

Variational integrator graph networks for learning energy-conserving dynamical systems

Shaan A. Desai 

*Machine Learning Research Group, University of Oxford Eagle House, Oxford OX2 6ED, United Kingdom
and John A. Paulson School of Engineering and Applied Sciences,
Harvard University, Cambridge, Massachusetts 02138, USA*

Marios Mattheakis 

John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA

Stephen J. Roberts 

Machine Learning Research Group, University of Oxford Eagle House, Oxford OX2 6ED, United Kingdom



(Received 16 July 2021; accepted 15 September 2021; published 30 September 2021)

Recent advances show that neural networks embedded with physics-informed priors significantly outperform vanilla neural networks in learning and predicting the long-term dynamics of complex physical systems from noisy data. Despite this success, there has only been a limited study on how to optimally combine physics priors to improve predictive performance. To tackle this problem we unpack and generalize recent innovations into individual inductive bias segments. As such, we are able to systematically investigate all possible combinations of inductive biases of which existing methods are a natural subset. Using this framework we introduce *variational integrator graph networks*—a novel method that unifies the strengths of existing approaches by combining an energy constraint, high-order symplectic variational integrators, and graph neural networks. We demonstrate, across an extensive ablation, that the proposed unifying framework outperforms existing methods, for data-efficient learning and in predictive accuracy, across both single- and many-body problems studied in the recent literature. We empirically show that the improvements arise because high-order variational integrators combined with a potential energy constraint induce coupled learning of generalized position and momentum updates which can be formalized via the partitioned Runge-Kutta method.

DOI: [10.1103/PhysRevE.104.035310](https://doi.org/10.1103/PhysRevE.104.035310)

I. INTRODUCTION

Accurately and efficiently learning the time evolution of energy-conserving dynamical systems from limited, noisy data is a crucial challenge in numerous domains including robotics [1], spatiotemporal dynamical systems [2], interacting particle systems [3], and materials [4]. To address this challenge, researchers have shown that enriching neural networks with well-chosen inductive biases such as Hamiltonians [5], integrators [6–8], and graphs [9–11] can significantly improve the learning of complex dynamical systems over vanilla neural networks. Fundamentally, physics-informed learning biases constrain neural networks to uncover and preserve the underlying physical process of a system under investigation. Most methods in this space typically combine multiple individual inductive biases to improve the overall predictive performance. However, no study extensively quantifies the performance uplift induced by an individual bias within these combinations. In addition, it remains an open challenge to identify the best combination.

In this paper, we unpack recent innovations by grouping their inductive biases into generalized segments. We then systematically investigate all possible combinations of these biases. In doing so, existing methods are naturally explored and generalized as they form a subset of the entire ablation. Using this we identify and develop *variational integrator graph networks* (VIGNs)—a novel method that brings together the core benefits of multiple inductive biases and unifies existing approaches which bring integrative, symplectic, and structural form to modeling energy-conserving physical systems. We show that higher-order variational integrators, formalized via the partitioned Runge-Kutta (PRK) method, can be used to couple position and momentum updates for more precise learning. To benchmark our method we conduct an extensive ablation study across recent developments in physics-informed learning biases and show that VIGNs consistently outperform existing baselines including Hamiltonian ODE graph networks (HOGNs) [12], ODE graph networks (OGNs) [12], Hamiltonian neural networks (HNNs) [5], and variational integrator networks (VINs) [6] across energy conserved noisy many-body dynamical systems. The outcomes of this study are therefore particularly useful for practitioners who are able to collect noisy position and momenta data of an energy-conserving many-body system for which the underlying equations of motion are unknown and they want (a) to

*Also at John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA; shaan@robots.ox.ac.uk

TABLE I. Three major methods used to approximate state derivatives $\dot{s} = [\dot{q}, \dot{p}]$. Delta: approximate state derivatives directly. Hamiltonian: approximate state derivatives by learning a Hamiltonian, then differentiating this with regard to input. Potential: approximate q via potential assuming that $\dot{q} = M^{-1}p$. Abbreviations centered above equations refer to the methods presented in this study.

Type	MLP-based method	Graph-based method
Delta	DN $[q_1, p_1, \dots, q_n, p_n] \rightarrow \text{MLP} \rightarrow [\dot{q}_1, \dot{p}_1, \dots, \dot{q}_n, \dot{p}_n]$	DGN $\begin{bmatrix} q_1, p_1 \\ \dots \\ q_n, p_n \end{bmatrix} \rightarrow \text{GN} \rightarrow \begin{bmatrix} \dot{q}_1, \dot{p}_1 \\ \dots \\ \dot{q}_n, \dot{p}_n \end{bmatrix}$
Hamiltonian	HNN $[q_1, p_1, \dots, q_n, p_n] \rightarrow \text{MLP} \rightarrow H(q, p) \xrightarrow{\text{autograd}} \left[\frac{\partial H}{\partial p_1}, \frac{-\partial H}{\partial q_1}, \dots, \frac{\partial H}{\partial p_n}, \frac{-\partial H}{\partial q_n} \right]$	HOGN $\begin{bmatrix} q_1, p_1 \\ \dots \\ q_n, p_n \end{bmatrix} \rightarrow \text{GN} \rightarrow H(q, p) \xrightarrow{\text{autograd}} \begin{bmatrix} \frac{\partial H}{\partial p_1}, \frac{-\partial H}{\partial q_1} \\ \dots \\ \frac{\partial H}{\partial p_n}, \frac{-\partial H}{\partial q_n} \end{bmatrix}$
Potential	PNN $[q_1, \dots, q_n] \rightarrow \text{MLP} \rightarrow E_{\text{potential}}(q) \xrightarrow{\text{autograd}} \left[\frac{-\partial E_{\text{pot}}}{\partial q_1}, \dots, \frac{-\partial E_{\text{pot}}}{\partial q_n} \right]$	PGN $\begin{bmatrix} q_1 \\ \dots \\ q_n \end{bmatrix} \rightarrow \text{GN} \rightarrow E_{\text{potential}}(q) \xrightarrow{\text{autograd}} \begin{bmatrix} \frac{-\partial E_{\text{pot}}}{\partial q_1} \\ \dots \\ \frac{-\partial E_{\text{pot}}}{\partial q_n} \end{bmatrix}$

