M. Tegmark, *Discovering candidates for integrable systems via backpropagation*, Physical Review E, 111 (2025), p. 025303. Introduces the OptPDE framework, which optimizes PDE coefficients via automatic differentiation to maximize the number of conserved quantities.

[117] A. A. Kaptanoglu, B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, *PySINDy: A comprehensive python package for robust sparse system identification*, Journal of Open Source Software, 7 (2022), p. 3994.

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

[119] M. E. Kavousanakis, G. Fabiani, A. Georgiou, C. Siettos, P. Kevrekidis, and I. G. Kevrekidis, *Solving symmetry-driven PDE dynamics with physics-informed neural networks.* Preprint, 2025.

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

[121] P. Kevrekidis, A. Bishop, and K. Rasmussen, *Parametric quantum resonances for bose–einstein condensates*, Journal of Low Temperature Physics, 120 (2000), pp. 205–212.

[122] P. G. Kevrekidis, *The Discrete Nonlinear Schrödinger Equation: Mathematical Analysis, Numerical Computations, and Physical Perspectives*, vol. 232 of Springer Tracts in Modern Physics, Springer, Berlin, Heidelberg, 2009.

[123] P. G. Kevrekidis, *Variational method for nonconservative field theories: Formulation and two $\mathcal{PT}$-symmetric case examples*, Phys. Rev. A, 89 (2014), p. 010102.

[124] S. Kim and I. Ahn, *Sparse identification of nonlinear gene regulatory networks from time series data*, Bioinformatics, 37 (2021), pp. 825–832.

[125] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114, (2014).

[126] G. Kissas, Y. Yang, E. Hwuang, W. R. T. Witschey, J. A. Detre, and P. Perdikaris, *Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri*, Scientific Reports, 10 (2020), pp. 1–11.

[127] Y. S. Kivshar and W. Królikowski, *Lagrangian formalism for dark solitons*, Optics Communications, 114 (1995), pp. 353–362.

[128] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, *Data-driven approximation of the koopman generator: Model reduction, system identification, and control*, Journal of Nonlinear Science, 28 (2018), pp. 985–1010.

[129] V. V. Konotop and L. Pitaevskii, *Landau dynamics of a grey soliton in a trapped condensate*, Phys. Rev. Lett., 93 (2004), p. 240403.

[130] N. Kovachki, Z. Li, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Neural operator: Learning maps between function spaces*, Journal of Machine Learning Research, 24 (2023), pp. 1–97.

[131] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Neural operator: Learning maps between function spaces*, Journal of Machine Learning Research, 24 (2023), pp. 1–97.

[132] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Neural operators: A review of algorithms and applications*, Acta Numerica, 32 (2023), pp. 877–1017.

[133] J. Krah, J. Sesterhenn, and J. Reiss, *Robust shifted pod for advection-dominated flows*, Journal of Computational Physics, 500 (2024), p. 112915.

[134] S. Krippendorf, D. Lüst, and M. Syvaeri, *Integrability ex machina*, Fortschritte der Physik, 69 (2021), p. 2100057.

[135] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, Society for Industrial and Applied Mathematics, 2016.

[136] N. Laskin and G. Zaslavsky, *Nonlinear fractional dynamics on a lattice with long range interactions*, Physica A: Statistical Mechanics and its Applications, 368 (2006), pp. 38–54.

[137] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

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

[139] K. Lee, K. Carlberg, and A. J. Chorin, *Coarse-grained model classification with diffusion maps and deep learning*, Physica D: Nonlinear Phenomena, 409 (2020), p. 132509.

[140] G. Lei, Z. Lei, and L. Shi, *Long-time integration of nonlinear wave equations with neural operators*, arXiv preprint arXiv:2410.15617, (2024).

[141] Z. Li and et al., *Learning nonlinear operators in latent spaces for real-time prediction of shallow water on spherical domains*, Nature Communications, (2024).

[142] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Fourier neural operator for parametric partial differential equations*, in Proceedings of the International Conference on Learning Representations (ICLR), 2021.

[143] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Dloss: Deep operator learning via spectral loss functions*, arXiv preprint arXiv:2010.08895, (2020).

[144] ——, *Fourier neural operator for parametric partial differential equations*, in International Conference on Learning Representations (ICLR), OpenReview.net, 2021. Poster.

[145] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Transformer neural networks for time-series learning and system identification of dynamical systems*, Nature Machine Intelligence, 5 (2023), pp. 53–66.

[146] S. Lin, L. Lu, and G. E. Karniadakis, *On universal approximation of nonlinear operators by deeponet*, Mathematics of Computation, 90 (2021), pp. 1743–1771.

