Hamiltonian & Lagrangian Neural Networks

Learning the Dynamics of Physical Systems

Artem Maevskiy
Research Fellow – Institute for Functional Intelligent Materials @ National University of Singapore

MLHEP-2023
Erice, Apr 11 – 18, 2023

NUS National University of Singapore | Institute for Functional Intelligent Materials

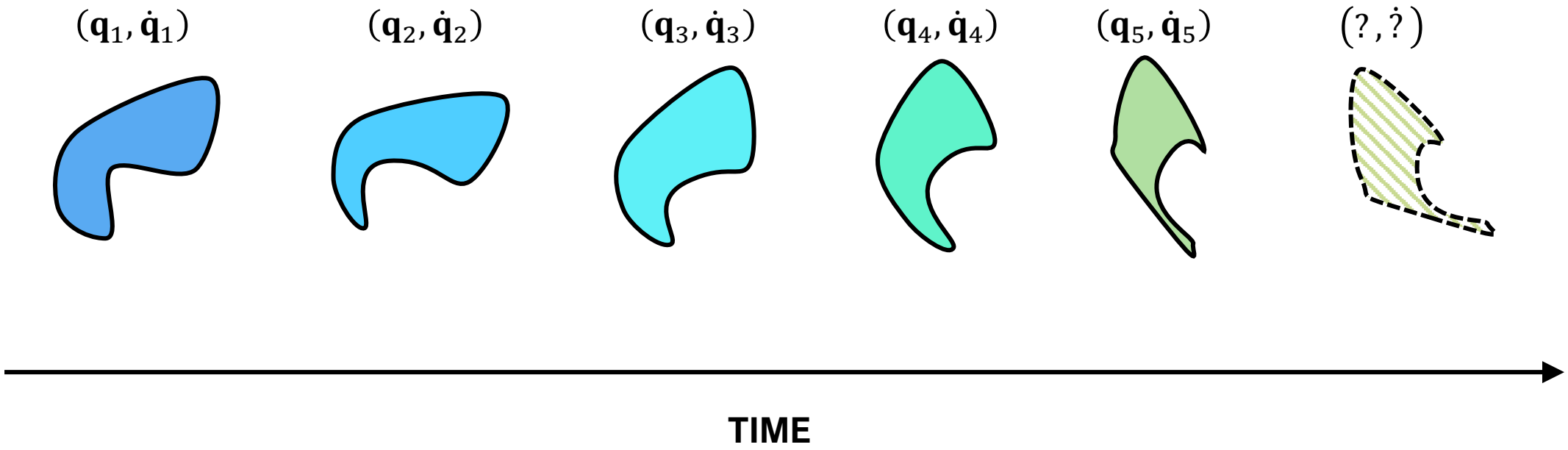
INFN National Institute for Nuclear Physics

Practicum

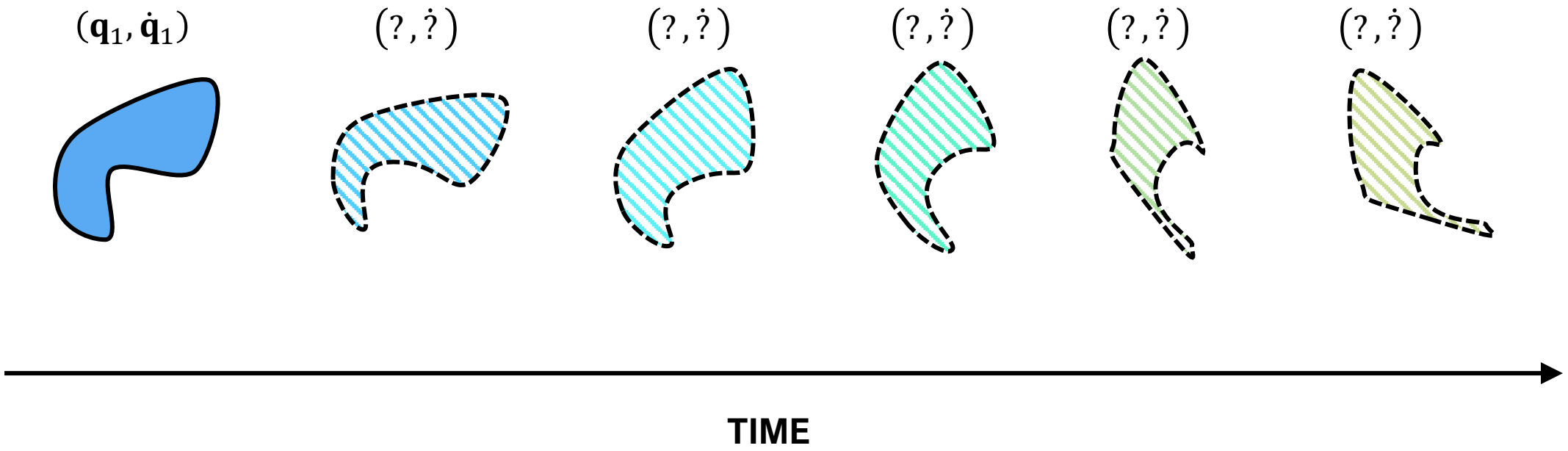
Observation → Knowledge

“Old man in glasses observing a boiling pot, silly 2D cartoon, white background” – as seen by Midjourney

