Variational Integrator Networks for Physically Meaningful Embeddings

Steindór Sæmundsson
Imperial College London
s.saemundsson16@imperial.ac.uk

Alexander Terenin
Imperial College London
a.terenin17@imperial.ac.uk

Katja Hofmann
Microsoft Research
katja.hofmann@microsoft.com

Marc Peter Deisenroth
University College London
m.deisenroth@ucl.ac.uk

Abstract

Learning workable representations of dynamical systems is becoming an increasingly important problem in a number of application areas. By leveraging recent work connecting deep neural networks to systems of differential equations, we propose *variational integrator networks*, a class of neural network architectures designed to ensure faithful representations of the dynamics under study. This class of network architectures facilitates accurate long-term prediction, interpretability, and data-efficient learning, while still remaining highly flexible and capable of modeling complex behavior. We demonstrate that they can accurately learn dynamical systems from both noisy observations in phase space and from image pixels within which the unknown dynamics are embedded.

1 Introduction

Deep learning has revolutionized a number of application areas, such as image classification and reinforcement learning, in part via its ability to obtain representations of data that generalize well and are useful for downstream tasks. Deep networks have accomplished this by simultaneously being highly expressive, yet capable of learning effectively from a finite amount of data. A key determinant in this efficiency is the inductive bias encoded by the architecture of the network, such as in convolutional networks for image data, as well as long short-term memory networks for text and other sequential data. These structural assumptions allow the network to learn efficiently, while still enabling it to capture complex relationships that would be prohibitively difficult to feature-engineer or otherwise write down manually.

We are interested in applying such networks to dynamical systems governed by the laws of physics. Such systems are highly flexible and capable of modeling complex phenomena – however, they also possess inherent structure such as *conservation of energy* and other physical quantities. In machine learning, this important structure is often ignored, due to the black-box nature of off-the-shelf algorithms. To perform well on a given task, deep neural networks must learn to conserve physical conserved quantities as effectively as possible. Owing to the precise form of their equations, such networks generally *cannot* learn to conserve these quantities exactly (Greydanus et al. 2019). It has been recently demonstrated by Greydanus et al. (2019) that this flaw harms the networks' capacity for accurate long-term prediction.

As a workaround, Greydanus et al. (2019) proposed to parameterize the dynamical system's Hamiltonian using a neural network, and to learn it directly from data. The specification of the Hamiltonian fully determines the dynamics, and the equations of motion are then reconstructed from the learned

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

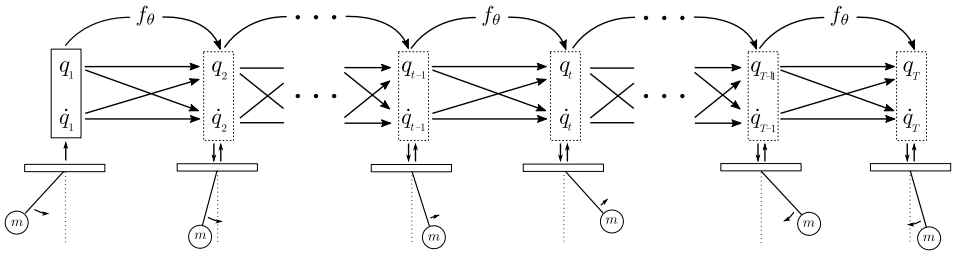

Figure 1: Learning the dynamics of a pendulum. Here, $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the latent states, and f_θ is a residual block. The full variational integrator network is built by stacking free-form residual blocks in the manner prescribed by a variational integrator to obtain a deep network. Here, a Hamiltonian variational integrator is used, which yields skip connections akin to those of a deep residual network.

Hamiltonian via standard techniques from mechanics. One downside to this approach is the black-box nature of the neural network, which makes it difficult to encode properties of the dynamical system, such as its constraints or symmetries.

The continuous-time equations of motion for a dynamical system are given by a set of differential equations that can be derived from its Lagrangian via variational calculus. These equations encode the underlying physical properties, such as conservation laws. In parallel, a deep residual network can be viewed as an Euler discretization of a system of ordinary differential equations, see Haber and Ruthotto (2017), E (2017), and Chen et al. (2018).

In this paper, we aim to bridge the viewpoint of neural ODEs (Haber and Ruthotto 2017; E 2017; Chen et al. 2018; Chang et al. 2018; Ruthotto and Haber 2018), where neural networks are seen as discretized dynamical systems, with the viewpoint of geometric embeddings (Chamberlain et al. 2017; Nickel and Kiela 2017; Ganea et al. 2018; Davidson et al. 2018), which impose structure on an embedding space. When data is concentrated on a manifold, Falorsi et al. (2018) argued that it is crucial to ensure the embedding space has the same topology as this manifold, motivating Lie group variational auto-encoders (Falorsi et al. 2018; de Haan and Falorsi 2018; Falorsi et al. 2019) and related methods (Raissi et al. 2017; Lutter et al. 2019).

We propose to model the dynamical system using a deep neural network, whose architecture is selected to match the discrete equations of motion governing the dynamical system. This allows us to re-interpret the embedding learned by the network as a dynamical system in its own right. We focus on a class of discretization methods called *variational integration* (Marsden et al. 2001). This gives rise to our proposed *variational integrator networks*: a class of flexible neural network architectures that encode physical laws and manifold constraints. These properties promote accurate long-term prediction, interpretability and more efficient learning than is possible with comparable black-box function approximators.

