α-separable graph Hamiltonian network: A robust model for learning particle interactions in lattice systems

Yixian Gao,[1,*] Ru Geng,[2,1,*] Panayotis Kevrekidis,[3] Hong-Kun Zhang,[4,3,†] and Jian Zu[1,‡]

[1] *Center for Mathematics and Interdisciplinary Sciences, School of Mathematics and Statistics,*
Northeast Normal University, Changchun 130024, People's Republic of China
[2] *School of Mathematics and Statistics, Changchun University, Changchun 130022, People's Republic of China*
[3] *Department of Mathematics and Statistics, University of Massachusetts, Amherst, Massachusetts 01003, USA*
[4] *School of Natural Science, Great Bay University, Guangdong 523000, People's Republic of China*

(Received 18 July 2024; accepted 13 January 2025; published 31 January 2025)

We propose an α-separable graph Hamiltonian network (α-SGHN) that reveals complex interaction patterns between particles in lattice systems. Utilizing trajectory data, α-SGHN infers potential interactions without prior knowledge about particle coupling, overcoming the limitations of traditional graph neural networks that require predefined links. Furthermore, α-SGHN preserves all conservation laws during trajectory prediction. Experimental results demonstrate that our model, incorporating structural information, outperforms baseline models based on conventional neural networks in predicting lattice systems. We anticipate that the results presented will be applicable beyond the specific on-site and intersite interaction lattices studied, including the Frenkel-Kontorova model, the rotator lattice, and the Toda lattice.

DOI: 10.1103/PhysRevE.111.015309

I. INTRODUCTION

The impact of lattice systems on the discoveries and technological progress of condensed matter physics, materials science, biochemistry, medicine, and other scientific fields is profound. For example, Fermi, Pasta, Ulam, and Tsingou attempted to study the thermalization process through a prototypical nonlinear lattice [1]. The Frenkel-Kontorova system is commonly used to study heat conduction theory [2], DNA dynamics [3–6], lattice defects/dislocations and crowdions [3,7,8], hydrogen-bonded chains [9], and so forth; see also [10]. The ϕ^4 system describes possible domain walls in early universe models in cosmology [11], studies structural phase transitions in the displacive limit in condensed matter physics [12–14], serves as a phenomenological model for nonlinear excitations with exotic spin-charge relations [15,16], and so forth; a recent recap of the relevant model and applications can be found in [17].

These types of lattice systems can be generally described by the Hamiltonian

$$\frac{d}{dt}\begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} = \mathbf{J}\begin{pmatrix}\frac{\partial H}{\partial \mathbf{q}}\\ \frac{\partial H}{\partial \mathbf{p}}\end{pmatrix}, \quad \mathbf{J} = \begin{pmatrix}\mathbf{O} & \mathbf{I}\\ -\mathbf{I} & \mathbf{O}\end{pmatrix}, \tag{1}$$

where $\mathbf{q} = (q_1, \dots, q_N)$ and $\mathbf{p} = (p_1, \dots, p_N)$ are the generalized positions and momenta, respectively, of the system of pointlike particles; N is the number of pointlike particles in the system. $\mathbf{I}$ represents the identity matrix. Here, $H = T + V$ denotes the Hamiltonian, with kinetic energy

$$T(\mathbf{p}) = \sum_{i=1}^{N} T(p_i) = \sum_{i=1}^{N} \frac{p_i^2}{2m}$$

and potential

$$V(\mathbf{q}) = \sum_{i=1}^{N} (V^{(1)}(q_{i+1} - q_i) + V^{(2)}(q_i)). \tag{2}$$

The intersite potential $V^{(1)}$ encapsulates particle interactions within the nearest neighbor, while the on-site potential $V^{(2)}$ accounts for potential interactions with an external environment, such as a substrate. Additionally, m denotes the mass of the pointlike particle, and we set $m = 1$ for simplicity.

Finding solutions to the differential equations based only on the observed trajectories is an important task in scientific fields. The emergence of artificial neural networks represents a revolutionary development in the analysis and interpretation of complex data. Energy interactions between paired or multiple particles in lattice systems can lead to interesting phenomena in the system's behavior, such as phase transitions, phonon vibrations, magnetic behavior, and soliton formation [10,17,18]. In the specific context of lattice systems, machine learning methods, especially deep learning, have made significant progress in describing governing equations, identifying phase transitions, and constructing predictive models [19–26]. Although they are useful, the methods relying on conventional neural networks still present challenges. Some require knowledge or understanding of system equations [24,25], while others necessitate the creation of an overcomplete operator library [19,21,22].

On the other hand, data that exhibit relationships between elements can be modeled as a graph. Elements are named

[*] These authors contributed equally to this work.
[†] Contact author: hongkun@math.umass.edu
[‡] Contact author: zuj100@nenu.edu.cn

TABLE I. Network performance comparison. α-SGHN has the least input conditions and the most widely applicable prediction content.

	Additional information required	Interaction prediction	Applicable scenarios
MLP		No	General
HNN		No	Widely applicable
HOGN	Particle interaction relationship	No	Specific (e.g., even symmetry system)
HGNN	Particle interaction relationship	No	Specific (e.g., even symmetry system)
α-SGHN (ours)		Yes	Widely applicable

graph nodes, and relationships are represented by edges. A graph can naturally serve as a comprehensive representation of a lattice system, where the particles of the system correspond to the nodes of the graph, and the interactions between particles are represented as edges of the graph. Therefore, graph neural networks (GNNs) have become a new field for studying lattice systems. When dealing with lattice systems involving many degrees of freedom and complex interactions, compared to conventional neural networks, GNNs exhibit a proficient ability to accurately identify key information around nodes, which ultimately improves the accuracy of the model. The outstanding performance of graph neural networks in some classic systems can be observed, for example, in the works cited in [27,28]. An important advantage of graph neural networks is that they encode the interaction relationships of the system (i.e., the underlying structure of the system). In some special cases, lattice systems such as gravitationally interacting celestial bodies can be modeled using fully connected graphs, due to interactions existing between all particles. The interactions in most lattice problems are complex and diverse, and may exhibit irregularities [29] or even demonstrate long-range interactions [30,31], and cannot be observed in advance. This complexity makes visual recognition of which particles interact (i.e., between which a link exists) challenging. Therefore, although methods based on graph neural networks are effective, they require more input information, especially the linking of system particles, compared to traditional neural networks.

To address these challenges, based on the previous work of a subgroup of the present authors [32], we propose the α-separable graph Hamiltonian network (α-SGHN) model, which can extract the underlying particle link (associated with each particle's interactions) of the lattice system based solely on its motion trajectory facilitating subsequent system predictions and other applications. To our knowledge, this is a prototypical example that can provide the underlying particle link of a lattice Hamiltonian system solely based on its trajectories. Table I compares our method with established models such as multilayer perceptrons (MLPs) [33] and Hamiltonian neural networks (HNNs) [34], Hamiltonian ODE graph networks (HOGNs) [27], and Hamiltonian graph neural networks (HGNNs) [28]. Among them, HOGNs and HGNNs are based on graph methods. Our method only requires trajectory data, providing the most widely applicable predictive content. See Sec. III for specific comparison. Considering that physics-informed neural networks (PINNs) and symbolic regression methods require prior knowledge of the governing equations, they are not included in our comparison.

Our model is divided into two parts. The first part focuses on learning the underlying particle interactions (links) in the system, while the second part utilizes this graph structure (including link information) for more accurate prediction, as shown in Fig. 1. In addition, we investigate whether the particle behavior predicted by α-SGHN preserves the system's conservation laws. The experiment shows that due to the embedding of particle interaction information in the GNN, it outperforms the baseline model based on conventional neural networks. In the Appendix, we also explore the predictive effect of α-SGHN on long-range interactions and high-dimensional system structures.

The main contributions of our article are as follows:

(1) Under the assumption of only knowing the trajectory of the system, our model learns the underlying interactions of the system particles. This overcomes the limitation of GNN applications, which require typically prior knowledge of graph structure information.

(2) Our model ensures maintenance of the system's conservation laws during trajectory prediction.

(3) We provide an explanation of why we anticipate that graph neural network models will generally outperform traditional neural network models when dealing with lattice systems.

(4) We provide a method for modeling directed graphs in learning non-even symmetric potential energy systems.

Our presentation is structured as follows. In Sec. II we present our methodology. Our numerical experiments are analyzed in detail in Sec. III. In Sec. IV we make a brief excursion to the intriguing realm of complete integrability (in lattices). Finally, in Sec. V we summarize our findings and present our conclusions, as well as some interesting directions for further work.

II. METHOD

Consider a dataset with N elements. When there are some relationships between these elements, it can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1 \cdots v_N\}$ being the set of nodes and $\mathcal{E} = \{e_{ij}\}$ the set of edges between nodes. Each data element corresponds to node v_i, and each edge $e_{i,j}$ represents a directed edge originating from node v_i and ending at node v_j reflecting the link between v_i and v_j. Obviously, the lattice system can be modeled as a graph, with particles corresponding to nodes and the interactions between particles implying the existence of edges between them. In this work, we are interested in the fact that we only have trajectory data $(\mathbf{q}^t, \mathbf{p}^t), t = 1, \dots, T$, for N lattice particles, and the interaction relationships between particles are *unknown*.

Our method is based on the previous work of [32], but the difference lies in that in this work, we can learn the interaction relationship (edge link) between particles in the lattice

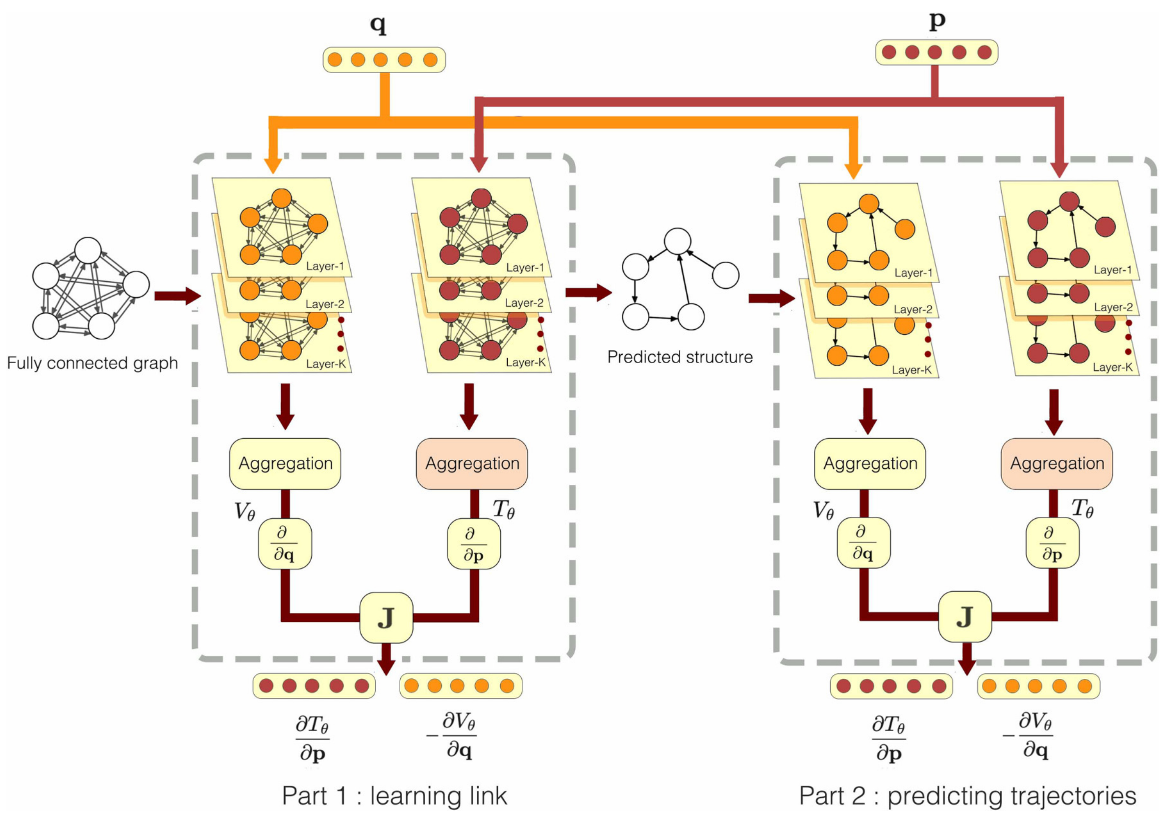
$\mathbf{q}$
$\mathbf{p}$
Fully connected graph
Layer-1
Layer-2
Layer-K
Layer-1
Layer-2
Layer-K
Predicted structure
Layer-1
Layer-2
Layer-K
Layer-1
Layer-2
Layer-K
Aggregation
Aggregation
Aggregation
Aggregation
V_θ
T_θ
V_θ
T_θ
$\frac{\partial}{\partial \mathbf{q}}$
$\frac{\partial}{\partial \mathbf{p}}$
$\frac{\partial}{\partial \mathbf{q}}$
$\frac{\partial}{\partial \mathbf{p}}$
$\mathbf{J}$
$\mathbf{J}$
$\frac{\partial T_\theta}{\partial \mathbf{p}}$
$-\frac{\partial V_\theta}{\partial \mathbf{q}}$
$\frac{\partial T_\theta}{\partial \mathbf{p}}$
$-\frac{\partial V_\theta}{\partial \mathbf{q}}$
Part 1 : learning link
Part 2 : predicting trajectories

FIG. 1. The architecture of α-SGHN.

system. Since we do not have any prior knowledge about the particle interaction relationship in lattice systems, we model the trajectory data observed at each time as a directed fully connected graph without self-loops, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; see Fig. 1(a). We first assume that each particle has interactions with other particles. Then, by learning the α-SGHN model, the true interaction relationships are preserved to obtain the underlying structure of the system which is denoted as $\mathcal{G}_\alpha = (\mathcal{V}, \mathcal{E}_\alpha)$.

Our method consists of two parts. The first part is to adjust the weight of the edge in the fully connected graph through trajectory data to extract the underlying interaction between particles. The second part utilizes the learned underlying interactions for more accurate and effective trajectory prediction. Next, we provide a detailed introduction to our model.

A. Part 1: Structural learning

Our network model consists of two components, one learning the potential energy at each sampling time named V net, and the other learning the kinetic energy at each sampling time named T net.

***V** net.* The V net is the parametrization of the potential energy, V_θ, within the lattice system. In the graph $\mathcal{G}$, for each sample time t, each node and edge is tied to a node feature vector, $\mathbf{h}_i^{(0)}$, and an edge feature vector, $\mathbf{e}_{i,j}$, respectively. To kickstart the process, the position q_i is utilized as the initial feature of node v_i within the lattice graph, thereby defining $\mathbf{h}_i^{(0)} = q_i$. In an adaptive fashion, edge features are learned through neural networks as follows:

$$\mathbf{e}_{i,j}^{(0)} = \mathcal{F}_e(q_i - q_j). \tag{3}$$

Here, $\mathcal{F}_e$ is a function approximated by neural networks to enhance the expression of edge features.