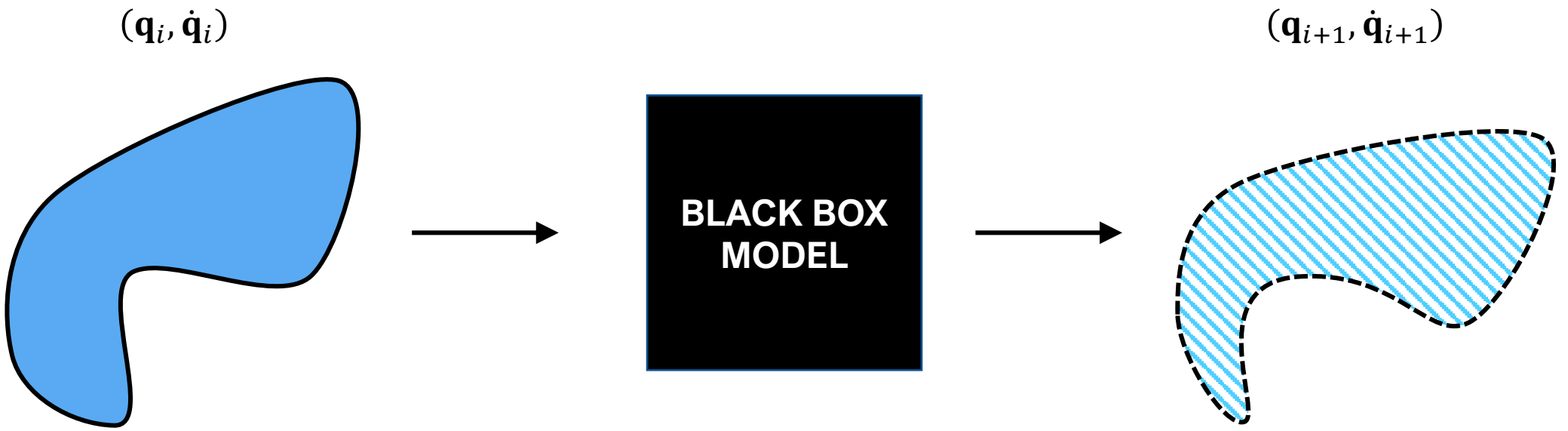
Observing a physical system



Observing a physical system

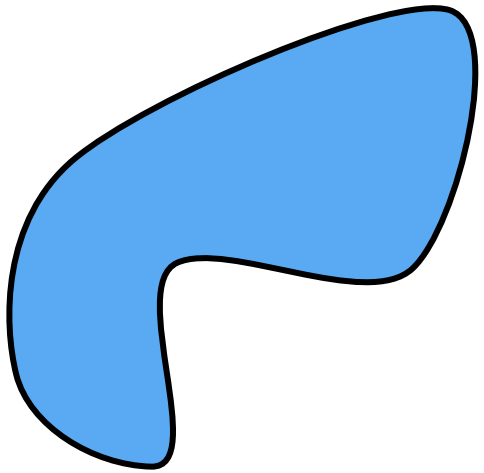


General approach



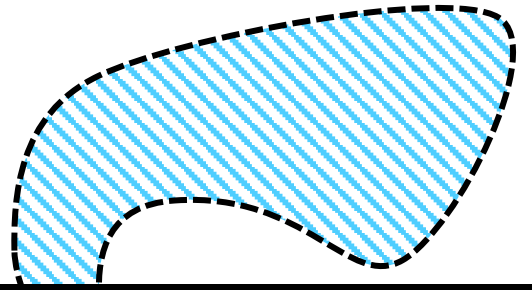
General approach

$(\mathbf{q}_i, \dot{\mathbf{q}}_i)$



BLACK BOX MODEL

$(\mathbf{q}_{i+1}, \dot{\mathbf{q}}_{i+1})$



How can we utilize our prior knowledge?

- Symmetries
- Invariants
- Conservation laws

Inductive bias

The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs of given inputs that it has not encountered. In machine learning, one aims to construct algorithms that are able to learn to predict a certain target output. **Wikipedia**

Lagrangian and Hamiltonian formalisms

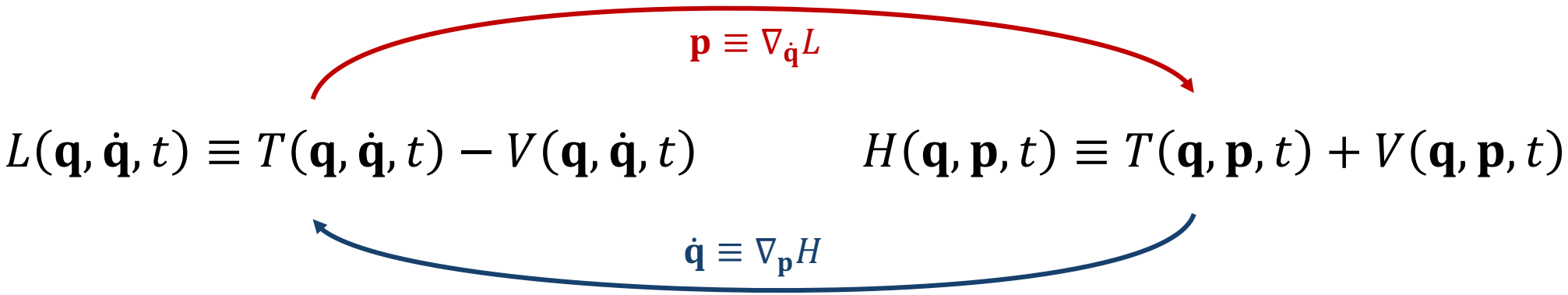
$$L(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, \dot{\mathbf{q}}, t)$$