[147] Z. Lin, J. Xu, P. Liu, and H. Zhang, *Data-driven prediction of rogue waves using long short-term memory neural networks*, Ocean Engineering, 229 (2021), p. 108971.

[148] Z. Liu, V. Madhavan, and M. Tegmark, *Machine learning conservation laws from differential equations*, Physical Review E, 106 (2022).

[149] Z. Liu, P. O. Sturm, S. Bharadwaj, S. J. Silva, and M. Tegmark, *Interpretable conservation laws as sparse invariants*, Physical Review E, 109 (2024), p. L023301.

[150] Z. Liu and M. Tegmark, *Machine learning conservation laws from trajectories*, Phys. Rev. Lett., 126 (2021), p. 180604.

[151] Z. Long, Y. Lu, X. Ma, and B. Dong, *Pde-net: Learning pdes from data*, in International conference on machine learning, PMLR, 2018, pp. 3208–3216.

[152] L. Lu, P. Jin, and G. E. Karniadakis, *Learning nonlinear operators via deeponet based on the universal approximation theorem of operators*, Nature Machine Intelligence, 3 (2021), pp. 218–229.

[153] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *Learning nonlinear operators via deeponet based on the universal approximation theorem of operators*, Nature Machine Intelligence, 3 (2021), pp. 218–229.

[154] ——, *Learning control of dynamical systems from data using operator neural networks*, Nature Communications, 13 (2022), p. 7462.

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

[156] P. Y. Lu, R. Dangovski, and M. Soljačić, *Discovering conservation laws using optimal transport and manifold learning*, Nature Communications, 14 (2023), p. 4744.

[157] B. Lusch, J. N. Kutz, and S. L. Brunton, *Deep learning for universal linear embeddings of nonlinear dynamics*, Nature Communications, 9 (2018), p. 4950.

[158] A. Majumdar et al., *Neural wave equation for irregularly sampled sequence data*, ICLR 2025, (2025).

[159] M. Malishava and S. Flach, *Lyapunov spectrum scaling for classical many-body dynamics close to integrability*, Phys. Rev. Lett., 128 (2022), p. 134102.

[160] B. A. Malomed, *Variational methods in nonlinear fiber optics and related fields*, in Progress in Optics, E. Wolf, ed., vol. 43, Elsevier, Amsterdam, 2002, pp. 71–193.

[161] S. V. Manakov, *Note on the integration of euler's equations of the dynamics of an n-dimensional rigid body*, Functional Analysis and Its Applications, 10 (1976), pp. 328–329. Original source of the trick for converting the Euler top Lax pair to yield integrability.

[162] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Inferring biological networks by sparse identification of nonlinear dynamics*, IEEE Transactions on Molecular, Biological and Multi-Scale Communications, 2 (2016), pp. 52–63.

[163] N. Margenberg, F. X. Kärtner, and M. Bause, *Optimal dirichlet boundary control by fourier neural operators applied to nonlinear optics*, arXiv preprint arXiv:2307.07292, (2023).

[164] J. E. Marsden, T. S. Ratiu, and A. Weinstein, *Semidirect products and reduction in mechanics*, Transactions of the American Mathematical Society, 281 (1984), pp. 147–177.

[165] R. Maulik, B. Lusch, P. Balaprakash, and S. L. Brunton, *Time-series learning of latent-space dynamics for reduced-order model closure*, Physical Review E, 103 (2021), p. 043307.

[166] L. D. McClenny and U. M. Braga-Neto, *Self-adaptive physics-informed neural networks*, Journal of Computational Physics, 474 (2023), p. 111722.

[167] Z. Meng, X. Li, and G. E. Karniadakis, *Ppinn: Parareal physics-informed neural network for time-dependent pdes*, Computer Methods in Applied Mechanics and Engineering, 370 (2020), p. 113250.

[168] D. A. Messenger and D. M. Bortz, *Weak sindy for partial differential equations*, Journal of Computational Physics, 443 (2021), p. 110525.

[169] ——, *Weak sindy: Galerkin-based data-driven model selection*, Multiscale Modeling & Simulation, 19 (2021), pp. 1474–1497.

[170] I. Mezić, *Spectral properties of dynamical systems, model reduction and decompositions*, Nonlinear Dynamics, 41 (2005), pp. 309–325.

[171] ——, *Analysis of fluid flows via spectral properties of the koopman operator*, Annual Review of Fluid Mechanics, 45 (2013), pp. 357–378.

[172] T. Mithun, Y. Kati, C. Danieli, and S. Flach, *Weakly nonergodic dynamics in the gross-pitaevskii lattice*, Phys. Rev. Lett., 120 (2018), p. 184101.