Further, we prescribe the K-layer node update and edge update operations as follows:

$$\mathbf{h}_i^{(k+1)} = \sum_{v_j \in \mathcal{N}_\mathcal{G}(i)} \alpha_{i,j} \mathcal{F}_{nu}^{(k)}\big(\mathbf{e}_{i,j}^{(k)}\big), \tag{4}$$

$$\mathbf{e}_{i,j}^{(k+1)} = 1/S \sum_{s=1}^{S} \Big(\mathbf{e}_{i,j}^{(k)} + \mathcal{F}_{eu}^{(k,s)}\big(\mathbf{h}_i^{(k)} \odot \mathbf{h}_j^{(k)}\big)\Big). \tag{5}$$

In these expressions, $\mathcal{F}_{nu}^{(k)}$ and $\mathcal{F}_{eu}^{(k,s)}$, with $0 \leqslant k \leqslant K-1$, are node update and edge update functions approximated by neural networks, respectively. k represents the kth layer of the graph neural network and the maximum number of layers is K. $\mathcal{N}_\mathcal{G}(i)$ indicates the neighbor of v_i, and $\odot$ signifies the Hadamard product used to assess node interactions. $\alpha = \{\alpha_{i,j}\}$ is the key to learning graph structure in the α-SGHN model. It is a parameter matrix initialized by a neural network that acts on each edge and adaptively labels the strength of the edges through the training process.

Subsequently, we amalgamate the node and edge information to depict the final node features:

$$\mathbf{a}_i = \mathcal{F}_a\big(\mathcal{F}_{v_1}\big(\mathbf{h}_i^{(0)}\big) \parallel \mathcal{F}_{v_2}\big(\mathbf{h}_i^{(K)}\big) \parallel \mathbf{x}_i\big), \tag{6}$$

where

$$\mathbf{x}_i = \sum_{v_j \in \mathcal{N}_\mathcal{G}(i)} \alpha_{i,j} \mathcal{F}_{v_3}\big(\mathbf{e}_{ij}^{(K)}\big). \tag{7}$$

$\mathcal{F}_{v_1}$ is used to enhance the expression of initial node features in the network, $\mathcal{F}_{v_2}$ is used to enhance the expression of node features in the last layer of the network, and $\mathcal{F}_{v_3}$ is used to enhance the expression of edge features in the last layer of the network. $\mathcal{F}_a$ is used to aggregate these features. They are all functions approximated by neural networks. $\|$ represents the concatenation operation.

Finally, we use a function approximated by neural networks to aggregate the features of all nodes and output a real number,

$$\hat{V} = \mathcal{F}_v\{\mathbf{a}_i | v_i \in \mathcal{V}\}, \tag{8}$$

where $\hat{V} \in \mathbb{R}$ is the approximate value of potential energy by the neural network model V_θ.

***T** net.* This involves the parametrization of the kinetic energy, T_θ. We define the node feature vector as $\hat{\mathbf{h}}_i^{(0)} = p_i$ for each time t. Then, we use K-layer node update operations as follows:

$$\hat{\mathbf{h}}_i^{(k+1)} = \mathcal{F}_{ut}^{(k)}\big(\hat{\mathbf{h}}_i^{(k)}\big). \tag{9}$$

Finally, we obtain the kinetic energy of the system by aggregating node features through addition:

$$\hat{T} = \sum_{v_i \in \mathcal{V}} \hat{\mathbf{h}}_i^{(K)}. \tag{10}$$

Loss function. The loss function of the first part is defined as

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \gamma \mathcal{L}_{GL}, \tag{11}$$

where

$$\mathcal{L}_{\text{pred}} = \left\| \frac{\partial T_\theta}{\partial \mathbf{p}} - \frac{d\mathbf{q}}{dt} \right\|_2 + \left\| -\frac{\partial V_\theta}{\partial \mathbf{q}} - \frac{d\mathbf{p}}{dt} \right\|_2. \tag{12}$$

This loss component is to make the output of the α-SGHN approximate the energy of the system. $\frac{d\mathbf{q}}{dt}$ and $\frac{d\mathbf{p}}{dt}$ are targets. Where possible, we use analytic time derivatives as targets. Otherwise, we calculate finite difference approximations. Notice that this selection of the loss function assumes the decomposition of the system's energy into T_θ and V_θ, presupposing the former as being a function of $\mathbf{p}$ and the latter as being a function of $\mathbf{q}$.

The graph learning loss $\mathcal{L}_{GL}$ is designed to prevent the model from only recognizing the nearest neighbors, specifically represented as

$$\mathcal{L}_{GL} = \|\alpha\|_F^2, \tag{13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. $\gamma \in \mathbb{R}$ is a hyperparameter that balances the two terms.

Link extraction. By reducing the loss function, if there is interaction between particles v_i and v_j, the corresponding $|\alpha_{i,j}|$ value is relatively large; otherwise, the $|\alpha_{i,j}|$ is set to zero. According to these rules, we can extract the link relationship of particles in the lattice system. Through linking relationships, we can get a new directed graph $\mathcal{G}_\alpha = \{\mathcal{V}, \mathcal{E}_\alpha\}$ for more accurate prediction in the next part. We take the direction with maximum value between $|\alpha_{i,j}|$ and $|\alpha_{j,i}|$ as the direction of the edge.

B. Part 2: Prediction

We repeat part 1 and replace $\mathcal{G}$ with $\mathcal{G}_\alpha$, where the following two formulas in V net need to be modified.

(1) Replace (4) with

$$\mathbf{h}_i^{(k+1)} = \sum_{v_j \in \mathcal{N}_{\mathcal{G}_\alpha}(i)} \mathcal{F}_{un}^{(k)}\big(\mathbf{e}_{i,j}^{(k)}\big). \tag{14}$$

(2) Replace (7) with

$$\mathbf{x}_i = \sum_{v_j \in \mathcal{N}_{\mathcal{G}_\alpha}(i)} \mathcal{F}_{v_3}\big(\mathbf{e}_{ij}^{(K)}\big). \tag{15}$$

The loss function of the second part is defined as $\mathcal{L} = \mathcal{L}_{\text{pred}}$. Figure 1 shows the architecture of α-SGHN.

After training, in order to obtain the predicted trajectory $(\hat{\mathbf{q}}^t, \hat{\mathbf{p}}^t)$, we integrate neural network models according to

$$(\hat{\mathbf{q}}^t, \hat{\mathbf{p}}^t) = (\mathbf{q}^0, \mathbf{p}^0) + \int_{t_0}^{t} \alpha\text{-SGHN}dt, \tag{16}$$

where $(\mathbf{q}^0, \mathbf{p}^0)$ represents any initial value.

III. EXPERIMENT

In this section, we use three test examples: the Frenkel-Kontorova lattice (one conserved quantity), the rotator lattice (two conserved quantities), and the Toda lattice (multiple—i.e., as many as the degrees of freedom—conserved quantities). To ascertain the effectiveness of our proposed model's prediction performance, we conduct experiments comparing α-SGHN with established network structures such as the MLP and the HNN. Due to our assumption that the particle linkage of the system is unknown, i.e., we do not know the graph structure of the system, the existing graph neural network models cannot serve as our primary baseline model as they all require knowledge of the exact particle link. But we also present the results of learning HOGN and HGNN using fully connected graphs (assuming that there are interactions between all particles) in Table III. We will verify through experiments that our graph neural network model α-SGHN can effectively learn multiple conserved quantities due to its ability to learn potential particle linkage relationships in the system, and is superior to conventional neural network models.

A. Lattice system

1. Frenkel-Kontorova lattice

The Frenkel-Kontorova (FK) model was introduced by Frenkel and Kontorova as a prototypical way to describe the structure and dynamics of a crystalline lattice near a dislocation. In solid-state physics, the model is, arguably, one of the first instances where a two-dimensional bulk defect is represented by a straightforward one-dimensional chain [35]. The characteristics of the FK model provide deep physical insights and significantly simplify the understanding of nonlinear dynamics in various problems of solid-state physics, classical mechanics, and biophysics, among many others [3,10]. Its Hamiltonian is given by

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + \frac{(q_{i+1} - q_i)^2}{2} + 1 - \cos(q_i) \right). \tag{17}$$

TABLE II. Piecewise learning rate.

	FK	Rotator	Toda-3	Toda-4	Toda-5
MLP-gelu	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}
MLP-silu	10^{-2}, 5×10^{-3}, 10^{-4}	10^{-3}, 10^{-4}, 10^{-5}	10^{-2}, 5×10^{-3}, 10^{-4}	10^{-2}, 5×10^{-3}, 10^{-4}	10^{-2}, 5×10^{-3}, 10^{-4}
MLP-tanh	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}
HNN-gelu	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}
HNN-silu	10^{-2}, 5×10^{-3}, 10^{-4}	10^{-3}, 10^{-4}, 10^{-5}	10^{-2}, 5×10^{-3}, 10^{-4}	10^{-2}, 5×10^{-3}, 10^{-4}	10^{-2}, 5×10^{-3}, 10^{-4}
HNN-tanh	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}
α-SGHN	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}	10^{-3}, 10^{-4}, 10^{-5}

2. *Rotator lattice*

The rotator lattice is, arguably, one of the simplest examples of a classical spin 1D model with nearest-neighbor interactions, and has a potential function governed by $1 - \cos(q_{i+1} - q_i)$. This model can also be viewed as an array of N-coupled pendulums. As an example of a chaotic dynamical system, when it is reduced to a harmonic chain and free rotator, it is integrable in both small and high energy limits and has been widely studied [36].

The Hamiltonian for the coupled rotator lattice dynamics amended with harmonic interactions reads [37]

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + \frac{(q_{i+1} - q_i)^2}{2} + 1 - \cos(q_{i+1} - q_i) \right). \quad (18)$$

It has two conserved quantities of energy and momentum, with the latter being equal to

$$P = \sum_{i=1}^{N} p_i = \sum_{i=1}^{N} \dot{q}_i.$$

3. *Toda lattice*

The Toda lattice model [38–41] has garnered significant attention in the physical sciences as a paradigm of a nonlinear lattice system where specific (solitonic) wave forms are known to propagate unaltered in shape, a property rigorously demonstrated in the literature [42]. The system's dynamics are governed by the Hamiltonian

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + \exp(q_i - q_{i+1}) \right). \quad (19)$$

This system is known to feature N constants of motion [43]. Under periodic boundary conditions (BCs), i.e., $q_{i+N} = q_i$ and $p_{i+N} = p_i$, the $2N$ phase-space coordinates can be used to define a symmetric, periodic tridiagonal $N \times N$ matrix (time dependent),

$$L = \begin{pmatrix} p_1 & v_1 & 0 & \cdots & v_N \\ v_1 & p_2 & v_2 & \cdots & 0 \\ 0 & v_2 & p_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ v_N & 0 & \cdots & v_{N-1} & p_N \end{pmatrix},$$

where $v_i = -e^{(q_i - q_{i+1})/2}$. All the quantities

$$\mathcal{C}_n \equiv \sum_{i=1}^{\mathcal{N}} \lambda_i^n = \mathrm{Tr}\, L^n$$

then represent the constants of the motion; see [43] for details. The first two are familiar and physically meaningful; namely, they represent the system's momentum and energy.

B. Dataset acquisition

For the FK and rotator lattices, we set $N = 32$. For the Toda lattice, we set $N = 3$, 4, and 5 to verify the conservation laws

TABLE III. The test results for FK system with $N = 32$. The best results are emphasized by bold font.

	Test loss	Trajectory$_{\mathrm{MSE}}$	Energy$_{\mathrm{MSE}}$
MLP-1-100-silu	$3.70 \times 10^{-3} \pm 1.20 \times 10^{-3}$	$1.32 \times 10^{-1} \pm 1.43 \times 10^{-1}$	$1.56 \times 10^{0} \pm 1.83 \times 10^{0}$
MLP-1-100-tanh	$2.36 \times 10^{-3} \pm 8.52 \times 10^{-4}$	$8.81 \times 10^{-2} \pm 1.04 \times 10^{-1}$	$3.70 \times 10^{-1} \pm 3.55 \times 10^{-1}$
MLP-1-50-gelu	$3.31 \times 10^{-2} \pm 5.95 \times 10^{-3}$	$7.91 \times 10^{-1} \pm 8.00 \times 10^{-1}$	$9.57 \times 10^{1} \pm 1.90 \times 10^{2}$
MLP-2-100-gelu	$3.58 \times 10^{-3} \pm 9.57 \times 10^{-4}$	$7.30 \times 10^{-2} \pm 7.45 \times 10^{-2}$	$4.18 \times 10^{-1} \pm 3.61 \times 10^{-1}$
MLP-1-200-gelu	$1.89 \times 10^{-2} \pm 4.72 \times 10^{-3}$	$6.15 \times 10^{-1} \pm 7.14 \times 10^{-1}$	$1.70 \times 10^{2} \pm 2.79 \times 10^{2}$
MLP-1-100-gelu	$2.06 \times 10^{-3} \pm 6.09 \times 10^{-4}$	$3.91 \times 10^{-2} \pm 3.72 \times 10^{-2}$	$1.33 \times 10^{-1} \pm 1.17 \times 10^{-1}$
HNN-1-100-silu	$1.91 \times 10^{-3} \pm 5.73 \times 10^{-4}$	$4.57 \times 10^{-2} \pm 4.27 \times 10^{-2}$	$6.76 \times 10^{-3} \pm 4.64 \times 10^{-3}$
HNN-1-100-tanh	$1.66 \times 10^{-3} \pm 6.79 \times 10^{-4}$	$5.06 \times 10^{-2} \pm 5.40 \times 10^{-2}$	$9.34 \times 10^{-3} \pm 1.21 \times 10^{-2}$
HNN-1-50-gelu	$3.68 \times 10^{-2} \pm 6.57 \times 10^{-3}$	$7.62 \times 10^{-1} \pm 9.03 \times 10^{-1}$	$2.53 \times 10^{2} \pm 5.30 \times 10^{2}$
HNN-2-100-gelu	$9.06 \times 10^{-4} \pm 2.90 \times 10^{-4}$	$1.41 \times 10^{-2} \pm 1.31 \times 10^{-2}$	$3.29 \times 10^{-3} \pm 2.90 \times 10^{-3}$
HNN-1-200-gelu	$1.43 \times 10^{-3} \pm 3.99 \times 10^{-4}$	$3.20 \times 10^{-2} \pm 3.08 \times 10^{-2}$	$4.61 \times 10^{-3} \pm 3.36 \times 10^{-3}$
HNN-1-100-gelu	$7.44 \times 10^{-4} \pm 2.62 \times 10^{-4}$	$1.36 \times 10^{-2} \pm 1.29 \times 10^{-2}$	$2.21 \times 10^{-3} \pm 1.65 \times 10^{-3}$
α-SGHN (ours)	$\mathbf{4.26 \times 10^{-7} \pm 1.22 \times 10^{-6}}$	$\mathbf{4.85 \times 10^{-6} \pm 2.37 \times 10^{-5}}$	$\mathbf{9.70 \times 10^{-7} \pm 3.13 \times 10^{-6}}$

TABLE IV. The test results for rotator system with $N = 32$. The best results are emphasized by bold font.