We demonstrate their effectiveness on a number of tasks, including inferring dynamical systems from noisy observations, and from the pixels of images, both in an interpretable and data-efficient manner.

2 Variational Integrators

In this section, we review variational integrators (VIs), a general class of discretization methods for dynamical systems. We study physical dynamical systems over a configuration space $\mathcal{Q}$, which are governed by the principle of least action, specified via the Lagrangian $L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$, and expressible in Hamiltonian form. A brief review of these and related concepts of classical mechanics is given in Appendix A.

VIs approximate the trajectory of a continuous-time dynamical system by discretizing its action integral

$$L^d(\boldsymbol{q}_t, \boldsymbol{q}_{t+1}, h) \approx \int_\tau^{\tau+h} L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\,\mathrm{d}t. \tag{1}$$

This is a discrete-time quadrature-based approximation, denoted by L^d, defined by $\boldsymbol{q}_t = \boldsymbol{q}(\tau)$ and $\boldsymbol{q}_{t+1} = \boldsymbol{q}(\tau + h)$ with step size h. From a Lagrangian perspective, we arrive at the discrete equations of motion

$$\frac{\partial L^d(\boldsymbol{q}_{t-1}, \boldsymbol{q}_t, h)}{\partial \boldsymbol{q}_t} + \frac{\partial L^d(\boldsymbol{q}_t, \boldsymbol{q}_{t+1}, h)}{\partial \boldsymbol{q}_t} = 0 \tag{2}$$

by using a discrete analog of Hamilton's principle (Marsden et al. 2001). From a Hamiltonian perspective, we arrive at

$$\frac{\partial L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h)}{\partial \dot{\boldsymbol{q}}_{t+1}} - h\boldsymbol{p}_t - h\frac{\partial L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h)}{\partial \boldsymbol{q}_t} = 0, \tag{3}$$

where $\boldsymbol{p}_t := \partial L / \partial \dot{\boldsymbol{q}}_t$ denotes generalized momenta (Kharevych et al. 2006). A short derivation is given in Appendix B.

VIs are *symplectic*: they conserve phase-space volume exactly. Symplectic integrators also approximately conserve energy, often only introducing third-order (and above) discretization error with respect to the energy. Such integrators yield discrete-time dynamical systems that closely resemble the continuous-time systems under study, and evolve in a way that is globally consistent with the true solution.

VIs are also *momentum-preserving*. This means that for any symmetry in the discrete system, there is a quantity that is exactly conserved. These properties help to ensure their accuracy. In the dissipative and forced cases, VIs have been both theoretically and empirically shown to produce stable long-term predictions and to capture statistically important quantities, even in chaotic regimes (Lew et al. 2004).

3 Variational Integrator Networks

To define a variational integrator network, we begin with the viewpoint of neural ODEs (Haber and Ruthotto 2017; Chen et al. 2018). In that setting, we obtain a deep residual network by specifying an ODE, whose right-hand-side is a one-layer neural network, together with an Euler discretization scheme. The resulting deep network's depth is determined by the number of discretization time steps selected.

We mirror this viewpoint with the goal of developing network architectures that learn dynamical systems faithfully, by having their learned embeddings be dynamical systems in their own right. We similarly begin with a single-layer neural network, whose weights are the parameters to be learned. Compared to neural ODEs, we introduce two key differences.

1. Rather than using this network to construct a free-form system of ODEs, we instead use it to construct a system of ODEs arising from the Euler-Lagrange equations governing a free-form dynamical system.
2. Instead of an Euler discretization, we use a structure-preserving discretization scheme given by a variational integrator.

This yields a deep neural network whose depth is determined by the number of discretization time steps. The structure-preserving nature of the resulting architectures facilitates accurate long-term prediction, interpretability and data-efficient learning and enables constraints to be added naturally. We call the network's embedding space $\boldsymbol{q}$ to emphasize that it is a dynamical system in its own right, whose laws of motion are flexible and determined by a neural network whose weights are learned from data.

For this strategy to constructively yield a network architecture, we need to be able to derive explicit update equations from the discrete equations of motion, given in Equation (2) and Equation (3). We illustrate a number of ways in which this can be done.

We first consider *Newtonian networks*, i.e. networks that follow Newton's laws of physics. These are constructed by considering a Lagrangian for a conservative systems of the form

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T(\dot{\boldsymbol{q}}) - U(\boldsymbol{q}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \mathbf{M} \dot{\boldsymbol{q}} - U(\boldsymbol{q}), \tag{4}$$

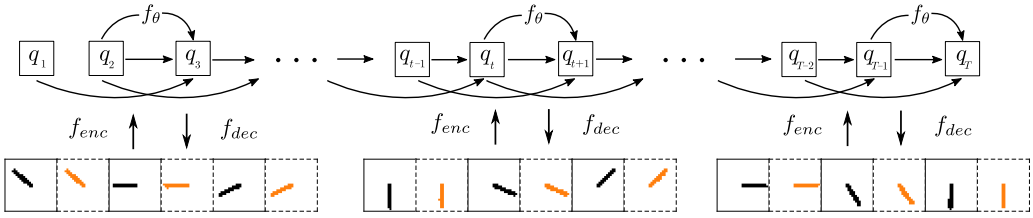

Figure 2: Learning the dynamics of a pendulum from pixel observations. Here, a variational autoencoder maps the pixels into the latent space $\boldsymbol{q}$ using f_{enc}, and maps the latent space back into pixels using f_{dec}. A Lagrangian variational integrator is used, for which $\boldsymbol{q}$ is the latent state. Unlike an ordinary residual network, the skip connections used are intertwined. We display the observations in black, and predicted values given by the decoder in orange. Experimental details for this setup are given in Section 4.