accurately learn the underlying dynamics such that they can predict trajectories for unseen initial conditions at inference time and (b) to learn a robust network that can maintain accurate predictions beyond the training time horizon, i.e., extrapolate beyond the training data range with formal guarantees that there is a valid dynamical system bridge between the last observed data and future predictions.

In Sec. II we describe the individual learning biases that comprise VIGNs. We then outline the details of the proposed architecture in Sec. III. In Sec. IV we demonstrate the performance of VIGNs across numerous well-known energy-conserving physical systems such as the simple pendulum and the many-body interacting spring particle system. Finally, in Sec. V we unpack the performance uplift and highlight some of the limitations of the existing method.

II. BACKGROUND

Numerous recent approaches tackle learning from physical data, but of them three methods stand out: graph networks [10], HNNs [5], and networks with embedded integrators [6,8]. VIGNs allow us to combine the major strengths of each approach and hence form a simple, unifying framework for learning the temporal behavior of dynamical systems. We briefly review the methods in the following sections.

To begin with, dynamical systems can be represented by systems of differential equations in the form

$$\dot{s} = f(s, t), \quad (1)$$

where $s = (\mathbf{q}, \mathbf{p})^T$ is a state vector composed of position and momentum, t is time, and f is an arbitrary function of time and the state vector. Physics-informed networks attempt to approximate f using a neural network (NN). In what follows we describe three main approaches used to approximate f : delta networks, Hamiltonian networks, and potential networks. The three machine learning methods are summarized in Table I.

A. Delta networks

A straightforward approach to learning dynamics from data is to use an NN to transform a current state vector s_t to a future state vector s_{t+1} . Early research [13], however, found that using a neural network to learn the gradients \dot{s} and then applying an integration resulted in better performance. In this paper we refer to networks that directly compute \dot{s}_t from s_t as delta networks (DNs). As an aside, these networks are referred to as “baseline” in [5]. In [12] these gradients are computed with graphs and referred to as ODE graph networks. To maintain consistency we refer to multilayer perceptron (MLP)–based methods as delta networks and graph-based methods as delta graph networks (DGNs).

B. Hamiltonian neural networks

In designing an NN, the typical operation of interest for many physical systems is one which accurately models the time evolution of the system. Recently, the work of [5] demonstrated that predictions through time can be improved using HNNs, which endow models with a Hamiltonian constraint. Given a system with N particles, the Hamiltonian \mathcal{H} is a scalar function of the canonical position $\mathbf{q} = (q_1, q_2, \dots, q_N)$ and momentum $\mathbf{p} = (p_1, p_2, \dots, p_N)$. In representing physical systems with a Hamiltonian, one can simply use Hamilton’s equations to extract the time derivatives of the inputs by differentiating the Hamiltonian with respect to its variables as

$$\dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \quad \dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}, \quad (2)$$

where $\dot{a} = \frac{da}{dt} \forall a(t)$. As a consequence, it is noted in [5] that by training a network to learn \mathcal{H} given inputs $[\mathbf{q}, \mathbf{p}]$, the system’s state-time derivatives can be naturally extracted through autodifferentiation of the predicted Hamiltonian with respect to the inputs. The Hamiltonian in most systems represents the total mechanical energy of the system and is therefore a powerful inductive bias that can be utilized to evolve a physical state while maintaining energy conservation. Indeed such approaches can be extended to damped and forced systems

via port Hamiltonians [14]. In addition, HNNs combined with graphs are referred to as HOGNs. [12].

C. Potential neural networks

Separable Hamiltonians found in many dynamical systems can be written as $\mathcal{H}(\mathbf{q}, \mathbf{p}) = E_{\text{kinetic}}(\mathbf{p}) + E_{\text{potential}}(\mathbf{q})$. Typically, for rigid-body systems, the form of the kinetic energy is $E_{\text{kinetic}} = \frac{M^{-1}}{2} \mathbf{p}^2$, where M is an inertial mass matrix that connects the generalized momenta $\dot{\mathbf{q}}$ to the canonical momenta \mathbf{p} such that $\dot{\mathbf{q}} = \frac{\partial E_{\text{kinetic}}}{\partial \mathbf{p}} = M^{-1} \mathbf{p}$. The authors of [15] and [6] exploit this simplification when dealing with generalized position \mathbf{q} and velocity $\dot{\mathbf{q}}$ to collapse Eq. (2) into

$$\frac{d\mathbf{q}}{dt} = \dot{\mathbf{q}}, \quad \frac{d\dot{\mathbf{q}}}{dt} = -M^{-1} \frac{\partial E_{\text{potential}}(\mathbf{q})}{\partial \mathbf{q}}. \quad (3)$$

Equation (3) allows us to learn a single function $E_{\text{potential}}$, with fewer network weights needed to learn a Hamiltonian and a single backpropagation with respect to \mathbf{q} as opposed to $[\mathbf{q}, \mathbf{p}]$ for HNNs. Further, it gives us the flexibility to learn the inertial mass matrix by explicitly learning M_θ rather than nesting it in $\tilde{U}_\theta = M^{-1}U$. As this type of network has not been introduced formally as an individual inductive bias, we coin the term potential neural networks (PNNs) in reference to them and potential graph networks (PGNs) in reference to their graph-based variants.