	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$	$\text{Momentum}_{\text{MSE}}$
MLP-1-100-gelu	$1.00 \times 10^{-3} \pm 3.94 \times 10^{-4}$	$1.93 \times 10^{-2} \pm 3.20 \times 10^{-2}$	$7.00 \times 10^{-2} \pm 7.27 \times 10^{-2}$	$1.03 \times 10^{-2} \pm 1.03 \times 10^{-2}$
MLP-1-100-tanh	$9.90 \times 10^{-4} \pm 4.67 \times 10^{-4}$	$4.30 \times 10^{-2} \pm 6.24 \times 10^{-2}$	$5.22 \times 10^{-2} \pm 5.09 \times 10^{-2}$	$8.28 \times 10^{-6} \pm 9.70 \times 10^{-6}$
MLP-1-50-silu	$1.36 \times 10^{-2} \pm 3.68 \times 10^{-3}$	$6.01 \times 10^{-1} \pm 5.82 \times 10^{-1}$	$9.52 \times 10^{1} \pm 1.57 \times 10^{2}$	$8.06 \times 10^{-5} \pm 1.41 \times 10^{-5}$
MLP-2-100-silu	$1.31 \times 10^{-3} \pm 4.91 \times 10^{-4}$	$1.53 \times 10^{-2} \pm 2.68 \times 10^{-2}$	$7.49 \times 10^{-2} \pm 8.44 \times 10^{-2}$	$2.56 \times 10^{-2} \pm 2.33 \times 10^{-2}$
MLP-1-200-silu	$1.11 \times 10^{-2} \pm 2.32 \times 10^{-3}$	$2.35 \times 10^{-1} \pm 2.72 \times 10^{-1}$	$1.76 \times 10^{1} \pm 1.73 \times 10^{1}$	$7.48 \times 10^{-5} \pm 9.21 \times 10^{-5}$
MLP-1-100-silu	$6.65 \times 10^{-4} \pm 1.94 \times 10^{-4}$	$6.14 \times 10^{-3} \pm 8.84 \times 10^{-3}$	$3.15 \times 10^{-2} \pm 2.86 \times 10^{-2}$	$3.95 \times 10^{-2} \pm 2.58 \times 10^{-2}$
HNN-1-100-silu	$8.61 \times 10^{-4} \pm 2.25 \times 10^{-4}$	$1.43 \times 10^{-2} \pm 1.90 \times 10^{-2}$	$2.84 \times 10^{-2} \pm 1.46 \times 10^{-2}$	$1.17 \times 10^{-1} \pm 3.65 \times 10^{-2}$
HNN-1-100-tanh	$1.47 \times 10^{-3} \pm 6.57 \times 10^{-4}$	$4.76 \times 10^{-0} \pm 2.94 \times 10^{1}$	$9.50 \times 10^{4} \pm 4.14 \times 10^{5}$	$1.46 \times 10^{1} \pm 6.35 \times 10^{1}$
HNN-1-50-gelu	$1.61 \times 10^{-2} \pm 4.03 \times 10^{-3}$	$2.63 \times 10^{1} \pm 1.60 \times 10^{2}$	$7.03 \times 10^{6} \pm 3.07 \times 10^{7}$	$2.74 \times 10^{-1} \pm 3.79 \times 10^{-2}$
HNN-2-100-gelu	$4.78 \times 10^{-3} \pm 1.14 \times 10^{-3}$	$9.38 \times 10^{-2} \pm 1.60 \times 10^{-1}$	$1.47 \times 10^{-1} \pm 6.84 \times 10^{-2}$	$7.25 \times 10^{-1} \pm 2.65 \times 10^{-1}$
HNN-1-200-gelu	$4.83 \times 10^{-3} \pm 1.17 \times 10^{-3}$	$1.26 \times 10^{-1} \pm 2.28 \times 10^{-1}$	$3.33 \times 10^{-1} \pm 1.18 \times 10^{-1}$	$1.68 \times 10^{-0} \pm 3.58 \times 10^{-1}$
HNN-1-100-gelu	$8.08 \times 10^{-4} \pm 2.60 \times 10^{-4}$	$1.55 \times 10^{-2} \pm 1.94 \times 10^{-2}$	$1.92 \times 10^{-2} \pm 1.05 \times 10^{-2}$	$7.38 \times 10^{-2} \pm 3.26 \times 10^{-2}$
α-SGHN(ours)	$\mathbf{4.71 \times 10^{-9} \pm 4.26 \times 10^{-9}}$	$\mathbf{1.97 \times 10^{-7} \pm 3.57 \times 10^{-7}}$	$\mathbf{1.32 \times 10^{-6} \pm 4.05 \times 10^{-7}}$	$\mathbf{4.25 \times 10^{-6} \pm 8.99 \times 10^{-7}}$

of the system in the cases of 3, 4, and 5 conserved quantities, respectively. We use the explicit, symplectic Runge-Kutta-Nyström algorithms (Sec. 8.5.3 in [44]) with 5 time units and a time step 0.0025 to find 50 trajectories. The initial conditions are

$$\mathbf{q}_0(i) \sim \mathcal{U}(0, 1), \quad i = 1, \dots, N, \tag{20}$$

$$\mathbf{p}_0(i) \sim \mathcal{U}(0, 1), \quad i = 1, \dots, N, \tag{21}$$

where $\mathcal{U}$ represents the uniform distribution. Their BCs are periodic, i.e., $q_{i+N} = q_i$ and $p_{i+N} = p_i$. We subsample the trajectories at a fixed time step of 0.05 as the training set.

We use the same method to generate 20 trajectories as the test set, where the integration time is three times that of the training set, i.e., 15 time units.

C. Network model settings

For the baseline model MLP and HNN, we carefully select the hyperparameters of the model to achieve the best predictive performance. We start from a layer of 200 hidden units, continuously adjusting the network width and depth, and replacing different activation functions tanh, SiLU, and GELU until the network performance reaches the level we estimated as optimal. For the α-SGHN model, Eq. (9) represents the feedforward neural network with 2 hidden layers, each containing 10 hidden units. Equation (8) denotes the feedforward neural network with 2 hidden layers, each having 60 hidden units. The neural network parametrization functions in Eqs. (3) to (6) are all represented by the feedforward neural network, each with 2 hidden layers and 5 units per layer. The activation functions used in this networks are all SiLU. In Eq. (6) and Eq. (10), the layers for node and edge updates are $K = 1$. $S = 2$ in Eq. (4). $\gamma = 0.05$ in Eq. (11).

We adopt a learning rate piecewise constant decay strategy [45]; see Table II. The piecewise learning rates are adjusted at 3500 and 5000 epoch points. We record the loss function during the training process to ensure the steady convergence. The total epoch for all model training is set to 10 000. The optimizer is Adam, and the batch size is set to 256. Through experiments we find that α-SGHN is far superior to the baseline model's optimal performance.

D. Test metrics

To evaluate our model in the test set, we log the following metrics: testing loss, the mean square error (MSE) of the predicted trajectories, and the MSE of the predicted conserved quantities in the system.

To determine the predicted conserved quantities and trajectories metric over long time spans, we integrate neural network models according to (16) by explicit, symplectic Runge-Kutta-Nyström algorithms with a time step 0.0025. $(\mathbf{q}^0, \mathbf{p}^0)$ is the initial value in the test set, and the initial time is set to 0 s. While we are cognizant of the interesting recent work of [46] which suggests that even such symplectic integrators can yield adverse (integrability or conservation law-breaking features), this is only relevant for times much longer (and time steps much bigger) than considered herein.

The MSE of the predicted trajectories is defined as

$$\text{MSE}_{\text{traj}} = \sum_{\text{i}=1}^{\text{N}} \left(\left(\text{q}_\text{i}^\text{t} - \hat{\text{q}}_\text{i}^\text{t} \right)^2 + \left(\text{p}_\text{i}^\text{t} - \hat{\text{p}}_\text{i}^\text{t} \right)^2 \right).$$

E. Link extraction

Figure 2(a) illustrates $|\alpha_{i,j}|$, where the x axis and y axis represent nodes, and the color indicates the strength of the interaction between nodes. We can define a hyperparameter as a threshold to obtain the interaction relationship between particles from Fig. 2(a). When $|\alpha_{i,j}|$ is less than this threshold, it indicates that there is no interaction between particles, and when $|\alpha_{i,j}|$ is greater than this threshold, it indicates that there is interaction between particles. In this paper, we set the threshold as $\frac{1}{2}(\max_{1 \leqslant i,j \leqslant N} |\alpha_{i,j}| + \min_{1 \leqslant i,j \leqslant N} |\alpha_{i,j}|)$. To ensure stability, if at least one of $|\alpha_{i,j}|$ and $|\alpha_{j,i}|$ is preserved by the threshold, we assume that there is an interaction relationship between i and j. The interaction (link) relationship obtained based on the threshold is shown in Fig. 2(b). Then, we apply a directed graph in the network model, where the direction of the edges is the direction of the $|\alpha_{i,j}|$ preserved by the threshold. If both $|\alpha_{i,j}|$ and $|\alpha_{j,i}|$ are preserved by the threshold, we will determine that the edges are directed toward the nodes with higher (or lower) $|\alpha_{i,j}|$ values, as shown in Fig. 2(c).

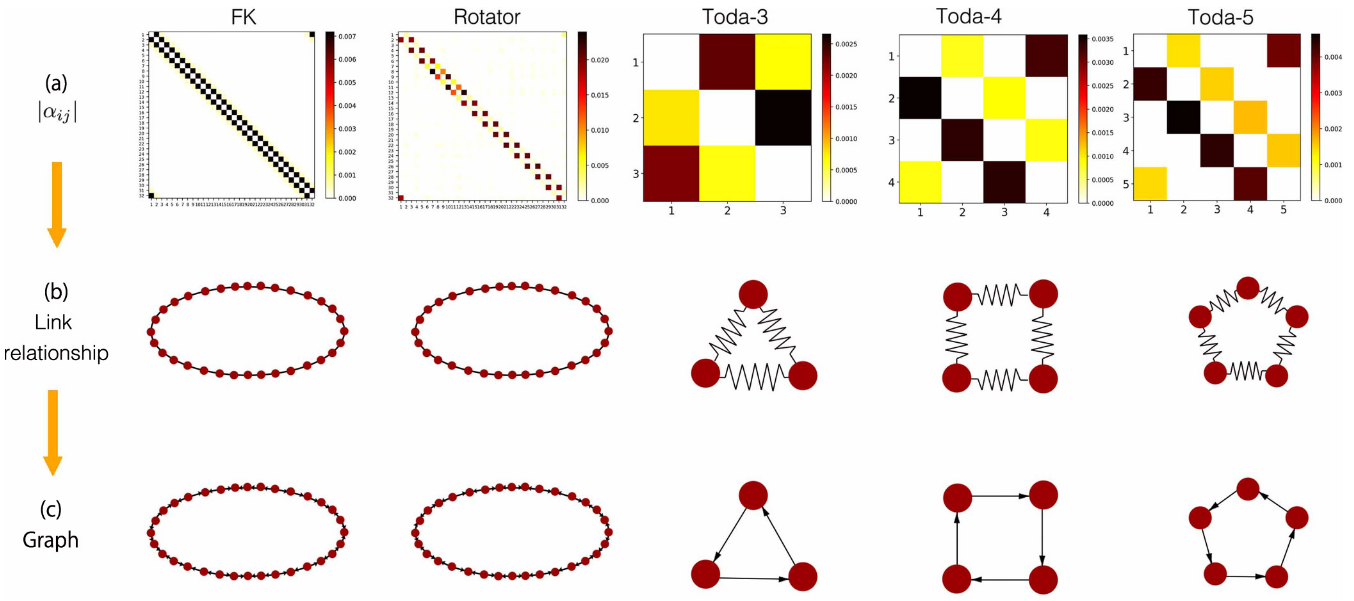

FIG. 2. (a). The x axis and y axis represent nodes, and the color-coded $|\alpha_{i,j}|$ values. (b) shows the particle link relationship extracted from (a). Obviously, this relationship satisfies the periodic boundary conditions. (c) is a directed graph constructed through (b), with edge directions corresponding to the maximum value between $|\alpha_{i,j}|$ and $|\alpha_{j,i}|$.

F. Prediction results

Table III compares the predictive performance of α-SGHN and baseline models for FK system. We trained the baseline model to its optimal performance and recorded the effects of changing the depth, width, and activation function of the network, which did not further improve the performance of the baseline model. It is evident that α-SGHN outperforms the baseline models considered herein in terms of testing loss, long-term trajectory prediction, and energy conservation properties. This is because the interior of the lattice system largely depends on the interactions between particles. Conventional neural networks can only process data for each particle equivalently, making it difficult to discover interactions between particles. α-SGHN can effectively predict the particle interactions within the system and learn the lattice system more effectively by utilizing the predicted interactions.

It can also be seen from Table IV that the performance of α-SGHN is still significantly superior to the baseline models we examined in the case of the rotator lattice. In this case, there exist two conservation laws and both of them are retrieved by numerous orders of magnitude better in the present setting rather than by the MLP or the HNN.

Table V shows the prediction comparison results of the Toda system with $N = 3$. Due to the fact that there are only three particles (in the setting of this table) and the interactions between particles are simple, the results of MLP and HNN training are not as poor as those of the FK and rotator systems with $N = 32$, and are indeed only slightly inferior to α-SGHN (e.g., lower only by 1 or 2 orders of magnitude with respect to α-SGHN as concerns the conservation laws). Compared to the FK and rotator models, when the number of particles is small, the width of the baseline model is correspondingly reduced to achieve the best effect. And an increase in the number of particles will greatly weaken the performance of the baseline model. However, based on graph models, the number of particles has little effect on the network model. This strongly suggests the scalability of the current approach to a larger scale conservative system and its comparative advantage in comparison to earlier methodologies.

Table VI shows the predicted comparison results of the Toda system with $N = 4$. α-SGHN once again outperforms the baseline models in terms of test loss and trajectory prediction. This Toda system has four conserved quantities, and, as can be seen, the baseline models' predictions for the third and fourth conserved quantities are progressively getting worse, while α-SGHN still performs better. This may be because α-SGHN retrieves the underlying structure of the system, making predictions for various quantities more stable. In an interesting outlier, in comparison to what was seen in earlier tables, for the momentum prediction, the MLP performs the best, but the corresponding result of α-SGHN is also satisfactory and not too far worse. Overall, however, the comparative advantage of α-SGHN when one views all the relevant diagnostics is demonstrable.