$$H(\mathbf{q}, \mathbf{p}, t) \equiv T(\mathbf{q}, \mathbf{p}, t) + V(\mathbf{q}, \mathbf{p}, t)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

$$\frac{d}{dt}\begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\nabla_{\mathbf{p}} H\\ -\nabla_{\mathbf{q}} H\end{pmatrix}$$

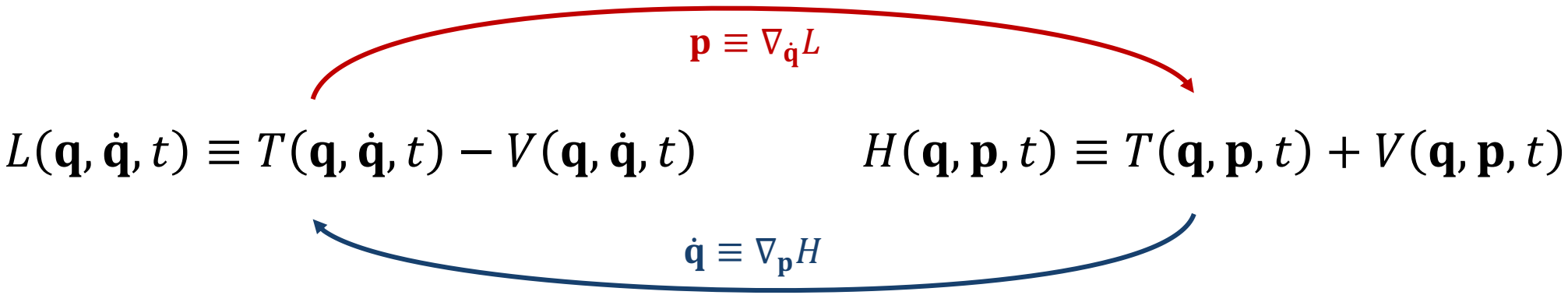
Lagrangian and Hamiltonian formalisms



$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

$$\frac{d}{dt}\begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\nabla_{\mathbf{p}} H\\ -\nabla_{\mathbf{q}} H\end{pmatrix}$$

Lagrangian and Hamiltonian formalisms



$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

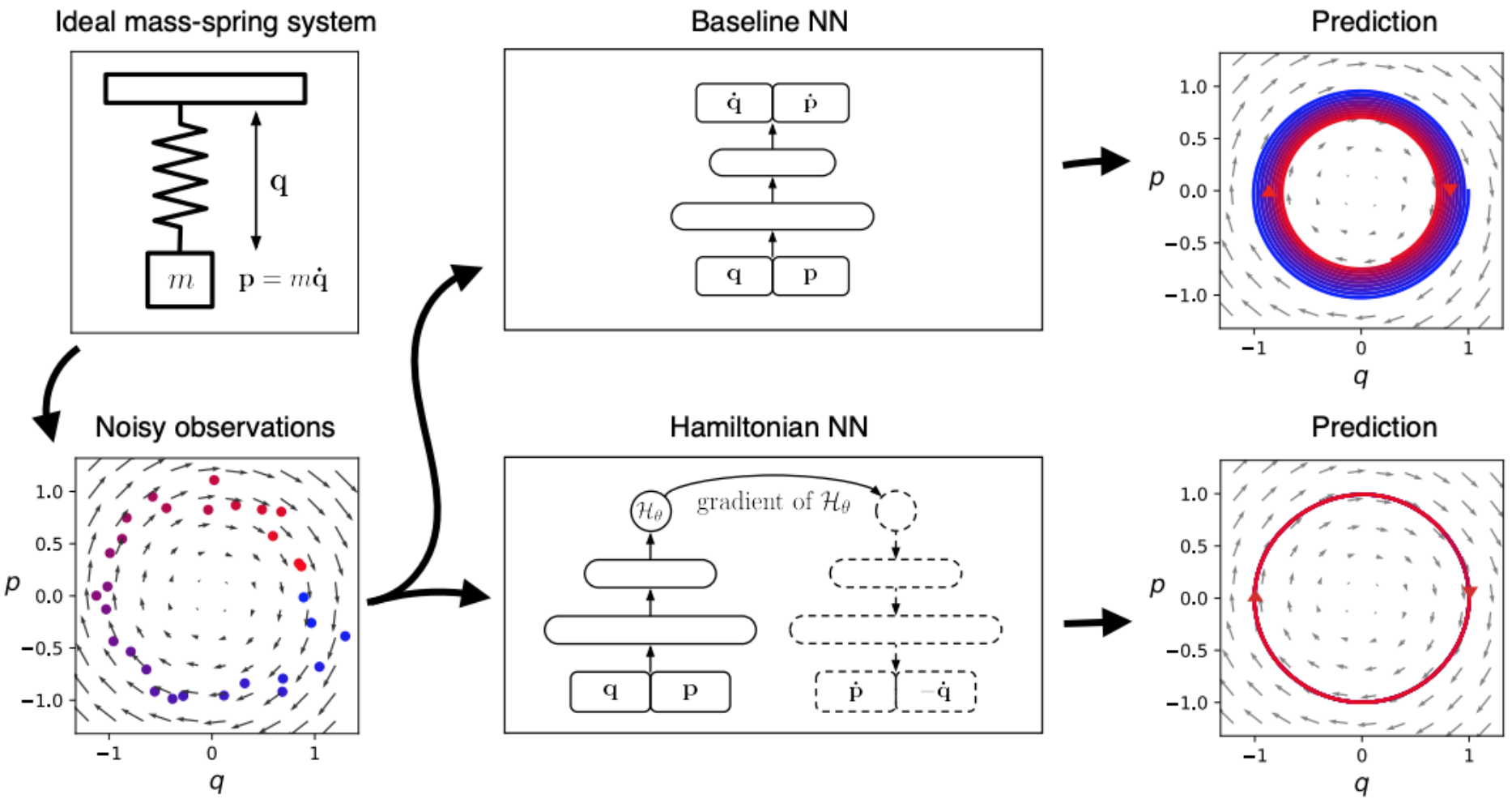
$$\frac{d}{dt}\begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\nabla_{\mathbf{p}} H\\ -\nabla_{\mathbf{q}} H\end{pmatrix}$$

