


Learning of discrete models of variational PDEs from data

Christian Offen ^{1, a)} and Sina Ober-Blöbaum¹

Paderborn University, Department of Mathematics, Warburger Str. 100, 33098 Paderborn, Germany

(*Electronic mail: christian.offen@uni-paderborn.de)

(Dated: 14 November 2023)

We show how to learn discrete field theories from observational data of fields on a space-time lattice. For this, we train a neural network model of a discrete Lagrangian density such that the discrete Euler–Lagrange equations are consistent with the given training data. We, thus, obtain a structure-preserving machine learning architecture. Lagrangian densities are not uniquely defined by the solutions of a field theory. We introduce a technique to derive regularisers for the training process which optimise numerical regularity of the discrete field theory. Minimisation of the regularisers guarantees that close to the training data the discrete field theory behaves robust and efficient when used in numerical simulations. Further, we show how to identify structurally simple solutions of the underlying continuous field theory such as travelling waves. This is possible even when travelling waves are not present in the training data. This is compared to data-driven model order reduction based approaches, which struggle to identify suitable latent spaces containing structurally simple solutions when these are not present in the training data. Ideas are demonstrated on examples based on the wave equation and the Schrödinger equation.

Citation of published version¹: Chaos 34 (1), 013104 (2024). <https://doi.org/10.1063/5.0172287>.

To learn practical models of dynamical systems from data, prior geometric knowledge and aspects of numerical integration theory can be exploited in the design of the data-driven architectures. The article shows how to utilise numerical analysis informed neural networks and discrete field theories to model dynamical systems governed by first principles.

I. INTRODUCTION

The identification of models of dynamical systems from observational data is an important task in machine learning: the identified models can be used to interpolate data points, predict motions, develop control strategies, or analyse long-term behaviour^{2,3}. Additionally, physical properties of the system can be discovered such as symmetries, conservation laws^{4,5}, stress tensors⁶, for instance. Moreover, data-driven models can be used as surrogates for analytic models when computations with explicit analytical models are prohibitively expensive, such as in molecular dynamics⁷.

To improve qualitative aspects of data-driven models, prior knowledge about physical aspects of the dynamics, such as symmetries, conservation laws, Hamiltonian or variational structure can be taken into account. The approach is referred to as *physics informed machine learning*. (Notice, however, that the term *physics informed neural networks* has also been used to describe neural network approximations of solutions to partial differential equations that represent physical laws. See Karniadakis et al⁸ for a review article.)

Variational principles are at the heart of physical theories and constitute first principles from which dynamical models

can be derived. In such dynamical systems motions or solutions to the field equations constitute stationary points of an action functional. For instance, classical mechanical systems, electromagnetic field theories, fluid dynamics, as well as quantum dynamical systems are governed by variational principles. Presence of variational structure in a dynamical system is related to many profound laws in physics such as the correspondence of symmetries and conserved quantities (Noether's theorem)^{9,10}.

To embed variational structure into machine learned models of dynamical systems, Greydanus et al propose to learn a model of a variational principle from data, called Lagrangian neural networks (LNNs)¹¹. This needs to be contrasted to learning of a model of a flow map or a vector field of a dynamical system without enforcing physical properties of the system¹². Similar ideas and extensions to LNN have been developed, such as combinations with model order reduction techniques¹³ or the inclusion of external forces¹⁴. The data-driven learning of variational principles needs to be contrasted to approaches that identify the analytic form of a (partial) differential equation via sparse regression^{15–17}. While the former aims at learning a model that can be used in numerical predictions, the latter focuses on interpretable system identification and requires a dictionary of candidate terms that can make up the differential equation.

Other approaches focus on exploiting Hamiltonian structure for learning models of dynamical systems. While Hamiltonian and variational structure are equivalent from a theoretical perspective (under non-degeneracy conditions), data-driven architectures for learning Hamiltonian systems, such as Hamiltonian Neural Networks¹⁸ or Symplectic Neural Networks¹⁹, differ from those architectures that learn models of action functionals as they require additional observation of momentum data rather than just position data (and possibly derivatives), or at least some prior knowledge of the symplectic phase space structure. However, much of the symplectic

^{a)}<https://www.uni-paderborn.de/en/person/85279>

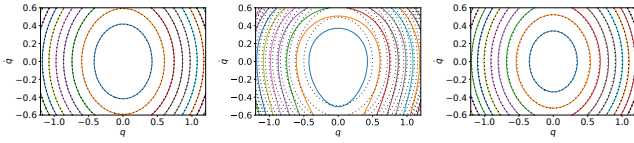


FIG. 1. Left: Motions of a data-driven model (solid lines) for the mathematical pendulum match motions of an analytic model (dotted). Centre: Numerically computed motions of the data-driven model are very inaccurate. Right: Applying the same numerical integrator to the analytic model yield much more accurate results. (Figure taken from²³ (Fig. 6). See²³ for details.)

structure can be learned from data as well^{20,21}.

Instead of learning a model of a continuous action functional to describe a dynamical system with variational structure, Qin²² proposes to learn a discrete variational principle instead. Discrete field theories can be trained on discrete observations of fields over a space-time lattice, while the training of continuous theories such as LNN¹¹ requires observations of velocities and acceleration data. Moreover, solutions of discrete field theories can be computed without further discretisation by numerical integrators such that discretisation errors are avoided.

However, the authors have demonstrated that numerical discretisation errors of learned continuous variational models of dynamical systems can be surprisingly big²³: the plot to the left of Fig. 1 shows that the exact trajectories of the learned dynamical system of the mathematical pendulum coincide with the exact trajectories of an analytic reference. However, an application of a standard variational numerical integrator (variational midpoint rule²⁴) to the learned model performs poorly (centre) while an application of the same integrator with the same step-size shows only small numerical errors when applied to the analytic reference (right). The reason is that variational principles are not uniquely determined by a system's motions. In machine learning, care is required to identify a model of a variational principle that is not only consistent with the motion data, but also suitable for numerical computations. This will be a central aspect of this article.

While learning a discrete field theory avoids discretisation errors in numerical computation, additional regularisation is essential to avoid learning degenerate principles for which discrete field equations become unsolvable or badly conditioned. While the authors have discussed a geometric regularisation strategy applying to models based on Gaussian Processes or kernel methods²³, different regularisation strategies are required when artificial neural networks are used. For variational dynamical systems, for which the underlying continuous system is an ordinary differential equation, we show first approaches in Lishkova et al.⁵. The present article extends the authors' conference contribution²⁵, in which machine learning architectures for discrete field theories are developed whose underlying dynamical system is a partial differential equation.

Machine learning of field theories governing infinite-dimensional dynamical systems needs to be contrasted with approaches that learn differential operators or a response func-

tion and do not exploit geometric structure or a discrete framework, such as DeepONet²⁶ or pde-net^{27,28}. Approaches that exploit Hamiltonian or Poisson structure can be found in some recent articles^{29–31}.

Further, identification of discrete field theories for partial differential equations needs to be contrasted with approaches based on model order reduction (MOR). Data-driven MOR-based techniques learn a reduction map of a spatially discretised system. Then, a full order model (FOM) is projected to the latent space to obtain a reduced order model (ROM). If no FOM is available, the learned reduction map is used to project dynamical data to the latent space. Subsequently, the ROM is learned, i.e. a data-driven model is fitted to describe the temporal dynamics of the system on the latent space. Several approaches have been developed to preserve geometric structure in this context^{13,32–38}.

In contrast to the data-driven MOR-based techniques, we learn the discrete field theory directly on a space-time lattice without a reduction step. Indeed, discrete field theories are based on stencils coupling spacial and temporal variables only locally. We exploit this locality to obtain a reduction of computational complexity for training, which would otherwise require MOR. (Notice that computing solutions of the trained model could still be done by MOR-based techniques, when locality of the stencils cannot be exploited due to boundary conditions or numerical stability issues.)

We demonstrate that avoiding projection of the dynamics to a latent space can be advantageous if, for instance, highly symmetric solutions of the model, such as travelling waves, need to be identified in the data-driven model. We develop a method to detect and preserve solutions of the discrete field theory that correspond to highly symmetric solutions in the underlying continuous dynamical system. As the mesh of a discrete field theory is typically not compatible with the symmetries of the continuous system, we use a suitable symmetric ansatz of the discrete action functional based on Palais' principle of symmetric criticality^{39,40} to design machine learning architectures for discrete field theories with travelling waves.

The main novelties of the article are

- the systematic design of numerical analysis informed regularisation strategies for machine learning models of discrete field theories,
- the preservation and detection of highly symmetric solutions in discrete field theories using data-driven architectures based on Palais' principle of symmetric criticality and comparison to data-driven model order reduction based techniques,
- extension of the approaches first introduced in our conference contribution²⁵ to different stencils and to degenerate Lagrangians that are linear in velocities.

The remainder of the article is structured as follows: Section II reviews the concept of variational principles in the continuous and discrete setting. The theory is illustrated on our main examples, the wave equation and the Schrödinger equation. Section III provides theoretical considerations motivating the regularisers that we will employ when training our

neural network models. Section IV introduces the machine learning set-up to learn a discrete field theory from data. Section V provides theoretical background on the notion of travelling waves and highly symmetric solutions in discrete field theories and describes their variational structure. Finally, Section VI provides numerical experiments based on the wave and Schrödinger equation. The article concludes with a summary and future work section (Section VII).

II. CONTINUOUS AND DISCRETE VARIATIONAL PRINCIPLES AND FIELD THEORIES

We recall briefly the concept of variational principles and their discretisations. We illustrate the theory on the wave equation and the Schrödinger equation, on which we will test our data-driven framework. For more detailed expositions, we refer to the literature for an introduction to variational calculus^{10,41,42} and discrete mechanics²⁴.

A. Continuous variational principles

Variational principles play a fundamental role in the derivation and analysis of field theories in physics⁴²: motions $u: X \rightarrow \mathbb{R}^d$ are described as stationary points of an action functional

$$S(u) = \int_X L(x, u(x), u_{x_0}(x), \dots, u_{x_n}(x)) dx. \quad (1)$$

Here the domain of definition of u, X , refers to a suitable manifold (with boundary) such as $X = [x_0^0, x_0^N] \times \dots \times [x_n^0, x_n^N]$ with coordinates $x = (x_0, x_1, \dots, x_n)$. Often, X represents a space-time domain. The expression u_{x_k} denotes the partial derivative $\frac{\partial u}{\partial x_k}$. More precisely, u constitutes a solution of the field theory, if the action functional S is stationary at u with respect to variations $\delta u: X \rightarrow \mathbb{R}^d$ vanishing at the boundary of X . The function L is called *Lagrangian density*. As it depends on the 1-jet of u only and not on higher derivatives, a field theory described by an action functional of the form (1) is called *1st order field theory*. Stationary points of the action functional are solutions to the Euler–Lagrange equation

$$0 = \text{EL}(L) = \frac{\partial L}{\partial u} - \sum_{j=0}^n \frac{d}{dx_j} \frac{\partial L}{\partial u_{x_j}}, \quad (2)$$

which constitutes a partial differential equation of 2nd order, unless L is degenerate.

Example 1 (Linear motions). Denote $x_0 = t$. For any $\theta: \mathbb{R}^d \rightarrow \mathbb{R}$ with nowhere degenerate Hesse matrix $\text{Hess}\theta$, the field theory described by the Lagrangian density $L(u, u_t) = \theta(u_t)$ yield the Euler–Lagrange equations

$$0 = \text{EL}(L) = \text{Hess}\theta(u_t)u_{tt}$$

that is solved by affine linear motions $u_{tt} = 0$ in \mathbb{R}^d .

Example 1 illustrates that Lagrangian densities are not uniquely determined by a system's motions. The level of ambiguity is higher than the ambiguity known as *gauge freedom*, which refers to transformations of the Lagrangians that leave the term $\text{EL}(L)$ invariant (Remark 9 - see Appendix A). This observation will become important later when we will learn field theories from data. Ambiguity of Lagrangians and the role of symmetry have been discussed in theoretical physics^{43–45}.

We proceed with introducing our main examples of field theories.

Example 2 (Wave equation). For a potential $V: \mathbb{R} \rightarrow \mathbb{R}$ the Euler–Lagrange equations to the Lagrangian

$$L(u, u_t, u_x) = \frac{1}{2}(u_t^2 - u_x^2) - V(u) \quad (3)$$

yield the non-linear wave equation

$$u_{tt}(t, x) - u_{xx}(t, x) + \nabla V(u(t, x)) = 0. \quad (4)$$

Here ∇V denotes the gradient of V and $u_{tt} = \frac{\partial^2 u}{\partial t^2}$, $u_{xx} = \frac{\partial^2 u}{\partial x^2}$ partial derivatives.

Example 3 (Degenerate Lagrangian). Let $J = \begin{pmatrix} 0 & -I_n \\ I_n & 0 \end{pmatrix}$, where $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix. For a function $H: \mathbb{R}^{2n} \rightarrow \mathbb{R}$ and the Lagrangian

$$L(u, u_t) = \frac{1}{2}(J^{-1}u)^\top u_t - H(u) \quad (5)$$

the Euler–Lagrange equations are

$$Ju_t = \nabla H(u). \quad (6)$$

Equation (6) are Hamilton's equations on \mathbb{R}^{2n} to the Hamiltonian H , with symplectic structure on \mathbb{R}^{2n} defined by J .

The Lagrangian (5) is degenerate, since the matrix $\frac{\partial^2 L}{\partial u_t^2} = 0$ is not regular. Indeed, the Euler–Lagrange equations (6) are first order differential equations rather than of second order.

Using a natural extension of the described variational principle (1) to complex valued fields, we present a variational principle for Schrödinger equation⁴⁶.

Example 4 (Schrödinger equation). For a potential $V: \mathbb{R} \rightarrow \mathbb{R}$ the Euler–Lagrange equations to the Lagrangian

$$L(\Psi, \Psi_t, \Psi_x) = -\frac{\hbar}{2}(\overline{\Psi}\Psi_t - \Psi\overline{\Psi}_t) - \overline{\Psi}_x\Psi_x - V(\overline{\Psi}\Psi) \quad (7)$$

yield the non-linear Schrödinger equation

$$\hbar i\Psi_t = (-\Delta + V'(|\Psi|^2))\Psi. \quad (8)$$

Here Ψ is a complex valued function defined on a space-time domain parametrised by the variables (t, x) , $\overline{\Psi}$ its complex conjugation, \hbar is the Plank constant, i the imaginary unit, and $\Delta = \frac{\partial^2}{\partial x^2}$ denotes the spatial Laplacian.