Table VII shows the predicted comparison results of the Toda system with $N = 5$. The system contains five conserved quantities, and it is evident that the baseline model cannot guarantee the conservation of the relevant quantities. This issue is especially prominent for the third, fourth, and fifth conserved quantities. Indeed, here we see a degradation of MLP and HNN the higher the relevant conservation law. α-SGHN predicts the interactions between system particles, making the learning of conserved quantities more stable. By the way, once again, the momentum conservation law is still not optimal in the case of the α-SGHN, yet the overall evidence is even more overwhelmingly in favor of our discovery of the network structure.

Figure 3 shows the evolution of the average true conservation law values and average predicted conservation values over time for 20 samples. The method used to obtain these

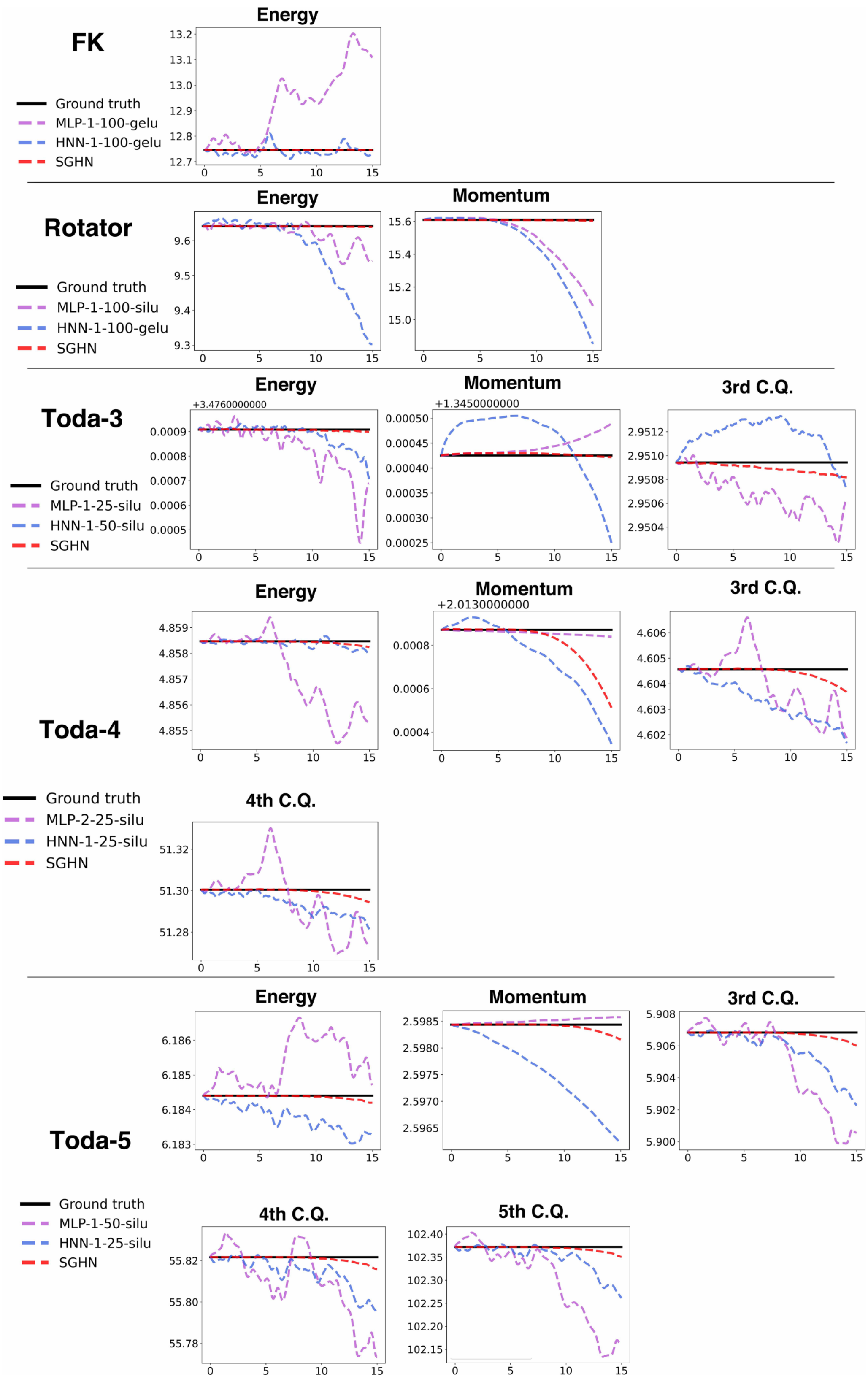

FIG. 3. The evolution over time of the average true conservation values and the average predicted conservation values of 20 samples. C.Q. represents conserved quantity.

TABLE V. The test results for Toda system with $N = 3$. The best results are emphasized by bold font.

	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$	$\text{Momentum}_{\text{MSE}}$	C3_{MSE}
MLP-1-25-gelu	$2.36 \times 10^{-4} \pm 2.93 \times 10^{-4}$	$3.00 \times 10^{-3} \pm 1.40 \times 10^{-2}$	$1.56 \times 10^{-3} \pm 2.80 \times 10^{-3}$	$3.14 \times 10^{-3} \pm 5.62 \times 10^{-3}$	$2.04 \times 10^{-2} \pm 3.85 \times 10^{-2}$
MLP-1-25-tanh	$3.06 \times 10^{-7} \pm 9.31 \times 10^{-7}$	$2.14 \times 10^{-5} \pm 8.34 \times 10^{-5}$	$1.46 \times 10^{-5} \pm 5.97 \times 10^{-5}$	$3.09 \times 10^{-6} \pm 4.35 \times 10^{-6}$	$3.32 \times 10^{-5} \pm 9.64 \times 10^{-5}$
MLP-1-10-silu	$1.02 \times 10^{-6} \pm 3.58 \times 10^{-6}$	$1.18 \times 10^{-5} \pm 4.14 \times 10^{-5}$	$5.57 \times 10^{-6} \pm 2.12 \times 10^{-5}$	$3.96 \times 10^{-9} \pm 4.40 \times 10^{-9}$	$4.05 \times 10^{-6} \pm 1.05 \times 10^{-5}$
MLP-2-25-silu	$1.58 \times 10^{-7} \pm 4.04 \times 10^{-7}$	$6.15 \times 10^{-6} \pm 2.95 \times 10^{-5}$	$2.28 \times 10^{-6} \pm 9.58 \times 10^{-6}$	$5.07 \times 10^{-10} \pm 1.03 \times 10^{-9}$	$4.58 \times 10^{-7} \pm 1.39 \times 10^{-6}$
MLP-1-50-silu	$4.35 \times 10^{-7} \pm 1.50 \times 10^{-6}$	$5.25 \times 10^{-6} \pm 2.27 \times 10^{-5}$	$1.02 \times 10^{-5} \pm 4.29 \times 10^{-5}$	$1.15 \times 10^{-8} \pm 1.65 \times 10^{-8}$	$4.61 \times 10^{-6} \pm 1.87 \times 10^{-5}$
MLP-1-25-silu	$2.53 \times 10^{-7} \pm 4.69 \times 10^{-7}$	$1.28 \times 10^{-6} \pm 1.69 \times 10^{-6}$	$6.27 \times 10^{-7} \pm 2.20 \times 10^{-6}$	$2.24 \times 10^{-9} \pm 3.85 \times 10^{-9}$	$1.67 \times 10^{-6} \pm 6.62 \times 10^{-6}$
HNN-1-50-gelu	$1.02 \times 10^{-4} \pm 1.47 \times 10^{-4}$	$2.91 \times 10^{-3} \pm 1.50 \times 10^{-2}$	$8.61 \times 10^{-4} \pm 1.58 \times 10^{-3}$	$1.96 \times 10^{-3} \pm 3.29 \times 10^{-3}$	$1.23 \times 10^{-2} \pm 2.06 \times 10^{-2}$
HNN-1-50-tanh	$5.88 \times 10^{-6} \pm 7.03 \times 10^{-6}$	$6.13 \times 10^{-5} \pm 1.83 \times 10^{-4}$	$2.20 \times 10^{-6} \pm 4.71 \times 10^{-6}$	$4.47 \times 10^{-6} \pm 8.55 \times 10^{-6}$	$4.72 \times 10^{-5} \pm 8.61 \times 10^{-5}$
HNN-1-25-silu	$8.78 \times 10^{-8} \pm 2.42 \times 10^{-7}$	$1.98 \times 10^{-6} \pm 7.29 \times 10^{-6}$	$5.82 \times 10^{-8} \pm 7.52 \times 10^{-8}$	$1.45 \times 10^{-7} \pm 1.38 \times 10^{-7}$	$4.89 \times 10^{-6} \pm 1.52 \times 10^{-5}$
HNN-2-50-silu	$3.04 \times 10^{-7} \pm 1.02 \times 10^{-6}$	$8.52 \times 10^{-6} \pm 4.60 \times 10^{-5}$	$5.43 \times 10^{-9} \pm 8.79 \times 10^{-9}$	$2.38 \times 10^{-8} \pm 4.28 \times 10^{-8}$	$1.71 \times 10^{-6} \pm 6.57 \times 10^{-6}$
HNN-1-100-silu	$9.07 \times 10^{-2} \pm 1.79 \times 10^{-1}$	$6.64 \times 10^{-1} \pm 8.55 \times 10^{-0}$	$7.30 \times 10^{1} \pm 3.18 \times 10^{2}$	$1.12 \times 10^{-0} \pm 4.30 \times 10^{-0}$	$1.11 \times 10^{4} \pm 4.85 \times 10^{4}$
HNN-1-50-silu	$1.02 \times 10^{-7} \pm 2.20 \times 10^{-7}$	$1.14 \times 10^{-6} \pm 3.57 \times 10^{-6}$	$1.31 \times 10^{-8} \pm 1.75 \times 10^{-8}$	$5.12 \times 10^{-8} \pm 7.22 \times 10^{-8}$	$7.96 \times 10^{-7} \pm 2.33 \times 10^{-6}$
α-SGHN(ours)	**$7.66\ \mathrm{x}\ 10^{-10} \pm 2.53\ \mathrm{x}\ 10^{-9}$**	**$1.25\ \mathrm{x}\ 10^{-8} \pm 2.95\ \mathrm{x}\ 10^{-8}$**	**$3.64\ \mathrm{x}\ 10^{-11} \pm 3.05\ \mathrm{x}\ 10^{-11}$**	**$6.39\ \mathrm{x}\ 10^{-11} \pm 1.09\ \mathrm{x}\ 10^{-10}$**	**$3.96\ \mathrm{x}\ 10^{-8} \pm 1.70\ \mathrm{x}\ 10^{-7}$**

20 samples can be found in Sec. III B. The baseline model adopts the network configuration with the best performance. Please note that the range of the y axis differs between plots. It can be seen that α-SGHN exhibits the best performance. The α-SGHN prediction of momentum in Toda with $N = 4$ appears to have a significant attenuation, but its error variation is around 1×10^{-4}. Generally, over the timescales of a few tens of units considered, it is clear that the α-SGHN has the weakest fluctuations. Figure 4 shows the evolution of Toda-4′s predicted conservation laws over time when the prediction time increases by 20 times the training time. It can be seen that overall, α-SGHN performs better than the baseline model (when taking into account all the conserved quantities and their deviation from the original value).

G. Why graph neural networks have advantages over traditional neural networks

In this section, we provide an intuitive explanation of why graph neural network–based models are superior to traditional neural networks. Figure 5 shows a comparison of network structures for learning energy part between HNN and SGHN.

On the left is an HNN, which is mainly learned by an MLP denoted as f_{HNN} to learn the Hamiltonian of the system. That is to say, it uses $f_{HNN}(q_1, \dots, q_N, p_1, \dots, p_N)$ to approximate $H = \sum_{i=1}^{N} T(p_i) + \sum_{i=1}^{N} V(q_i, q_{N_i})$, where the input is $q_1, \dots, q_N, p_1, \dots, p_N$. From the left panel of Fig. 5, it can be seen that there is no connection between the input data and they are connected to the neurons of the next layer of the network in an identical manner.

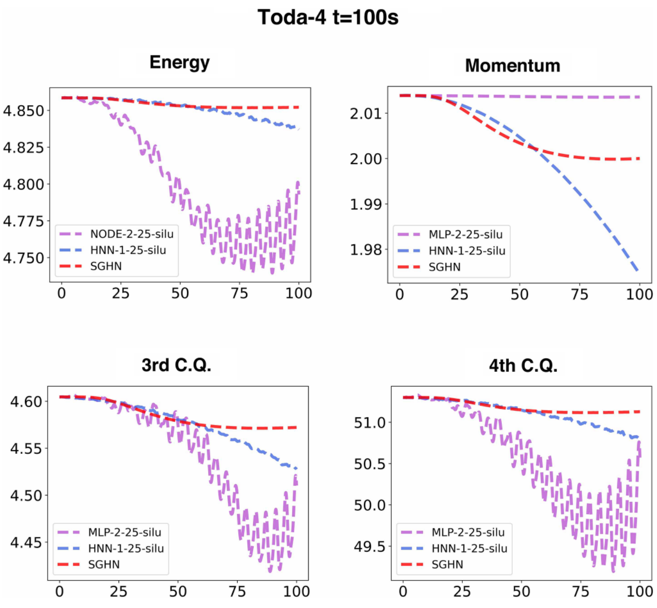

FIG. 4. The evolution over time of the average true conservation values and the average predicted conservation values of 20 samples. C.Q. represents conserved quantity.

TABLE VI. The test results for Toda system with $N = 4$. The best results are emphasized by bold font.