In the case where we only have canonical coordinates, potential networks can still be used but a separate NN needs to be designed to learn the inertial matrix M [6].

D. Embedded integrators

One approach to solving Eq. (1) is to parametrize the function f by one of the methods just described and minimize the euclidean distance between the predicted state-time derivatives $\hat{\mathbf{s}}$ and the ground truth \mathbf{s}_{gt} derivatives. One challenge in doing this is that it assumes access to the ground truth state-time derivatives, which can be hard to extract. To circumvent this problem, researchers embed a numerical integrator into the training process [8,16]. Formally this equates to integrating both sides of Eq. (1) such that for *short-range integration*,

$$\mathbf{s}_{t+\Delta t} = \mathbf{s}_t + \int_t^{t+\Delta t} f_\theta(\mathbf{s}, t) dt, \quad (4)$$

and for *long-range integration*,

$$\mathbf{s}_{t+n} = \mathbf{s}_t + \int_t^{T_{\text{max}}} f_\theta(\mathbf{s}, t) dt, \quad (5)$$

where θ are the weights of the neural network. The short-range integration involves a single discrete step Δt , whereas the long-range integration involves a sequence of discrete steps to the final time $T_{\text{max}} = n\Delta t$. It can be shown that if we integrate the system from t to T_{max} , the network above ends up being a composition of multiple transformations as would be found in recurrent neural networks and residual networks [8].

1. Symplecticity

While the embedded integrator resolves the challenge of having to obtain state-time derivatives, it introduces a new

complexity—the choice of integrator. Numerical integrators are chosen based on their precision and truncation error, however, when dealing with Hamiltonian systems an additional factor to consider is whether the integrator preserves the energy of the system. It has been shown that symplectic integrators can preserve the energy during integration, making them versatile for long-range integrations of conserved quantities [17]. The performance of these integrators on a range of different systems is outlined in Supplemental Material Ref. [18], where we see that for long-range integrations even low-order symplectic integrators are as performant as high-order Runge-Kutta (RK) methods in terms of energy conservation.

While VINs [6] and symplectic recurrent neural networks [19] both illustrate how an embedded symplectic integrator improves learning over traditional RK methods, they only do so for low-order methods. To extend our investigation to higher-order symplectic integrators, we need to study PRK methods.

Typically, RK methods can be described by Butcher tables (see Supplemental Material Ref. [18]), and if the coefficients satisfy certain conditions then they can be made symplectic [17]. However, the additional symplecticity constraint on the table of coefficients forces the integration scheme to be implicit. While implicit integrators are powerful, they require a root finding approach. Introducing such complexity into an embedded NN is possible but significantly complicates the backpropagation technique. As such, it is important to establish whether explicit symplectic methods can be developed. Fortunately, by creating separate Butcher tables for position and momentum it is indeed possible to describe a PRK method with coefficients that result in an explicit symplectic integration scheme. Note that variational integrators, derived through variational calculus, can be described by PRK methods [17]. As a consequence, it is possible to generalize the result of VINs to higher-order methods (see Supplemental Material Ref. [18] for details).

E. Graph neural networks

Until this point, we have strictly been speaking of NNs as multilayer perceptrons. However, interacting particle systems have structure and therefore the state of a physical system can be represented by a graph $G = (u, V, E)$ [9]. For example, a node (V) can be a particle in a many-body problem. These nodes can be used to represent the core features of the particle, like its position, momentum, and mass, and other particle constants. Edges (E) can represent forces between the particles, and the ‘globals’ (u) can represent universal constants such as the air density and gravitational constant. By representing physical systems this way, we are able to preserve the structure of our data and find solutions that conform to this prior structure using graph NNs [9–11,20–25]. Graph neural networks carry out a sequence of transformations to the graph nodes and edges to update the graph parameters. The representation is therefore powerful for many-body systems primarily because the graph networks can operate within the known constraints of physical systems.