Alternatively, with $\Psi = \phi + ip$, $u = (\phi, p)^\top$, J as in Example 3, the Schrödinger equation can be obtained from the a real variational principle with Lagrangian

$$L(u, u_t, u_x) = \hbar(J^{-1}u)^\top u_t - \|u_x\|^2 - V(\|u\|^2) \quad (9)$$

which yields

$$\hbar J u_t = (-\Delta + V'(\|u\|^2))u, \quad (10)$$

which is equivalent to (8).

B. Discrete variational principles

1. Set-Up

Let us introduce discrete actions based on various types of discrete Lagrangian densities. These can be interpreted as approximations to exact discrete Lagrangians (see Appendix B). While we provide a rather general setup and expression of discrete Lagrangian densities in the following, Example 6 introduces the types of densities used in our numerical experiments. For a detailed introduction we refer to the article⁴⁷ by Marsden, Pekarsky, Shkoller, West.

First we introduce some notation for meshes. Consider

- the $n + 1$ -dimensional cube $X = [x_0^0, x_0^{N_0}] \times \dots \times [x_n^0, x_n^{N_n}] \subset \mathbb{R}^{n+1}$,
- the mesh $X_d = \{x_0^0, \dots, x_0^{N_0}\} \times \dots \times \{x_n^0, \dots, x_n^{N_n}\} \subset X$,
- the grid's interior points $\hat{X}_d = \{x_0^1, \dots, x_0^{N_0-1}\} \times \dots \times \{x_n^1, \dots, x_n^{N_n-1}\} \subset X_d$.

Assume for simplicity that the grid X_d is uniform with spacing $\Delta x_r = (x_r^{N_r} - x_r^0)/N_r$ ($r = 0, \dots, n$). Let $\Delta x = \Delta x_0 \cdot \dots \cdot \Delta x_n$ denote the discrete volume element. Consider

- the index set $\mathcal{J} = \{0, \dots, N_0\} \times \dots \times \{0, \dots, N_n\}$,
- the multi-index $l = (l_0, \dots, l_n) \in \mathcal{J}$,
- the small cube $X^l = [x_0^{l_0}, x_0^{l_0+1}] \times \dots \times [x_n^{l_n}, x_n^{l_n+1}]$ to $l \in \mathcal{J}$ with $l_r < N_r$ for all $0 \leq r \leq n$,
- and the set of vertices $X_{\text{vtx}}^l = \{x_0^{l_0}, x_0^{l_0+1}\} \times \dots \times \{x_n^{l_n}, x_n^{l_n+1}\}$ of a small cube X^l . Each X_{vtx}^l has 2^{n+1} elements.

A discrete action functional S_d now assigns real values to functions $U: X_d \rightarrow \mathbb{R}$ defined on the grid X_d : a discrete action functional is given as $S_d: (\mathbb{R}^d)^{X_d} \rightarrow \mathbb{R}$ with

$$S_d(U) = \sum_{l \in \mathcal{J}, l_r < N_r} L_d(X_{\text{vtx}}^l, U(X_{\text{vtx}}^l)) \Delta x, \quad (11)$$

where $L_d: (\mathbb{R}^{n+1})^{2^{n+1}} \times (\mathbb{R}^d)^{2^{n+1}} \rightarrow \mathbb{R}$ can be evaluated from the values of U on each vertex set X_{vtx}^l . L_d is called discrete Lagrangian density. In many cases L_d is *autonomous*, i.e. does not explicitly depend on X_{vtx}^l and is, thus, of the form

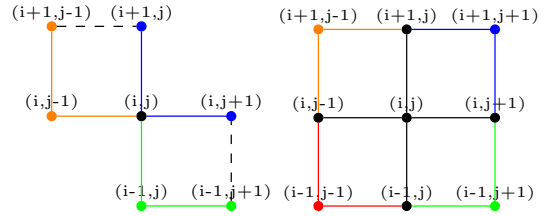


FIG. 2. Visualisation of the 7 and 9 point stencil of Example 6. Colours indicate that points are part of the same summand in (13) or (14). Black vertices are present in several summands.

$L_d(U(X_{\text{vtx}}^l))$. The discrete action S_d of (11) is a discretised version of (1).

The discrete Euler–Lagrange equations are obtained by discrete variations fixing the boundary as $\nabla_{U(\hat{X}_d)} S_d(U) = 0$. This gives for any interior vortex $v \in \hat{X}_d$ the discrete Euler–Lagrange equation

$$\text{DEL}(L_d)_v := \nabla_{U(v)} \sum_{l \in \mathcal{J}_v} L_d(X_{\text{vtx}}^l, U(X_{\text{vtx}}^l)) = 0, \quad (12)$$

where \mathcal{J}_v contains all indices l of cubes that contain v , i.e. $v \in X^l$. The derivative $\nabla_{U(v)}$ describes the derivative with respect to the u -variable at the vortex v .

2. Examples

Example 5 (Linear motions). Denote $x_0 = t$. Let $A \in \mathbb{R}^{d \times d}$ be a symmetric, non-degenerate matrix. Consider a uniform mesh $X_d = \{t^0, \dots, t^N\}$ on $[t^0, t^N] \subset \mathbb{R}$. We identify $(\mathbb{R}^d)^{X_d}$ with the space $(\mathbb{R}^d)^{N+1}$. In analogy to Example 1, for any $\theta: \mathbb{R}^d \rightarrow \mathbb{R}$ with nowhere degenerate Hessian the discrete Lagrangian

$$L_d(u^i, u^{i+1}) = \theta(u^{i+1} - u^i)$$

yields discrete Euler–Lagrange equations

$$0 = \text{DEL}(L_d)_{u^i} = \nabla \theta(u^i - u^{i-1}) - \nabla \theta(u^{i+1} - u^i)$$

to which $u^{i+1} = u^i + (u^i - u^{i-1})$ is the locally unique solution.

Example 5 demonstrates that discrete Lagrangian densities are not uniquely determined by a system's motion. Depending on the choice of L_d , the discrete Euler–Lagrange equations can be easier or more difficult to solve numerically. This will need to be considered when learning a model of L_d from data. The following example introduces the two main stencils used in our numerical experiments.

Example 6 (3 and 4 point discrete Lagrangian). We denote $x_0 = t$, $x_1 = x$ and identify $\mathbb{R}^{X_d} \cong \mathbb{R}^{(N_0+1) \times (N_1+1)}$ by the space of $(N_0+1) \times (N_1+1)$ matrices with elements $U = (u_j^i)_{0 \leq i \leq N_0, 0 \leq j \leq N_1}$. For a discrete Lagrangian of the form $L_d(u_j^i, u_j^{i+1}, u_{j+1}^i)$ (3 input arguments, autonomous) the discrete Euler–Lagrange

equations (12) yield the 7 point stencil

$$\begin{aligned} \text{DEL}(L_d)_j^i = \frac{\partial}{\partial u_j^i} & \left(L_d(u_j^i, u_j^{i+1}, u_{j+1}^i) \right. \\ & + L_d(u_j^{i-1}, u_j^i, u_{j+1}^{i-1}) \\ & \left. + L_d(u_{j-1}^i, u_{j-1}^{i+1}, u_j^i) \right) = 0, \end{aligned} \quad (13)$$

denoting $\text{DEL}(L_d)_j^i = \text{DEL}(L_d)_{u_j^i}$. If L_d is of the form $L_d(u_j^i, u_j^{i+1}, u_{j+1}^i, u_{j+1}^{i+1})$ (4 input arguments, autonomous), then the discrete Euler–Lagrange equations yield the 9 point stencil

$$\begin{aligned} \text{DEL}(L_d)_j^i = \frac{\partial}{\partial u_j^i} & \left(L_d(u_j^i, u_j^{i+1}, u_{j+1}^i, u_{j+1}^{i+1}) \right. \\ & + L_d(u_j^{i-1}, u_j^i, u_{j+1}^{i-1}, u_{j+1}^i) \\ & + L_d(u_{j-1}^i, u_{j-1}^{i+1}, u_j^i, u_j^{i+1}) \\ & \left. + L_d(u_{j-1}^{i-1}, u_{j-1}^i, u_j^{i-1}, u_j^i) \right) = 0. \end{aligned} \quad (14)$$

The stencils are visualised in Fig. 2

We proceed by introducing the two main discrete field theories used in the numerical experiments.

Example 7 (Discrete 2d wave equation). We denote $x_0 = t$, $x_1 = x$ and identify $\mathbb{R}^{x_d} \cong \mathbb{R}^{(N_0+1) \times (N_1+1)}$ by the space of $(N_0 + 1) \times (N_1 + 1)$ matrices with elements $U = (u_j^i)_{0 \leq i \leq N_0, 0 \leq j \leq N_1}$. Let L be the Lagrangian of Example 2. Consider the autonomous discrete Lagrangian density

$$\begin{aligned} L_d(u_j^i, u_j^{i+1}, u_{j+1}^i) &= L \left(u_j^i, \frac{u_j^{i+1} - u_j^i}{\Delta t}, \frac{u_{j+1}^i - u_j^i}{\Delta x} \right) \\ &= \frac{1}{2} \left(\frac{u_j^{i+1} - u_j^i}{\Delta t} \right)^2 - \frac{1}{2} \left(\frac{u_{j+1}^i - u_j^i}{\Delta x} \right)^2 - V(u_j^i). \end{aligned} \quad (15)$$

The discrete Euler–Lagrange equation (13) is given as

$$\frac{u_j^{i-1} - 2u_j^i + u_j^{i+1}}{\Delta t^2} - \frac{u_{j-1}^i - 2u_j^i + u_{j+1}^i}{\Delta x^2} + \nabla V(u_j^i) = 0. \quad (16)$$

It can be interpreted as a 2nd order discretisation of Example 2 by finite differences. While for general L_d the discrete Euler–Lagrange equation (13) constitutes a 7-point stencil, on L_d of (15) it simplifies to a 5-point stencil. The stencil can be used conveniently to compute forward propagations of solutions as u_j^{i+1} can be computed in terms of u_j^i , u_j^{i-1} , u_{j+1}^i , which all belong to previous time-steps.

The following example uses a straight forward extension of the discrete variational principle to complex valued fields.

Example 8 (Discrete Schrödinger equation). Let L be the Lagrangian (7) of Example 4. Consider

$$L_d(\Psi_{j-1}^{i-1}, \Psi_{j-1}^i, \Psi_j^{i-1}, \Psi_j^i) = L(m_j^i \Psi, \Delta t_j^i \Psi, \Delta x_j^i \Psi) \quad (17)$$

with

$$m_j^i \Psi = \frac{\Psi_{j-1}^{i-1} + \Psi_{j-1}^i + \Psi_j^{i-1} + \Psi_j^i}{4} \quad (18)$$

$$\Delta t_j^i \Psi = \frac{(\Psi_{j-1}^i - \Psi_{j-1}^{i-1}) + (\Psi_j^i - \Psi_j^{i-1})}{2\Delta t} \quad (19)$$

$$\Delta x_j^i \Psi = \frac{(\Psi_j^{i-1} - \Psi_{j-1}^{i-1}) + (\Psi_j^i - \Psi_{j-1}^i)}{2\Delta x} \quad (20)$$

that approximate the field Ψ and its derivative in t and x in the centre of a square $X^{(i,j)}$ of the grid. Further, consider the following approximations of derivatives at the node (i, j)

$$\begin{aligned} D_{t_j}^i \Psi &= \frac{(\Psi_{j-1}^{i+1} - \Psi_{j-1}^{i-1}) + 2(\Psi_j^{i+1} - \Psi_j^{i-1}) + (\Psi_{j+1}^{i+1} - \Psi_{j+1}^{i-1})}{8\Delta t} \\ D_{x_j}^{2i} \Psi &= \frac{1}{4\Delta x^2} \left((\Psi_{j-1}^{i-1} - 2\Psi_j^{i-1} + \Psi_{j+1}^{i-1}) \right. \\ &\quad + (\Psi_{j-1}^i - 2\Psi_j^i + \Psi_{j+1}^i) \\ &\quad \left. + (\Psi_{j-1}^{i+1} - 2\Psi_j^{i+1} + \Psi_{j+1}^{i+1}) \right). \end{aligned}$$

As discrete Euler–Lagrange equations (14) we obtain

$$\begin{aligned} i\hbar D_{t_j}^i \Psi &= -D_{x_j}^{2i} \Psi \\ &+ \frac{1}{4} \left(V'(m_{j-1}^{i-1} \Psi) m_{j-1}^{i-1} \Psi + V'(m_{j-1}^i \Psi) m_{j-1}^i \Psi \right. \\ &\quad \left. + V'(m_{j-1}^{i+1} \Psi) m_{j-1}^{i+1} \Psi + V'(m_j^i \Psi) m_j^i \Psi \right), \end{aligned} \quad (21)$$

which is a 9 point stencil.

3. Forward propagation

In Example 6 assume that boundary conditions $(u_0^i)_{0 \leq i \leq N_0}$, $(u_{N_1}^i)_{0 \leq i \leq N_0}$, initial data $(u_j^0)_{0 \leq j \leq N_1}$, and initial velocities $(\dot{u}_j^0)_{0 \leq j \leq N_1}$ are given. Let

$$L_{\Delta x}^{\Delta t}(U^i, U^{i+1}) = \sum_{j=0}^{N_1-1} L_d(u_j^i, u_j^{i+1}, u_{j+1}^i) \quad (22)$$

or

$$L_{\Delta x}^{\Delta t}(U^i, U^{i+1}) = \sum_{j=0}^{N_1-1} L_d(u_j^i, u_j^{i+1}, u_{j+1}^i, u_{j+1}^{i+1}), \quad (23)$$

respectively. Here $U^i = (u_j^i)_{0 \leq j \leq N_1}$ refers to values at interior spatial grid points. Let $L_{\Delta x}^{\Delta t}(U, \dot{U}) = L_{\Delta x}^{\Delta t}(U - \Delta t/2 \dot{U}, U + \Delta t/2 \dot{U})$. Solutions to the discrete field theory can be computed as follows: U^1 is computed as the solution to

$$\nabla_{\dot{U}} L_{\Delta x}^{\Delta t}(U^0, \dot{U}^0) = -\nabla_U L_{\Delta x}^{\Delta t}(U^0, U^1). \quad (24)$$

Subsequent U^i are computed as solutions to the DEL for $L_{\Delta x}^{\Delta t}$

$$\nabla_{U^i} \left(L_{\Delta x}^{\Delta t}(U^{i-1}, U^i) + L_{\Delta x}^{\Delta t}(U^i, U^{i+1}) \right) = 0. \quad (25)$$

This follows classical variational integration theory²⁴. We refer to the literature on low-rank approximations and integration schemes^{48–51} in case fine meshes or higher spacial dimensions are considered and the systems (24), (25) become prohibitively high-dimensional.