	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$
MLP-2-25-gelu	$1.03 \times 10^{-4} \pm 9.57 \times 10^{-5}$	$1.22 \times 10^{-3} \pm 3.36 \times 10^{-3}$	$2.39 \times 10^{-4} \pm 6.61 \times 10^{-4}$
MLP-2-25-tanh	$2.49 \times 10^{-5} \pm 3.43 \times 10^{-5}$	$8.08 \times 10^{-4} \pm 3.33 \times 10^{-3}$	$7.73 \times 10^{-5} \pm 1.83 \times 10^{-4}$
MLP-2-10-silu	$7.71 \times 10^{-6} \pm 1.69 \times 10^{-5}$	$2.11 \times 10^{-4} \pm 7.35 \times 10^{-4}$	$2.07 \times 10^{-5} \pm 3.87 \times 10^{-5}$
MLP-1-25-silu	$5.04 \times 10^{-6} \pm 5.37 \times 10^{-6}$	$3.92 \times 10^{-5} \pm 8.15 \times 10^{-5}$	$2.00 \times 10^{-5} \pm 3.95 \times 10^{-5}$
MLP-3-25-silu	$3.80 \times 10^{-6} \pm 6.97 \times 10^{-6}$	$4.28 \times 10^{-5} \pm 1.08 \times 10^{-4}$	$4.47 \times 10^{-5} \pm 1.69 \times 10^{-4}$
MLP-2-50-silu	$5.44 \times 10^{-6} \pm 1.11 \times 10^{-5}$	$3.56 \times 10^{-5} \pm 8.72 \times 10^{-5}$	$1.31 \times 10^{-5} \pm 2.99 \times 10^{-5}$
MLP-2-25-silu	$4.34 \times 10^{-6} \pm 6.36 \times 10^{-6}$	$3.00 \times 10^{-5} \pm 8.03 \times 10^{-5}$	$4.00 \times 10^{-5} \pm 1.33 \times 10^{-4}$
HNN-1-25-gelu	$4.56 \times 10^{-6} \pm 1.05 \times 10^{-5}$	$8.85 \times 10^{-5} \pm 3.74 \times 10^{-4}$	$1.15 \times 10^{-6} \pm 1.41 \times 10^{-6}$
HNN-1-25-tanh	$2.32 \times 10^{-5} \pm 7.00 \times 10^{-5}$	$6.83 \times 10^{-4} \pm 3.38 \times 10^{-3}$	$3.79 \times 10^{-6} \pm 1.28 \times 10^{-5}$
HNN-1-10-silu	$3.95 \times 10^{-6} \pm 8.89 \times 10^{-6}$	$3.68 \times 10^{-5} \pm 1.02 \times 10^{-4}$	$2.70 \times 10^{-6} \pm 3.43 \times 10^{-6}$
HNN-2-25-silu	$3.86 \times 10^{-6} \pm 1.42 \times 10^{-5}$	$6.23 \times 10^{-5} \pm 3.15 \times 10^{-4}$	$5.00 \times 10^{-7} \pm 1.86 \times 10^{-6}$
HNN-1-50-silu	$2.91 \times 10^{-6} \pm 9.00 \times 10^{-6}$	$1.76 \times 10^{-5} \pm 8.77 \times 10^{-5}$	$7.46 \times 10^{-7} \pm 2.66 \times 10^{-6}$
HNN-1-25-silu	$1.17 \times 10^{-6} \pm 3.36 \times 10^{-6}$	$8.98 \times 10^{-6} \pm 2.66 \times 10^{-5}$	$1.95 \times 10^{-7} \pm 4.22 \times 10^{-7}$
α-SGHN(ours)	$\mathbf{1.96 \times 10^{-9} \pm 6.09 \times 10^{-9}}$	$\mathbf{3.41 \times 10^{-8} \pm 1.23 \times 10^{-7}}$	$\mathbf{1.03 \times 10^{-8} \pm 1.81 \times 10^{-8}}$
	$\text{Momentum}_{\text{MSE}}$	C3_{MSE}	C4_{MSE}
MLP-2-25-gelu	$4.17 \times 10^{-5} \pm 8.30 \times 10^{-5}$	$6.78 \times 10^{-4} \pm 7.61 \times 10^{-4}$	$6.82 \times 10^{-2} \pm 9.65 \times 10^{-2}$
MLP-2-25-tanh	$3.40 \times 10^{-5} \pm 7.44 \times 10^{-5}$	$2.14 \times 10^{-4} \pm 4.07 \times 10^{-4}$	$2.06 \times 10^{-2} \pm 4.33 \times 10^{-2}$
MLP-2-10-silu	$7.29 \times 10^{-10} \pm 1.83 \times 10^{-10}$	$5.93 \times 10^{-5} \pm 9.14 \times 10^{-5}$	$1.09 \times 10^{-2} \pm 2.20 \times 10^{-2}$
MLP-1-25-silu	$6.88 \times 10^{-9} \pm 5.02 \times 10^{-9}$	$9.75 \times 10^{-5} \pm 2.27 \times 10^{-4}$	$1.38 \times 10^{-2} \pm 3.31 \times 10^{-2}$
MLP-3-25-silu	$\mathbf{1.15 \times 10^{-10} \pm 1.48 \times 10^{-10}}$	$8.02 \times 10^{-5} \pm 3.19 \times 10^{-4}$	$1.94 \times 10^{-2} \pm 7.53 \times 10^{-2}$
MLP-2-50-silu	$1.02 \times 10^{-9} \pm 1.08 \times 10^{-9}$	$1.57 \times 10^{-4} \pm 3.93 \times 10^{-4}$	$1.11 \times 10^{-2} \pm 2.91 \times 10^{-2}$
MLP-2-25-silu	$4.59 \times 10^{-10} \pm 5.68 \times 10^{-10}$	$5.50 \times 10^{-5} \pm 9.84 \times 10^{-5}$	$8.03 \times 10^{-3} \pm 1.80 \times 10^{-2}$
HNN-1-25-gelu	$1.77 \times 10^{-6} \pm 1.11 \times 10^{-6}$	$4.04 \times 10^{-5} \pm 6.32 \times 10^{-5}$	$1.48 \times 10^{-3} \pm 1.72 \times 10^{-3}$
HNN-1-25-tanh	$7.87 \times 10^{-7} \pm 7.56 \times 10^{-7}$	$6.53 \times 10^{-4} \pm 1.91 \times 10^{-3}$	$2.59 \times 10^{-2} \pm 8.09 \times 10^{-2}$
HNN-1-10-silu	$5.85 \times 10^{-6} \pm 4.47 \times 10^{-7}$	$1.51 \times 10^{-4} \pm 3.17 \times 10^{-4}$	$4.70 \times 10^{-3} \pm 7.58 \times 10^{-3}$
HNN-2-25-silu	$1.29 \times 10^{-7} \pm 1.18 \times 10^{-7}$	$9.15 \times 10^{-6} \pm 2.61 \times 10^{-5}$	$7.34 \times 10^{-4} \pm 2.70 \times 10^{-3}$
HNN-1-50-silu	$1.19 \times 10^{-7} \pm 1.24 \times 10^{-7}$	$1.64 \times 10^{-5} \pm 4.67 \times 10^{-5}$	$8.54 \times 10^{-4} \pm 2.38 \times 10^{-3}$
HNN-1-25-silu	$6.46 \times 10^{-8} \pm 5.70 \times 10^{-8}$	$3.17 \times 10^{-5} \pm 1.31 \times 10^{-4}$	$7.46 \times 10^{-4} \pm 2.64 \times 10^{-3}$
α-SGHN(ours)	$2.11 \times 10^{-8} \pm 3.17 \times 10^{-8}$	$\mathbf{1.64 \times 10^{-7} \pm 2.30 \times 10^{-7}}$	$\mathbf{8.33 \times 10^{-6} \pm 1.30 \times 10^{-5}}$

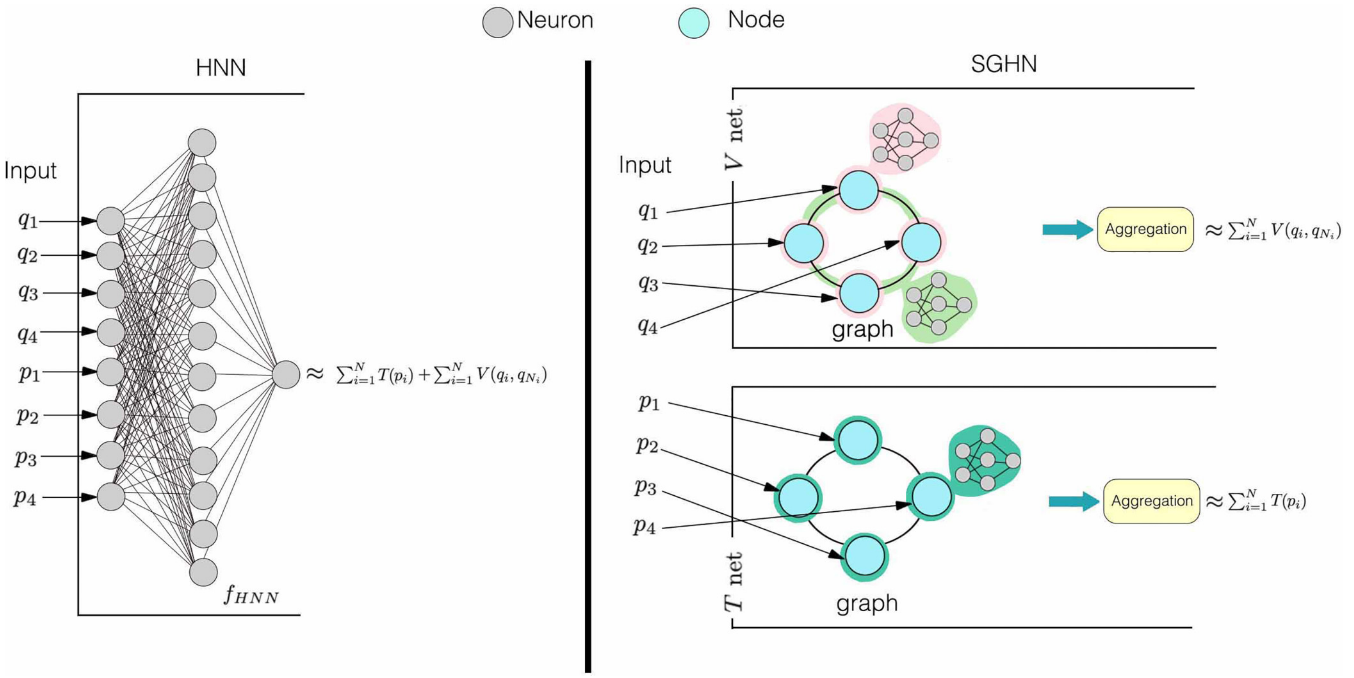

FIG. 5. Comparison of network structures for learning energy part between HNN and SGHN.

TABLE VII. The test results for Toda system with $N = 5$. Nan = not a number. The best results are emphasized by bold font.

	Test loss	Trajectory$_{\mathrm{MSE}}$	Energy$_{\mathrm{MSE}}$	Momentum$_{\mathrm{MSE}}$
MLP-1-50-gelu	$1.87 \times 10^{-5} \pm 2.60 \times 10^{-5}$	$1.44 \times 10^{-4} \pm 3.19 \times 10^{-4}$	$9.38 \times 10^{-5} \pm 2.75 \times 10^{-4}$	$2.42 \times 10^{-5} \pm 2.71 \times 10^{-5}$
MLP-1-50-tanh	$3.82 \times 10^{-5} \pm 4.72 \times 10^{-5}$	$8.90 \times 10^{-4} \pm 3.68 \times 10^{-3}$	$6.44 \times 10^{-5} \pm 7.88 \times 10^{-5}$	$6.31 \times 10^{-5} \pm 5.59 \times 10^{-5}$
MLP-1-25-silu	$1.23 \times 10^{-5} \pm 1.14 \times 10^{-5}$	$7.41 \times 10^{-5} \pm 1.59 \times 10^{-4}$	$1.74 \times 10^{-5} \pm 2.06 \times 10^{-5}$	$2.32 \times 10^{-8} \pm 2.80 \times 10^{-8}$
MLP-2-50-silu	$9.59 \times 10^{-6} \pm 1.13 \times 10^{-5}$	$5.59 \times 10^{-5} \pm 1.25 \times 10^{-4}$	$5.23 \times 10^{-5} \pm 1.91 \times 10^{-4}$	$\mathbf{5.35 \times 10^{-9} \pm 5.95 \times 10^{-9}}$
MLP-1-100-silu	$3.29 \times 10^{-1} \pm 1.41 \times 10^{-0}$	Nan	Nan	Nan
MLP-1-50-silu	$6.39 \times 10^{-6} \pm 1.60 \times 10^{-5}$	$5.25 \times 10^{-5} \pm 1.25 \times 10^{-4}$	$2.69 \times 10^{-5} \pm 4.05 \times 10^{-5}$	$1.09 \times 10^{-8} \pm 8.26 \times 10^{-9}$
HNN-1-25-gelu	$1.81 \times 10^{-4} \pm 2.00 \times 10^{-4}$	$2.20 \times 10^{-3} \pm 7.00 \times 10^{-3}$	$1.73 \times 10^{-3} \pm 3.07 \times 10^{-3}$	$4.16 \times 10^{-3} \pm 5.77 \times 10^{-3}$
HNN-1-25-tanh	$4.17 \times 10^{-5} \pm 8.75 \times 10^{-5}$	$1.77 \times 10^{-3} \pm 9.10 \times 10^{-3}$	$1.74 \times 10^{-5} \pm 2.85 \times 10^{-5}$	$1.72 \times 10^{-5} \pm 2.22 \times 10^{-5}$
HNN-1-10-silu	$7.26 \times 10^{-5} \pm 1.21 \times 10^{-4}$	$3.18 \times 10^{-3} \pm 1.36 \times 10^{-2}$	$2.32 \times 10^{-5} \pm 5.05 \times 10^{-5}$	$3.67 \times 10^{-6} \pm 2.80 \times 10^{-6}$
HNN-2-25-silu	$5.25 \times 10^{-6} \pm 1.99 \times 10^{-5}$	$1.26 \times 10^{-4} \pm 5.90 \times 10^{-4}$	$9.85 \times 10^{-7} \pm 3.43 \times 10^{-6}$	$4.15 \times 10^{-7} \pm 3.16 \times 10^{-7}$
HNN-1-50-silu	$6.28 \times 10^{-6} \pm 2.05 \times 10^{-5}$	$1.06 \times 10^{-4} \pm 5.45 \times 10^{-4}$	$1.32 \times 10^{-6} \pm 4.32 \times 10^{-6}$	$2.15 \times 10^{-7} \pm 1.57 \times 10^{-7}$
HNN-1-25-silu	$2.14 \times 10^{-6} \pm 5.47 \times 10^{-6}$	$3.08 \times 10^{-5} \pm 9.12 \times 10^{-5}$	$1.07 \times 10^{-6} \pm 1.53 \times 10^{-6}$	$1.36 \times 10^{-6} \pm 9.82 \times 10^{-7}$
α-SGHN(ours)	$\mathbf{2.98 \times 10^{-9} \pm 5.19 \times 10^{-9}}$	$\mathbf{4.14 \times 10^{-8} \pm 1.79 \times 10^{-7}}$	$\mathbf{7.53 \times 10^{-9} \pm 1.19 \times 10^{-8}}$	$1.25 \times 10^{-8} \pm 1.81 \times 10^{-8}$
	C3$_{\mathrm{MSE}}$	C4$_{\mathrm{MSE}}$	C5$_{\mathrm{MSE}}$	
MLP-1-50-gelu	$1.93 \times 10^{-4} \pm 1.59 \times 10^{-4}$	$2.52 \times 10^{-2} \pm 5.80 \times 10^{-2}$	$1.92 \times 10^{-1} \pm 2.38 \times 10^{-1}$	
MLP-1-50-tanh	$5.34 \times 10^{-4} \pm 5.24 \times 10^{-4}$	$2.96 \times 10^{-2} \pm 3.24 \times 10^{-2}$	$3.84 \times 10^{-1} \pm 3.95 \times 10^{-1}$	
MLP-1-25-silu	$5.43 \times 10^{-5} \pm 7.75 \times 10^{-5}$	$7.96 \times 10^{-3} \pm 1.04 \times 10^{-2}$	$7.48 \times 10^{-2} \pm 1.24 \times 10^{-1}$	
MLP-2-50-silu	$4.33 \times 10^{-5} \pm 6.74 \times 10^{-5}$	$1.68 \times 10^{-2} \pm 5.20 \times 10^{-2}$	$8.22 \times 10^{-2} \pm 1.68 \times 10^{-1}$	
MLP-1-100-silu	Nan	Nan	Nan	
MLP-1-50-silu	$5.92 \times 10^{-5} \pm 1.11 \times 10^{-4}$	$6.62 \times 10^{-3} \pm 1.33 \times 10^{-2}$	$5.78 \times 10^{-2} \pm 1.05 \times 10^{-1}$	
HNN-1-25-gelu	$2.71 \times 10^{-2} \pm 3.97 \times 10^{-2}$	$1.29 \times 10^{-0} \pm 2.32 \times 10^{-0}$	$1.66 \times 10^{1} \pm 2.78 \times 10^{1}$	
HNN-1-25-tanh	$3.90 \times 10^{-4} \pm 5.42 \times 10^{-4}$	$3.11 \times 10^{-2} \pm 6.44 \times 10^{-2}$	$4.78 \times 10^{-1} \pm 8.33 \times 10^{-1}$	
HNN-1-10-silu	$6.72 \times 10^{-4} \pm 1.44 \times 10^{-3}$	$4.28 \times 10^{-2} \pm 9.68 \times 10^{-2}$	$6.94 \times 10^{-1} \pm 1.65 \times 10^{-0}$	
HNN-2-25-silu	$2.74 \times 10^{-4} \pm 1.12 \times 10^{-3}$	$2.14 \times 10^{-2} \pm 8.97 \times 10^{-2}$	$2.96 \times 10^{-1} \pm 1.21 \times 10^{-0}$	
HNN-1-50-silu	$6.32 \times 10^{-5} \pm 1.86 \times 10^{-4}$	$3.20 \times 10^{-3} \pm 8.26 \times 10^{-3}$	$6.14 \times 10^{-2} \pm 1.73 \times 10^{-1}$	
HNN-1-25-silu	$2.97 \times 10^{-5} \pm 4.69 \times 10^{-5}$	$1.27 \times 10^{-3} \pm 2.06 \times 10^{-3}$	$1.96 \times 10^{-2} \pm 3.07 \times 10^{-2}$	
α-SGHN(ours)	$\mathbf{1.13 \times 10^{-7} \pm 1.64 \times 10^{-7}}$	$\mathbf{5.91 \times 10^{-6} \pm 9.53 \times 10^{-6}}$	$\mathbf{7.99 \times 10^{-5} \pm 1.22 \times 10^{-4}}$	