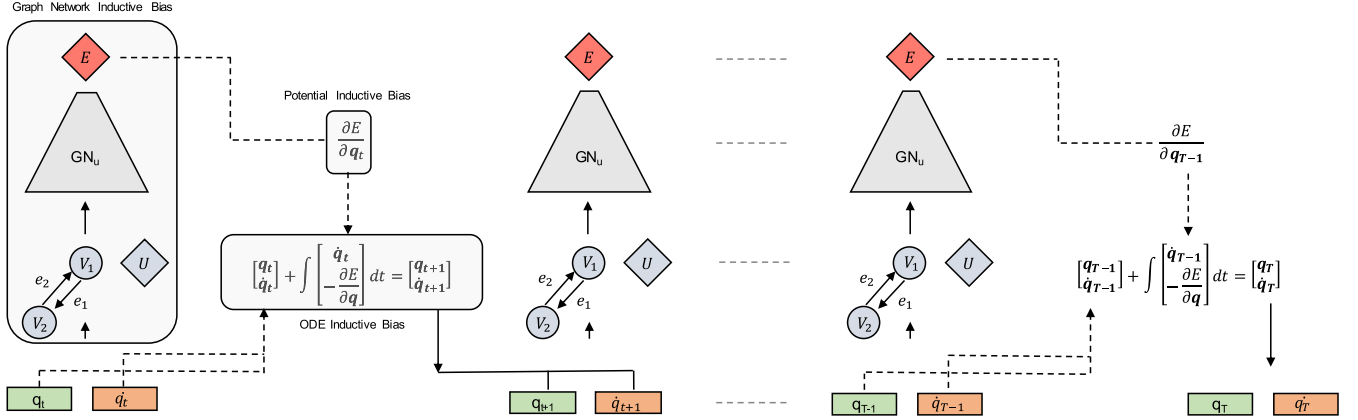


FIG. 1. The architecture for our method takes as input the position vector $[\mathbf{q}]$ and feeds it through a graph network to compute the potential energy $E(\mathbf{q})$. Using backpropagation, the update for $\dot{\mathbf{q}}$ is computed and the input state is integrated one step. Continuing this sequence yields a multistep integration scheme.

F. Related work

The notion of embedding physically informed inductive biases in NNs can be found in numerous early works aimed at modeling materials [13,26–29]. For example, early efforts by Witkoskie and Doren [26] demonstrate that in contrast to directly learning a potential energy surface, the inclusion of gradients in the learning process can drive a network to accurately model the forces. However, most materials modeling frameworks are task specific and usually do not generalize well.

More general approaches that capture physical laws include search algorithms [30], symbolic learning [22], and regressive techniques [31–33]. In addition, graphs have also been presented as natural inductive biases in modeling physics [9,20].

Neural ODEs [8] have also resparked interest in inductive biases for differential equations. Inspired by this work, [5] and [34] show that a neural network can be used to predict a Hamiltonian which can be differentiated with respect to the input (\mathbf{p} and \mathbf{q}) to obtain the time derivatives of the system. With these derivatives accurately learned, a neural ODE-type integration scheme can be used to evolve a system. This general approach has formed the basis for many advancements within physical learning [6,11,12,16,35].

Although HNNs predict dynamics for few-body systems well (e.g., a swinging pendulum or mass-spring system), they are not readily adaptable to large N -body problems when the input dimension increases. The work in [12] shows that graph networks are ideal for resolving this type of system because they can operate on structured data, i.e., the system does not need to be vectorized as would be the case for multilayer feed-forward neural networks.

Inspired by neural ODEs, variational integrator networks [6] propose a neural network whose architecture matches the discrete equation of motion governing the dynamical system, as derived by applying the Euler-Lagrange equations to a discretized action integral. The paper indicates major benefits when using the method for noisy data, as well as providing precise energy and momentum conservation.

While it is clear that the constrained HNN [36] is capable of solving Hamiltonian systems more efficiently, it assumes

that we have access to Cartesian coordinates for all systems and requires explicit rigid-body constraints.

Our method brings together the inductive biases presented in all these papers and leverages them to solve large many-body problems in noisy data settings.

III. METHOD

The architecture for our method is shown in Fig. 1. At training time it is assumed that, for any given system, we have access to noisy state variables $\mathbf{s}(t) = [\mathbf{q}(t), \mathbf{p}(t)]$ for every particle and we know that the total energy of the system is conserved. The unknowns are the Hamiltonian, the underlying differential equations, and the noise. The objective for all the networks is therefore to approximate the underlying equations of motion. Unlike DNs or Hamiltonians, our network takes as input state vectors $\mathbf{s} = (\mathbf{q})$ representing generalized coordinates and learns to predict the potential function and its derivatives with respect to the inputs. Note that we adopt the potential neural network so we require generalized coordinates. However, the transition to a Hamiltonian NN is straightforward. Since the input training data can be described by a graph, we show vertices V_i , edges E_{ij} , and globals u as input to the graph network GN_u to predict the potential energy $E_{\text{potential}}$. The key difference between our graph approach and the HOGN [12] is that our network only takes the position \mathbf{q} as input, i.e., the nodes of the graph only have position.

Succinctly, for DNs and Hamiltonian networks the input to the network is a state vector into the MLP or graph network of the form $[\mathbf{q}, \mathbf{p}]$, i.e., both position and momenta are fed in to learn $[\dot{\mathbf{q}}, \dot{\mathbf{p}}]$. For potential networks an assumption is made about the form of $\dot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{p}$, and therefore, the only input required is the position since the only thing that needs to be learned is $\ddot{\mathbf{q}}$.

In the noiseless setting, for all systems the training loss is defined as the mean-squared error (MSE) across all time steps and across all state vectors such that

$$\mathcal{L} = \frac{1}{N} \sum_{t=1}^{T_{\max}} (\mathbf{s}(t) - \hat{\mathbf{s}}(t))^2, \quad (6)$$

where $\mathbf{s} = [\mathbf{q}, \mathbf{p}]$. Note that this is a standard approach in data-driven discovery of Hamiltonian and Lagrangian systems.

In the noisy setting, we follow an approach similar to that in [6] and compute the full log-likelihood of the predicted state vector \mathbf{s}_{pred} as

$$P(\mathbf{s}_{\text{pred}}|\mathbf{s}, \sigma^2) = \prod_{t=1}^{T_{\text{max}}} \mathcal{N}(\mathbf{s}_{\text{pred}}(t)|\mathbf{s}(t), \sigma^2 I), \quad (7)$$

where \mathcal{N} is a Gaussian distribution, σ^2 reflects the variance, and I is an identity matrix.