where T is the kinetic energy, U is the potential energy of the system, and $\mathbf{M}$ is a diagonal mass matrix. We omit time dependence for ease of notation. From a Lagrangian perspective, using a first-order approximation $L^d(\boldsymbol{q}_t, \boldsymbol{q}_{t+1}, h) = L\Big(\boldsymbol{q}_t, \frac{(\boldsymbol{q}_{t+1}-\boldsymbol{q}_t)}{h}\Big)$ of the action, we arrive at the architecture

$$\boldsymbol{q}_{t+1} = 2\boldsymbol{q}_t - \boldsymbol{q}_{t-1} - h^2\mathbf{M}^{-1}\frac{\partial f_\theta(\boldsymbol{q}_t)}{\partial \boldsymbol{q}_t}, \tag{5}$$

From a Hamiltonian perspective, where our state space is now expanded to $(\boldsymbol{q}, \dot{\boldsymbol{q}})$, the approximation $L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h) = L(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1})$, yields the architecture

$$\dot{\boldsymbol{q}}_{t+1} = \dot{\boldsymbol{q}}_t - h\mathbf{M}^{-1}\frac{\partial f_\theta(\boldsymbol{q}_t)}{\partial \boldsymbol{q}_t} \tag{6}$$

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + h\dot{\boldsymbol{q}}_{t+1}, \tag{7}$$

which is equivalent to the architecture in Equation (5) if we take $\dot{\boldsymbol{q}}_t = (\boldsymbol{q}_t - \boldsymbol{q}_{t-1})/h$ and combine equations in order to reduce the state from $(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t)$ to $(\boldsymbol{q}_{t-1}, \boldsymbol{q}_t)$.

These architectures enjoy the following properties: **1)** They automatically enforce the underlying physical properties, such as approximate conservation of energy and symmetries, since they are VIs; **2)** The setup retains flexibility to model complex phenomena, as the potential function has not been restricted: the residual block f_θ can be a black-box neural network. We opt to use a single-layer fully connected network; **3)** The mass can either be modeled explicitly for additional interpretability, or absorbed into f_θ; **4)** Momentum can be modeled explicitly if need be. Dealing with position variables only can be particularly useful in situations where velocities are not directly measured, such as pixel observations of dynamical systems embedded in images or video.

As a more complicated example, we consider a *Newtonian rotation network* in 2D. This illustrates how we can improve efficiency by incorporating this structure into the network if we know that a system's evolution takes place entirely on a manifold $\mathcal{M} \subseteq \mathcal{Q}$. For example, the movement of a pendulum is uniquely characterized by its angle with respect to some reference point, and for a given initial state its trajectories in phase space $(q_t, \dot{q}_t)$ are not dense in $\mathbb{R}^2$. Instead, they concentrate on a submanifold determined by the length of the pendulum.

To address this issue, we consider a particular class of variational integrators: *Lie group variational integrators* (LGVIs). LGVIs use the properties of Lie groups to construct integrators that automatically evolve on a specified Lie group. The key idea is to approximate the change in position over integration steps using group elements (Leok 2007). Since the state space is closed under the group action (e.g. matrix multiplication when represented by matrices), the constraints are automatically enforced. For instance, the Lie group $SO(2)$ (with matrix multiplication as the group action) is a natural way to encode the underlying manifold of 2D rotations, like the evolution of the angle of a pendulum. A Newtonian network in a uniform gravitational potential that evolves automatically on $SO(2)$ can be derived as follows. Denoting the angle by ϑ, the corresponding rotation network is given by

$$\sin\Delta\vartheta_t = \sin\Delta\vartheta_{t-1} + h^2 r_\varphi(\vartheta) \tag{8}$$

$$\vartheta_{t+1} = \vartheta_t + \Delta\vartheta_t, \tag{9}$$

where $r_\varphi(\vartheta_t)$ is a neural network with $\sin(\cdot)$ activations at the last layer. Appendix C provides further details.

3.1 Learning VINs from noisy observations

Given initial conditions and the equations of motion of a system, the state evolution is given by a solution $\boldsymbol{q}(t)$ to the equations of motion. VINs represent an approximation to the solution between the initial condition $(\boldsymbol{q}_1, \boldsymbol{q}_2)$ from the Lagrangian perspective and an end point $(\boldsymbol{q}_{T-1}, \boldsymbol{q}_T)$, or alternatively $(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t)$ pairs in the Hamiltonian case. We represent an embedding in the network by

$$\boldsymbol{q}_t = q_\theta(\boldsymbol{q}_1, \boldsymbol{q}_2, h, t), \tag{10}$$

which denotes layer t in the VIN, see Figure 1 for an illustration of a VIN. Given a dataset of noisy observations of the state of a system, denoted by $\boldsymbol{y}_{1:T}$, we specify a Gaussian likelihood

$$p(\boldsymbol{y}_{1:T}|\boldsymbol{q}_{1:T}, \sigma^2) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{q}_t, \sigma^2\mathbf{I}), \tag{11}$$

which lets us train the model by maximizing the log-likelihood of the parameters θ of the VIN, the initial condition $(\boldsymbol{q}_1, \boldsymbol{q}_2)$, and the error variance σ^2. The parameters to be learned are $\Theta = (\theta, \boldsymbol{q}_1, \boldsymbol{q}_2, \sigma^2)$. Training proceeds by maximizing

$$\sum_{t=1}^{T} \log p(\boldsymbol{y}_t|\Theta) \tag{12}$$

with respect to Θ using any variation of minibatch stochastic gradient ascent.