On the right is the α-SGHN part 2. It can be seen that the input data are first assigned to the system graph structure learned in the first part (the graph structure encodes the interaction relationships between particles in the system). Subsequently, the information of the nodes and edges is aggregated through MLPs placed on the nodes and edges. In other words, α-SGHN has a deep understanding of the interaction information within the system compared to HNN, which makes it more effective.

H. Comparison with graph neural networks

In this section, we will compare our method with the two graph neural network methods Hamiltonian ODE graph network (HOGN) [27] and Hamiltonian graph neural network (HGNN) [28], and the parameter settings of the graph neural network model are the same as those in [28]. We define the initial conditions as follows:

$$\mathbf{q}_0(i) = \lambda_i \sin\left(\frac{(i-1)\pi}{N-1}\right), \tag{22}$$

$$\mathbf{p}_0(i) = 0, \quad i = 1, \ldots, N, \tag{23}$$

where $\lambda_i \sim \mathcal{U}(0, 1)$. Under the same input conditions, i.e., without knowing the links of the particles, a fully connected graph is used like α-SGHN (assuming that particles all have interactions between them), and the test results are shown in Table VIII. It can be seen that other graph-based methods cannot work without knowing the particle links within the system (i.e., the particle interactions). Our method works because it can predict the particle links of the system, uncovering the matrix of the interparticle interactions. In the case of known lattice particles links, the accuracy of HOGN and HGNN cannot support learning systems with non-even symmetric potential energy; see Table IX.

IV. THE RELATIONSHIP BETWEEN THE COMPLETE INTEGRABILITY OF A SYSTEM AND THE LEARNING PERFORMANCE OF NEURAL NETWORKS

In this section, we explore the impact of the system's conservation laws on the predictive performance of neural network models. We construct a hybrid system of FK and Toda (FK-Toda), as shown here:

$$H = \sum_{i=1}^{N} \frac{p_i^2}{2} + \mu \sum_{i=1}^{N} \exp(q_i - q_{i+1}) + (1-\mu)\left(\frac{(q_{i+1} - q_i)^2}{2} + 1 - \cos(q_i)\right). \tag{24}$$

When $\mu = 0$, it is an FK system with a conserved quantity, which is energy. When $\mu = 1$, it is a Toda system, completely integrable, with N conserved quantities. Notice that this is an interesting dynamical system in its own right, inspired by the

TABLE VIII. The test results for HGNN, HOGN, and α-SGHN. The best results are emphasized by bold font. The link relationship of the lattice system is unknown, so a fully connected network is used.

FK system with $N = 32$				
	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$	$\text{Momentum}_{\text{MSE}}$
HOGN	$3.35 \times 10^{-2} \pm 5.82 \times 10^{-2}$	$1.26 \times 10^{-0} \pm 4.59 \times 10^{-0}$	$4.77 \times 10^{3} \pm 1.43 \times 10^{4}$	
HGNN	$3.44 \times 10^{-2} \pm 6.44 \times 10^{-3}$	$1.82 \times 10^{-1} \pm 9.33 \times 10^{-2}$	$8.60 \times 10^{-2} \pm 1.06 \times 10^{-1}$	
α-SGHN	$\mathbf{2.46 \times 10^{-9} \pm 1.61 \times 10^{-9}}$	$\mathbf{2.73 \times 10^{-8} \pm 3.67 \times 10^{-8}}$	$\mathbf{1.72 \times 10^{-8} \pm 2.14 \times 10^{-8}}$	

Rotator system with $N = 32$				
	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$	$\text{Momentum}_{\text{MSE}}$
HOGN	$1.90 \times 10^{-1} \pm 5.23 \times 10^{-2}$	Nan	Nan	Nan
HGNN	$1.43 \times 10^{-1} \pm 3.04 \times 10^{-2}$	$3.02 \times 10^{-1} \pm 1.21 \times 10^{-1}$	$8.92 \times 10^{-1} \pm 8.31 \times 10^{-1}$	$2.38 \times 10^{1} \pm 7.93 \times 10^{-0}$
α-SGHN	$\mathbf{3.90 \times 10^{-9} \pm 1.66 \times 10^{-9}}$	$\mathbf{3.93 \times 10^{-8} \pm 5.08 \times 10^{-8}}$	$\mathbf{5.27 \times 10^{-8} \pm 2.13 \times 10^{-8}}$	$\mathbf{8.07 \times 10^{-8} \pm 3.03 \times 10^{-8}}$

Toda system with $N = 5$				
	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$	$\text{Momentum}_{\text{MSE}}$
HOGN	$3.44 \times 10^{-2} \pm 3.11 \times 10^{-2}$	$1.89 \times 10^{-1} \pm 1.94 \times 10^{-1}$	$2.92 \times 10^{-2} \pm 5.20 \times 10^{-2}$	$1.92 \times 10^{-1} \pm 2.62 \times 10^{-1}$
HGNN	$1.37 \times 10^{-2} \pm 1.01 \times 10^{-2}$	$2.00 \times 10^{-1} \pm 1.84 \times 10^{-1}$	$4.81 \times 10^{-3} \pm 5.92 \times 10^{-3}$	$6.17 \times 10^{-3} \pm 8.51 \times 10^{-3}$
α-SGHN	$\mathbf{4.80 \times 10^{-9} \pm 6.40 \times 10^{-9}}$	$\mathbf{8.61 \times 10^{-8} \pm 1.10 \times 10^{-7}}$	$\mathbf{7.56 \times 10^{-10} \pm 6.83 \times 10^{-10}}$	$\mathbf{1.60 \times 10^{-9} \pm 1.50 \times 10^{-9}}$

well-known Salerno model in the context of discrete nonlinear Schrödinger equations [47]. The latter has been explored recently in the vicinity of the integrable limit as concerns the sensitivity of higher-order conserved quantities [48] and the "detectability" of a model's integrable nature. Here, we explore the predictive performance of the FK-Toda system and how it changes for the model when $\mu \in [0, 1]$.

For evaluation, we use the mean absolute percentage error (MAPE), which is defined as

$$\text{MAPE} = \frac{1}{N_T}\sum_{t=1}^{N_T}\frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{\mathbf{F}}_{t,i} - \mathbf{F}_{t,i}}{\mathbf{F}_{t,i}}\right|, \tag{25}$$

where $\hat{\mathbf{F}}_{t,i}$ and $\mathbf{F}_{t,i}$ are the predicted value and the corresponding ground truth at the tth time for the ith particle.

Figure 6 shows the changes in the predicted MAPE of trajectory and energy of the network model as μ increases when $N = 5$. To display this clearly, the y axis uses a semilog scale. It can be seen that the MAPE of α-SGHN has always been the lowest. As μ increases, the system becomes more complex and the predictive performance may decrease. However, when μ approaches 1, the predictive performance of the model improves. When $\mu = 1$, we retrieve the completely integrable model, and there we can see how the α-SGHN predictions visibly capture the integrable structure; we observe similar MAPE trends for HNN and MLP. That is to say, when complex systems approach integrability, the predictive performance of the model will improve, and when the system is completely integrable, the predictive performance of the model will reach its best. This is an interesting feature worth exploring further in the future (e.g., its potential validity in other settings).

To avoid the above conclusion being an exception, we also constructed in the same spirit a hybrid system of the Fermi-Pasta-Ulam-Tsingou [1,49,50] and Toda (FPUT-Toda):

$$H = \sum_{i=1}^{N}\frac{p_i^2}{2} + \mu\sum_{i=1}^{N}\exp(q_i - q_{i+1}) + (1-\mu)\left(\frac{(q_{i+1}-q_i)^2}{2} + \frac{(q_{i+1}-q_i)^3}{6}\right). \tag{26}$$

When $\mu = 0$, it is an FPUT system with two conserved quantities, which are energy and momentum. When $\mu = 1$, it is a Toda system, completely integrable, with N conserved quantities.

Figure 7 shows the results of the FPUT-Toda system testing. The MAPE of α-SGHN has always been the lowest. When the system is fully integrable, i.e., $\mu = 1$, the performance of the three neural network models becomes optimal. Overall, our results indicate that neural networks have better

TABLE IX. In the case of known lattice system links, HOGN and HNN cannot learn systems with non-even symmetric potential energy.

Toda system with $N = 5$				
	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$	$\text{Momentum}_{\text{MSE}}$
HOGN with graph	$2.42 \times 10^{-3} \pm 2.66 \times 10^{-3}$	$1.20 \times 10^{-2} \pm 1.62 \times 10^{-2}$	$2.61 \times 10^{-3} \pm 3.78 \times 10^{-3}$	$2.67 \times 10^{-3} \pm 2.45 \times 10^{-3}$
HGNN with graph	$2.62 \times 10^{-3} \pm 2.93 \times 10^{-3}$	$5.30 \times 10^{-3} \pm 7.25 \times 10^{-3}$	$1.97 \times 10^{-3} \pm 3.35 \times 10^{-3}$	$1.88 \times 10^{-4} \pm 2.90 \times 10^{-4}$
α-SGHN	$\mathbf{4.80 \times 10^{-9} \pm 6.40 \times 10^{-9}}$	$\mathbf{8.61 \times 10^{-8} \pm 1.10 \times 10^{-7}}$	$\mathbf{7.56 \times 10^{-10} \pm 6.83 \times 10^{-10}}$	$\mathbf{1.06 \times 10^{-9} \pm 1.50 \times 10^{-9}}$

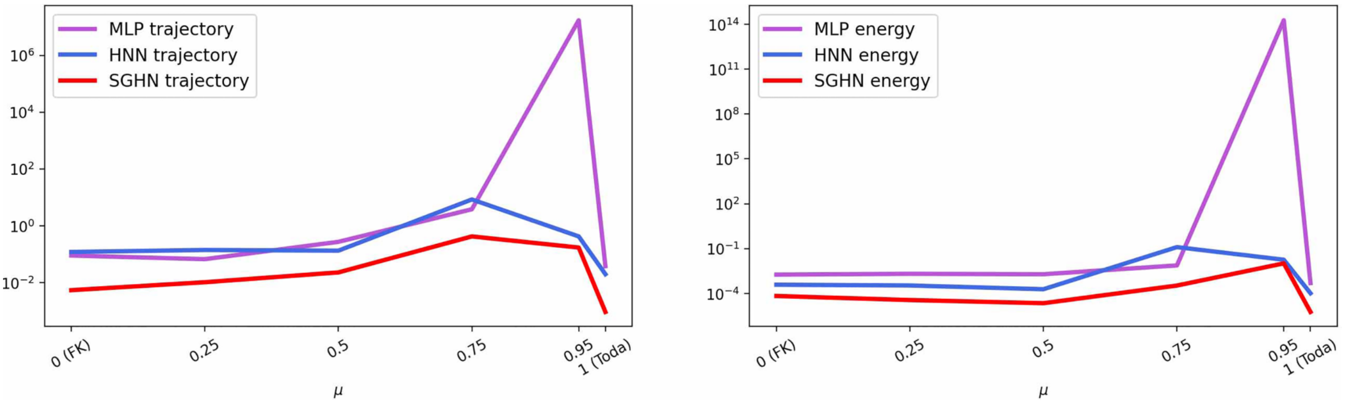

FIG. 6. The mean of MAPE of 20 test samples when μ changes from 0 to 1 for FK-Toda system with $N = 5$. HNN has a hidden layer with 25 units per layer, while MLP has a hidden layer with 50 units per layer. The activation functions are all SiLU. In order to display this clearly, the y axis uses semilog scale.

performance in modeling fully integrable systems, as also discussed above.

V. CONCLUSIONS AND FUTURE CHALLENGES

In this work, we introduced our model α-SGHN for studying lattice systems. First, it can capture the interactions between particles in complex, strongly nonlinear Hamiltonian lattice systems with multiple degrees of freedom. Second, α-SGHN can utilize the predicted particle interaction for further and more accurate system behavior prediction. It eliminates the limitation of knowing the system structure in advance when studying lattice systems based on graph neural networks, and provides a method for inferring particle interactions based solely on particle trajectories. α-SGHN can discover particle interactions in lattice systems solely from their motion trajectories, without requiring any prior knowledge of structural information. In addition, we also investigated whether the particle behavior predicted by α-SGHN preserves the system's conservation laws. We tested a variety of systems ranging from one and two conserved quantities, to controllably many conserved quantities and ultimately to (in the case of as many as the degrees of freedom, the scenario of) complete integrability. The experimental results show that the trajectory predicted by α-SGHN preserves all the conserved variables of the system, and generically features a performance that is far superior to the baseline model, although we did find isolated exceptions of other networks performing better (e.g., the momentum conservation in some of the Toda lattice examples). Nevertheless, overall, the performance of α-SGHN was found to be considerably superior to that of the baseline models to which it was compared herein, lending considerable promise to it for future applications. This may be due to the graph structure in α-SGHN implying information about particle interactions.