In the special case that $L_{\Delta x}(U, \dot{U})$ is linear in the velocities \dot{U} , (24) is independent of \dot{U}^0 such that no initial velocity profile \dot{U}^0 is needed to obtain U^1 via (24). In particular, if $L_{\Delta x}(U, \dot{U})$ is of the form (5), then the computation is numerically stable⁵². Indeed, independence of \dot{U}^0 of the numerical scheme is expected since the EL equations (6) are 1st order ordinary differential equations. While numerical stability of the scheme is guaranteed for $L_{\Delta x}$ of the form (5), for more general Lagrangians linear in velocities other techniques are required for numerically stable computations^{53–55}.

Remark 1. Depending on boundary conditions, the solution procedure of Section II B 3 can often be drastically simplified by using the local stencils (13), (14) to propagate solutions forward in time, as we made explicit in Example 7. Use of the local stencils avoids solving the potentially high dimensional system (25).

III. PREPARATION OF REGULARISATION STRATEGY

In the following, we will train a neural network model of a discrete Lagrangian density L_d such that L_d defines a discrete field theory that is consistent with observations. Discrete Lagrangian densities are not uniquely determined by the system's motions as illustrated in Example 5. (Also see Remark 10.) The secondary goal is to identify a model of L_d that is not only consistent with the true dynamics but is also computationally efficient when numerical methods are used to solve the discrete Euler–Lagrange equations.

To prepare our design of regularisers to be employed in the training phase, we analyse the convergence properties of Newton iterations when used to compute solutions to discrete Euler–Lagrange equations for the 3 and 4 point discrete Lagrangians of Example 6. The following Proposition refers to the DEL for 3-point Lagrangians (13).

Proposition 1. Let $u_j^i, u_j^{i+1}, u_{j+1}^i, u_{j+1}^{i+1}, u_{j-1}^i, u_{j-1}^{i+1}$ such that (13) holds. Let $\mathcal{O} \subset \mathbb{R}^d$ be a convex, neighbourhood of $u^* = u_j^{i+1}$, $\|\cdot\|$ a norm of \mathbb{R}^d inducing an operator norm on $\mathbb{R}^{d \times d}$. Define $p(u) := \frac{\partial^2 L_d}{\partial u_j^i \partial u} (u_j^i, u, u_{j+1}^i)$ and let θ and $\bar{\theta}$ be Lipschitz constants on \mathcal{O} for p and for $\text{inv} \circ p$, respectively, where inv denotes matrix inversion. Let

$$\rho^* := \|\text{inv}(p(u^*))\| = \left\| \left(\frac{\partial^2 L_d}{\partial u_j^i \partial u^*} (u_j^i, u^*, u_{j+1}^i) \right)^{-1} \right\| \quad (26)$$

and let $f(u^{(n)})$ denote the left hand side of (13) with u_j^{i+1} replaced by $u^{(n)}$. If $\|u^{(0)} - u^*\| \leq \min\left(\frac{\rho^*}{\theta}, \frac{1}{2\theta\rho^*}\right)$ for $u^{(0)} \in \mathcal{O}$, then the Newton iterations $u^{(n+1)} := u^{(n)} - \text{inv}(p(u^{(n)}))f(u^{(n)})$ converge quadratically against u^* , i.e.

$$\|u^{(n+1)} - u^*\| \leq \rho^* \theta \|u^{(n)} - u^*\|^2. \quad (27)$$

A proof can be found in Appendix C.

Remark 2. Proposition 1 is a restatement of Proposition 1 of our conference contribution²⁵, where it is stated without proof.

Proposition 1 clarifies numerical convergence properties when computing solutions on a grid with the stencil visualised to the left of Fig. 2. It applies when values u_0^i at the left boundary and initial values u_j^i, u_j^2 are known for all indices i, j such that remaining values can be computed subsequently. However, sometimes it is required to compute all spatial data at a time step at once due to boundary conditions, particularities of the considered stencils or numerical stability issues. This is the setting of the computations mentioned in Section II B 3. To obtain a field theory optimised for such computations, we present the following Proposition.

Proposition 2. Let $U^{i-1}, U^i, U^{i+1} \in (\mathbb{R}^d)^{(N_1-1)}$ such that (25) holds. Let $\mathcal{O} \subset (\mathbb{R}^d)^{(N_1-1)}$ be a convex neighbourhood of $u^* = U^{i+1}$, $\|\cdot\|$ a norm on $(\mathbb{R}^d)^{(N_1-1)}$ inducing an operator norm on $(\mathbb{R}^d)^{(N_1-1) \times (N_1-1)}$. Define $p(U) := \frac{\partial^2 L_{\Delta x}}{\partial U^i \partial U} (U^i, U)$ and let θ and $\bar{\theta}$ be Lipschitz constants on \mathcal{O} for p and for $\text{inv} \circ p$, respectively, where inv denotes matrix inversion. Let

$$\rho^* := \|\text{inv}(p(U^*))\| = \left\| \left(\frac{\partial^2 L_{\Delta x}}{\partial U^i \partial U^*} (U^i, U^*) \right)^{-1} \right\| \quad (28)$$

and let $f(U^{(n)})$ denote the left hand side of (25) with U^{i+1} replaced by $U^{(n)}$. If $\|U^{(0)} - U^*\| \leq \min\left(\frac{\rho^*}{\theta}, \frac{1}{2\theta\rho^*}\right)$ for $u^{(0)} \in \mathcal{O}$, then the Newton iterations $U^{(n+1)} := U^{(n)} - \text{inv}(p(U^{(n)}))f(U^{(n)})$ converge quadratically against U^* , i.e.

$$\|U^{(n+1)} - U^*\| \leq \rho^* \theta \|U^{(n)} - U^*\|^2. \quad (29)$$

Proof. The proof is analogous to the proof of Proposition 1, which can be found in Appendix C. \square

For computational details of how to approximate ρ^* numerically, see Appendix A 3.

IV. LOSS FUNCTION AND REGULARISER FOR TRAINING DISCRETE LAGRANGIAN DENSITIES

Assume we are given observations $U^{(k)}: X_d \rightarrow \mathbb{R}^d$ $k = 1, \dots, K$ of solutions to an unknown discrete field theory and the goal is to identify a model of the discrete field theory. We pick a type of a discrete Lagrangian density L_d such as the 3 or 4 points discrete Lagrangians introduced in Example 6 or any discretisation of (B3) by a linear quadrature formula. The discrete Lagrangian L_d is modelled as a neural network whose parameters are optimised such that

1. the discrete Euler–Lagrange equations (12) are consistent with the observations,
2. computations based on (12) are numerically well conditioned.

For this, we consider a loss function $\ell = \ell_{\text{data}} + w\ell_{\text{reg}}$ consisting of the data consistency term ℓ_{data} and the regulariser ℓ_{reg} and a weight $w \in \mathbb{R}_+$. These are specified in the following.

A. Data consistency

We introduce the data consistency term

$$\ell_{\text{data}} = \sum_{k=1}^K \sum_v |\text{DEL}(L_d)_v(U^{(k)})|^2. \quad (30)$$

In the inner sum, the sum is taken over all vertices for which the summand is defined. The term ℓ_{data} measures how consistent the model L_d is with the training data.

Remark 3 (Partial observations and batch learning). The computation of a summand $|\text{DEL}(L_d)_v(U^{(k)})|^2$ in (30) does not involve the full observation $U^{(k)}(X_d)$ but only the observation on a stencil such as the ones illustrated in Fig. 2. Therefore, partial observations can be incorporated in ℓ_{data} . Moreover, when batch learning⁵⁶ is applied, batches can contain tuples of stencil data coming from different observed solutions. This can be expected to prevent over-fitting phenomena.

Remark 4 (Low dimensionality). Although solutions to the discrete field theory can require solving high dimensional systems such as $|\text{DEL}(L_{\Delta x}^\Delta)^i(U)|^2 = 0$ in (25), in the training phase to compute ℓ_{data} each summand only requires evaluating the low-dimensional Euler–Lagrange equations (12).

Remark 5 (Locality). In the setting of the article, we assume that there exists a continuous field theory underlying our observational data $U^{(k)}$ on a mesh X_d (spacetime lattice). The underlying continuous theory can be described by a partial differential equation (2) and whether a function $u: X \rightarrow \mathbb{R}$ fulfils the differential equation can be checked point-wise, i.e. it is a local condition. This locality is reflected in our discrete model by the locality of the stencil defining the term $\text{DEL}(L_d)_v(U^{(k)})$. In this sense, the data-driven model will be local in space and time by construction. In contrast, if in the setting of a space-time lattice $X = [t^0, t^N] \times [x^0, x^N]$ we were to learn $L_{\Delta x}^\Delta$ (see (23)) rather than the discrete Lagrangian density L_d , potentially after model order reduction in the spatial directions, then locality would only be guaranteed in the time-direction.

B. Regulariser

Since any constant function L_d constitutes a minimiser of ℓ_{data} , a carefully chosen regulariser ℓ_{reg} is required to obtain a regular discrete field theory. Indeed, as seen in Example 5 much freedom in the choice of L_d can be expected. In view of Proposition 1 and 2, we consider regularisers that aim to minimise ρ^* of (27) or (29), respectively. If successfully optimised, we can expect solvability and rapid convergence of Newton iterations when computing solutions to the field theory, at least for initial data close to training data.

A regulariser ℓ_{reg} for the 3-point discrete Lagrangian of Example 6 is given by

$$\ell_{\text{reg}} = \alpha \sum_{k=1}^K \sum_{i,j} \left\| \left(\frac{\partial^2 L_d}{\partial u_j^i \partial u_{j+1}^{i+1}} (u_j^{i(k)}, u_{j+1}^{i+1(k)}, u_{j+1}^{i(k)}) \right)^{-1} \right\|^2 \quad (31)$$

with normalisation $\alpha = \frac{1}{KN_0N_1}$ by the number of summands. In the summation in (31) the indices i, j run over all interior mesh points. The expression is motivated by Proposition 1: each summand corresponds to $(\rho^*)^2$ of the Proposition which is related to the convergence speed of Newton iterations. We obtain the bound $\rho^* < \sqrt{\ell_{\text{reg}}}$. The resulting field theory will be optimised for propagating solutions forward in space and time using the 7-point stencil Eq. (13) visualised to the left of Fig. 2.

To obtain a field theory based on the 3- or 4-point discrete Lagrangian of Example 6 which is optimised for forward propagation of solutions as described in Section II B 3, we consider the regularisation

$$\ell_{\text{reg}} = \frac{1}{KN_0} \sum_{k=1}^K \sum_{i=1}^{N_0} \left\| \Lambda_i^{(k)-1} \right\|^2, \quad (32)$$

with

$$\Lambda_i^{(k)} = \frac{\partial^2 L_{\Delta x}^\Delta}{\partial U^i \partial U^{i+1}} (U^{i(k)}, U^{i+1(k)}) \quad (33)$$

The expression is based on Proposition 2. Here $U^{i(k)} = (u_j^{i(k)})_{0 < j < N_1}$. See Remark 12 and above for a description of $\Lambda^{(k)}$. In practise, we found that the tamed version

$$\ell_{\text{reg}} = \frac{1}{KN_0} \sum_{k=1}^K \sum_{i=1}^{N_0} \text{relu} \left(-10 \left\| \Lambda^{(k)-1} \right\|^{-2} + 1 \right) \leq 1 \quad (34)$$

is more suitable for training. Here $\text{relu}(x) = \max(0, x)$ is the rectified linear unit function. For numerical computation of $\left\| \Lambda^{(k)-1} \right\|^{-1}$ see Remark 13. The regularisers (32) and (34) both get large when the singular values of $\Lambda^{(k)}$ become small on the training data. However, (34) is tamed in the sense that the summands take values in $[0, 1)$ and are zero when the smallest singular values of $\Lambda^{(k)}$ is bigger than or equal to 0.1. On the training data set, ρ^* of Proposition 2 is bounded by $\rho^* \leq \sqrt{10(1 - \ell_{\text{reg}})^{-1}}$. Figure 3 shows the behaviour of the summands of various regularisers in terms of the square of the smallest singular value σ^2 .

For a better intuition of the action of the considered regularisers, Fig. 4 shows the effect of a regulariser in a simplified learning task: a neural network model of a function $f: \mathbb{R} \rightarrow \mathbb{R}$ is fitted to observations $f(x) = 0$ for $x \in R = \{-8, -5, 1.2, 3\}$. While a model trained with the mean squared error $\ell_{\text{data}}^{\text{illustration}} = \sum_{x \in R} f(x)^2$ is almost flat after 1100 optimisation steps with adam⁵⁷ (left of Fig. 4), the presence of the regulariser $\ell_{\text{reg}}^{\text{illustration}} = \sum_{x \in R} (0.01 + f'(x)^2)^{-1}$ drives the model of f to a representation which has nondegenerate roots at R that are easy to find using Newton iterations (right of

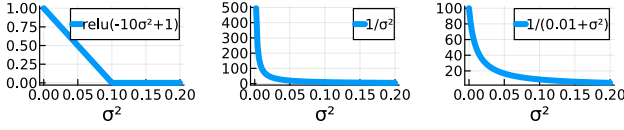


FIG. 3. Behaviour of summands in regularisers ℓ_{reg} of (31), (32), (34), and $\ell_{\text{illustration}}$.

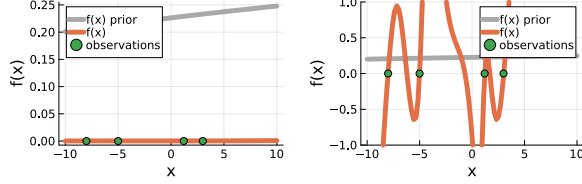


FIG. 4. Illustration of degeneracy of roots when learning without (left) or with (right) regularising terms. The orange curve refers to the trained model, the grey curve to the untrained model of f . The regulariser makes sure that the roots are non-degenerate and numerical root finding is well-conditioned.

Fig. 4). Indeed, the loss of the regulariser $\ell_{\text{reg}}^{\text{illustration}} < 0.855$ guarantees that $f'(x) > 3.38$ for $x \in R$ and thus the regularity of the roots problem for f close to R is guaranteed.