3.2 Learning VINs from pixels

It is possible that the dynamical system of interest is not observed directly, but indirectly through a set of intermediate data not of primary interest. For example, we can observe a swinging pendulum by seeing images of its location at a given set of time points. Here, we propose to address this problem using variational autoencoders (VAEs) (Kingma and Welling 2014; Rezende et al. 2014).VAEs let us scale to high-dimensional observations of physical systems that are related to the configuration space by an unknown nonlinear mapping. We focus here on images.

We define a distribution over paths in configuration space by placing a standard Gaussian $p_\theta(\boldsymbol{q}_i) = \mathcal{N}(\boldsymbol{q}_i \mid \mathbf{0}, \mathbf{I})$ for $i = 1, 2$ over the initial condition. The joint distribution over trajectories is

$$p_\theta(\boldsymbol{q}_{1:T}) = p_\theta(\boldsymbol{q}_1)p_\theta(\boldsymbol{q}_2)p_\theta(\boldsymbol{q}_{3:T}|\boldsymbol{q}_1, \boldsymbol{q}_2), \tag{13}$$

which we can sample from by sampling $\boldsymbol{q}_i^s \sim p(\boldsymbol{q}_i)$ for $i = 1, 2$, and feeding the samples through the network $\boldsymbol{q}_t^s = q_\theta(\boldsymbol{q}_1^s, \boldsymbol{q}_2^s, h, t)$.

We specify the joint distribution over observations and paths in latent space as

$$p_\theta(\boldsymbol{y}_{1:T}, \boldsymbol{q}_{1:T}) = p_\theta(\boldsymbol{q}_{1:T}) \prod_{t=1}^{T} p_\theta(\boldsymbol{y}_t|\boldsymbol{q}_t), \tag{14}$$

where the distribution over latent paths is given by Equation (13). The likelihood $p_\theta(\boldsymbol{y}_t|\boldsymbol{q}_t)$ is parameterized by a neural network $f_{dec}(\boldsymbol{q}_t)$, called the decoder.

We aim to approximate the posterior distribution $p(\boldsymbol{q}_{1:T}|\boldsymbol{y}_{1:T})$, which is intractable due to the nonlinear relationships introduced by the decoder f_{dec} and the dynamics of q_θ. VAEs do this by learning a Kullback-Leibler approximation $q_\phi(\boldsymbol{q}_{1:T}|\boldsymbol{y}_{1:T})$ to the posterior, parameterized by the encoder $f_{enc}(\boldsymbol{y}_{1:T})$. Figure 2 illustrates the VIN-VAE setup.

Training minimizes the Kullback-Leibler divergence between the approximate posterior and the true one, which is equivalent to maximizing the evidence lower bound (ELBO)

$$\log p(\boldsymbol{y}_{1:T}) \geq \mathbb{E}_{q_\phi(\boldsymbol{q}_{1:T}|\cdot)} \left[\log \frac{p_\theta(\boldsymbol{y}_{1:T}, \boldsymbol{q}_{1:T})}{q_\phi(\boldsymbol{q}_{1:T}|\boldsymbol{y}_{1:T})} \right] \tag{15}$$

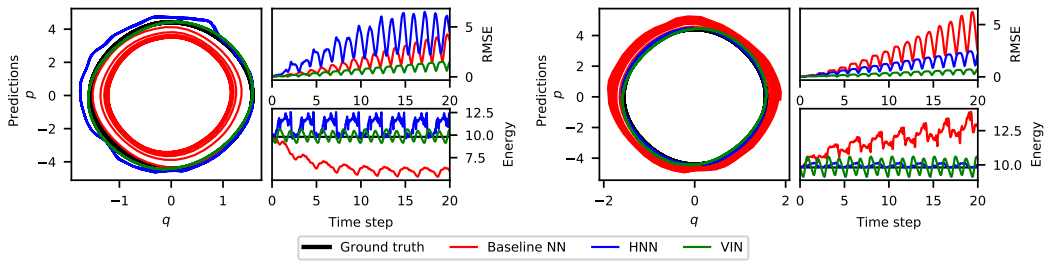

Figure 3: Learning physics from noisy observations for the ideal pendulum. Given a set of initial conditions, we forecast a path in configuration space and compare against the ground truth. We show model predictions, total root-mean-squared error between coordinates and the total energy of the dynamical system in the embedding.

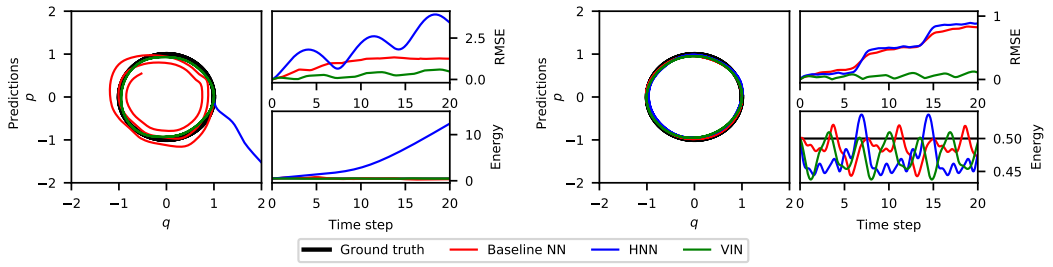

Figure 4: Learning physics from noisy observations for the ideal mass-spring. Given a set of initial conditions, we forecast a path in configuration space and compare against the ground truth. We show model predictions, total root-mean-squared error between coordinates and the total energy of the dynamical system in the embedding.