- Can we use NNs to directly model L or H?

Outline

- Hamiltonian Neural Networks
  - Learning dynamics in the Hamiltonian framework
- Lagrangian Neural Networks
  - Learning dynamics in the Lagrangian framework
- Hamiltonian Generative Networks
  - Combining HNN with unsupervised representation learning
- Neural Hamiltonian Flow
  - An unexpected applications of HNN as a Normalizing Flow

Hamiltonian Neural Networks

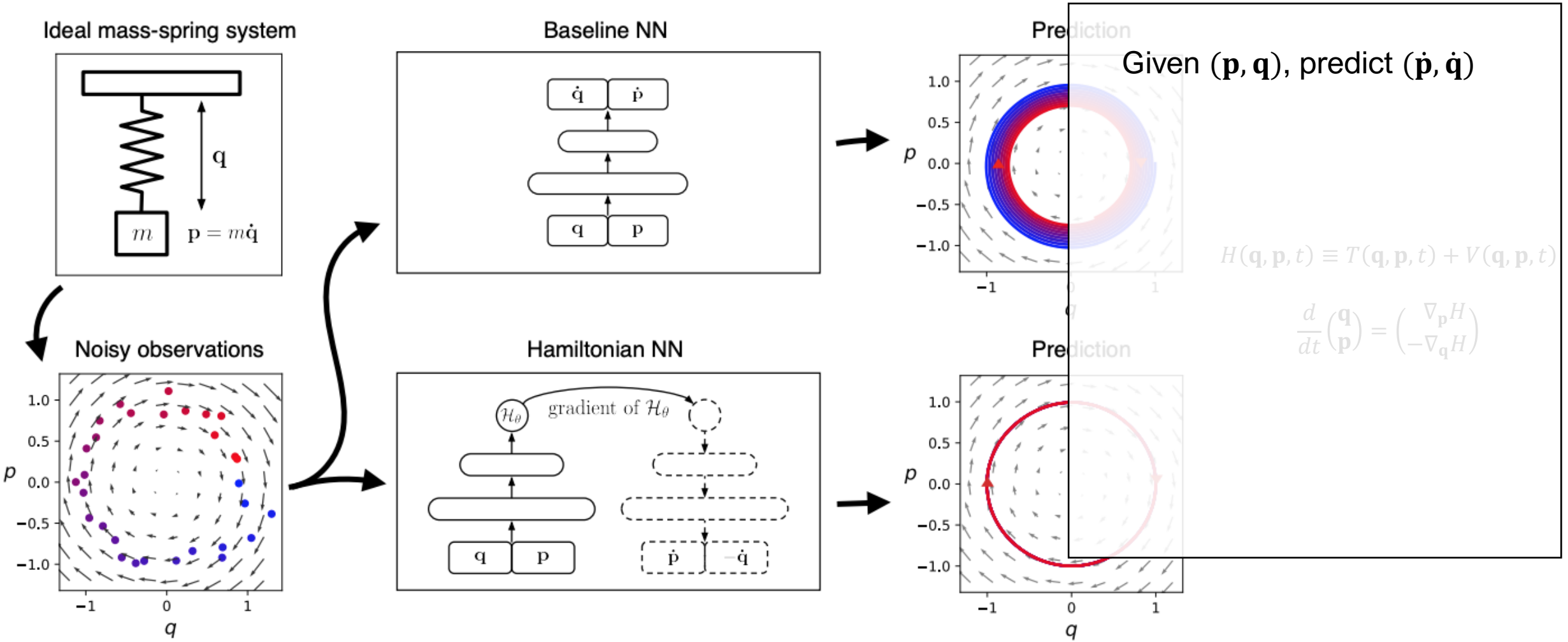


$$H(\mathbf{q}, \mathbf{p}, t) \equiv T(\mathbf{q}, \mathbf{p}, t) + V(\mathbf{q}, \mathbf{p}, t)$$