Remark 6 (Comparison with Gaussian Processes). When Gaussian Processes are used instead of neural networks, the problem of degenerate roots are not as pronounced: Fig. 5 shows the posterior distribution and its mean of a Gaussian Process for observations $f(x) = 0$, $x \in R$ (constant prior $m_0(x) = 0.5$, squared exponential kernel with unit length scale and unit standard deviation). The roots of the mean function at 1.2 and 3 are non-degenerate, while roots at -8 and -5 appear tangential. This behaviour is quite distinct from that of neural networks Fig. 4 (left). We conclude that each learning approaches requires adapted regularisation. We have developed regularisation strategies for Lagrangians modelled as Gaussian Processes in previous work²³.

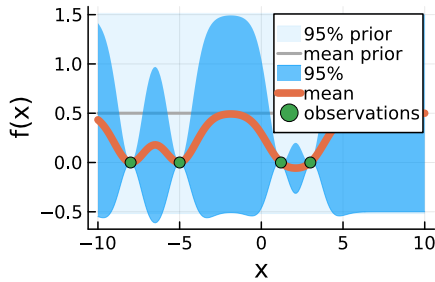


FIG. 5. As comparison to Fig. 4 (left), a posterior distribution for a Gaussian Processes for the same observations is computed without regularisation. The problem of degenerate roots of the mean is less pronounced but does occur for sparse training data.

V. VARIATIONAL PRINCIPLES FOR TRAVELLING WAVES IN DISCRETE AND CONTINUOUS THEORIES

This section discusses the notion of travelling waves in discrete field theories. As reference solutions for our numerical experiments in Section VI, we report analytic expressions for travelling waves of the discrete wave equation and discrete Schrödinger equation. Further, a central motivation for learning (discrete) field theories by learning a (discrete) Lagrangian density is the presence of variational structure of the learned model. We invoke Palais' principle of symmetric criticality³⁹. It implies that variational structure is passed on to the governing equations of motions of travelling waves if the discrete Lagrangian is autonomous. In the broader context of learning field theories and trying to predict highly symmetric solutions with the trained model, this is used as a motivation for learning discrete Lagrangian densities with the correct invariances. Readers mainly interested in the machine learning architecture may skip to the numerical experiments (Section VI) directly.

Highly symmetric solutions u of a continuous field theory, such as travelling waves, are of special interest due to their simple structure. When the same dynamical system is described by a discrete field theory, however, the values $u(X_d)$ of u over the mesh X_d might not have symmetry properties unless the symmetries of u happen to align with the symmetries of the mesh X_d . For instance, let $X = \mathbb{R} \times [0, b] / \sim$. Here $[0, b] / \sim$ denotes the identification of endpoints $0 \sim b$ of an interval $[0, b]$ (periodic boundary conditions). Functions $u: X \rightarrow \mathbb{R}$ with $u(t, x) = u(t + s, x + cs)$ for all $s \in \mathbb{R}$ are of the form $u(t, x) = f(x - ct)$ for $f: [0, b] / \sim \rightarrow \mathbb{R}$ (travelling wave with wave speed $c \in \mathbb{R}$). Consider a rectangular mesh $X_d \subset X$ with mesh widths $\Delta t, \Delta x > 0$. If and only if $\frac{c\Delta t}{\Delta x} \in \mathbb{Q}$, the values of the mesh $u(X_d)$ are periodic in time and $u(X_d)$ is a discrete travelling wave, i.e. $u_j^i = u_{j+z k_1}^{i+z k_0}$ for all $z \in \mathbb{Z}$, where $\frac{c\Delta t}{\Delta x} = \frac{k_0}{k_1}$ with $k_0 \in \mathbb{Z}$, $k_1 \in \mathbb{N}$. The theory of lattice differential equations^{58,59} provides a way to treat $u(X_d)$ as a travelling wave even if $\frac{c\Delta t}{\Delta x} \notin \mathbb{Q}$.

To design machine learning architectures for discrete field theories that effectively capture solutions corresponding to travelling waves in the continuous theory, we pay close attention to invariances of the field theory that relate to structural properties of the dynamical system that governs travelling waves. In particular, autonomy of a field theory relates to the fact that travelling waves themselves are governed by a variational principle. This is made precise in the following Proposition.

Proposition 3 (Symmetric criticality for travelling waves). *Let $c \in \mathbb{R} \setminus \{0\}$, $b > 0$, $X = ([0, b/c] / \sim) \times ([0, b] / \sim)$ be a torus (periodic boundary conditions), $W = C^1(X, \mathbb{R}^d)$ continuously differentiable functions and $\Sigma = \{u \in W \mid u(t + s, x + cs) = u(t, x)\}$ be travelling waves with wave speed c . Let $S: W \rightarrow \mathbb{R}$*

$$S(u) = \int_0^{b/c} \int_0^b L(u(t, x), u_t(t, x), u_x(t, x)) dx dt \quad (35)$$

be a continuously differentiable action functional with an au-

onomous Lagrangian $L: (\mathbb{R}^d)^3 \rightarrow \mathbb{R}$. Denote the restriction of S to Σ by $S_\Sigma: \Sigma \rightarrow \mathbb{R}$. Then a travelling wave $u \in \Sigma$ is a stationary point of S if and only if u is a stationary point of S_Σ . Using the identification $\Sigma \cong C^1([0, b]/\sim, \mathbb{R}^d)$, $u(t, x) = f(x - ct) = f(\xi)$ for $u \in \Sigma$, we have

$$S_\Sigma(f) = \frac{b}{c} \int_0^b L(f(\xi), -cf_\xi(\xi), f_\xi(\xi)) d\xi. \quad (36)$$

Remark 7. The statement of Proposition 3 holds true for standing travelling waves ($c = 0$), when the fraction b/c in the expression of the domain X as well as in (35) and in (36) are substituted by 1. To include the case $c = 0$ we will omit the irrelevant factor b/c when reporting S_Σ .

Notice that for $u \in \Sigma$ to be a stationary point of S_Σ , the action needs to be stationary at u with respect to variations through travelling waves. To be a stationary point of S , however, u needs to be stationary with respect to all variations $\delta u: X \rightarrow \mathbb{R}^d$. The Proposition says that stationarity of S_Σ is already sufficient to conclude stationarity of S .

The proof checks invariance of $S(u)$ under the transformation $u \mapsto s.u$ with $s.u(t, x) = u(t + s, x + cs)$ and verifies the conditions of Palais' principle of symmetric criticality (PSC)^{39,40}. A proof is contained in Appendix D.

In particular, Proposition 3 says that travelling waves are again governed by a variational principle, namely by the restricted functional $S_\Sigma: \Sigma \rightarrow \mathbb{R}$. We illustrate this on the wave equation and the Schrödinger equation.

Example 9 (PSC for wave equation). Consider the wave equation (Example 2) with periodic boundary conditions. Let $\xi = x - ct$. By Proposition 3, travelling wave solutions $u(t, x) = f(x - ct)$ of (4) exactly correspond to the stationary points of $S_\Sigma(f): [0, b]/\sim \rightarrow \mathbb{R}$ with

$$S_\Sigma = \int_0^b \left(\frac{1}{2} (c^2 - 1) f_\xi^2 - V(f) \right) d\xi. \quad (37)$$

Although Proposition 3 and Remark 7 guarantee variational structure of travelling wave solutions for any $c \in \mathbb{R}$, they do not imply existence of non-trivial solutions: for $V(f) = \frac{1}{2} f^2$ a Fourier ansatz in the Euler–Lagrange equation $(1 - c^2) f_{\xi\xi} = \nabla V(f)$ reveals the condition

$$c^2 = 1 + \frac{l^2}{4\pi^2 m^2}, \quad m \in \mathbb{Z} \setminus \{0\}$$

with solutions $u(t, x) = \alpha_1 \sin(\kappa(x - ct)) + \alpha_2 \cos(\kappa(x - ct))$, where $\kappa = 2\pi m/b$, $\alpha_1, \alpha_2 \in \mathbb{R}$.

Example 10 (PSC in Schrödinger equation). In analogy to Example 9, travelling waves $\Psi(t, x) = f(x - ct) = f(\xi) \in \mathbb{C}$ with wave speed $c \in \mathbb{R}$ in the Schrödinger equation (Example 4) with periodic boundary conditions are governed by the variational principle $S_\Sigma: [0, b]/\sim \rightarrow \mathbb{R}$ with $S_\Sigma(f) = \int_0^b L_\Sigma(f, f_\xi) d\xi$ with

$$L_\Sigma(f, f_\xi) = \frac{c\hbar}{2} (\bar{f} f_\xi - f \bar{f}_\xi) - f_\xi \bar{f}_\xi - V(f \bar{f}). \quad (38)$$

A Fourier ansatz for f reveals that for any $m \in \mathbb{Z} \setminus \{0\}$, $\alpha \in \mathbb{R}$ there is a travelling wave $\Psi(t, x) = f(\xi) = \alpha e^{i \frac{2\pi m}{b} \xi}$ with $\xi = x - ct$ and

$$c = \frac{2\pi m}{\hbar b} - \frac{b}{2\pi \hbar} \nabla V(\alpha^2). \quad (39)$$

Let us now obtain corresponding statements for discrete field theories. To make sense of travelling waves in discrete equations such as (13) and (14), standard approaches in lattice differential equations^{58,59} replace discrete equations by functional equations. We recover this interpretation from a variational perspective.

Lemma 1 (Continuous action for discrete field theory). *Let $X = ([0, b_0]/\sim) \times ([0, b_1]/\sim)$ be a torus and L_d the 3-point discrete Lagrangian density of Example 6. Let $\Delta t, \Delta x > 0$. Consider the action $S: X \rightarrow \mathbb{R}$,*

$$S(u) = \int_X L_d(u(t, x), u(t + \Delta t, x), u(t, x + \Delta x)) dx dt. \quad (40)$$

Stationary points $u: X \rightarrow \mathbb{R}^d$ are the solutions of the functional equation

$$\begin{aligned} \frac{\partial}{\partial u} \big(& L_d(u(t, x), u(t + \Delta t, x), u(t, x + \Delta x)) \\ & + L_d(u(t - \Delta t, x), u(t, x), u(t - \Delta t, x + \Delta x)) \\ & + L_d(u(t, x - \Delta x), u(t + \Delta t, x - \Delta x), u(t, x)) \big) = 0. \end{aligned} \quad (41)$$

Notice that (41) is (13) with u_{j+r}^{i+s} substituted by $u(t + s\Delta t, x + r\Delta x)$. The proof follows by a standard variational calculus. Analogous statements hold for all autonomous discrete Lagrangians (11), in particular for the 4-point discrete Lagrangian of Example 6. In analogy to Proposition 3 we have the following Proposition.

Proposition 4 (PSC for TW in discrete theory). *Let $c \in \mathbb{R} \setminus \{0\}$, $b > 0$, $X = ([0, b/c]/\sim) \times ([0, b]/\sim)$ be a torus, $W = C^1(X, \mathbb{R}^d)$ continuously differentiable functions and $\Sigma = \{u \in W \mid u(t + s, x + cs) = u(t, x)\}$ be travelling waves with wave speed c . Let $S: W \rightarrow \mathbb{R}$ be defined as in (40). Denote the restriction of S to Σ by $S_\Sigma: \Sigma \rightarrow \mathbb{R}$. Then a travelling wave $u \in \Sigma$ is a stationary point of S , if and only if u is a stationary point of S_Σ .*

The proof is analogous to the proof of Proposition 3 since the computation in (D3) of Appendix D works for all autonomous Lagrangians. An analogous statement holds in the case of standing travelling waves $c = 0$.

Corollary 1 (TW in discrete wave equation). *Travelling waves of the discrete wave equation (Example 7) with periodic boundary conditions in space (period b) are governed by $S_\Sigma: \Sigma \rightarrow \mathbb{R}$ with*

$$\begin{aligned} S_\Sigma(f) = \int_0^b \bigg(& \frac{1}{2} \left(\frac{f(\xi + \Delta t) - f(\xi)}{\Delta t} \right)^2 \\ & - \frac{1}{2} \left(\frac{f(\xi + \Delta x) - f(\xi)}{\Delta x} \right)^2 - V(f(\xi)) \bigg) d\xi \end{aligned}$$

using again the identification $\Sigma \cong C^1([0, b]/\sim, \mathbb{R})$.

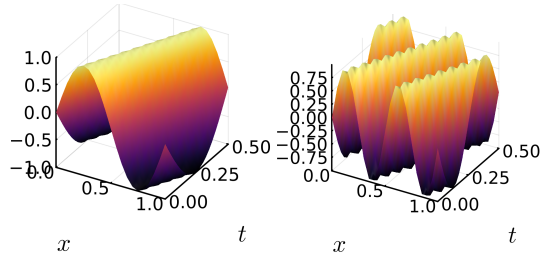


FIG. 6. First two travelling waves ($m = 1, 2$ in Example 11) of the discrete wave equation with quadratic potential. The wave crest appears rough due to the mesh resolution. This is expected in discrete theories.

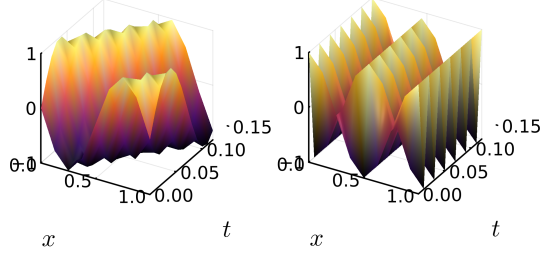


FIG. 7. Left: Travelling waves in Example 12 of the discrete Schrödinger equation ($m = 1$). The wave crest appears rough due to the mesh resolution. Right: Spurious travelling wave.

Corollary 2 (TW in discrete Schrödinger). *Travelling waves of the discrete Schrödinger equation (Example 8) are governed by S_Σ : $\Sigma \rightarrow \mathbb{R}$ with*

$$S_\Sigma(f) = \int_0^b L_d(f(\xi - \Delta t - \Delta x), f(\xi - \Delta t), f(\xi), f(\xi - \Delta x)) d\xi$$

and with L_d of (17), using again the identification $\Sigma \cong C^1([0, b]/\sim, \mathbb{R})$.

The expressions $S_\Sigma(f)$ follows from a calculation analogous to (D3). In the case $c \neq 0$ we have omitted the irrelevant constant factor $\frac{b}{c}$ when reporting S_Σ .