on the log-marginal likelihood with respect to the model parameters θ and the variational parameters ϕ. We choose the variational family

$$q_\phi(\boldsymbol{q}_{1:T}|\boldsymbol{y}_{1:T}) = q_\phi(\boldsymbol{q}_1)q_\phi(\boldsymbol{q}_2)\prod_{t=3}^{T} p_\theta(\boldsymbol{q}_t|\boldsymbol{q}_1,\boldsymbol{q}_2), \tag{16}$$

where we emphasize that the conditional $p_\theta(\boldsymbol{q}_t|\boldsymbol{q}_1,\boldsymbol{q}_2)$ is the same as the model, and we specify

$$q(\boldsymbol{q}_t) = \mathcal{N}(\boldsymbol{q}_t \mid \boldsymbol{m}_t, \boldsymbol{s}_t^2), \qquad t = 1, 2. \tag{17}$$

The mean and variance of the initial condition are estimated from the full trajectory $\boldsymbol{y}_{1:T}$ by the encoder f_{enc}. Using Equations (14), (16) in (15), the ELBO is

$$\mathcal{L} = \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{q}_t|\cdot)}\big[\log p(\boldsymbol{y}_t|\boldsymbol{q}_t)\big] - \sum_{t=1}^{2} \mathbb{KL}\big[q(\boldsymbol{q}_t) \,||\, p(\boldsymbol{q}_t)\big],$$

where $\mathbb{KL}\big[q \,||\, p\big]$ denotes the Kullback-Leibler divergence between q and p. This is maximized using any variation of minibatch stochastic gradient ascent.

4 Experiments

To study the performance of VINs, we implemented them for an ideal pendulum and (b) an ideal mass-spring system. In the former case, the dynamics are given by a set of differential equations that do not admit an analytic solution. We generate trajectories for both systems, each consisting of 30 transitions (data points), using numerical integration. We study the ability of VINs to infer a useful representation of the system when given a limited quantity of data, in cases where the dynamical system is observed both directly and indirectly. Network architectures and hyperparameters are given in Appendix D.

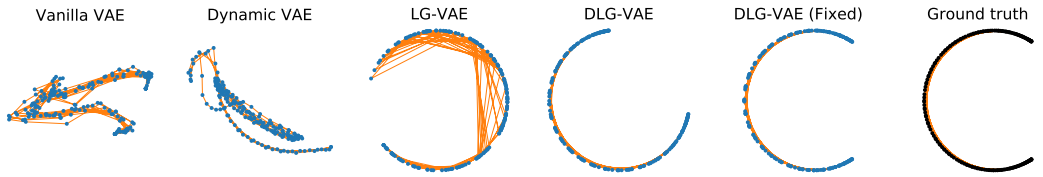

Figure 5: Example embedded representations of an ideal pendulum system. The blue dots represent points in embedded space corresponding to image observations, the orange lines connect points sequentially in time. The figure shows how the embeddings learned by the baseline models either fail to capture the global structure (a)–(b) and/or are discontinuous with respect to the time dimension (c). The DLG-VAE (d)–(e), learns an embedding that is consistent with the ground truth (f).

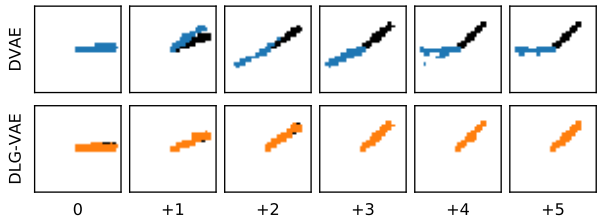

Figure 6: Reconstructions of up to 5-step ahead predictions in latent space. Black: ground truth, blue: DVAE, orange: DLG-VAE. Top: DVAE predicts physically meaningless trajectories. Bottom: DLG-VAE predicts physically meaningful trajectories in latent space and captures the evolution accurately.

4.1 Learning from noisy observations

Here we consider VINs in a noisy setting. Specifically, the model is given noisy position and velocity measurements from which it needs to learn the dynamics. We compare our proposed VINs with Hamiltonian neural networks (HNNs) (Greydanus et al. 2019) and standard feed-forward neural networks (NNs) without additional structure that would explicitly incorporate physical or mechanical constraints. We use the VIN given by Equation (6) in Section 3. HNNs are trained on observations of the form $(\boldsymbol{q}_t, \boldsymbol{p}_t, \dot{\boldsymbol{q}}_t, \dot{\boldsymbol{p}}_t)$. We replicate the setup from Greydanus et al. (2019) with one key difference: we introduce noise in all observations, rather than only introducing it in $(\boldsymbol{q}_t, \boldsymbol{p}_t)$ and observing $(\dot{\boldsymbol{q}}_t, \dot{\boldsymbol{p}}_t)$ noise free. This makes the setting more realistic, but system identification more challenging. To account for the noise, we add a noise variable to all models and maximize the log-likelihood, rather than the mean-squared error.