To benchmark the performance of our method we conduct an extensive ablation study. Our ablation iterates across all combinations of the inductive biases described in Sec. II. Namely, it includes both graph and nongraph methods that learn either the state derivatives directly [5,12], the Hamiltonian (Hamiltonian networks), or the potential function (potential networks). We use first- through fourth-order integrators that are symplectic and nonsymplectic. We also iterate over a multistep integration scheme with step sizes of 1, 5, and 10 to account for short-, mid-, and long-range integrations during training. Note that we can indeed integrate for more than 10 steps but this increases the memory requirement. In addition, since the ablation iterates over all possible combinations of inductive biases, existing methods in the literature are naturally covered. For example, VINs can be described as low-order, long-range symplectic integrators coupled with potential networks, while HOGNs couple low- and high-order, short-range integrators with Hamiltonians and graphs (see Supplemental Material Ref. [18] for details).

Since our work iterates across all these methods we adopt a new naming convention for convenience. We refer to networks that combine graphs with potential networks as potential graph networks (PGNs). Note that VIGNs are PGNs under symplectic integration.

IV. EXPERIMENTS

We carry out our experiments on numerous data sets used in the recent literature and describe their configurations here (see Supplemental Material Ref. [18] for full training and testing configurations).

A. Training

For all the systems we investigate the training data are generated using an eighth-order RK method with $r_{\text{tol}} = 10^{-12}$ so that the ground truth is precise and conserves energy. The noise model for all systems is chosen to maintain a noise-to-signal ratio of less than 30%, which allows us to investigate which architecture is the most robust to noisy data. For all noisy training configurations, the noise source is a Gaussian, $\mathcal{N}(0, \sigma)$. The noise is added to each state vector similarly to the approach taken in [5,6].

B. Testing

To evaluate the performance of our models, we sample 50 initial conditions and integrate these systems to three times the training time horizon $3T_{\text{max}}$. In other words, the true performance of the model is tested by evaluating points beyond the

training regime. For each set of initial conditions we compute the MSE across the entire trajectory between the prediction and the ground truth states. Note that some of our 50 sampled initial conditions can be slightly outside the training regime, which can lead to a few poorly predicted trajectories by all the models. To prevent this skewing our final reported results, we compute the geometric mean, a measure of central tendency, of the MSEs computed across the 50 initial conditions.

Here, we describe the systems investigated. A list of all the experimental results can be found in Supplemental Material Ref. [18]. In the following discussion, we only report the results of our ablation with fourth-order methods, for clarity. The results for these systems are summarized in Fig. 2 and the others are summarized in Supplemental Material Ref. [18].

C. Mass-spring system

We start by considering a simple frictionless one-dimensional mass-spring system modeled by the Hamiltonian as

$$\mathcal{H} = \frac{p^2}{2m} + k \frac{q^2}{2}. \quad (8)$$

For simplicity we set the mass m and spring constants to 1 without loss of generality. As is done in [5], the training data are sampled uniformly in the energy range 0.5 to 4.5.

D. Pendulum system

We carry out testing on a one-dimensional pendulum, which is more complex than the simple mass spring because it is a nonlinear system. The Hamiltonian is modeled as

$$\mathcal{H} = \frac{p^2}{2ml^2} + mgl(1 - \cos(q)), \quad (9)$$

where the mass and lengths are set to 1 and g is set to 9.81. We use 25 initial conditions which satisfy the condition that the total energy lies in [1.3,2.3] for training. Note that this energy yields strong nonlinear behavior.

E. Two-body gravitational system

The two-body system represents a particle system in which the forces between particles are modeled by a gravitational force. The system can be represented by the Hamiltonian

$$\mathcal{H} = \sum_{i=1}^2 \frac{|p_i|^2}{2m_i} - \sum_{1 \leq i < j \leq 2} g \frac{m_i m_j}{|q_j - q_i|^2}, \quad (10)$$

where we set masses to 1 and g to 1 without loss of generality. The coordinates are assumed to be scaled by the reduced mass μ ; in addition, the center of mass is assumed to be fixed at 0. We use 20 initial conditions which satisfy the condition that the radius of a particle's trajectory is uniformly sampled between [0.5,1.5] as is done in [5]. We illustrate the rollout of a single test point in Fig. 3.

F. Three-body gravitational system

The three-body system represents a particle system in which the force between particles is modeled by a