Example 11 (TW in discrete wave equation). Consider $V(u) = \frac{1}{2}u^2$ in the discrete wave equation (Example 7) with periodic boundary conditions in space (period b). A Fourier ansatz for f reveals (away from some resonant or degenerate cases) that $u(t, x) = f(\xi) = \alpha_1 \sin(\kappa \xi) + \alpha_2 \cos(\kappa \xi)$, $\xi = x - ct$, $\kappa = 2\pi m/b$, ($m \in \mathbb{Z} \setminus \{0\}$, $\alpha_1, \alpha_2 \in \mathbb{R}$) is a solution with wave speed c a real solution of

$$\cos(\kappa c \Delta t) = 1 - \frac{\Delta t^2}{2} + \frac{\Delta t^2}{\Delta x^2} (\cos(\kappa \Delta x) - 1).$$

See Fig. 6 for plots of the first two travelling waves ($m = 1, 2$).

Example 12 (TW in discrete Schrödinger). Consider a linear potential $V(u) = \beta u$ in the discrete Schrödinger equation (Example 8) with periodic boundary conditions in space (period b). A Fourier ansatz for f reveals (away from some resonant or degenerate cases) that $u(t, x) = f(\xi) = \alpha e^{i\kappa \xi}$, $\xi = x - ct$,

$\kappa = 2\pi m/b$, ($m \in \mathbb{Z} \setminus \{0\}$, $\alpha \in \mathbb{R}$) is a solution with wave speed

$$c = \frac{2}{\kappa \Delta t} \left(\arctan \left(\frac{2}{\hbar} \frac{\Delta t}{\Delta x^2} \tan^2 \left(\frac{1}{2} \kappa \Delta x \right) + \frac{\beta}{2\hbar} \Delta t \right) + s\pi \right),$$

where $s \in \mathbb{Z}$. Asymptotic expansion in $\Delta t, \Delta x$ recovers the expression (39) up to higher order terms. Additional travelling wave solutions of the discrete theory are obtained with wave speed $c = \frac{b(2\tilde{m}+1)}{2m\Delta t}$, $\tilde{m} \in \mathbb{Z}$. These are regarded as spurious since they do not converge to solutions of the continuous theory as $\Delta t \rightarrow 0$. (See Fig. 7.)

Remark 8 (Blended backward error analysis). Using the technique of blended backward error analysis^{60,61} we show that the discrete action S_Σ can formally be written as a continuous first order theory $S_\Sigma(f) = \int \tilde{L}(f, f_\xi) d\xi$, where \tilde{L} is a formal power series in $\Delta t, \Delta x$.

VI. NUMERICAL EXPERIMENTS

We report numerical experiments based on learning discrete field theories for the discrete wave equation (Example 7) and the discrete Schrödinger equation (Example 8) from data. The discrete field theories with the discrete Lagrangian densities (15), (17) are regarded as the true theories rather than as approximations of continuous theories in this context.

The use-case of our machine learning architecture is to learn effective models of variational dynamical systems whose underlying dynamics is an unknown variational partial differential equation. The observational data is discrete and lies on a spacetime lattice. We, therefore, restrict ourselves to explore models which capture solutions whose discrete Fourier modes exhibit an exponential decay. This disregards solutions that do not have a limit as lattice parameters are made asymptotically small. These are sometimes referred to as spurious solutions in numerical integration theory⁶² as they do not correspond to solutions of an underlying continuous theory.

We evaluate the performance of the data-driven models by computing or extending solutions of the trained models to initial data and checking consistency against the true models, compare performance to alternative reduced order modelling approaches (ROMs), and analyse how well travelling waves are captured. In the experiments, travelling waves will not be part of the training data but will be discovered using the trained model.

A. Wave equation

Consider the space-time domain $X = [0, T] \times [0, b]/\sim$ (periodic boundary conditions in space), $T = 0.5$, $b = 1$, discretisation parameters $\Delta t = 0.025$, $\Delta x = 0.05$, and quadratic potential $V(u) = \frac{1}{2}u^2$. The discrete lattice is denoted by X_d and $M = b/\Delta x = 20$ is the number of spatial interior grid points and $N = T/\Delta t = 20$ the number of time steps.

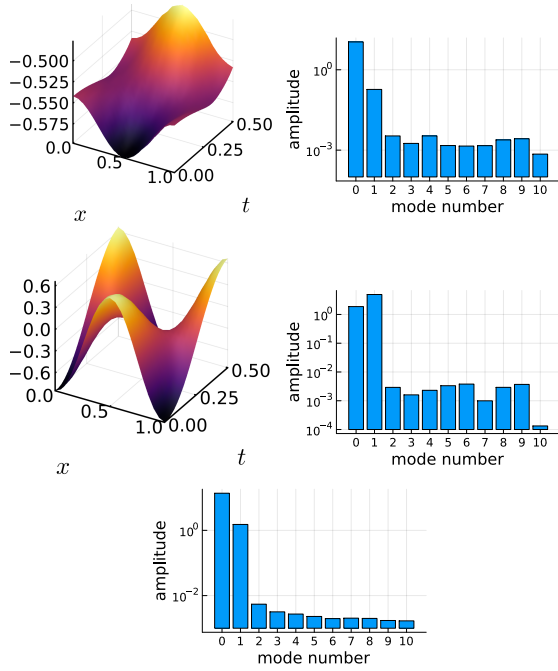


FIG. 8. Typical elements in the training data set (left) and their absolute value of amplitudes of Fourier modes averaged over time (right). Bottom: Absolute values of amplitudes of Fourier modes averaged over time and all training data.

1. Training data

To generate initial data, $M^{\text{rft}} = 11$ Fourier coefficients $\gamma_j = M e^{-2j^4} \eta_j$ ($j = 0, \dots, M^{\text{rft}}$) are computed where η_j are samples from a standard normal distribution. An application of the inverse of the discrete Fourier transform for real input (`irfft` of Julia package `FFTW.jl`⁶³) yields initial data $(u_j^0)_{j=0}^{M-1}$. Moreover, initial velocities $(\dot{u}_j^0)_{j=0}^{M-1}$ are drawn from a standard normal distribution. Then $K = 80$ solutions of the field theory are computed (Section II B 3). Figure 8 displays two typical solutions and the absolute values of the amplitudes of their discrete Fourier modes averaged over all time steps. A corresponding average over all times and all solutions is displayed at the bottom of Fig. 8. Each solution $U^{(k)} = (u_j^{i,(k)})_{0 \leq i \leq N, 0 \leq j < M}$ yields $(N-1)M = 380$ tuples $(u_j^{i,(k)}, u_j^{i+1,(k)}, u_{j+1}^{i,(k)}, u_{j+1}^{i-1,(k)}, u_{j+1}^{i,(k)}, u_{j-1}^{i,(k)}, u_{j-1}^{i+1,(k)})_{0 \leq i < N, 0 \leq j < M}$ of data corresponding to the 7-point stencil. The 7-point tuples constitute the training data set \mathcal{D} . \mathcal{D} contains 30400 tuples of stencil data. For batch training⁵⁶, \mathcal{D} is partitioned into batches of size 10.

2. Training

The 3 point discrete Lagrangian of Example 6 is modelled by a neural networks (3 layers, interior layer with 10 nodes, 160 parameters in total, activation function \tanh). Consider

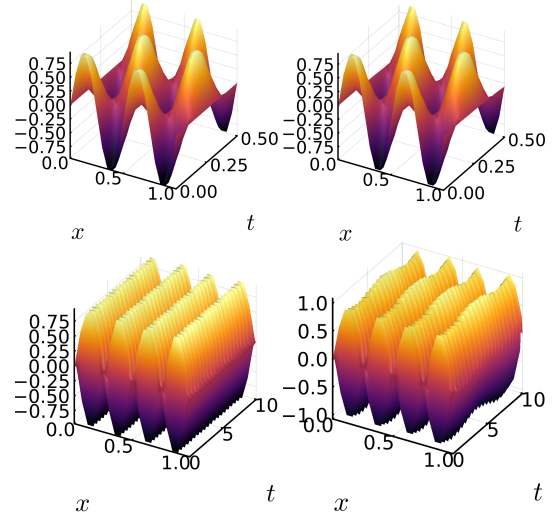


FIG. 9. True solution of the model (left) versus computed solution from the learned model (right). The initial data $u_j^0 = u_j^1 = \sin(4\pi j \Delta x)$ was not seen during training. Top line shows predictions until $T = 0.5$ (as in training), while the bottom line shows extrapolation until $20T = 10$.

ℓ_{data} of (30)

$$\ell_{\text{data}} = \sum_{\text{stencil data} \in \text{batch of } \mathcal{D}} |\text{DEL}(L_d)(\text{stencil data})|^2 \quad (42)$$

with $\text{DEL}(L_d)$ of (13), and ℓ_{reg} from (31)

$$\ell_{\text{reg}} = \alpha \sum \left\| \left(\frac{\partial^2 L_d}{\partial u_j^i \partial u_{j+1}^{i+1}}(u_j^{i,(k)}, u_j^{i+1,(k)}, u_{j+1}^{i,(k)}) \right)^{-1} \right\|^2, \quad (43)$$

with normalisation $\alpha = \frac{1}{\text{batch size}} = \frac{1}{10}$. The sum is taken over all stencil data $(u_j^{i,(k)}, u_j^{i+1,(k)}, u_{j+1}^{i,(k)}, u_{j+1}^{i-1,(k)}, u_{j+1}^{i,(k)}, u_{j-1}^{i,(k)}, u_{j-1}^{i+1,(k)})$ of a batch.

For each batch of \mathcal{D} one optimisation step with the optimisation method `adam`⁵⁷ with the default learning parameters of the Julia/Flux.jl^{64,65} implementation is applied to the parameters of the neural network using the loss function $\ell = \ell_{\text{data}} + \ell_{\text{reg}}$. This is repeated 1320 times (number of epochs). For the trained model we have $\ell_{\text{data}} \approx 8.6 \cdot 10^{-8}$ and $\ell_{\text{reg}} \approx 1.4 \cdot 10^{-7}$. (Here $\ell_{\text{data}}, \ell_{\text{reg}}$ are evaluated on the full data set \mathcal{D} .)

3. Evaluation

a. Extrapolation The trained neural network model of a discrete Lagrangian density is used to compute solutions of the learned field theory initialised from the unseen initial value $u_j^0 = u_j^1 = \sin(4\pi j \Delta x)$, $j = 0, \dots, M-1$ (Fig. 9). Computations are performed as explained in Section II B 3. On the training domain the absolute difference between a true

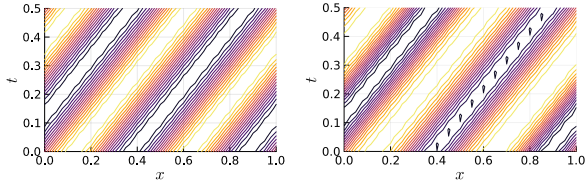


FIG. 10. Contour plot of true travelling wave (left) with wave number 2 and the located travelling wave (right).

solution is $\|U_{\text{true}} - U_{\text{predicted}}\|_{\infty} \approx 0.057$. On the extrapolated domain with final time $20T$ the error rises to $\|U_{\text{true}} - U_{\text{predicted}}\|_{\infty} \approx 0.343$ and to ≈ 1.41 for $100T$.

b. Travelling waves The travelling waves of Fig. 6 fulfil the discrete Euler-Lagrange equations (13) for the trained discrete Lagrangian density L_d with a maximal error of $\max_{i,j} |\text{DEL}(L_d)_j^i| < 0.004$ ($m = 1$, left plot of Fig. 6) or $\max_{i,j} |\text{DEL}(L_d)_j^i| < 0.033$ ($m = 2$, right plot of Fig. 6), respectively. Therefore, these true travelling waves constitute solutions of the learned discrete field theory (up to small error). Travelling waves can be found numerically in the trained model as well: for a guess of a wave speed c and a periodic function $f: [0, b] / \sim \rightarrow \mathbb{R}$

$$f(\xi) = \sum_{m=-(1+\lfloor \frac{M}{2} \rfloor)}^{1+\lfloor \frac{M}{2} \rfloor} \hat{f}_m e^{\frac{2\pi i m}{b} \xi} \quad (44)$$

represented by $1 + \lfloor \frac{M}{2} \rfloor$ Fourier coefficients $\hat{f}_m \in \mathbb{C}$, the objective function

$$\ell_{\text{wave}}(c, (\hat{f}_m)_{m=0}^{1+\lfloor \frac{M}{2} \rfloor}) = \sum_{i=1}^{N-1} \sum_{j=0}^{M-1} \|\text{DEL}(L_d)_j^i\|^2$$

is minimised with $\text{DEL}(L_d)_j^i$ from (13) with $u_j^i = f(i\Delta t - c j\Delta x)$ for all i, j . To avoid finding the trivial solution, the normalisation condition $\ell_{\text{wave}}^{\text{unit}} = \|\|f\|_{L_2}^2 - 1\|$ is added to the objective. Here $\|f\|_{L_2}^2 = \Delta x \sum |\hat{f}_m|^2$ is the discrete L_2 norm of f .

For the experiment, we initialise the optimisation procedure from the true Fourier coefficients $(\hat{f}_m)_m$ and wave speed c corresponding to the true travelling waves of Example 11 (wave number $m = 1, 2$) perturbed by random noise (identically independently normally distributed, standard deviation $\sigma = 0.5$). The located travelling wave U^{TW} has maximal norm error $\|U^{\text{TW}} - U_{\text{ref}}^{\text{TW}}\|_{\infty} < 0.116$ (wave number $m = 1$) or $\|U^{\text{TW}} - U_{\text{ref}}^{\text{TW}}\|_{\infty} < 0.31$ (wave number $m = 2$), respectively. The wave speeds are found up to an error of $|c^{\text{TW}} - c_{\text{ref}}^{\text{TW}}| < 0.0009$ or $|c^{\text{TW}} - c_{\text{ref}}^{\text{TW}}| < 0.0082$, respectively. Figure 10 compares the contours of the located and the true discrete travelling wave for the case $m = 2$.