$$\frac{d}{dt}\begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\nabla_{\mathbf{p}} H\\ -\nabla_{\mathbf{q}} H\end{pmatrix}$$

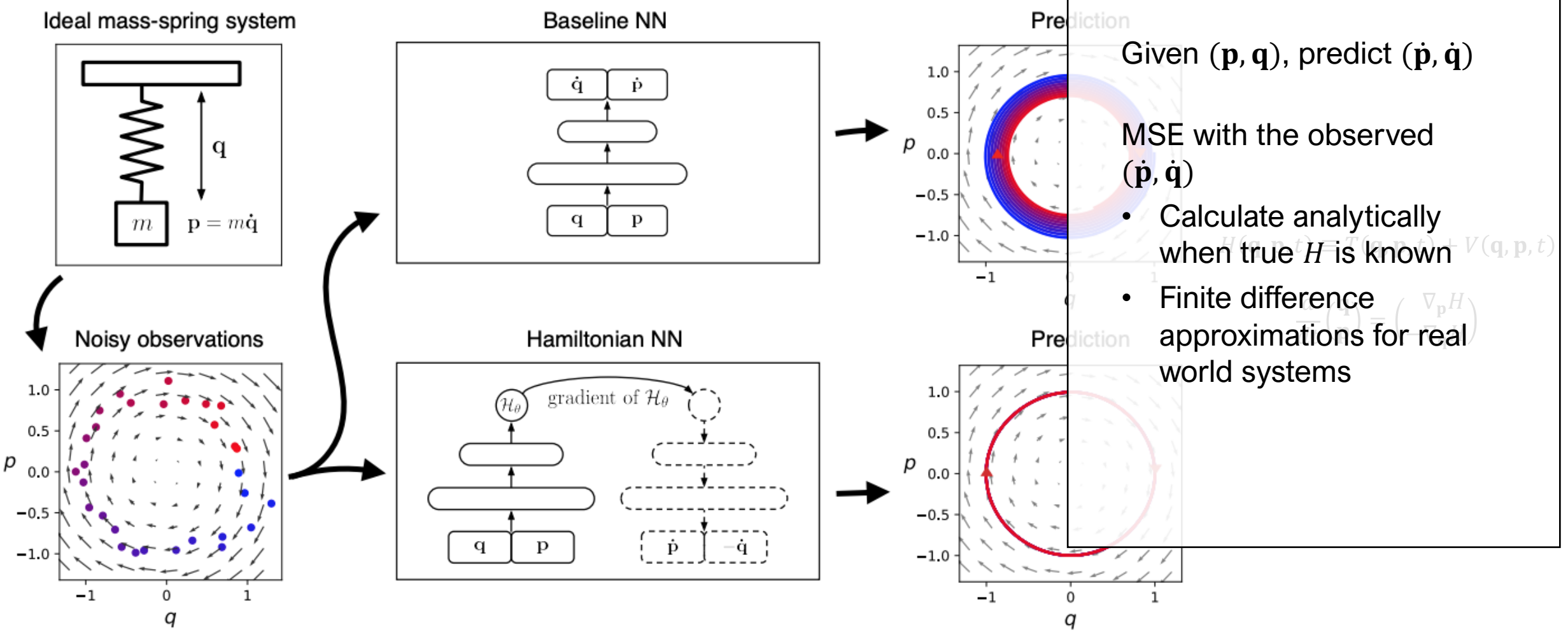
Sam Greydanus, Misko Dzamba, Jason Yosinski. Hamiltonian Neural Networks. Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html, arXiv:1906.01563