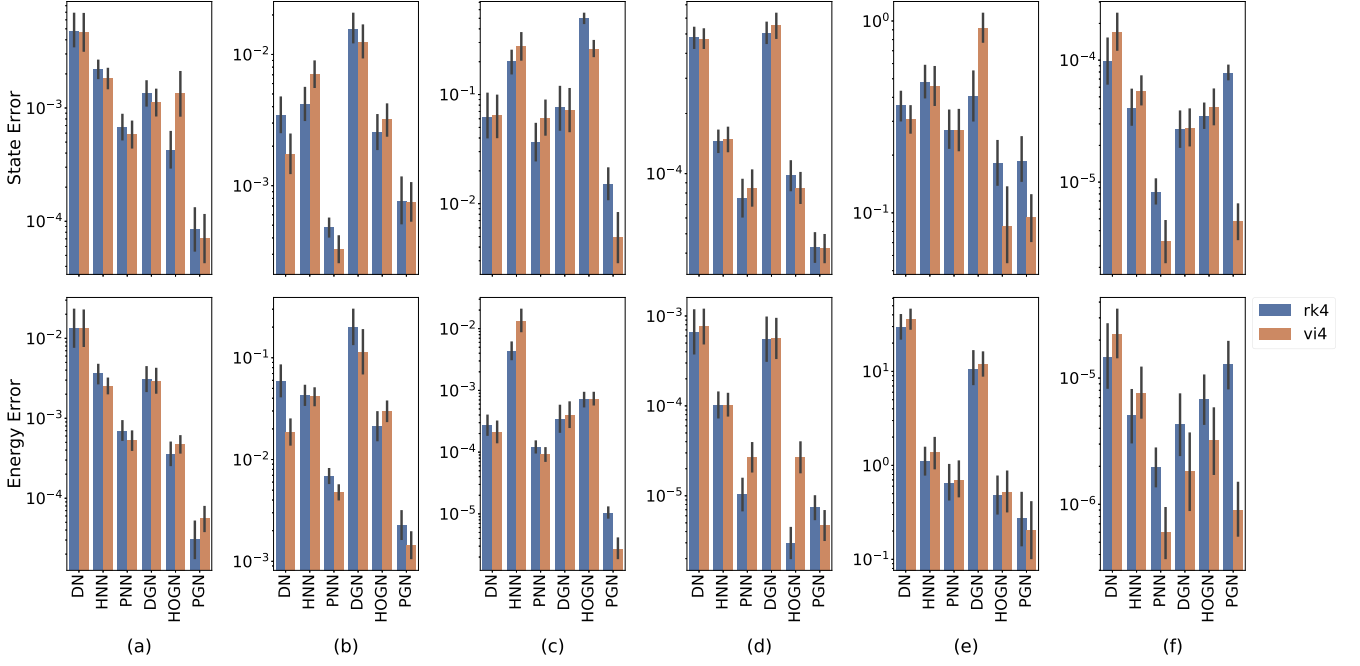


FIG. 2. State and energy geometric mean (and $\pm\sigma$ standard errors) of the rollout MSE for 50 initial conditions of the (a) mass-spring, (b) pendulum, (c) two-body gravitational, (d) three-body gravitational, (e) five-body spring force, and (f) Henon-Heiles systems. The results are reported for models trained on noisy data. We see that high-order, long-range integrators coupled with potential networks perform the best, with their graph variants showing added versatility in both single- and many-body systems.

gravitational force. The system can be represented by

$$\mathcal{H} = \sum_{i=1}^3 \frac{|p_i|^2}{2m_i} - \sum_{1 \leq i < j \leq 3} g \frac{m_i m_j}{|q_j - q_i|^2}, \quad (11)$$

where we set masses to 1 and g to 1.

G. N -body spring force system

We also carry out our experiments on a data set similar to that found in [12]. We develop an N -body data set, where the interaction force between particles is modeled by $\mathbf{F}_{ij} = -k_i k_j (\mathbf{q}_i - \mathbf{q}_j)$ following the same sampling procedure as in

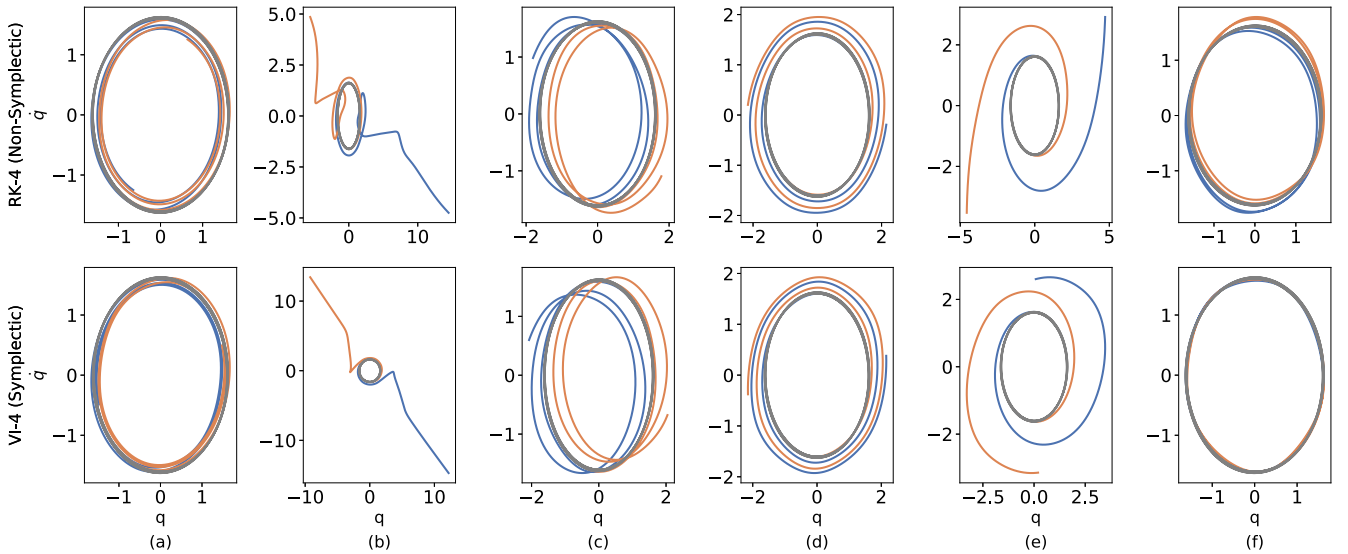


FIG. 3. Qualitative evolution of a single test state of the two-body gravitational problem trained on noisy data for (a) DNs, (b) HNNs, (c) PNNs, (d) DGNs, (e) HOGNs, and (f) PGNs. We see that the PGN (f) is the most performant method, as it stays close to the ground truth lines (shown in black), with the variational integrator variant of PGN (VIGN) doing the best (bottom right).