These findings seem to suggest the relevance of extending considerations of the α-SGHN approach to a variety of other models, including ones with beyond-nearest-neighbor interactions and even ones associated with long-range interactions and the related fractional derivatives, a number of examples of which have been recently explored in [51]. This may be quite useful in not only discovering such models and reconstructing their trajectories, but also toward examining their potential conservation laws and exploring their integrability, as was done herein in a range of benchmark cases. Such studies are currently in progress and will be reported in future publications.

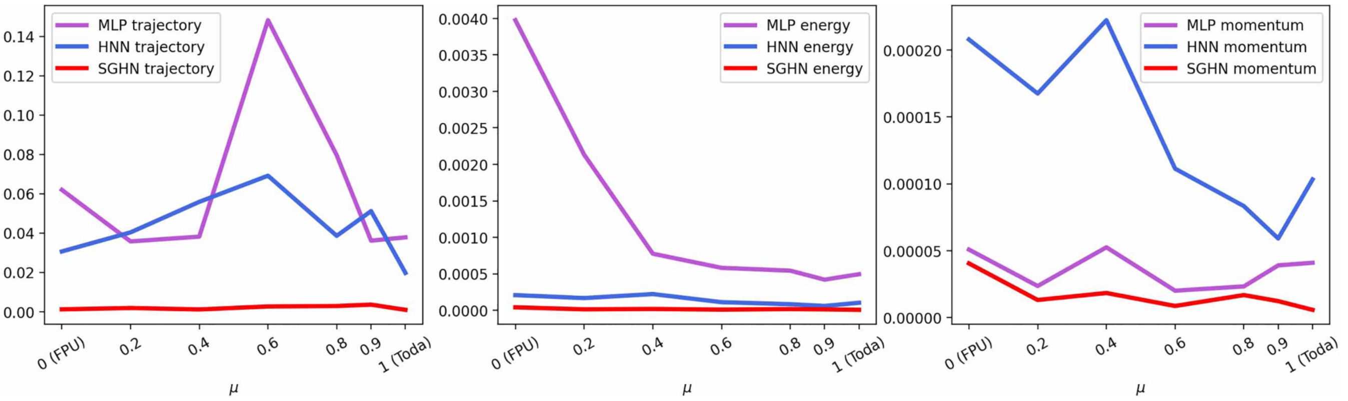

FIG. 7. The mean of MAPE of 20 test samples when μ changes from 0 to 1 for FPU-Toda system with $N = 5$. HNN has a hidden layer with 25 units per layer, while MLP has a hidden layer with 50 units per layer. The activation functions are all SiLU.

TABLE X. The test results for a system with multiple beyond-nearest-neighbor interactions, where $N = 32$. The best results are emphasized by bold font.

	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$
MLP-1-100-silu	$1.83 \times 10^{-2} \pm 7.18 \times 10^{-3}$	$5.66 \times 10^{-1} \pm 1.08 \times 10^{-0}$	$3.18 \times 10^{2} \pm 1.07 \times 10^{3}$
MLP-1-100-tanh	$3.49 \times 10^{-2} \pm 1.47 \times 10^{-2}$	$1.26 \times 10^{-0} \pm 1.18 \times 10^{-0}$	$1.27 \times 10^{2} \pm 1.27 \times 10^{2}$
MLP-1-50-gelu	$1.28 \times 10^{-1} \pm 3.78 \times 10^{-2}$	$3.68 \times 10^{1} \pm 4.99 \times 10^{1}$	$9.88 \times 10^{5} \pm 2.46 \times 10^{6}$
MLP-2-100-gelu	$2.45 \times 10^{-2} \pm 8.77 \times 10^{-3}$	$7.89 \times 10^{-1} \pm 9.77 \times 10^{-1}$	$1.01 \times 10^{2} \pm 1.61 \times 10^{2}$
MLP-1-200-gelu	$7.20 \times 10^{-2} \pm 1.99 \times 10^{-2}$	$8.14 \times 10^{-1} \pm 8.74 \times 10^{-1}$	$1.78 \times 10^{2} \pm 1.88 \times 10^{2}$
MLP-1-100-gelu	$1.96 \times 10^{-2} \pm 8.07 \times 10^{-3}$	$3.75 \times 10^{-1} \pm 4.43 \times 10^{-1}$	$1.24 \times 10^{1} \pm 1.34 \times 10^{1}$
HNN-1-100-silu	$1.63 \times 10^{-2} \pm 5.49 \times 10^{-3}$	$3.49 \times 10^{-1} \pm 2.93 \times 10^{-1}$	$7.36 \times 10^{-2} \pm 4.54 \times 10^{-2}$
HNN-1-100-tanh	$1.83 \times 10^{-2} \pm 1.11 \times 10^{-2}$	$4.13 \times 10^{-1} \pm 3.88 \times 10^{-1}$	$7.77 \times 10^{-2} \pm 4.56 \times 10^{-2}$
HNN-1-50-gelu	$1.23 \times 10^{-1} \pm 3.79 \times 10^{-2}$	$9.27 \times 10^{1} \pm 1.05 \times 10^{2}$	$5.08 \times 10^{6} \pm 9.09 \times 10^{6}$
HNN-2-100-gelu	$8.58 \times 10^{-3} \pm 2.71 \times 10^{-3}$	$1.20 \times 10^{-1} \pm 1.32 \times 10^{-1}$	$3.57 \times 10^{-2} \pm 2.50 \times 10^{-2}$
HNN-1-200-gelu	$9.67 \times 10^{-3} \pm 3.05 \times 10^{-3}$	$1.75 \times 10^{-1} \pm 1.77 \times 10^{-1}$	$3.31 \times 10^{-2} \pm 2.46 \times 10^{-2}$
HNN-1-100-gelu	$6.07 \times 10^{-3} \pm 1.74 \times 10^{-3}$	$8.89 \times 10^{-2} \pm 9.10 \times 10^{-2}$	$2.07 \times 10^{-2} \pm 1.26 \times 10^{-2}$
α-SGHN(ours)	$\mathbf{1.19 \times 10^{-7} \pm 9.34 \times 10^{-8}}$	$\mathbf{5.52 \times 10^{-7} \pm 5.89 \times 10^{-7}}$	$\mathbf{1.81 \times 10^{-6} \pm 5.98 \times 10^{-7}}$

ACKNOWLEDGMENTS

This material is based on work supported by the U.S. National Science Foundation through Grants No. PHY-2110030 and No. DMS-2204702 (P.G.K.), as well as by the National Natural Science Foundation of China through Grants No. 12371187 and Science and Technology Development Plan Project of Jilin Province 20240101006JJ.

APPENDIX: SUPPLEMENTARY EXPERIMENT

1. Supplementary experiment on the system with long-range interaction

To further validate the effectiveness and generality of our method, we construct an irregular lattice system with multiple beyond-nearest-neighbor interactions, whose Hamiltonian is as follows: $H = \sum_{i=1}^{32} (\frac{1}{2}p_i^2 + \frac{1}{2}q_i^2 + \frac{1}{4}q_i^4 + \frac{1}{2}(q_i - q_{i+1})^2 + \frac{1}{2}(q_i - q_{i+3})^2 + \frac{1}{2}(q_i - q_{i+5})^2 + \frac{1}{2}(q_i - q_{i+7})^2) + \frac{1}{2}(q_{17} - q_{30})^2 + \frac{1}{2}(q_{11} - q_{30})^2$. In this example, each particle interacts with its 1st, 3rd, 5th, and 7th order neighbors, and the 30th particle also interacts with the 17th and 11th particles. Figure 8 shows the structure matrix learned by α-SGHN and the system particle interaction relationships inferred from the structure matrix, indicating that our method correctly captures the interaction relationships of the system.

The second step of training our model is to approximate the differential control equation of the unknown system. By integrating the trained model, we can predict the motion trajectory of the system under any initial conditions. Table X records the comparison of our model's predictive ability with the baseline model, indicating that our method has the best performance. In addition, the results also show that compared to previous examples, regardless of whether there is long-range interaction between system particles, the system's predictive ability is robust.

2. Supplementary experiment on structural prediction

To explore the universality of α-SGHN, we conducted data analysis on systems with long-range interactions in two dimensions to see if particle interactions can be accurately learned.

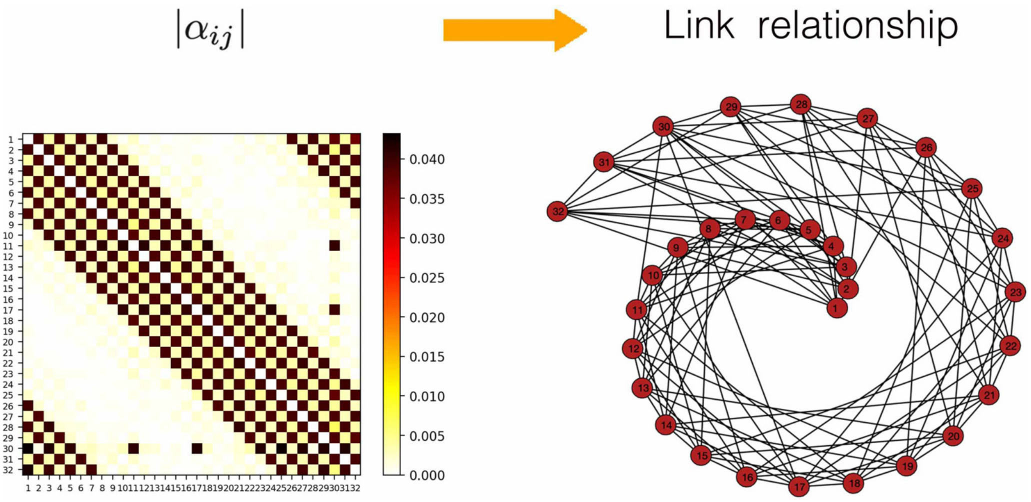

FIG. 8. Particle interaction learning results for a system with multiple beyond-nearest-neighbor interactions.

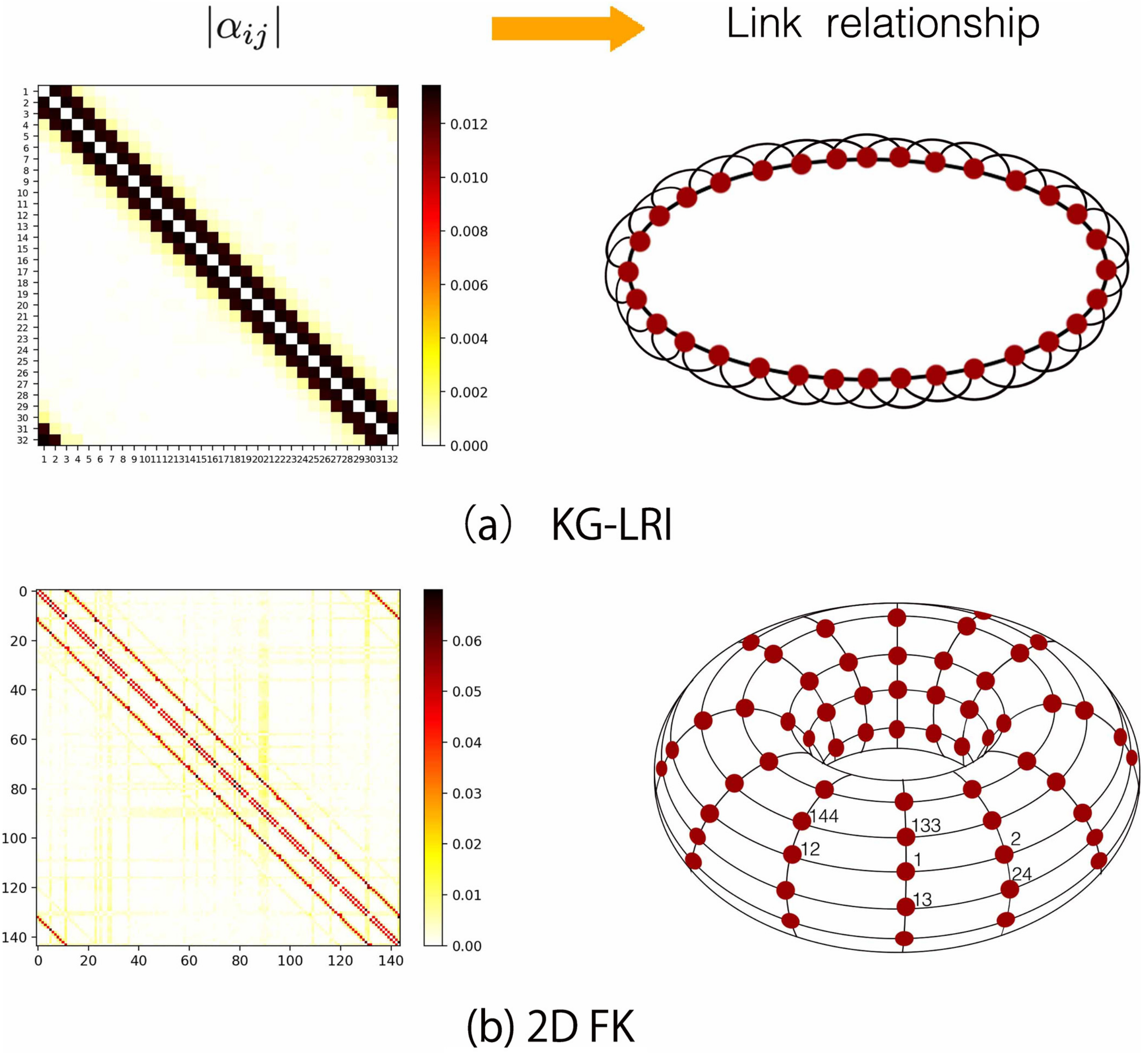

FIG. 9. Left: The x axis and y axis represent nodes, and the color-coded $|\alpha_{i,j}|$ values. Right: The particle link relationship is extracted from the left. It satisfies the periodic boundary conditions.

TABLE XI. Taking the FK system as an example, as the number of particles N increases, the performance of the model changes while the model parameters remain unchanged.