Following Greydanus et al. (2019), we examine two scenarios: (a) a moderate-data regime, where models are trained using 25 training trajectories with a total of 750 data points, (b) a low-data regime using 5 training trajectories with a total of 150 data points. Figures 4 and 3 show that prediction performance differs between the models. In the low-data regime, despite learning to approximately conserve the system's energy, the HNN does not capture the correct dynamics, and performs poorly on prediction in terms of RMSE on both systems. On the mass-spring system (Figure 4), with sufficient data, the HNN prediction error is low over a small horizon, but exhibits two large jumps as the trajectory evolves. We suggest that in both cases the HNN fits the noise in the training data (overfits) and fails to identify the underlying system. The NN baseline performs better than the HNN in the low-data regime, whereas the HNN demonstrates better predictive performance in the moderate-data regime on the pendulum system (Figure 3). The VIN exhibits good predictive performance, outperforming the baselines on both systems, in both the low-data and moderate-data regimes.

From Figures 4 and 3, we observe that the energy behavior of HNNs, VINs, and NNs differs. Given sufficient data, both the HNN and VIN learn a model that conserves a quantity that approximates the

energy of the system. However, the HNN overfits in the low-data regime on both systems. The NN baseline incorrectly dissipates/adds energy in both scenarios for the pendulum system, particularly as time passes, but learns to approximately conserve energy for the mass-spring system given 25 training trajectories. This contributes to the worse predictive performance of the NN baseline.

We conclude that VINs can effectively identify the system from noisy observations, even in small-data scenarios where HNNs and NNs struggle. We attribute this to their architecture: their embedded space is a dynamical system in its own right, which enforces the physical constraints automatically when forecasting, enabling their long-term predictions to better match the true dynamical systems. In contrast, the HNN relies on generalization to conserve energy, as demonstrated by the difference in performance in the low-data and moderate-data regimes.

4.2 Learning from pixel observations

Next, we study VINs in a VAE setting, which adds an auxiliary image processing task in addition to prediction. Here, we observe 28×28 pixel images depicting the pendulum system. In this setting, we add the manifold constraints imposed by Equation (8). We call this setup the *dynamic Lie group VAE* (DLG-VAE), see Appendix D for details. We assess whether our approach learns representations with good interpolative and predictive properties, and whether it is data-efficient compared to less-structured baselines.

To assess the DLG-VAE, we evaluate the structure of its latent space and compare it with representations learned by a standard VAE (Kingma and Welling 2014; Rezende et al. 2014), a VAE with free-form dynamics governed by a feed-forward network (DVAE), and a Lie group VAE (LG-VAE) (Falorsi et al. 2018) with no dynamic structure. Figure 5 visualizes the latent spaces after training on 20 images and mapping an additional 80 test images into latent space using the encoder f_{enc}. The vanilla VAE captures local structure: observations close together in image space are mapped to points close together in latent space. However, it fails to capture the global structure of the state space and has discontinuities with respect to the sequential nature of the dataset. Figure 5(b) shows that adding an unrestricted neural network to capture the dynamics does not solve the problem. The Lie group VAE captures the correct global structure by restricting the manifold, but still exhibits discontinuities with respect to the time dimension, since it does not model the dynamics. The DLG-VAE's latent space does not have such discontinuities: it learns both the global structure and respects the sequential nature of the data due to the structure encoded by the VIN.

The Euler-Lagrange equations governing motion are invariant to linear transformations, and thus the true dynamics are inherently unidentifiable from image data alone. In Figure 5(d), we observe that the model converged at a lower value for gravitational acceleration than the ground-truth value, and has not covered the same number of periods in latent space as the pendulum has in observation space. Note here that the learned embedding is still useful for downstream prediction, since the decoder accounts for this difference. To explore this further, we examined a variation with the magnitude and direction of the gravitational acceleration set to their ground-truth values. Details on how the gravitational acceleration enters Equation (8) can be found in Appendix C. Here, we see that the DLG-VAE learns to map to-and-from pixel space to a system nearly identical to the ground truth.

5 Conclusion

In this work, we introduced *variational integrator networks*, a class of deep network architectures for creating neural embeddings, which encode and represent dynamical systems. They ensure faithful representation of dynamical system by using an embedding that forms a dynamical system in its own right. This facilitates data-efficient learning, enhances interpretability, and allows for accurate long-term prediction when compared to other classes of networks.

Recent trends in deep learning have sought to improve the performance of deep networks on physical systems by designing networks whose behavior is more understandable and better matched to the underlying physics. Variational integrator networks take a step toward progressing this line of work.

Acknowledgments

This work was supported by Microsoft Research through its PhD scholarship program.

References

[1] B. P. Chamberlain, J. Clough, and M. P. Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv:1705.10359*, 2017. Cited on page 2.

[2] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham. Reversible architectures for arbitrarily deep residual neural networks. In *AAAI Conference on Artificial Intelligence*, 2018. Cited on page 2.

[3] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018. Cited on pages 2, 3.

[4] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical variational auto-encoders. *arXiv:1804.00891*, 2018. Cited on page 2.

[5] P. de Haan and L. Falorsi. Topological constraints on homeomorphic auto-encoding. *arXiv:1812.10783*, 2018. Cited on page 2.

[6] W. E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017. Cited on page 2.

[7] L. Falorsi, P. de Haan, T. R. Davidson, N. De Cao, M. Weiler, P. Forré, and T. S. Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv:1807.04689*, 2018. Cited on pages 2, 8.

[8] L. Falorsi, P. de Haan, T. R. Davidson, and P. Forré. Reparameterizing distributions on Lie groups. In *International Conference on Artificial Intelligence and Statistics*, 2019. Cited on page 2.

[9] O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. *Advances in Neural Information Processing Systems*, 2018. Cited on page 2.

[10] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages 1, 7, 11.

[11] E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017. Cited on pages 2, 3.