B. Coarse Mesh

Even though training data $(U^{(k)})_{k=1}^K$ (see Section VIA 1) is available on the fine mesh X_d with $M(N+1)$ mesh

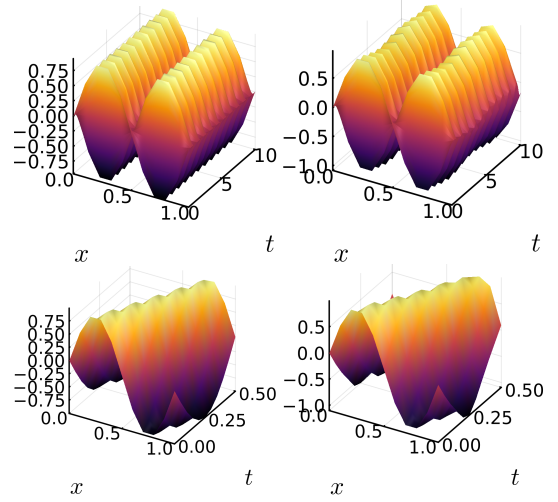


FIG. 11. Model for discrete wave equation trained for a coarser grid. Top: Computation of solution to unseen initial data $u_j^0 = u_j^1 = \sin(2\pi/T j \Delta x^{\text{coarse}})$ on extended domain $20T$ (left: reference, right: data-driven model). Bottom: Initialisation with exact data u_j^0, u_j^1 of a travelling wave. (left: reference, right: data-driven model)

points, the discrete model L_d can be trained on a coarser mesh. This can be useful to save computational costs when only low resolution predictions are required. Consider the mesh X_d^{coarse} for X with $(\Delta t^{\text{coarse}}, \Delta x^{\text{coarse}}) = (2\Delta t, 2\Delta x)$. We find $(N-3) * M * K = 27200$ tuples $(u_j^{i,(k)}, u_j^{i+2,(k)}, u_{j+2}^{i,(k)}, u_j^{i-2,(k)}, u_{j-2}^{i,(k)}, u_{j-2}^{i+2,(k)})_{0 \leq i < N-1, 0 \leq j < M}$ with data over the coarse 7-point stencil in the observational data. Notice that this makes efficient use also of data over mesh points in X_d that do not lie in X_d^{coarse} . Let $\mathcal{D}^{\text{coarse}}$ be a set of the 27200 training data tuples. The discrete Lagrangian density L_d^{coarse} is parametrised with a neural network of the same architecture as before. For training, the network parameters are randomly initialised. The same expressions for the loss function (42), (43) and the same batching and optimisation methods are used. L_d is trained for 63800 epochs with final losses $\ell_{\text{data}} \approx 1.1 \cdot 10^{-5}$, $\ell_{\text{reg}} \approx 2.5 \cdot 10^{-5}$.

Figure 11 shows numerical experiments with the trained coarse model. In the top row of the figure, the (coarse) 7-point stencil is used to compute a solution $U_{\text{predicted}}$ on X_d^{coarse} initialised from the unseen initial data $u_j^0 = u_j^1 = \sin(2\pi/T j \Delta x^{\text{coarse}})$. The prediction contains an extrapolation until $20T = 10$. A reference is computed with the 7 point stencil (reducing to a 5 point stencil) w.r.t. the original, fine mesh X_d applied to the discrete Lagrangian density (15) of Example 7 (the true model). Subsampling over X_d^{coarse} yields the reference U_{true} . We obtain $\|U_{\text{predicted}} - U_{\text{true}}\|_{\infty} \approx 0.33$. The bottom row of Fig. 11 shows the prediction $U_{\text{predict}}^{\text{TW}}$ of the model when initialised from data $u_j^0 = u_j^1$ of a travelling wave ($m = 1$) of the true system. We have $\|U_{\text{predict}}^{\text{TW}} - U_{\text{true}}^{\text{TW}}\|_{\infty} \approx 0.24$.

C. Comparison to MOR based technique

For comparison, we employ model order reduction in the spatial variable and then learn the temporal dynamics. This approach was explored in a recent articles in the continuous Lagrangian setting³⁶ and in related settings and problems^{13,32–35,37,38}. However, for consistency we will continue using discrete theory and continue using the training data of the discrete wave equation model (Section VI A 1).

We use standard principle component analysis^{66,67} to identify a linear projection map $\text{pr}: \mathbb{R}^M \rightarrow \mathbb{R}^{M_{\text{red}}} =: Q$ and a recovery map $R: Q \rightarrow \mathbb{R}^M$ such that $\text{pr} \circ R = \text{id}_Q$ with $M_{\text{red}} < M$. Then a neural network model of a discrete Lagrangian $L_d^Q: Q \times Q \rightarrow \mathbb{R}$ is trained to recover the temporal dynamics. To identify pr , let

$$\mathcal{U}^{\text{space}} = (U^{(1)}, U^{(2)}, \dots, U^{(K)}) \in \mathbb{R}^{MK(N+1)}$$

denote a matrix arranging the matrices $U^{(k)} = (u_j^i)_{0 \leq j < M}^{0 \leq i \leq N}$ of training data horizontally. Consider its singular value decomposition $\mathcal{U}^{\text{space}} = A \Sigma B^T$. Let $A' = (a_1, a_2) \in \mathbb{R}^{M \times M_{\text{red}}}$ contain the first M_{red} columns of A and set $\text{pr}(u) = A'^T u$ and $R(q) = A'q$. For $M_{\text{red}} = 2$ the average relative reconstruction error is

$$\frac{1}{MK(N+1)} \sum_{k=1}^{MK(N+1)} \frac{\|u_k - R(\text{pr}(u_k))\|_2}{\|u_k\|_2} \approx 0.0026,$$

where u_k denote the columns of $\mathcal{U}^{\text{space}}$.

To learn the temporal dynamics on the latent space, consider a neural network model of $L_d^Q: Q \times Q \rightarrow \mathbb{R}$ (3 layers, interior layer of size 10, 170 parameters, activation function `softplus`). Let $q^{i,(k)} \in Q$ denote the projected training data $q_{(k)}^i = \text{pr}((u_j^{i,(k)})_{0 \leq j < M})$ ($1 \leq k \leq K, 0 \leq i \leq N$) and let

$$\mathcal{D}^Q = \{(q_{(k)}^{i-1}, q_{(k)}^i, q_{(k)}^{i+1}) \mid 1 \leq k \leq K, 0 < i \leq N\}$$

be the collection of tipples in the projected training data. Consider the loss function $\ell^Q = \ell_{\text{data}}^Q + \ell_{\text{reg}}^Q$ with

$$\ell_{\text{data}}^Q = \sum_{q_{\text{stencil}} \in \mathcal{D}^Q} \text{DEL}(L_d^Q)(q_{\text{stencil}})^2 \quad (45)$$

with

$$\text{DEL}(L_d^Q)(q_{\text{stencil}}) = \frac{\partial}{\partial q^i} \left(L_d^Q(q^{i-1}, q^i) + L_d^Q(q^i, q^{i+1}) \right) \quad (46)$$

where $q_{\text{stencil}} = (q^{i-1}, q^i, q^{i+1})$ and regularisation

$$\ell_{\text{reg}}^Q = \sum_{q_{\text{stencil}} \in \mathcal{D}^Q} \left\| \left(\frac{\partial^2 L_d^Q}{\partial q^i \partial q^{i+1}} L_d^Q(q^i, q^{i+1}) \right)^{-1} \right\|^2.$$

The regularisation is an analogy to (31) for discrete Lagrangians of two input arguments optimised for forward propagation by solving (46) for q^{i+1} . The summands are approximated with 3 inverse vector iterations (see Remark 13).

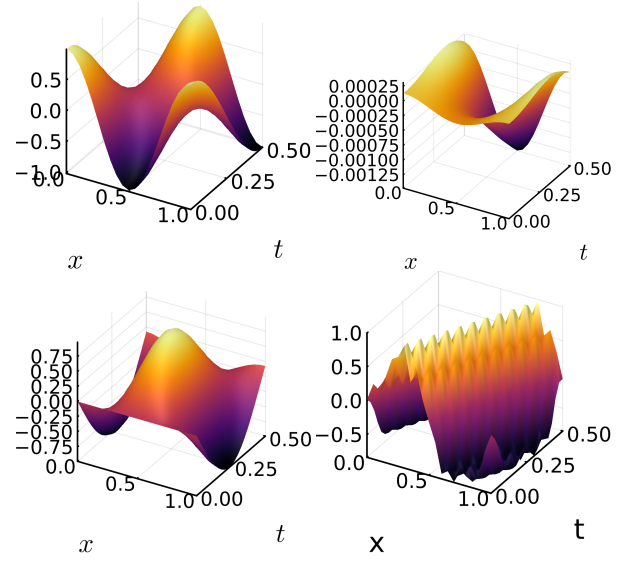


FIG. 12. While the ROM model successfully predicts motions initialised close to training data (top left) it struggles to generalise (top right compare to top of Fig. 9). Moreover, initialising with data of a true travelling wave does not produce a travelling wave (bottom left), while spurious travelling waves fulfil the discrete Euler-Lagrange equations when projected to the latent space (bottom right).

Again, batch training is performed with a batch size of 10. The regularisation term ℓ_{reg}^Q was scaled by 10^{-8} such that both terms of the loss function are of the same order of magnitude at the beginning of training with randomly initialised parameters of the neural network. After 54750 epochs of optimisation with adam, $\ell_{\text{data}}^Q < 1.19 \cdot 10^{-5}$, $\ell_{\text{reg}}^Q < 1.54 \cdot 10^{-5}$.

The trained model can recover the training data shown in Fig. 8 in max-error norm $\|\cdot\|_\infty$ up to 0.003 or 0.006 when initialised with the exact data at time $t = 0$, $t = \Delta t$. Figure 12 shows that while the ROM model successfully predicts dynamics with initial data close to training data (top left), it fails on initial data such as $u_j^0 = u_j^1 = \sin(4\pi j \Delta x)$, $j = 0, \dots, M-1$ (top right). This can be compared to the proposed discrete field theory model, which shows good performance on this task (see Fig. 9).

Unsurprisingly, as the training data does not contain any travelling waves, it fails to produce a good prediction when initialised with the exact initial data of a travelling wave. Indeed, the reconstructed solution has no similarity to a travelling waves (Fig. 12 bottom left). In contrast, the proposed stencil based scheme produces an output that is visually undistinguishable (not displayed) from the exact travelling wave (left of Fig. 6).

The travelling wave locator for discrete field theories described in paragraph VI A 3 b can easily be adapted to the ROM setting: as a loss function we use

$$\ell_{\text{wave}}^Q(c, (\hat{f})_{m=0}^{1+\lfloor \frac{M}{2} \rfloor}) = \sum_{i=1}^{N-1} \left\| \frac{\partial}{\partial q^i} \left(L_d^Q(q^{i-1}, q^i) + L_d^Q(q^i, q^{i+1}) \right) \right\|^2$$

with $q^i = \text{pr}((u_j^i)_{0 \leq j < M})$, with $u_j^i = f(i\Delta t - c j \Delta x)$ for all i, j , and f as in (44). We obtain the result U_Q^{TW} to the bottom right

of Fig. 12: indeed, the projection $\text{pr}(U_Q^{\text{TW}})$ of U_Q^{TW} fulfils the discrete Euler–Lagrange equation for L_d^Q on the latent space Q up to

$$\left| \sum_{q_{\text{stencil}} \in \mathcal{D}_{\text{TW}}^Q} \text{DEL}(L_d^Q)(q_{\text{stencil}})^2 \right| < 3 \cdot 10^{-9}, \quad (47)$$

where $\mathcal{D}_{\text{TW}}^Q$ denotes the triples in $\text{pr}(U_Q^{\text{TW}})$. However, it is apparently not an accurate approximation of the exact travelling wave (left of Fig. 6).

We conclude that the ROM-based method achieves accurate predictions for initial values close to its training data but can struggle in generalisation tasks as well as in the travelling wave task, in contrast to the proposed stencil based model. This is expected since the latent space was identified based on training data which does not contain travelling waves.

D. Schrödinger equation

Consider the space-time domain $X = [0, T] \times [0, b] / \sim$, $T = 0.12$, $b = 1$ with periodic boundary conditions in space, discretisation parameters $\Delta t = 0.01$, $\Delta x = 0.125$, and linear potential $V(r) = r$. The discrete lattice is denoted by X_d and $M = 1/\Delta x = 8$ is the number of spatial interior grid points and $N = T/\Delta t = 12$ the number of time steps.

To obtain training data, we use the 4-point Lagrangian density of Example 8 and compute 80 solutions: identifying $\mathbb{C} \cong \mathbb{R}^2$, $\Psi = \phi + ip \mapsto u = (\phi, p)^\top$, the corresponding Lagrangian $L_{\Delta x}^{\Delta t}$ of (23) is degenerate and of the form as in Example 3. Therefore, position data at time $t = 0$ is sufficient to initialise the numerical scheme of Section II B 3 and no velocity data is required. Initial position data is generated based on randomly sampled, exponentially decaying Fourier modes in analogy to Section VI A 1. Figure 13 (top left) shows the real part of such a solution.

A 4-point Lagrangian $L_d : (\mathbb{R}^2)^4 \rightarrow \mathbb{R}$ (see Example 6) is modelled as a neural network (2 hidden layers of width 12, activation `softplus`) and fitted to the training data. As a loss function $\ell = \ell_{\text{data}} + \ell_{\text{reg}}$ we adapt the data consistency term ℓ_{data} of (42) to 4-point Lagrangians. As regularisation term ℓ_{reg} we use the tamed regulariser (34) to obtain a field theory optimised for temporal forward propagation. A loss of $\ell_{\text{data}} < 8.04 \cdot 10^{-5}$ and $\ell_{\text{reg}} = 0$ is reached after 1544 epochs of batch learning, where a batch consists of 2 blocks and a block contains all stencil data of 3 consecutive time steps.

Figure 13 shows that the trained model successfully extrapolates training data with an extrapolation error $\|\Psi_{\text{true}} - \Psi_{\text{predicted}}\|_{\mathbb{C}} \approx 0.01$ with $T^{\text{extrapolate}} = 50\Delta t > 12\Delta t = T$ (bottom figures). Here we have used the notation $\|\Phi\|_{\mathbb{C}} = \max_{i,j} |\Phi_j^i|_{\mathbb{C}}$ for a complex matrix $\Phi \in \mathbb{C}^{(N+1) \times M}$.