[173] P. J. Morrison, *Bracket formulation for irreversible classical fields*, Physics Letters A, 100 (1984), pp. 423–427.

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

[175] J. Moser, *Three integrable hamiltonian systems connected with isospectral deformations*, Advances in Mathematics, 16 (1975), pp. 197–220.

[176] F. Noé, S. Olsson, J. Köhler, and H. Wu, *Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning*, Science, 365 (2019), p. eaaw1147.

[177] M. Nonino, G. Scovazzi, and C. Farhat, *Reduced-order models for hyperbolic pdes via arbitrary lagrangian–eulerian (ale) mappings*, Computer Methods in Applied Mechanics and

Engineering, 421 (2024), p. 116813.
[178] H. C. Öttinger and M. Grmela, *Dynamics and thermodynamics of complex fluids. ii. illustrations of a general formalism*, Physical Review E, 56 (1997), pp. 6633–6655.
[179] J. Page and R. R. Kerswell, *Koopman analysis of burgers equation*, Phys. Rev. Fluids, 3 (2018), p. 071901.
[180] J. S. Park, *Wgfinns: Weak-form generic formalism informed neural networks*. KIAS Workshop presentation (slides), 2024. Accessed Nov 6, 2024.
[181] J. P. Parker and J. Page, *Koopman analysis of isolated fronts and solitons*, SIAM Journal on Applied Dynamical Systems, 19 (2020), pp. 2803–2828.
[182] J. P. Parker and C. Valva, *Koopman analysis of the periodic korteweg–de vries equation*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 33 (2023), p. 043102.
[183] R. Pascanu, T. Mikolov, and Y. Bengio, *On the difficulty of training recurrent neural networks*, Proceedings of the 30th International Conference on Machine Learning (ICML), 28 (2013), pp. 1310–1318.
[184] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *Pytorch: An imperative style, high-performance deep learning library*, 2019.
[185] B. Peherstorfer, *Breaking the kolmogorov barrier with nonlinear model reduction*, Nature Communications, 13 (2022), p. 7010.
[186] W. Peng, S. Qin, S. Yang, et al., *Fourier neural operator for real-time simulation of 3d dynamic urban microclimate*, arXiv preprint arXiv:2308.03985, (2023).
[187] L. Pitaevskii and S. Stringari, *Bose-Einstein condensation*, Oxford University Press, Oxford, 2003.
[188] B. Prinari, *Discrete solitons of the focusing Ablowitz–Ladik equation with nonzero boundary conditions via inverse scattering*, Journal of Mathematical Physics, 57 (2016), p. 083510.
[189] A. F. Psaros, X. Meng, J. Zou, and G. E. Karniadakis, *Operator learning for control of dynamical systems*, Proceedings of the Royal Society A, 479 (2023), p. 20230417.
[190] J. Pu and Y. Chen, *Lax pairs informed neural networks solving integrable systems*, Journal of Computational Physics, 510 (2024), p. 113090.
[191] V. Pérez-García, P. Torres, and G. Montesinos, *The method of moments for nonlinear schrödinger equations: Theory and applications*, SIAM Journal of Applied Mathematics, 67 (2007), pp. 990–1015.
[192] M. Raissi, *Deep hidden physics models: Deep learning of nonlinear partial differential equations*, Journal of Machine Learning Research, 19 (2018), pp. 1–24.
[193] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations*, 2017.
[194] ——, *Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations*, 2017.
[195] 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 (2019), pp. 686–707.
[196] M. Raissi, A. Yazdani, and G. E. Karniadakis, *Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations*, Science, 367 (2020), pp. 1026–1030.
[197] V. Ravoson, L. Gavrilov, and R. Caboz, *Separability and lax pairs for hénon–heiles system*, Journal of mathematical physics, 34 (1993), pp. 2385–2393.
[198] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann, *Shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena*, SIAM Journal on Scientific Computing, 40 (2018), pp. A1322–A1344.
[199] M. Remoissenet, *Waves Called Solitons*, Springer-Verlag, Berlin, 1999.
[200] P. I. Renn, C. Wang, S. Lale, et al., *Forecasting subcritical cylinder wakes with fourier*

neural operators, arXiv preprint arXiv:2301.08290, (2023).

[201] A. G. Reyman and M. A. Semenov-Tian-Shansky, *Group-theoretical methods in the theory of finite-dimensional integrable systems*, Encyclopaedia of Mathematical Sciences, 16 (1987), pp. 116–225. Introduces and discusses the concept of fake Lax pairs.