Hamiltonian Neural Networks



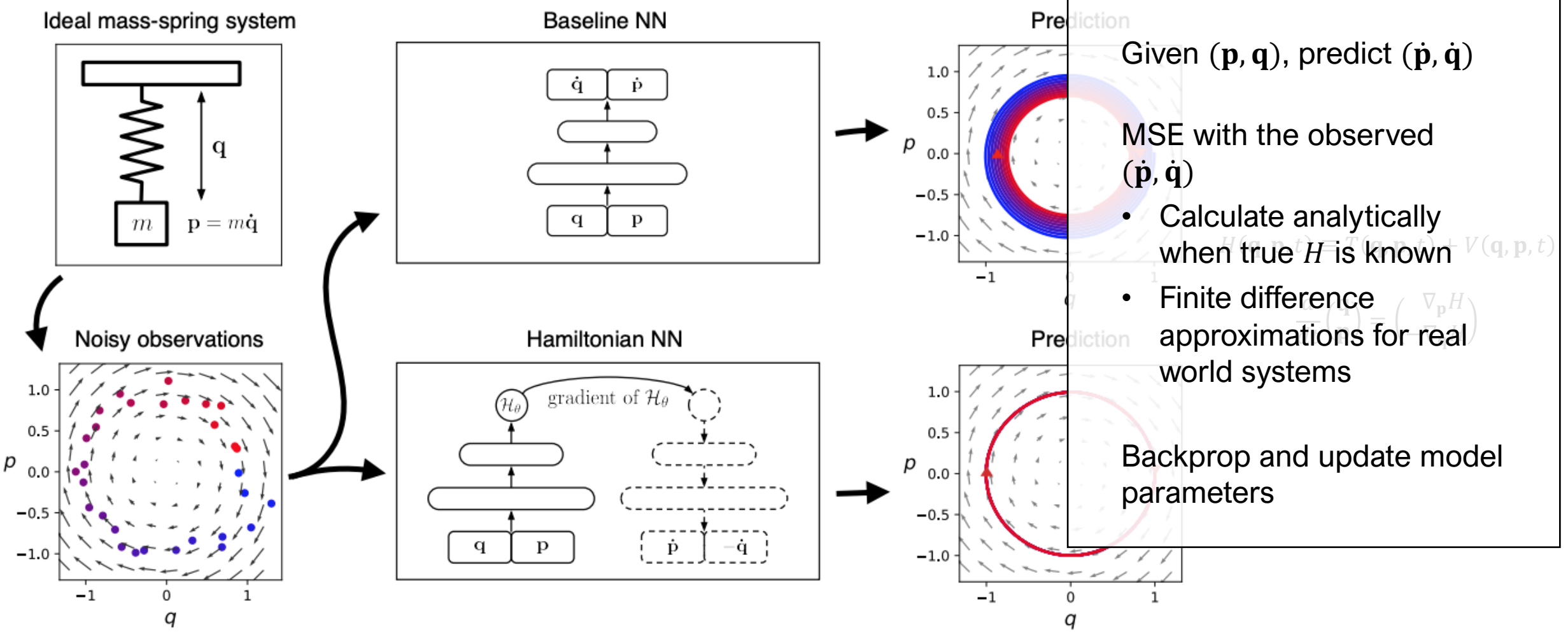
Sam Greydanus, Misko Dzamba, Jason Yosinski. Hamiltonian Neural Networks. Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html, arXiv:1906.01563

Hamiltonian Neural Networks



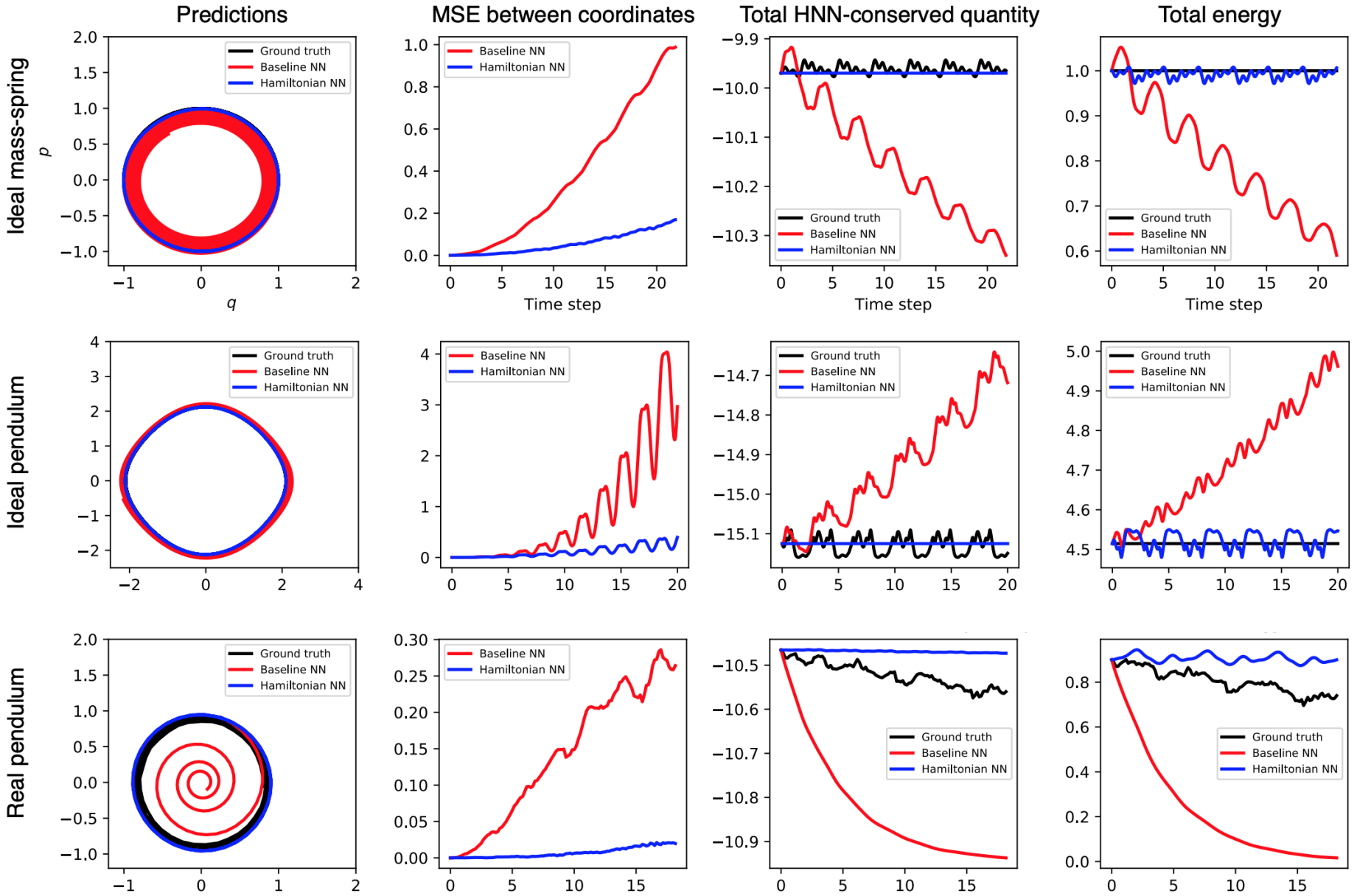
Sam Greydanus, Misko Dzamba, Jason Yosinski. Hamiltonian Neural Networks. Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html, arXiv:1906.01563