Moreover, when the learned model is initialised with data $u^0 = (u_j^0)_{0 \leq j < M}$, $u^1 = (u_j^1)_{0 \leq j < M}$ of an exact travelling wave (Example 12, $m = 1$), then it reproduces the exact travelling wave up to an error of $\|\Psi_{\text{true}} - \Psi_{\text{predicted}}\|_{\mathbb{C}} \approx 0.19$. (Compare top right of Fig. 13 showing Ψ_{true} with the left of Fig. 7

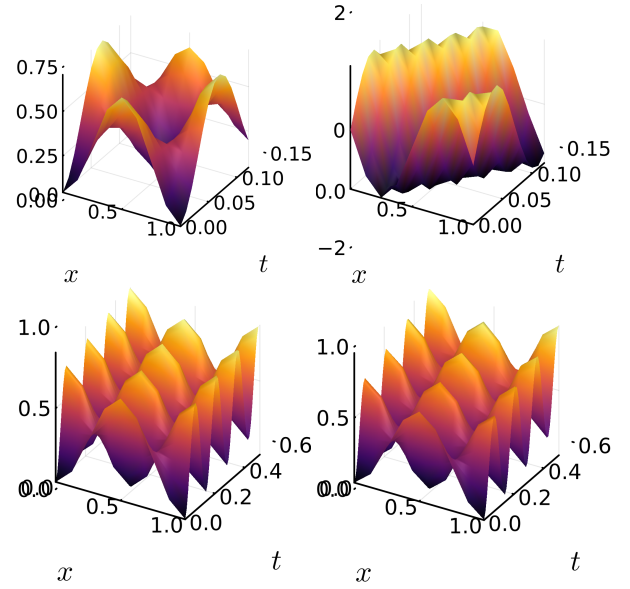


FIG. 13. Learned discrete Schrödinger equation. Top left: Sample of solution in training data (real part). Top right: Learned model reproduces travelling wave when initialised from exact travelling wave data (compare to left of Fig. 7). Bottom: Extrapolation of the solution with true discrete Lagrangian (left) and with the learned discrete Lagrangian (right) (error 0.01 in infinity norm).

showing Ψ_{true} .) Notice that travelling wave solutions are captured well in the trained model although the training data does not contain any travelling waves.

VII. SUMMARY AND FUTURE WORK

We have developed a machine learning framework to learn discrete field theories from discrete observation data. Neural network models of discrete Lagrangian densities are fitted such that the discrete Euler Lagrange equations are fulfilled on the training data. The discrete Lagrangian densities are based on local stencils motivated by the theory of variational integration methods²⁴. Thus, the local nature of differential equations is reflected in the structure of our approach.

As Lagrangian densities are not uniquely determined by a dynamical system's motions, we present a way to systematically derive regularisation terms for the training procedure. The regularisers guide the optimisation process to a numerical analysis informed model of a discrete Lagrangian density with good convergence properties when used in numerical simulations.

In comparison to architectures that project spatial dimensions to a data-driven latent space (data-driven reduced order models - ROMs) and then learn the temporal dynamics, our framework learns the discrete field theory on a space-time lattice. A reduction of computational complexity of the learning task in our approach is obtained by the use of local stencils rather than by a contraction of dimensionality. By a comparison with a simple, data-driven ROM based method, we demonstrate that learning on the lattice can be advantageous

when highly symmetric solutions such as travelling waves are sought in the data-driven model but are not present in the training data set. Moreover, locality, symmetries, and fundamental principles such as Palais' principle of symmetric criticality can be easier to preserve.

However, while locality of the stencil can be exploited during training in the evaluation of a data consistency term ℓ_{data} , depending on boundary conditions, numerical stability concerns, or particularities of an employed stencil, numerical integration of the trained model of a discrete field theory can require solving high-dimensional problems. Moreover, when the regulariser employed during training is based on the mentioned numerical method, then it can inherit the computational complexity. While we have shown that one can train the field theory on a coarse mesh while efficiently using the observed training data from the finer mesh, another option is to develop machine learning architectures tailored to numerical integrators developed for high-dimensional problems such as low-rank integrators^{49,50}. This will be the subject of future work.

Further, the machine learning architecture based on discrete Lagrangian densities for local stencils learns structurally simple solutions well and extrapolates correctly to highly symmetric solutions, such as travelling waves, even when these are not in the training data set. Numerical integration theory offers a diverse set of discretisation methods suitable for a variety of purposes including handling of shock waves or highly oscillatory solutions. In future work, we plan to exploit these methods within the machine learning context to develop reliable architectures for systems with challenging dynamical features.

Additionally, we propose to learn a representation of a field theory that is as symmetric as possible to improve qualitative features of numerical simulations as we have done in the ode case⁵. A combination with this technique will allow for the identification of conservation laws of the underlying dynamical system.

ACKNOWLEDGMENTS

C. Offen acknowledges the Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen and computing time provided by the Paderborn Center for Parallel Computing (PC2).

AUTHOR DECLARATIONS

The authors have no conflicts to disclose.

DATA AVAILABILITY

The data that support the findings of this study are openly available in the GitHub repository Christian-Offen/DLNN_pde at https://github.com/Christian-Offen/DLNN_pde. An archived version⁶⁸ of

release v1.0 of the GitHub repository is openly available at <https://dx.doi.org/10.5281/zenodo.8245861>.

Appendix A: Complementary remarks

1. Gauge freedom

Remark 9 (Gauge freedom - Continuous setting). For a continuously differentiable function $F = (F^1, \dots, F^n)$ in (x, u) , denote the total divergence

$$\nabla_x F(x, u(x)) = \frac{\partial}{\partial x_0}(F^1(x, u(x))) + \dots + \frac{\partial}{\partial x_n}(F^n(x, u(x))).$$

When $\nabla_x F(x, u(x))$ is added to a Lagrangian L , then the action $S(1)$ can at most change by a constant, according to the divergence theorem. Therefore, the Euler–Lagrange equations (2) are not effected, i.e.

$$\text{EL}(L + \nabla_x F(x, u(x))) = \text{EL}(L).$$

Remark 10 (Gauge freedom - Discrete setting). In analogy to Remark 9, discrete Lagrangians are not uniquely determined by a system's discrete motions. Indeed, two 3-point Lagrangians $L_d(a, b, c)$ and $\tilde{L}_d(a, b, c)$ as in Example 6 yield the same discrete Euler–Lagrange equations (13) if they differ by a discrete divergence, i.e. if $L_d(a, b, c) - \tilde{L}_d(a, b, c)$ is of the form

$$\chi_1(a) - \chi_1(b) + \chi_2(a) - \chi_2(c) + \chi_3(b) - \chi_3(c) \quad (\text{A1})$$

for a differentiable function $\chi = (\chi_1, \chi_2, \chi_3) : X \rightarrow \mathbb{R}^3$. Analogous observations apply to other types of discrete Lagrangians.

2. Use of 7 and 9 point stencil for forward propagations

Below additional remarks on the computation with the considered 7 and 9 point numerical stencils (Example 6) as described in Section II B 3.

Remark 11 (Numerical stability of discrete SE). Solutions to the discrete Schrödinger equation (21) can be computed as explained in Section II B 3. Notice that when the discrete Lagrangian L_d of (17) is expressed in the variable (u, ϕ) , where $\Psi = u + i\phi$, then $L_{\Delta x}^{\Delta t}$ of Section II B 3 is of the form (5) of Example 3. Therefore, the computation can be initialised from initial data $(\Psi_j^0)_{0 \leq j \leq N_1}$ and boundary conditions $(\Psi_0^i)^{0 < i \leq N_0}$, $(\Psi_{N_1}^i)^{0 < i \leq N_0}$ and is numerically stable⁵². No velocity data is required for the initialisation.

3. Numerical approximation of constant in Newton iterations convergence results

We provide details on how to approximate the constant ρ^* in Proposition 2, which relates to the convergence speed of Newton iterations.

Remark 12. The matrix $\Lambda = \frac{\partial^2 L_{\Lambda}^M}{\partial U^i \partial U^{i+1}}(U^i, U^{i+1})$ of (28) has the following block-tri-diagonal structure

$$\Lambda = \begin{pmatrix} A_1 & B_1 & & & \\ C_2 & A_2 & B_2 & & \\ & \ddots & \ddots & \ddots & \\ & & C_{N_1-2} & A_{N_1-2} & B_{N_1-2} \\ & & & C_{N_1-1} & A_{N_1-1} \end{pmatrix} \in (\mathbb{R}^d)^{(N_1-1) \times (N_1-1)}, \quad (\text{A2})$$

where

$$\begin{aligned} A_j &= \frac{\partial^2}{\partial u_j^i \partial u_j^{i+1}} (L_d(u_j^i, u_j^{i+1}, u_{j+1}^i, u_{j+1}^{i+1}) \\ &\quad + L_d(u_{j-1}^i, u_{j-1}^{i+1}, u_j^i, u_j^{i+1})) \\ B_j &= \frac{\partial^2}{\partial u_j^i \partial u_{j+1}^{i+1}} (L_d(u_j^i, u_j^{i+1}, u_{j+1}^i, u_{j+1}^{i+1})) \\ C_j &= \frac{\partial^2}{\partial u_j^i \partial u_{j-1}^{i+1}} (L_d(u_{j-1}^i, u_{j-1}^{i+1}, u_j^i, u_j^{i+1})). \end{aligned} \quad (\text{A3})$$

If periodic boundary conditions rather than Dirichlet boundary conditions are considered then the spatial mesh has N_1 interior mesh points and

$$\Lambda = \begin{pmatrix} A_0 & B_0 & & & C_0 \\ C_1 & A_1 & B_1 & & \\ & \ddots & \ddots & \ddots & \\ & & C_{N_1-2} & A_{N_1-2} & B_{N_1-2} \\ B_{N_1-1} & & & C_{N_1-1} & A_{N_1-1} \end{pmatrix} \in (\mathbb{R}^d)^{N_1 \times N_1} \quad (\text{A4})$$

with A_j, B_j, C_j as in (A3)

Remark 13 (Numerical computation of $\|\Lambda^{-1}\|$). Let $\|\Lambda\|$ denote the spectral norm of Λ . The value $\rho^* = \|\Lambda^{-1}\|^{-1}$ is the smallest singular value σ of Λ . Its square σ^2 can be approximated using inverse vector iterations⁶⁹ applied to $M = \Lambda^\top \Lambda$: given a random start vector v' , iterate $v \leftarrow v'$, $v \leftarrow M \setminus v$, $v \leftarrow v/\|v\|$, $\sigma^2 \leftarrow (v^\top v')^{-1}$. Here $M \setminus v$ denotes the solution of the linear equation $Mx = v$. For this a Cholesky factorisation of M can be computed and used throughout the iteration. While in our case sparse structure of Λ can be exploited, we refer to the literature⁶⁹ for methods that avoid the explicit computation of the matrix product $\Lambda^\top \Lambda$ in the computation of singular values. Further, more elaborate tensor representations⁷⁰ can be used, especially when generalising to higher-dimensional space-time lattices. Approximations to $\sigma^2 = (\rho^*)^2$ will be used in the machine learning framework introduced in the following sections.

Appendix B: Exact discrete Lagrangians

We can view the various discrete Lagrangians introduced in Section II B as discretisations of exact discrete Lagrangians. An exact discrete Lagrangian to a continuous field theory has the property that solutions to the discrete Euler–Lagrange equations coincide with the motions of the continuous theory

on the grid. For this we extend our definition of discrete Lagrangians from Section II B. For a more detailed introduction and further information we refer to⁴⁷.

With the notation of Section II B, let ∂X^l denote the boundary of the hypercube X^l . Let $\partial X_d = \bigcup_{l \in \mathcal{J}} \partial X^l \subset \mathbb{R}^{n+1}$ denote the n -skeleton of X .

Definition 1 (Discrete Lagrangian and action). A *discrete Lagrangian* $L_d: \bigcup_l (\partial X^l \times \mathbb{R}^k) \rightarrow \mathbb{R}$ maps any graph of a boundary function $u_{\partial X^l}: \partial X^l \rightarrow \mathbb{R}^k$ to a real number. The corresponding *discrete action* $S_d: \partial X_d \times \mathbb{R}^k \rightarrow \mathbb{R}$ maps any graph of a boundary function $u_{\partial X_d}: \partial X_d \rightarrow \mathbb{R}^k$ on the n -skeleton to

$$S_d(u_{\partial X_d}) = \sum_{l \in \mathcal{J}} L_d(u_{\partial X_d}|_{\partial X^l}). \quad (\text{B1})$$

Here $u_{\partial X_d}|_{\partial X^l}$ denotes the restriction of $u_{\partial X_d}$ to the boundary ∂X^l of the cube X^l .

Notice that in Definition 1 L_d is allowed to depend on the graph of the boundary function rather than just its values, which is implicit in the notation.

Definition 2 (Discrete Euler–Lagrange equations). Let L_d be a discrete Lagrangian. For each interior vertex $v \in \dot{X}_d$ let $\mathcal{J}_v \subset \mathcal{J}$ contain the indices of hypercubes X^l containing v . The *discrete Euler–Lagrange equations* are given as

$$\sum_{l \in \mathcal{J}_v} \frac{\delta L_d}{\delta u_{\partial X^l}}(u_{\partial X_d}|_{\partial X^l}) = 0 \quad (\text{B2})$$

for $u_{\partial X_d}: \partial X_d \rightarrow \mathbb{R}^k$. Here $\frac{\delta L_d}{\delta u_{\partial X^l}}$ denotes the variational derivative.

Example 13 (Exact discrete Lagrangian). For a Lagrangian L as in (1) we define the *exact discrete Lagrangian*

$$L_d^{\text{exact}}(u_{\partial X^l}) = \int_{X^l} L(x, u(x), u_{x_0}(x), \dots, u_{x_n}(x)) dx, \quad (\text{B3})$$

where $u: X^l \rightarrow \mathbb{R}^k$ is the solution (assuming existence/uniqueness) of the Euler–Lagrange equation $\text{EL}(L) = 0$ subject to the boundary condition $u|_{\partial X^l} = u_{\partial X^l}$.

We conclude with the following observations.

- If $u: X \rightarrow \mathbb{R}^k$ solves the Euler–Lagrange equation (2) then $S(u) = S_d^{\text{exact}}(u|_{\partial X_d})$, when S_d is the discrete action (B1) for the exact discrete Lagrangian $L_d = L_d^{\text{exact}}$.
- In the ode case $\dim X = 1$ we have $\partial X_d = X_d = \{x_0^0, \dots, x_0^{N_0}\}$ and the exact discrete action is defined for functions on the mesh X_d .
- The discrete Lagrangians (15) of Example 7 and (17) of Example 8 can be obtained from (B3) by quadrature.