[12], leading to a Hamiltonian as

$$\mathcal{H} = \frac{1}{2} \sum_i^N \frac{|\mathbf{p}_i|^2}{2m_i} + \sum_i^N \sum_{i < j}^N \frac{1}{2} k_{ij} (\mathbf{q}_i - \mathbf{q}_j)^2. \quad (12)$$

The overall mechanism closely aligns with important problems in N -particle systems used to model complex materials in solid state physics.

H. Hénon-Heiles system

The systems investigated so far do not exhibit chaotic motion. Hénon-Heiles is a system used to describe the nonlinear motion of a star around a galactic center and defined by the Hamiltonian

$$\mathcal{H} = \frac{1}{2} \mathbf{p}^2 + \frac{1}{2} \mathbf{q}^2 + \lambda \left(q_x^2 q_y - \frac{q_y^3}{3} \right), \quad (13)$$

which exhibits chaotic motion, i.e., small perturbations on initial conditions lead to drastically different trajectories. It has been shown that Hamiltonian neural networks can be used to capture the dynamics in this setting [35,37].

We show the test results of all systems in Fig. 2 with models trained on noisy data using 10-step integration during training. Most notable is that potential-based networks consistently perform the best, with potential graph networks doing the best in terms of the state and energy MSEs for most systems. In addition, the performances of RK-4 and the symplectic fourth-order integrator are relatively comparable across most systems, indicating that fourth-order symplecticity constraints are more relevant for very large integration time steps such as in our two-body problem or chaotic trajectories like Hénon-Heiles.

V. CONCLUSION

From our extensive ablation across both noisy and non-noisy training data we have found that the VIGN is consistently the best or next-best method in the noisy data setting. We believe that the reason inductive biases are not as successful with noiseless data is that they overfit to the training set in addition to the networks attempting to compensate for the error induced by numerical integration. Noise naturally reduces the overfitting and thus allows the VIGN/PGN to do well. Once we identified the VIGN as the most performant we needed to establish which biases were most useful. In Fig. 2 we show that a potential network bias drives the largest performance increase against other approaches. We see that graph-based methods are more performant for larger many-body systems as is expected but remain robust in the single-body settings too. We also find that using a long-range integration scheme for training in the noisy data setting

tends to help the overall performance of all methods, as it encourages the network to learn the underlying dynamics using multiple noisy points rather than one. Indeed, we note that the performance depends heavily on the system under investigation. For example, in Fig. 3, we observe that using a Hamiltonian constraint can lead to diverging trajectories at inference time for the two-body system, when noisy training data are considered. A similar performance of HNNs trained under noisy data has been reported in [6], implying that networks with Hamiltonian constraints are capable of overfitting to the noise at training time and thus, occasionally, fail to identify the underlying dynamics. Hence, the choice of the prior, defining the embedded neural network physics, can play a dominant role.

Although we do note that symplectic integrators are good for long-range integration and energy preservation, their performance in many of the systems we investigate is only marginally better than that of the Runge-Kutta. In preserving the energy, symplectic integrators are capable of drifting from the ground truth state while ensuring energy conservation, which explains why we occasionally see RK methods doing better at state and energy conservation. However, we do note that symplectic integrators of low order are much better at preserving the dynamics than low-order Runge Kutta methods. This result is consistent with the theory of symplectic integrators.

We have shown that learning dynamics from data strongly benefits from well-chosen inductive biases. We present VIGNs as one such method capable of learning from scarce, noisy data across a diverse array of domains. We highlight that VIGNs are able (1) to unify graph networks, ODEs, potential networks, and symplectic inductive biases for learning precise trajectories in large many-body systems, (2) to make learning data efficient, (3) to maintain flexibility in learning from generalized momenta and easy extension to canonical coordinates, and (4) to build higher-order variational integrators through PRK methods. We note that different networks produce variable results in different settings, as would be expected. In spite of this, we see that potential-based networks are consistently the best at achieving the lowest state and energy MSE and, more importantly, that the VIGN is always first or second among these methods. We believe this is an important outcome for practitioners, as the VIGN can reliably be used across systems to attain excellent performance in terms of both state and energy error.

ACKNOWLEDGMENT

We would like to thank the Rhodes Trust for supporting this research.

- [1] M. Lutter, C. Ritter, and J. Peters, Deep lagrangian networks: Using physics as model prior for deep learning, in *International Conference on Learning Representations*, New Orleans, LA, May 6–9 (2019); <https://iclr.cc/Conferences/2019>.
- [2] G. Barmbaris, G. Neofotistos, M. Mattheakis, J. Hizanidis, G. Tsironis, and E. Kaxiras, Robust prediction of complex

spatiotemporal states through machine learning with sparse sensing, *Phys. Lett. A* **384**, 126300 (2020).

- [3] Z. Li, S. Yang, G. Song, and L. Cai, Conformation-guided molecular representation with Hamiltonian neural networks, in *International Conference on Learning Representations* (2021); <https://papertalk.org/papertalks/28603>.