		Energy$_{\text{MSE}}$		
	$N = 32$	$N = 64$	$N = 132$	$N = 200$
MLP-1-100-gelu	$1.33 \times 10^{-1} \pm 1.17 \times 10^{-1}$	$2.13 \times 10^{3} \pm 2.14 \times 10^{3}$	$7.82 \times 10^{3} \pm 1.03 \times 10^{4}$	$5.61 \times 10^{4} \pm 3.56 \times 10^{4}$
HNN-1-100-gelu	$2.21 \times 10^{-3} \pm 1.65 \times 10^{-3}$	$5.61 \times 10^{4} \pm 3.56 \times 10^{4}$	$1.54 \times 10^{5} \pm 2.49 \times 10^{5}$	$5.12 \times 10^{5} \pm 1.11 \times 10^{6}$
α-SGHN	**$9.70 \times 10^{-7} \pm 3.13 \times 10^{-6}$**	**$1.39 \times 10^{-7} \pm 1.28 \times 10^{-7}$**	**$8.66 \times 10^{-7} \pm 2.39 \times 10^{-7}$**	**$2.43 \times 10^{-6} \pm 2.54 \times 10^{-6}$**
		Trajectory$_{\text{MSE}}$		
MLP-1-100-gelu	$3.91 \times 10^{-2} \pm 3.72 \times 10^{-2}$	$1.46 \times 10^{-0} \pm 1.27 \times 10^{-0}$	$1.75 \times 10^{-0} \pm 1.27 \times 10^{-0}$	$2.73 \times 10^{-0} \pm 1.59 \times 10^{-0}$
HNN-1-100-gelu	$1.36 \times 10^{-2} \pm 1.29 \times 10^{-2}$	$2.73 \times 10^{-0} \pm 1.59 \times 10^{-0}$	$2.91 \times 10^{-0} \pm 4.58 \times 10^{-0}$	$2.33 \times 10^{-0} \pm 5.95 \times 10^{-0}$
α-SGHN	**$4.85 \times 10^{-6} \pm 2.37 \times 10^{-5}$**	**$1.93 \times 10^{-7} \pm 6.91 \times 10^{-7}$**	**$1.70 \times 10^{-7} \pm 2.62 \times 10^{-7}$**	**$8.24 \times 10^{-7} \pm 2.10 \times 10^{-6}$**

TABLE XII. For the FK system, the performance of the model with $K = 1$ and $K = 2$, where $N = 132$.

	Train loss	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$
$K = 1$	1.02×10^{-8}	$2.62 \times 10^{-8} \pm 2.52 \times 10^{-8}$	$1.70 \times 10^{-7} \pm 2.62 \times 10^{-7}$	$8.66 \times 10^{-7} \pm 2.39 \times 10^{-7}$
$K = 2$	2.40×10^{-8}	$1.52 \times 10^{-7} \pm 1.78 \times 10^{-7}$	$7.02 \times 10^{-7} \pm 1.86 \times 10^{-6}$	$4.33 \times 10^{-6} \pm 2.60 \times 10^{-6}$

We consider the Klein-Gordon (KG) lattice system with long-range interactions (KG-LRI) [52] and the 2D FK system [53]. The methods for obtaining datasets and predicting trajectory results have been extensively studied in previous work; see [32]. Here we will only explore the results of structural learning.

The KG-LRI system is described by

$$H = \sum_{i=1}^{N} \left(\frac{p_i^2}{2} + a\frac{(q_{i+1} - q_i)^2}{2} + b\frac{(q_{i+2} - q_i)^2}{2} + \frac{q_i^2}{2} + \frac{q_i^4}{4} \right), \tag{A1}$$

where $a > 0$, $b \geqslant 0$. We extend our investigation to the periodic KG-LRI model, presuming $a = b = 1$ and a total of $N = 32$ nodes.

The 2D FK system is described by

$$H = \sum_{i,j=1}^{N} \left(\frac{p_{i,j}^2}{2} + a\frac{(q_{i+1,j} - q_{i,j} - \rho)^2}{2} + b\frac{(q_{i,j+1} - q_{i,j})^2}{2} - \cos(q_{i,j}) \right), \tag{A2}$$

where $a > 0$, $b > 0$, and ρ denotes the average particle distance. In our 2D FK model framework, we set a, b, and ρ to 1. Assuming $M = N = 12$, this model portrays a square grid with a total of 144 particles.

In fact, in our model, the edge link is easily learned. We set the number of epochs to 1200, and the learning rate is reduced to 1×10^{-5} after 4600 epochs. The remaining settings of the model are unchanged, and remain the same as before.

Figure 9 shows the value of $|\alpha|$ and the particle interaction relationship inferred from them. We have displayed partial particle IDs in the 2D FK system. Obviously, these systems satisfy the periodic boundary conditions.

3. The influence of particle number on model performance

In this section, we take the FK system as an example to explore the variation of model performance with the number of particles while keeping the model parameter settings unchanged. The parameters of the models are the ones with the best performance in Table III. The results are shown in Table XI. Note that the degree of freedom of the system is $2N$. Obviously, as N increases, the performance of the α-SGHN model remains relatively stable.

4. The influence of the layer of graph neural networks on model performance

In this section, we investigate the impact of the number of layers K of graph neural networks on model performance. Table XII records the performance prediction of the FK system for $K = 1$ and $K = 2$ when $N = 132$. Table XIII records the performance prediction of the rotator system for $K = 1$ and $K = 2$ when $N = 32$. The effect is better for $K = 1$ than for $K = 2$. It can be seen that using only one layer of graph neural network can effectively learn the features of the system, and the model performance is robust and not affected by the number of particles. From the results of training loss and testing loss, $K = 2$ shows a slight overfitting phenomenon.

TABLE XIII. For the rotator system, the performance of the model with $K = 1$ and $K = 2$, where $N = 32$.

	Train loss	Test loss	$\text{Trajectory}_{\text{MSE}}$	$\text{Energy}_{\text{MSE}}$
$K = 1$	4.78×10^{-9}	$4.71 \times 10^{-9} \pm 4.26 \times 10^{-9}$	$1.97 \times 10^{-7} \pm 3.57 \times 10^{-7}$	$1.32 \times 10^{-6} \pm 4.05 \times 10^{-7}$
$K = 2$	4.18×10^{-9}	$2.09 \times 10^{-8} \pm 1.11 \times 10^{-8}$	$9.52 \times 10^{-7} \pm 3.12 \times 10^{-6}$	$5.13 \times 10^{-5} \pm 4.36 \times 10^{-5}$

[1] E. Fermi, P. Pasta, S. Ulam, and M. Tsingou, *Studies of the Nonlinear Problems* (Los Alamos National Lababoratory, Los Alamos, NM, 1955).

[2] B. Hu and L. Yang, Heat conduction in the Frenkel-Kontorova model, Chaos **15**, 015119 (2005).

[3] O. M. Braun and Y. S. Kivshar, *The Frenkel-Kontorova Model: Concepts, Methods, and Applications* (Springer, Berlin, 2004).

[4] S. Yomosa, Soliton excitations in deoxyribonucleic acid (DNA) double helices, Phys. Rev. A **27**, 2120 (1983).

[5] S. Homma and S. Takeno, A coupled base-rotator model for structure and dynamics of DNA: Local fluctuations in helical twist angles and topological solitons, Prog. Theor. Phys. **72**, 679 (1984).

[6] G.-f. Zhou and C.-T. Zhang, A short review on the nonlinear motion in DNA, Phys. Scr. **43**, 347 (1991).

[7] A. S. Kovalev, One-dimensional model for describing complex topological defects in two-dimensional antiferromagnets, Low Temp. Phys. **20**, 815 (1994).

[8] A. Landau, A. Kovalev, and A. Kondratyuk, Model of interacting atomic chains and its application to the description of the crowdion in an anisotropic crystal, Phys. Status Solidi B **179**, 373 (1993).

[9] V. Y. Antonchenko, A. Davydov, and A. Zolotariuk, Solitons and proton motion in ice-like structures, Phys. Status Solidi B **115**, 631 (1983).

[10] J. Cuevas-Maraver, P. G. Kevrekidis, and F. L. Williams, *The sine-Gordon Model and its Applications: From Pendula and Josephson Junctions to Gravity and High-Energy Physics* (Springer, Berlin, 2014).

[11] Y. B. Zel'dovich, I. Y. Kobzarev, and L. B. Okun, Cosmological consequences of a spontaneous breakdown of a discrete symmetry, Zh. Eksp. Teor. Fiz. **67**, 3 (1974).

[12] T. Schneider and E. Stoll, Observation of cluster waves and their lifetime, Phys. Rev. Lett. **35**, 296 (1975).

[13] J. F. Currie, J. A. Krumhansl, A. R. Bishop, and S. E. Trullinger, Statistical mechanics of one-dimensional solitary-wave-bearing scalar fields: Exact results and ideal-gas phenomenology, Phys. Rev. B **22**, 477 (1980).

[14] G. F. Mazenko and P. S. Sahni, Statistical-mechanical treatment of kinks in a one-dimensional model for displacive phase transitions, Phys. Rev. B **18**, 6139 (1978).

[15] W.-P. Su, J. R. Schrieffer, and A. J. Heeger, Solitons in polyacetylene, Phys. Rev. Lett. **42**, 1698 (1979).

[16] R. Jackiw and J. R. Schrieffer, Solitons with fermion number 12 in condensed matter and relativistic field theories, Nucl. Phys. B **190**, 253 (1981).

[17] J. Cuevas-Maraver and P. Kevrekidis (eds.), *A Dynamical Perspective on the ϕ^4 Model*, Nonlinear Systems and Complexity (Springer, Cham, 2019).

[18] P. Kevrekidis, J. Cuevas-Maraver, and A. Saxena (eds.), *Emerging Frontiers in Nonlinear Science* (Springer, Cham, 2020).

[19] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. **378**, 686 (2019).

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

[21] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA **113**, 3932 (2016).

[22] S. Saqlain, W. Zhu, E. G. Charalampidis, and P. G. Kevrekidis, Discovering governing equations in discrete systems using PINNs, Commun. Nonlinear Sci. Numer. Simul. **126**, 107498 (2023).

[23] D. Zvyagintseva, H. Sigurdsson, V. K. Kozin, I. Iorsh, I. A. Shelykh, V. Ulyantsev, and O. Kyriienko, Machine learning of phase transitions in nonlinear polariton lattices, Commun. Phys. **5**, 8 (2022).

[24] J. Li, J. Chen, and B. Li, Gradient-optimized physics-informed neural networks (GOPINNs): A deep learning method for solving the complex modified KdV equation, Nonlinear Dyn. **107**, 781 (2022).

[25] W. Zhu, W. Khademi, E. G. Charalampidis, and P. G. Kevrekidis, Neural networks enforcing physical symmetries in nonlinear dynamical lattices: The case example of the Ablowitz-Ladik model, Phys. D (Amsterdam, Neth.) **434**, 133264 (2022).

[26] P. Jin, Z. Zhang, I. G. Kevrekidis, and G. E. Karniadakis, Learning Poisson systems and trajectories of autonomous systems via Poisson neural networks, IEEE Trans. Neural Networks Learn. Syst. **34**, 8271 (2022).

[27] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia, Hamiltonian graph networks with ODE integrators, arXiv:1909.12790.

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

[29] M. Yip, J. Mohle, and J. E. Bolander, Automated modeling of three-dimensional structural components using irregular lattices, Comput.-Aided Civ. Infrastruct. Eng. **20**, 393 (2005).

[30] N. Laskin and G. Zaslavsky, Nonlinear fractional dynamics on a lattice with long range interactions, Phys. A (Amsterdam, Neth.) **368**, 38 (2006).

[31] N. Go and H. Taketomi, Respective roles of short- and long-range interactions in protein folding, Proc. Natl. Acad. Sci. USA **75**, 559 (1978).

[32] R. Geng, J. Zu, Y. Gao, and H.-K. Zhang, Separable graph Hamiltonian network: A graph deep learning model for lattice systems, Phys. Rev. Res. **6**, 013176 (2024).

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, USA, 2016).

[34] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian neural networks, in *32nd Conference on Neural Information Processing Systems (NeurIPS 2019)* (Curran Associates, Red Hook, NY, 2020).

[35] J. Tekić, P. Mali, D. Kapor, Z. Ivić, M. Raković, and A. Rockov, *The ac Driven Frenkel-Kontorova Model* (Institut za Nuklearne Nauke Vinča, Belgrade, 2016).

[36] C. Giardina, R. Livi, A. Politi, and M. Vassalli, Finite thermal conductivity in 1D lattices, Phys. Rev. Lett. **84**, 2144 (2000).

[37] Y. Li, S. Liu, N. Li, P. Hänggi, and B. Li, 1D momentum-conserving systems: The conundrum of anomalous versus normal heat transport, New J. Phys. **17**, 043064 (2015).

[38] M. Toda, Vibration of a chain with nonlinear interaction, J. Phys. Soc. Jpn. **22**, 431 (1967).

[39] M. Toda, Mechanics and statistical mechanics of nonlinear chains, J. Phys. Soc. Jpn. **27**, 1704B (1969).

[40] M. Toda, Waves in nonlinear lattice, Prog. Theor. Phys. Suppl. **45**, 174 (1970).

[41] M. Toda, *Theory of Nonlinear Lattices* (Springer, Berlin, 1989).

[42] J. Ford, S. D. Stoddard, and J. S. Turner, On the integrability of the Toda lattice, Prog. Theor. Phys. **50**, 1547 (1973).

[43] H. Flaschka, The Toda lattice. II. Existence of integrals, Phys. Rev. B **9**, 1924 (1974).

[44] J.-M. Sanz-Serna and M.-P. Calvo, *Numerical Hamiltonian Problems* (Dover Publications, Mineola, 2018).

[45] G. Montavon, G. Orr, and K.-R. Müller, *Neural Networks: Tricks of the Trade* (Springer, Berlin, 2012).

[46] C. Danieli, E. A. Yuzbashyan, B. L. Altshuler, A. Patra, and S. Flach, Dynamical chaos in the integrable Toda chain induced by time discretization, Chaos **34**, 033107 (2024).

[47] M. Salerno, Quantum deformations of the discrete nonlinear Schrödinger equation, Phys. Rev. A **46**, 6856 (1992).

[48] T. Mithun, A. Maluckov, A. Mančić, A. Khare, and P. G. Kevrekidis, How close are integrable and nonintegrable models: A parametric case study based on the Salerno model, Phys. Rev. E **107**, 024202 (2023).

[49] G. Berman and F. Izrailev, The Fermi-Pasta-Ulam problem: Fifty years of progress, Chaos **15**, 015104 (2005).

[50] A. Vainchtein, Solitary waves in FPU-type lattices, Phys. D (Amsterdam, Neth.) **434**, 133252 (2022).

[51] P. G. Kevrekidis and J. Cuevas-Maraver (Eds.), *Fractional Dispersive Models and Applications* (Springer, Berlin, 2024).

[52] V. Koukouloyannis, P. G. Kevrekidis, J. Cuevas, and V. Rothos, Multibreathers in Klein-Gordon chains with interactions beyond nearest neighbors, Phys. D (Amsterdam, Neth.) **242**, 16 (2013).

[53] E. Granato and S. C. Ying, Dynamical transitions and sliding friction in the two-dimensional Frenkel-Kontorova model, Phys. Rev. B **59**, 5154 (1999).