[202] P. Rivera-Casillas, S. Dutta, S. Cai, and et al., *A neural operator-based emulator for regional shallow water dynamics*, arXiv preprint arXiv:2502.14782, (2025).

[203] J. Rossi, S. Chandramouli, R. Carretero-González, and P. G. Kevrekidis, *On the temporal tweezing of cavity solitons*, Journal of Nonlinear Mathematical Physics, 31 (2024), p. 43.

[204] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust, *Reduction and reconstruction for self-similar dynamical systems*, Nonlinearity, 16 (2003), p. 1257.

[205] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, Journal of Fluid Mechanics, 641 (2009), pp. 115–127.

[206] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Data-driven discovery of partial differential equations*, Science Advances, 3 (2017), p. e1602614.

[207] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Data-driven discovery of partial differential equations*, Science Advances, 3 (2017), p. e1602614.

[208] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, Nature, 323 (1986), pp. 533–536.

[209] S. Saemundsson, A. Terenin, K. Hofmann, and M. Deisenroth, *Variational integrator networks for physically structured embeddings*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 3078–3087.

[210] O. Sallam and M. Fürth, *On the use of fourier features-physics informed neural networks (ff-pinn) for forward and inverse fluid mechanics problems*, Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment, 237 (2023), pp. 846–866.

[211] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia, *Hamiltonian graph networks with ODE integrators*, arXiv preprint arXiv:1909.12790, (2019).

[212] S. Saqlain, W. Zhu, E. G. Charalampidis, and P. G. Kevrekidis, *Discovering governing equations in discrete systems using pinns*, Communications in Nonlinear Science and Numerical Simulation, 126 (2023), p. 107498.

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

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

[215] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, Journal of Fluid Mechanics, 656 (2010), pp. 5–28.

[216] M. Schmidt and H. Lipson, *Distilling free-form natural laws from experimental data*, science, 324 (2009), pp. 81–85.

[217] M. L. Shahab, F. A. Suheri, R. Kusdiantara, and H. Susanto, *Physics-informed neural networks for high-dimensional solutions and snaking bifurcations in nonlinear lattices*, 2025.

[218] M. L. Shahab and H. Susanto, *Neural networks for bifurcation and linear stability analysis of steady states in partial differential equations*, 2025.

[219] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, *Convolutional lstm network: A machine learning approach for precipitation nowcasting*, Advances in Neural Information Processing Systems, 28 (2015).

[220] Y. Starosvetsky, K. Jayaprakash, M. A. Hasan, and A. Vakakis, *Dynamics and Acoustics of Ordered Granular Media*, World Scientific, Singapore, 2017.

[221] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press, 2018.

[222] C. Sulem and P.-L. Sulem, *The Nonlinear Schrödinger Equation: Self-Focusing and Wave Collapse*, Springer, 1999.

[223] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, *Modal analysis of fluid flows: An overview*, AIAA Journal, 55 (2017), pp. 4013–4041.

[224] N. Takeishi, Y. Kawahara, and T. Yairi, *Learning koopman invariant subspaces for dynamic mode decomposition*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[225] N. Thuerey, K. Weissenow, L. Prantl, and X. Hu, *Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows*, AIAA Journal, 58 (2020), pp. 25–36.

[226] M. Toda, *Theory of nonlinear lattices*, Springer-Verlag, Berlin, 1981.

[227] M. Tomasetto, J. P. Williams, F. Braghin, A. Manzoni, and J. N. Kutz, *Reduced order modeling with shallow recurrent decoder networks*, 2025.

[228] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, *Hamiltonian generative networks*, in International Conference on Learning Representations, 2020.

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

[230] A. van der Schaft and B. Maschke, *An l_2-norm preserving discretization of port-hamiltonian systems*, SIAM Journal on Control and Optimization, 39 (2000), pp. 1–27.

[231] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, in Advances in Neural Information Processing Systems, vol. 30, 2017.

[232] S. Wang, Z. Dou, T. Liu, and L. Lu, *Fundiff: Diffusion models over function spaces for physics-informed generative modeling*, arXiv preprint arXiv:2506.07902, (2025).

[233] S. Wang, Y. Teng, and P. Perdikaris, *Eigenvector physics-informed neural networks for schrödinger equations*, Computer Methods in Applied Mechanics and Engineering, 384 (2021), p. 113938.

[234] S. Wang, X. Yu, and P. Perdikaris, *Physics-informed deeponets for learning operators that are invariant to nonlinear coordinate transformations*, Journal of Computational Physics, 429 (2021), p. 110492.

[235] S. Wang, Z. Zhang, L. Lu, and G. E. Karniadakis, *Physics-informed machine learning for data-driven wave modeling: A survey*, Wave Motion, 113 (2022), p. 102632.