- [4] H. Zhai and G. Hu, Inferring micro-bubble dynamics with physics-informed deep learning, [arXiv:2105.07179](https://arxiv.org/abs/2105.07179) [physics].
- [5] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian Neural Networks, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Red Hook, NY, 2019), pp. 15379–15389.
- [6] S. Saemundsson, A. Terenin, K. Hofmann, and M. Deisenroth, Variational integrator networks for physically meaningful embeddings, in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (PMLR, 2020), pp. 3078–3087.
- [7] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, Reversible architectures for arbitrarily deep residual neural networks, in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, edited by S. A. McIlraith and K. Q. Weinberger (AAAI Press, 2018).
- [8] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Neural ordinary differential equations, in *Advances in Neural Information Processing Systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Red Hook, NY, 2018), pp. 6571–6583.
- [9] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer *et al.*, Relational inductive biases, deep learning, and graph networks, [arXiv:1806.01261](https://arxiv.org/abs/1806.01261) [cs, stat].
- [10] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. A. Riedmiller, R. Hadsell, and P. W. Battaglia, Graph networks as learnable physics engines for inference and control, in *ICML*, edited by G. Jennifer Dy and A. Krause, Proceedings of Machine Learning Research Vol. 80 (PMLR, 2018), pp. 4467–4476.
- [11] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, Learning to simulate complex physics with graph networks, in *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 119 (PMLR, 2020), pp. 8459–8468.
- [12] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia, Hamiltonian graph networks with ODE integrators, [arXiv:1909.12790](https://arxiv.org/abs/1909.12790) [physics].
- [13] A. Pukrittayakamee, M. Malshe, M. Hagan, L. M. Raff, R. Narulkar, S. Bukkapatnum, and R. Komanduri, Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks, *J. Chem. Phys.* **130**, 134101 (2009).
- [14] S. Desai, M. Mattheakis, D. Sondak, P. Protopapas, and S. J. Roberts, Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems, [arXiv:2107.08024](https://arxiv.org/abs/2107.08024).
- [15] H. Yu, X. Tian, W. E, and Q. Li, OnsagerNet: Learning stable and interpretable dynamics using a generalized Onsager principle, [arXiv:2009.02327](https://arxiv.org/abs/2009.02327) [physics].
- [16] Y. D. Zhong, B. Dey, and A. Chakraborty, Symplectic ODE-Net: Learning Hamiltonian dynamics with control, in *International Conference on Learning Representations* (2020).
- [17] J. E. Marsden and M. West, Discrete mechanics and variational integrators, *Acta Numer.* **10**, 357 (2001).
- [18] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevE.104.035310> for training regimes, ablations, and symplectic integrators.
- [19] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, Symplectic recurrent neural networks, in *International Conference on Learning Representations* (2020).
- [20] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu, Interaction networks for learning about objects, relations and physics, in *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16* (Curran Associates Inc., Barcelona, Spain, 2016), pp. 4509–4517.
- [21] S. Seo and Y. Liu, Differentiable physics-informed graph networks, [arXiv:1902.02950](https://arxiv.org/abs/1902.02950) [cs, stat].
- [22] M. Cranmer, A. Sanchez Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, Learning symbolic physics with graph networks, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 17429–17442.
- [23] S. Seo, C. Meng, and Y. Liu, Differentiable physics-aware difference graph networks for sparsely-observed dynamics, in *International Conference on Learning Representations* (2020).
- [24] C. L. Lamb, A. Garcez, M. Gori, O. R. M. Prates, P. H. C. Avelar, and M. Y. Vardi, Graph neural networks meet neural-symbolic computing: A survey and perspective, in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (International Joint Conferences on Artificial Intelligence Organization, 2020), pp. 4877–4884.
- [25] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, Lagrangian neural networks, [arXiv:2003.04630](https://arxiv.org/abs/2003.04630) [physics, stat].
- [26] J. B. Witkoskie and D. J. Doren, Neural network models of potential energy surfaces: Prototypical examples, *J. Chem. Theory Comput.* **1**, 14 (2005).
- [27] J. S. Smith, O. Isayev, and A. E. Roitberg, ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost, *Chem. Sci.* **8**, 3192 (2017).
- [28] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning, *Phys. Rev. Lett.* **108**, 058301 (2012).
- [29] K. Yao, J. E. Herr, D. Toth, R. Mckintyre, and J. Parkhill, The TensorMol-0.1 model chemistry: A neural network augmented with long-range physics, *Chem. Sci.* **9**, 2261 (2018).
- [30] D. J. Hills, A. M. Grütter, and J. J. Hudson, An algorithm for discovering Lagrangians automatically from data, *Peer J. Comput. Sci.* **1**, e31 (2015).
- [31] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, Discovering Physical Concepts with Neural Networks, *Phys. Rev. Lett.* **124**, 010508 (2020).
- [32] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science* **324**, 81 (2009);
- [33] B. de Silva, D. M. Higdon, S. L. Brunton, and J. N. Kutz, Discovery of physics from data: Universal laws and discrepancy models, [arXiv:1906.07906](https://arxiv.org/abs/1906.07906) [physics, stat].
- [34] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, Hamiltonian generative networks, in *International Conference on Learning Representations* (2020).

- [35] A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, Physics enhanced neural networks predict order and chaos, [Phys. Rev. E **101**, 062207 \(2020\)](#).
- [36] M. Finzi, K. A. Wang, and A. G. Wilson, Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 13880–13889.
- [37] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, Hamiltonian neural networks for solving differential equations, [arXiv:2001.11107](#) [physics].