[12] L. Kharevych, W. Yang, Y. Tong, E. Kanso, J. E. Marsden, P. Schröder, and M. Desbrun. Geometric, variational integrators for computer animation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 43–51, 2006. Cited on pages 3, 10.

[13] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014. Cited on pages 5, 8.

[14] M. Leok. An overview of Lie group variational integrators and their applications to optimal control. In *International Conference on Scientific Computation and Differential Equations*, page 1, 2007. Cited on page 4.

[15] A. J. Lew, J. E. Marsden, M. Ortiz, and M. West. An Overview of variational integrators. In *Finite Element Methods: 1970s and Beyond*. CIMNE, 2004. Cited on pages 3, 10.

[16] M. Lutter, C. Ritter, and J. Peters. Deep Lagrangian networks: using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2019. Cited on page 2.

[17] J. E. Marsden, S. Pekarsky, S. Shkoller, and M. West. Variational methods, multisymplectic geometry and continuum mechanics. *Journal of Geometry and Physics*, 38(3-4):253–284, 2001. Cited on pages 2, 3.

[18] R. A. Meyers. *Mathematics of complexity and dynamical systems*. Springer, 2009. Cited on page 11.

[19] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6338–6347, 2017. Cited on page 2.

[20] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (Part I): data-driven solutions of nonlinear partial differential equations. *arXiv:1711.10561*, 2017. Cited on page 2.

[21] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014. Cited on pages 5, 8.

[22] L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *arXiv:1804.04272*, 2018. Cited on page 2.

A Appendix: Short review of Lagrangian and Hamiltonian mechanics

Hamiltonian and Lagrangian mechanics are two intricately related formulations of classical mechanics. In classical mechanics, we assume that we are given a continuous-time dynamical system defined on a space $\mathcal{Q} \subseteq \mathbb{R}^d$, which we call the *configuration space*. A state of the system is taken to be a set of parameters $\boldsymbol{q} \in \mathcal{Q}$ that uniquely identify the configuration of the system. Continuous-time evolution of the dynamics in $\mathcal{Q}$ yields a path in configuration space. Lagrangian and Hamiltonian mechanics formulate the laws of physics in terms of properties of these paths.

Specifically, *Hamilton's principle*, also called the *Principle of Least Action*, states that there exists a real-valued function L such that all paths in configuration space which occur in nature minimize the path integral

$$A(L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))) = \int_0^T L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\,\mathrm{d}t \tag{18}$$

where $\dot{\boldsymbol{q}}$ is the velocity, which is the time-derivative of position. For a given L, it can be shown using the calculus of variations that minimization of A is equivalent to solving a system of partial differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big(\frac{\partial L}{\partial \dot{\boldsymbol{q}}^q}\Big) - \frac{\partial L}{\partial \boldsymbol{q}^q} = 0, \tag{19}$$

called the *Euler-Lagrange Equations*, or the *equations of motion*. Given a set of initial conditions $(\boldsymbol{q}(0), \dot{\boldsymbol{q}}(0))$, the solutions to the equations of motion describe the trajectory of the system.

This gives the starting point of Lagrangian mechanics – physical phenomena that satisfy it are called *classical*, and span virtually all areas of physics. The behavior of particular phenomena varies according to choice of the Lagrangian L, which fully characterizes how the system evolves over time.

For example, for $\boldsymbol{q} \in \mathbb{R}^d$, take $L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T(\boldsymbol{q}, \dot{\boldsymbol{q}}) - U(\boldsymbol{q})$ where T is the kinetic energy, and U is the potential energy of the system. This describes a conservative Newtonian system.

B Appendix: Discrete equations of motion and variational integrators

From the Lagrangian perspective, the discrete action sum is

$$\sum_{t=1}^{T} L^d(\boldsymbol{q}_t, \boldsymbol{q}_{t+1}, h). \tag{20}$$

By using the discrete version of Hamilton's principle in the special case of three points $\boldsymbol{q}_{t-1}, \boldsymbol{q}_t, \boldsymbol{q}_{t+1}$, and taking variations with respect to $\boldsymbol{q}_t$, one arrives at Equation (2) (Lew et al. 2004).

In the Hamiltonian case, the discrete Hamilton-Pontryagin sum is

$$\sum_{t=1}^{T} \Big[\boldsymbol{p}_{t+1}\big(\frac{\boldsymbol{q}_{t+1} - \boldsymbol{q}_t}{h} - \dot{\boldsymbol{q}}_{t+1}\big)h + L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h)\Big]. \tag{21}$$

Taking discrete variations with respect to each state variable $(\boldsymbol{p}, \boldsymbol{q}, \dot{\boldsymbol{q}})$ with fixed endpoints one arrives at for

$$\boldsymbol{p}:\ \boldsymbol{q}_{t+1} - \boldsymbol{q}_t = h\dot{\boldsymbol{q}}_{t+1} \quad \dot{\boldsymbol{q}}:\ \boldsymbol{p}_{t+1} - \boldsymbol{p}_t = \frac{\partial L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h)}{\partial \boldsymbol{q}_t} \quad \boldsymbol{q}:\ h\boldsymbol{p}_{t+1} = \frac{\partial L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h)}{\partial \dot{\boldsymbol{q}}_{t+1}} \tag{22}$$

and using Equation (22), to replace $\boldsymbol{p}_{t+1}$ with $\boldsymbol{p}_t$ and $\frac{\partial L^d(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_{t+1}, h)}{\partial \boldsymbol{q}_t}$, one arrives at Equation (3) (Kharevych et al. 2006).