Hamiltonian Neural Networks



Sam Greydanus, Misko Dzamba, Jason Yosinski. Hamiltonian Neural Networks. Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html, arXiv:1906.01563

Results



Sam Greydanus, Misko Dzamba, Jason Yosinski. Hamiltonian Neural Networks. Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html, arXiv:1906.01563

Learning dynamics from pixel data

- Autoencoder to convert pixels to latent reprezentation (z_q, z_p)

- Additional loss to enforce the structure of the latent:

$$\mathcal{L}_{CC} = \left\| \mathbf{z}_\mathbf{p}^t - (\mathbf{z}_\mathbf{q}^t - z_\mathbf{q}^{t+1}) \right\|_2$$

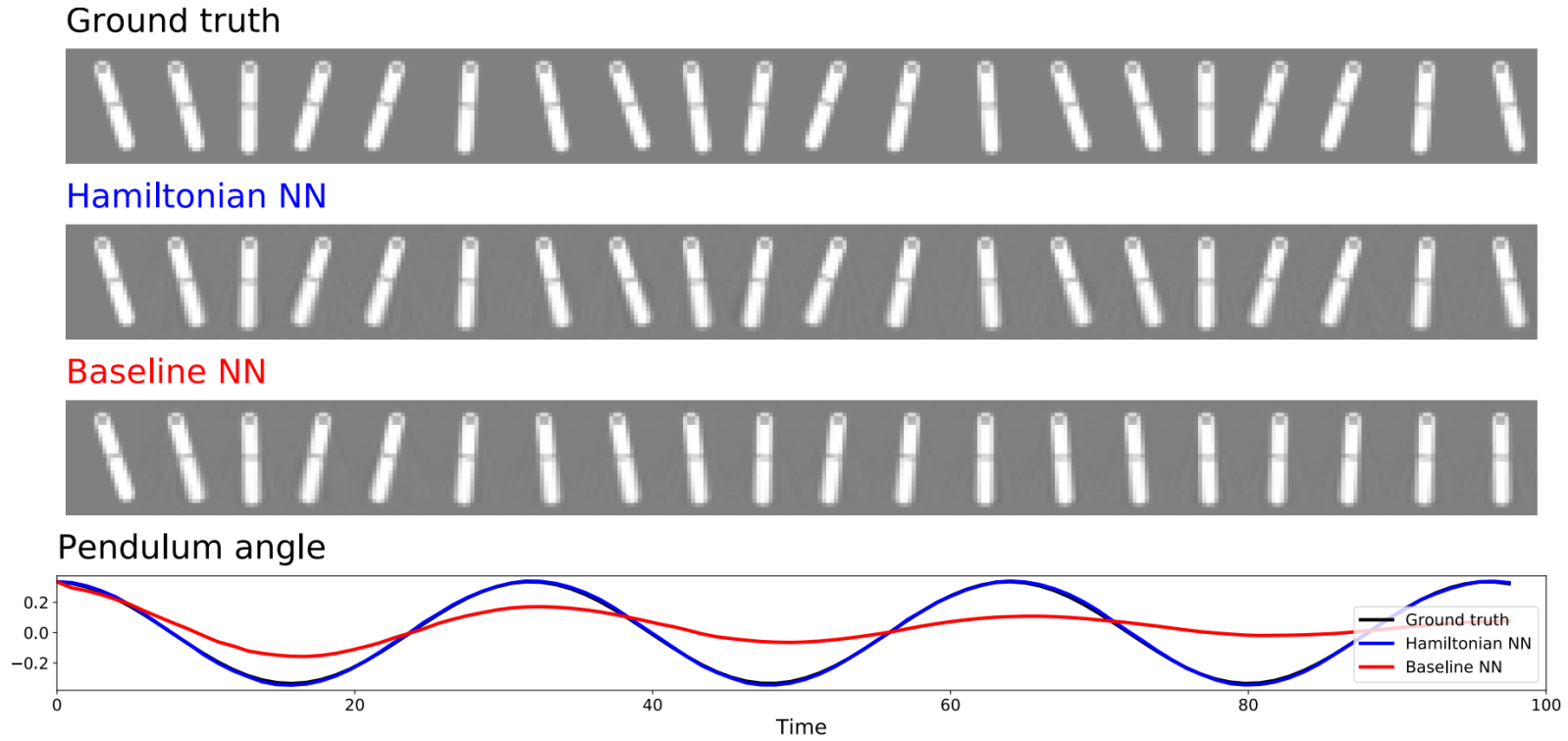


Figure 4: Predicting the dynamics of the pixel pendulum. We train an HNN and its baseline to predict dynamics in the latent space of an autoencoder. Then we project to pixel space for visualization. The baseline model rapidly decays to lower energy states whereas the HNN remains close to ground truth even after hundreds of frames. It mostly obscures the ground truth line in the bottom plot.

Sam Greydanus, Misko Dzamba, Jason Yosinski. Hamiltonian Neural Networks. Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html, arXiv:1906.01563

Lagrangian Neural Networks

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, \dot{\mathbf{q}}, t)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

- Motivation:
  - Relation between $\mathbf{p}$ and $(\mathbf{q}, \dot{\mathbf{q}})$ may be complicated
  - $\mathbf{q}$ and $\dot{\mathbf{q}}$ are easy to realte ($\dot{\mathbf{q}} \equiv \partial_t \mathbf{q}$)

M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Lagrangian Neural Networks

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, \dot{\mathbf{q}}, t)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

- Motivation:
  - Relation between $\mathbf{p}$ and $(\mathbf{q}, \dot{\mathbf{q}})$ may be complicated
  - $\mathbf{q}$ and $\dot{\mathbf{q}}$ are easy to realte ($\dot{\mathbf{q}} \equiv \partial_t \mathbf{q}$)
- To integarte the E.-L. equations, we need to solve for $\ddot{\mathbf{q}}$

M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Lagrangian Neural Networks

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, \dot{\mathbf{q}}, t)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

- Motivation:
  - Relation between $\mathbf{p}$ and $(\mathbf{q}, \dot{\mathbf{q}})$ may be complicated
