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

$$S(\boldsymbol{q}) = \int_0^T L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\,\mathrm{d}t \tag{19}$$

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big(\frac{\partial L}{\partial \dot{\boldsymbol{q}}^q}\Big) - \frac{\partial L}{\partial \boldsymbol{q}^q} = \mathbf{0}, \tag{20}$$

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: 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

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}. \tag{21}$$

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 $SO(2)$

$$\mathbf{R}(\omega) = \exp \mathbf{S}(\omega) = \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix}. \tag{22}$$

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

$$\dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\omega), \tag{23}$$

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

$$L(\mathbf{R}, \mathbf{S}(\omega)) = \frac{1}{2} m l^2 \omega^2 + mg\boldsymbol{e}_2^T \mathbf{R} \boldsymbol{e}_1 \tag{24}$$

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

$$\mathbf{R}_{t+1} = \mathbf{R}_t \mathbf{F}_t. \tag{25}$$

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

$$L^d(\mathbf{R}_k, \mathbf{F}_k) = \frac{1}{2h} ml^2 \langle \mathbf{F}_k - \mathbf{I}, \mathbf{F}_k - \mathbf{I} \rangle + \frac{hmgl}{2} \big(\boldsymbol{e}_2^T \mathbf{R}_t \boldsymbol{e}_1 + \boldsymbol{e}_2^T \mathbf{R}_{t+1} \boldsymbol{e}_1 \big), \tag{26}$$

which approximates the angular velocity as

$$\mathbf{S}(\dot{\theta}) = \frac{\mathbf{F}_k - \mathbf{I}}{h}. \tag{27}$$

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

$$(\mathbf{F}_t - \mathbf{F}_t^T) - (\mathbf{F}_{t+1} - \mathbf{F}_{t+1}^T) - \frac{2h^2 g}{l} \mathbf{S}(\boldsymbol{e}_2^T \mathbf{R}_{t+1} \boldsymbol{e}_1) = \mathbf{0}, \tag{28}$$

which, when taken with (25), defines the Lie group variational integrator. One arrives at (11), written in terms of the elements of the matrices, by subsuming the force terms into the neural network.

C Appendix: Hyperparameters for Experiments

C.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.

C.2 Pixel Observations

In all VAE experiments we used the same encoder and decoder structure. Both the encoder and decoder consisted of two fully connected hidden layers with a 1000 hidden units and ReLU activation functions.

- Encoder: two fully-connected hidden layers with 1000 units and ReLU activation functions, followed by an LSTM with a 50 dimensional hidden state that processed the embedded sequence in reverse to give the variational parameters for the initial condition.
- Decoder: two fully-connected hidden layers with 1000 units and ReLU activation functions.

The dynamics networks (i.e. ResRNN, VIN-VV, VIN-$SO(2)$, VIN-SV) all had a depth of 10 and used 10 observations as input to the encoder. The step size for the networks was chosen to be 1.0 in latent space. The underlying fully connected network had 1000 hidden units and tanh activation functions.

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 6000 epochs through the datasets and use the parameters with the highest ELBO for evaluation.