C Appendix: Lie group variational integrator for $SO(2)$

We start by formulating a Lagrangian with the Lie group $SO(2)$ using matrix representations. First, define the map from scalars $\omega \in \mathbb{R}$ to 2×2 skew-symmetric matrices

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}. \tag{23}$$

The set of 2×2 skew-symmetric matrices forms the Lie algebra $\mathfrak{so}(2)$. The matrix exponential map, takes elements of the Lie algebra to elements of the group $SO(2)$

$$\mathbf{R}(\omega) = \exp \mathbf{S}(\omega) = \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix}. \tag{24}$$

Kinematics for group elements $R \in SO(2)$ can be written in terms of Lie algebra elements as

$$\dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\omega), \tag{25}$$

where ω is analogous to angular velocity. A conservative Newtonian Lagrangian in a uniform gravitational potential can be written in terms of the Lie group $SO(2)$ as

$$L(\mathbf{R}, \mathbf{S}(\omega)) = \frac{1}{2} ml^2\omega^2 + mgl\boldsymbol{e}_2^T \mathbf{R}\boldsymbol{e}_1 \tag{26}$$

where $\mathbf{R} = \mathbf{R}(\theta)$ is a rotation matrix parameterized by θ, g is the gravitational acceleration and $\boldsymbol{e}_1, \boldsymbol{e}_2$ are orthogonal unit vectors in the inertial frame of reference, $\boldsymbol{e}_1 = [1, 0], \boldsymbol{e}_2 = [0, 1]$.

To develop a Lie group variational integrator, define $\mathbf{F}_t \in SO(2)$ such that

$$\mathbf{R}_{t+1} = \mathbf{R}_t \mathbf{F}_t. \tag{27}$$

Since $\mathbf{F}_t \in SO(2)$, the update enforces $\mathbf{R}_{t+1} \in SO(2)$ since Lie groups are closed under the group action. Here, group action is given by matrix multiplication. Then define the discretization of the action integral as

$$L^d(\mathbf{R}_k, \mathbf{F}_k) = \frac{1}{2h} ml^2 \langle \mathbf{F}_k - \mathbf{I}, \mathbf{F}_k - \mathbf{I} \rangle + \frac{hmgl}{2} \big(\boldsymbol{e}_2^T \mathbf{R}_t \boldsymbol{e}_1 + \boldsymbol{e}_2^T \mathbf{R}_{t+1} \boldsymbol{e}_1 \big), \tag{28}$$

which approximates the angular velocity as

$$\mathbf{S}(\dot{\theta}) = \frac{\mathbf{F}_k - \mathbf{I}}{h}. \tag{29}$$

Using the discrete form of Hamilton's principle, one obtains (Meyers 2009) the equation

$$(\mathbf{F}_t - \mathbf{F}_t^T) - (\mathbf{F}_{t+1} - \mathbf{F}_{t+1}^T) - \frac{2h^2 g}{l} \mathbf{S}(\boldsymbol{e}_2^T \mathbf{R}_{t+1} \boldsymbol{e}_1) = 0 \tag{30}$$

which, when taken with Equation (27), defines the Lie group variational integrator. One arrives at Equation (8), written in terms of the elements of the matrices, by subsuming the force terms into the neural network.

D Appendix: Hyperparameters for experiments

D.1 Noisy system observations

The setup resembles the one of Greydanus et al. (2019) closely. The neural network architecture for the baseline NN, the network that parameterizes the Hamiltonian in HNNs and the one that parameterizes the VIN was the same throughout. This was a single hidden layer feed-forward network with 200 hidden units and $\tanh(\cdot)$ activations on the hidden layer. The noise added to the observations was sampled from a standard Gaussian with standard deviation $\sigma = 0.1$. For the mass-spring system, we set the spring constant and mass to $k = m = 1$, as was done by Greydanus et al. (2019). For the pendulum, unlike the original work, we use $m = l = 1$, and $g = 9.81$. Training trajectories were sampled uniformly from energies ranging from $[0.2, 1]$ for the mass-spring system and $[1.3, 2.3]$ for the pendulum. We trained the models using ADAM with a learning rate of 10^{-3}. We did a hyperparameter search over $[2000, 5000, 10000]$ training steps and chose the best performing models for comparison.

For predictions with the baseline NN and HNN, we use the procedure of Greydanus et al. (2019), which uses fourth order Runga-Kutta with an error tolerance of 10^{-9}, implemented in SCIPY.INTEGRATE.SOLVE_IVP. For the VIN we simply predict forwards in time using the trained network.

D.2 Pixel observations

In all VAE experiments we used the same encoder and decoder structure. This consisted of:

- Encoder: two convolutional layers with $\{32, 64\}$ filters, respectively, and the third one is a fully-connected layer that maps to the parameters of the variational distributions.
- Decoder: fully-connected layer and two de-convolutional layers with filters $\{64, 32\}$.
- Encoder and decoder use kernel sizes of 3 and strides of 2×2.

The Dynamic VAE models the dynamic prior with a neural network, composed of one hidden layer with 200 units. Additionally, we use ℓ^2 regularization of the weights of the dynamic network. All activations (excluding output layers) are ReLUs. The rotational VIN uses the a feed-forward network with one hidden layer composed of 200 units and uses a $\tanh(\cdot)$ activation function for the hidden layer and a $sin(\cdot)$ activation on the last layer, as explained in the paper (see Equation (8) and Appendix C). We train using ADAM with a learning rate of 3.0×10^{-4} until the ELBO converges on the training set, up to a maximum of 10000 epochs through the datasets.