Proof. Equipped with the norm $\|\cdot\|_W$ with $\|u\|_W = \max_{(t,x) \in X} |u(t,x)| + \max_{(t,x) \in X} \|Du(t,x)\|_{\mathbb{R}^d \times \mathbb{R}^2}$ the space W is a Banach space since the torus X is compact. The group

$([0, b]/\sim, +)$ is compact and acts on the Banach space W by $u \mapsto s.u$ with $s.u(t, x) = u(t + s, x + cs)$. The fixpoints of the group action are Σ . The considered action S is invariant under the action, i.e. $S(u) = S(s.u)$ for all $s \in [0, b]/\sim$ since the Lagrangian is autonomous: let $u \in W$, $s \in [0, b]$.

$$\begin{aligned} S(s.u) &= \int_0^{b/c} \int_0^b L(s.u(t, x), (s.u)_t(t, x), (s.u)_x(t, x)) dx dt \\ &= \int_{-s/c}^{(b-s)/c} \int_{-s}^{b-s} L(u(t, x), u_t(t, x), u_x(t, x)) dx dt \\ &= \int_0^{b/c} \int_0^b L(u(t, x), u_t(t, x), u_x(t, x)) dx dt \\ &= S(u). \end{aligned} \quad (D3)$$

By compactness of the group $([0, b]/\sim, +)$, Palais' principle of symmetric criticality³⁹ guarantees that critical points of the restricted functional $S_\Sigma: \Sigma \rightarrow \mathbb{R}$ constitute critical points of S .

To proof validity of (D2), consider coordinates $(x - ct, x + ct) = (\xi, \theta)$ and a travelling wave $u(t, x) = f(x - ct) = f(\xi)$, $f: [0, b]/\sim \rightarrow \mathbb{R}^d$. We have

$$\begin{aligned} S_\Sigma(f) &= S(u) = \int_0^{b/c} \int_0^b L(f(\xi), -cf_\xi(\xi), f_\xi(\xi)) dx dt \\ &= \int_{-b}^b \int_{|\xi|}^{2b-|\xi|} L(f(\xi), -cf_\xi(\xi), f_\xi(\xi)) \frac{d\theta d\xi}{2c} \\ &\stackrel{(*)}{=} \int_0^b \int_0^{2b} L(f(\xi), -cf_\xi(\xi), f_\xi(\xi)) \frac{d\theta d\xi}{2c} \\ &= \frac{b}{c} \int_0^b L(f(\xi), -cf_\xi(\xi), f_\xi(\xi)) d\xi \end{aligned} \quad (D4)$$

where we have used periodicity $f(\xi + b) = f(\xi)$ in $(*)$. \square

Remark 14. For standing travelling waves $c = 0$, the torus $X = [0, 1]/\sim \times [0, b]/\sim$ is considered. An analogous statement is obtained with b/c substituted by 1 in (D1) and (D2).

¹C. Offen and S. Ober-Blöbaum, "Learning of discrete models of variational PDEs from data," *Chaos* **34**, 013104 (2024).

²K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks* **1**, 4–27 (1990).

³J. Han, C. Ma, Z. Ma, and W. E, "Uniformly accurate machine learning-based hydrodynamic models for kinetic equations," *Proceedings of the National Academy of Sciences* **116**, 21983–21991 (2019).

⁴E. Dierkes, C. Offen, S. Ober-Blöbaum, and K. Flaßkamp, "Hamiltonian neural networks with automatic symmetry detection," *Chaos: An Interdisciplinary Journal of Nonlinear Science* **33** (2023), 10.1063/5.0142969, 063115.

⁵Y. Lishkova, P. Scherer, S. Ridderbusch, M. Jamnik, P. Liò, S. Ober-Blöbaum, and C. Offen, "Discrete Lagrangian neural networks with automatic symmetry discovery (accepted)," in *Proceedings of IFAC World Congress 2023, Yokohama, Japan (9-14/07/2023)* (IFAC-PaperOnLine, 2023) 2211.10830.

⁶J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Phys. Rev. Fluids* **3**, 074602 (2018).

⁷J. Kadupitiya, F. Sun, G. Fox, and V. Jadhao, "Machine learning surrogates for molecular dynamics simulations of soft materials," *Journal of Computational Science* **42**, 101107 (2020).

⁸G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics* **3**, 422–440 (2021).

⁹J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems* (Springer New York, New York, NY, 1999).

¹⁰E. L. Mansfield, *A Practical Guide to the Invariant Calculus*, Cambridge Monographs on Applied and Computational Mathematics (Cambridge University Press, 2010).

¹¹M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," (2020).

¹²B. Hamzi and H. Owhadi, "Learning dynamical systems from data: A simple cross-validation perspective, part i: Parametric kernel flows," *Physica D: Nonlinear Phenomena* **421**, 132817 (2021).

¹³C. Allen-Blanchette, S. Veer, A. Majumdar, and N. E. Leonard, "LagNetViP: A Lagrangian neural network for video prediction (AAAI 2020 symposium on physics guided ai)," (2020).

¹⁴M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian Networks: Using physics as model prior for deep learning," in *7th International Conference on Learning Representations (ICLR)* (ICLR, 2019).

¹⁵H. Schaeffer, "Learning partial differential equations via data discovery and sparse optimization," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **473**, 20160446 (2017).

¹⁶S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances* **3**, e1602614 (2017), <https://www.science.org/doi/pdf/10.1126/sciadv.1602614>.

¹⁷T. Tripura and S. Chakraborty, "A bayesian framework for discovering interpretable lagrangian of dynamical systems from data," (2023), arXiv:2310.06241 [stat.ML].

¹⁸S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian Neural Networks," in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).

¹⁹P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis, "SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems," *Neural Networks* **132**, 166–179 (2020).

²⁰T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis, "On learning Hamiltonian systems from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**, 121107 (2019).

²¹Y. Chen, B. Xu, T. Matsubara, and T. Yaguchi, "Variational principle and variational integrators for neural symplectic forms," in *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems* (2023).

²²H. Qin, "Machine learning and serving of discrete field theories," *Scientific Reports* **10** (2020), 10.1038/s41598-020-76301-0.

²³S. Ober-Blöbaum and C. Offen, "Variational learning of Euler–Lagrange dynamics from data," *Journal of Computational and Applied Mathematics* **421**, 114780 (2023).

²⁴J. E. Marsden and M. West, "Discrete mechanics and variational integrators," *Acta Numerica* **10**, 357–514 (2001).

²⁵C. Offen and S. Ober-Blöbaum, "Learning discrete Lagrangians for variational pdes from data and detection of travelling waves," in *Geometric Science of Information*, Vol. 14071, edited by F. Nielsen and F. Barbaresco (Springer Nature Switzerland, Cham, 2023) pp. 569–579.

²⁶L. 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**, 218–229 (2021).

²⁷Z. Long, Y. Lu, X. Ma, and B. Dong, "PDE-net: Learning PDEs from data," in *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 80, edited by J. Dy and A. Krause (PMLR, 2018) pp. 3208–3216.

²⁸Z. Long, Y. Lu, and B. Dong, "PDE-Net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network," *Journal of Computational Physics* **399**, 108925 (2019).

²⁹T. Matsubara, A. Ishikawa, and T. Yaguchi, "Deep energy-based modeling of discrete-time physics," in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 13100–13111.

³⁰P. 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**,

- 8271–8283 (2023).
- ³¹ S. Eidnes and K. O. Lye, “Pseudo-Hamiltonian neural networks for learning partial differential equations,” (2023), arXiv:2304.14374 [cs.LG].
 - ³² J. Mason, C. Allen-Blanchette, N. Zolman, E. Davison, and N. Leonard, “Learning interpretable dynamics from images of a freely rotating 3d rigid body,” (2022).
 - ³³ P. Buchfink, S. Glas, and B. Haasdonk, “Symplectic model reduction of Hamiltonian systems on nonlinear manifolds and approximation with weakly symplectic autoencoder,” *SIAM Journal on Scientific Computing* **45**, A289–A311 (2023).
 - ³⁴ K. Carlberg, R. Tuminaro, and P. Boggs, “Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics,” *SIAM Journal on Scientific Computing* **37**, B153–B184 (2015).
 - ³⁵ H. Sharma, H. Mu, P. Buchfink, R. Geelen, S. Glas, and B. Kramer, “Symplectic model reduction of Hamiltonian systems using data-driven quadratic manifolds,” (2023), arXiv:2305.15490 [math.NA].
 - ³⁶ H. Sharma and B. Kramer, “Preserving lagrangian structure in data-driven reduced-order modeling of large-scale dynamical systems,” (2022), arXiv:2203.06361 [math.NA].
 - ³⁷ H. Sharma, Z. Wang, and B. Kramer, “Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical Hamiltonian systems,” *Physica D: Nonlinear Phenomena* **431**, 133122 (2022).
 - ³⁸ T. M. Tyranski and M. Kraus, “Symplectic model reduction methods for the Vlasov equation,” *Contributions to Plasma Physics*, e202200046 (2022).
 - ³⁹ R. S. Palais, “The principle of symmetric criticality,” *Comm. Math. Phys.* **69**, 19–30 (1979).
 - ⁴⁰ C. G. Torre, “Symmetric Criticality in Classical Field Theory,” *AIP Conference Proceedings* **1360**, 63–74 (2011).
 - ⁴¹ P. J. Olver, *Applications of Lie Groups to Differential Equations* (Springer US, 1986).
 - ⁴² J.-L. Basdevant, *Variational Principles in Physics* (Springer New York, New York, NY, 2007).
 - ⁴³ M. Henneaux, “Equations of motion, commutation relations and ambiguities in the lagrangian formalism,” *Annals of Physics* **140**, 45–64 (1982).
 - ⁴⁴ G. Marmo and C. Rubano, “On the uniqueness of the Lagrangian description for charged particles in external magnetic field,” *Il Nuovo Cimento A* **98**, 387–399 (1987).
 - ⁴⁵ G. Marmo and G. Morandi, “On the inverse problem with symmetries, and the appearance of cohomologies in classical Lagrangian dynamics,” *Reports on Mathematical Physics* **28**, 389–410 (1989).
 - ⁴⁶ A. Deriglazov, “On singular Lagrangian underlying the Schrödinger equation,” *Physics Letters A* **373**, 3920–3923 (2009).
 - ⁴⁷ J. E. Marsden, S. Pekarsky, S. Shkoller, and M. West, “Variational methods, multisymplectic geometry and continuum mechanics,” *Journal of Geometry and Physics* **38**, 253–284 (2001).
 - ⁴⁸ E. Kieri, C. Lubich, and H. Walach, “Discretized dynamical low-rank approximation in the presence of small singular values,” *SIAM Journal on Numerical Analysis* **54**, 1020–1038 (2016).
 - ⁴⁹ C. Lubich, B. Vandereycken, and H. Walach, “Time integration of rank-constrained tucker tensors,” *SIAM Journal on Numerical Analysis* **56**, 1273–1290 (2018), <https://doi.org/10.1137/17M1146889>.
 - ⁵⁰ H. M. Walach, *Time integration for the dynamical low-rank approximation of matrices and tensors*, Ph.D. thesis, Eberhard Karls Universität Tübingen (2019).
 - ⁵¹ S. Schrammer, *On dynamical low-rank integrators for matrix differential equations*, Ph.D. thesis, Karlsruher Institut für Technologie (KIT) (2022).
 - ⁵² C. Rowley and J. Marsden, “Variational integrators for degenerate lagrangians, with application to point vortices,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002., Vol. 2 (2002) pp. 1521–1527.
 - ⁵³ M. Kraus, “Projected variational integrators for degenerate lagrangian systems,” (2017), arXiv:1708.07356 [math.NA].
 - ⁵⁴ C. L. Ellison, J. M. Finn, J. W. Burby, M. Kraus, H. Qin, and W. M. Tang, “Degenerate variational integrators for magnetic field line flow and guiding center trajectories,” *Physics of Plasmas* **25** (2018), 10.1063/1.5022277, 052502.
 - ⁵⁵ J. W. Burby, J. M. Finn, and C. L. Ellison, “Improved accuracy in degenerate variational integrators for guiding centre and magnetic field line flow,” *Journal of Plasma Physics* **88**, 835880201 (2022).
 - ⁵⁶ S. Marsland, *Machine Learning: An Algorithmic Perspective, Second Edition*, 2nd ed. (Chapman & Hall/CRC, 2014).
 - ⁵⁷ D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2015).
 - ⁵⁸ J. Mallet-Paret, “Traveling waves in spatially discrete dynamical systems of diffusive type,” in *Dynamical Systems: Lectures given at the C.I.M.E. Summer School held in Cetraro, Italy, June 19-26, 2000*, edited by J. W. Macki and P. Zecca (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003) pp. 231–298.
 - ⁵⁹ H. J. Hupkes and E. S. V. Vleck, “Travelling waves for adaptive grid discretizations of reaction diffusion systems III: Nonlinear theory,” *Journal of Dynamics and Differential Equations* (2022), 10.1007/s10884-022-10143-4.
 - ⁶⁰ R. I. McLachlan and C. Offen, “Backward error analysis for conjugate symplectic methods,” *Journal of Geometric Mechanics* **15**, 98–115 (2023).
 - ⁶¹ R. I. McLachlan and C. Offen, “Backward error analysis for variational discretisations of pdes,” *Journal of Geometric Mechanics* **14**, 447–471 (2022).
 - ⁶² E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer Series in Computational Mathematics (Springer Berlin Heidelberg, 2013).
 - ⁶³ M. Frigo and S. G. Johnson, “The design and implementation of FFTW3,” *Proceedings of the IEEE* **93**, 216–231 (2005), special issue on “Program Generation, Optimization, and Platform Adaptation”.
 - ⁶⁴ M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, “Fashionable modelling with Flux,” *CoRR abs/1811.01457* (2018), arXiv:1811.01457.
 - ⁶⁵ M. Innes, “Flux: Elegant machine learning with Julia,” *Journal of Open Source Software* (2018), 10.21105/joss.00602.
 - ⁶⁶ S. L. Brunton and J. N. Kutz, “Singular value decomposition (svd),” in *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2019) p. 3–46.
 - ⁶⁷ L. N. Trefethen and D. Bau, *Numerical Linear Algebra* (SIAM, 1997).
 - ⁶⁸ C. Offen, “Software: Release v1.0 of GitHub repository Christian-Offen/DLNN_pde. identifier: Christian-Offen/DLNN_pde: DLNN_pde, publisher: Zenodo,” (2023).
 - ⁶⁹ P. Deuffhard and A. Hohmann, “Linear eigenvalue problems,” in *Numerical Analysis in Modern Scientific Computing: An Introduction* (Springer New York, New York, NY, 2003) pp. 119–150.
 - ⁷⁰ M. Röhrig-Zöllner, J. Thies, and A. Basermann, “Performance of the low-rank TT-SVD for large dense tensors on modern multicore CPUs,” *SIAM Journal on Scientific Computing* **44**, C287–C309 (2022).
 - ⁷¹ P. Deuffhard and A. Hohmann, *Numerical Analysis in Modern Scientific Computing* (Springer New York, 2003).