[236] Y. Wang, C.-Y. Lai, J. Gómez-Serrano, and T. Buckmaster, *Asymptotic self-similar blow-up profile for three-dimensional axisymmetric euler equations using neural networks*, Physical Review Letters, 130 (2023), p. 244002.

[237] S. J. Wetzel, R. G. Melko, J. Scott, M. Panju, and V. Ganesh, *Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks*, Phys. Rev. Res., 2 (2020), p. 033499.

[238] G. B. Whitham, *Linear and nonlinear waves*, Wiley-Interscience, (1974).

[239] G. B. Whitham, *Linear and nonlinear waves*, John Wiley & Sons, 2011.

[240] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, *A data-driven approximation of the koopman operator: Extending dynamic mode decomposition*, Journal of Nonlinear Science, 25 (2015), pp. 1307–1346.

[241] S. Xiong, Y. Tong, X. He, S. Yang, C. Yang, and B. Zhu, *Nonseparable symplectic neural networks*, in International Conference on Learning Representations, 2021.

[242] H. Xu, H. Liu, Y. He, G. Lin, and X. Wang, *Deep reinforcement learning for optimal control of nonlinear dynamical systems governed by partial differential equations*, Computer Methods in Applied Mechanics and Engineering, 398 (2022), p. 115248.

[243] L. Yang, X. Meng, and G. E. Karniadakis, *B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data*, Journal of Computational

Physics, 425 (2021), p. 109913.

[244] S. Yang, S. Chen, W. Zhu, and P. G. Kevrekidis, *Identification of moment equations via data-driven approaches in nonlinear schrödinger models*, Frontiers in Photonics, Volume 5 - 2024 (2024).

[245] S. Yang, S. Chen, W. Zhu, and P. G. Kevrekidis, *Sparse identification of variational approximations*. Preprint, in preparation, 2025.

[246] D. Yarotsky, *Optimal approximation of continuous functions by very deep relu networks*, in Conference on learning theory, PMLR, 2018, pp. 639–649.

[247] H. Yu, X. Tian, W. E, and Q. Li, *Onsagernet: Learning stable and interpretable dynamics using a generalized onsager principle*, Physical Review Fluids, 6 (2021), p. 114402.

[248] V. E. Zakharov and L. D. Faddeev, *Korteweg–de vries equation: A completely integrable hamiltonian system*, Funktsional'nyi Analiz i ego Prilozheniya, 5 (1971), pp. 18–27.

[249] V. E. Zakharov, V. S. L'vov, and G. Falkovich, *Kolmogorov spectra of turbulence i: Wave turbulence*, Springer Series in Nonlinear Dynamics, (1992).

[250] M. Zhang, Q. Meng, D. Zhang, Y. Wang, Z. Ma, L. Chen, and T. Liu, *Complex-valued neural operator assisted soliton identification*, arXiv preprint arXiv:2305.18209, (2023).

[251] Z. Zhang, Y. Shin, and G. E. Karniadakis, *Gfinns: Generic formalism informed neural networks for deterministic and stochastic dynamical systems*, Philosophical Transactions of the Royal Society A, 380 (2022), p. 20210207.

[252] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, *A unified framework for sparse relaxed regularized regression: Sr3*, IEEE Access, 7 (2019), pp. 1404–1423.

[253] Y. D. Zhong, B. Dey, and A. Chakraborty, *Symplectic ode-net: Learning hamiltonian dynamics with control*, in International Conference on Learning Representations, 2020.

[254] e. a. Zhou, *Pi-fusion: Physics-informed diffusion model for learning fluid dynamics*, arXiv preprint arXiv:2406.03711, (2024).

[255] J. Zhou, G. Cui, Z. Hu, C. Zhang, Z. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, *Graph neural networks: A review of methods and applications*, AI Open, 1 (2020), pp. 57–81.

[256] D. Zhu, W. Chen, and Z. Lu, *Deep learning for design and retrieval of complex nanophotonic structures*, ACS Photonics, 6 (2019), pp. 2041–2049.

[257] 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 (2022), p. 133264.

[258] W. Zhu, H.-K. Zhang, and P. Kevrekidis, *Machine learning of independent conservation laws through neural deflation*, Physical Review E, 108 (2023), p. L022301.

[259] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, *Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data*, Journal of Computational Physics, 394 (2019), pp. 56–81.

[260] C. Zou, K. Azizzadenesheli, Z. E. Ross, and R. W. Clayton, *Deep neural helmholtz operators for 3-d elastic wave propagation and inversion*, Geophysical Journal International, 239 (2024), pp. 1469–1484.