  - $\mathbf{q}$ and $\dot{\mathbf{q}}$ are easy to realte ($\dot{\mathbf{q}} \equiv \partial_t \mathbf{q}$)
- To integarte the E.-L. equations, we need to solve for $\ddot{\mathbf{q}}$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}$$

Assuming no explicit time dependence, i.e.:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \sum_j \left[\frac{\partial^2 L}{\partial \dot{q}_i \partial \dot{q}_j}\ddot{q}_j + \frac{\partial^2 L}{\partial \dot{q}_i \partial q_j}\dot{q}_j\right]$$

M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Lagrangian Neural Networks

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, \dot{\mathbf{q}}, t)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

- Motivation:
  - Relation between $\mathbf{p}$ and $(\mathbf{q}, \dot{\mathbf{q}})$ may be complicated
  - $\mathbf{q}$ and $\dot{\mathbf{q}}$ are easy to realte ($\dot{\mathbf{q}} \equiv \partial_t \mathbf{q}$)
- To integarte the E.-L. equations, we need to solve for $\ddot{\mathbf{q}}$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}$$

Assuming no explicit time dependence, i.e.:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \sum_j \left[\frac{\partial^2 L}{\partial \dot{q}_i \partial \dot{q}_j}\ddot{q}_j + \frac{\partial^2 L}{\partial \dot{q}_i \partial q_j}\dot{q}_j\right]$$

Using matrix notation:

$$\left(\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}^{\mathrm{T}} L\right)\ddot{\mathbf{q}} + \left(\nabla_{\dot{\mathbf{q}}}\nabla_{\mathbf{q}}^{\mathrm{T}} L\right)\dot{\mathbf{q}} = \nabla_{\mathbf{q}} L$$

M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Lagrangian Neural Networks

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv T(\mathbf{q}, \dot{\mathbf{q}}, t) - V(\mathbf{q}, \dot{\mathbf{q}}, t)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

- Motivation:
  - Relation between $\mathbf{p}$ and $(\mathbf{q}, \dot{\mathbf{q}})$ may be complicated
  - $\mathbf{q}$ and $\dot{\mathbf{q}}$ are easy to realte ($\dot{\mathbf{q}} \equiv \partial_t \mathbf{q}$)
- To integarte the E.-L. equations, we need to solve for $\ddot{\mathbf{q}}$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}$$

Assuming no explicit time dependence, i.e.:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \sum_j \left[\frac{\partial^2 L}{\partial \dot{q}_i \partial \dot{q}_j}\ddot{q}_j + \frac{\partial^2 L}{\partial \dot{q}_i \partial q_j}\dot{q}_j\right]$$

Using matrix notation:

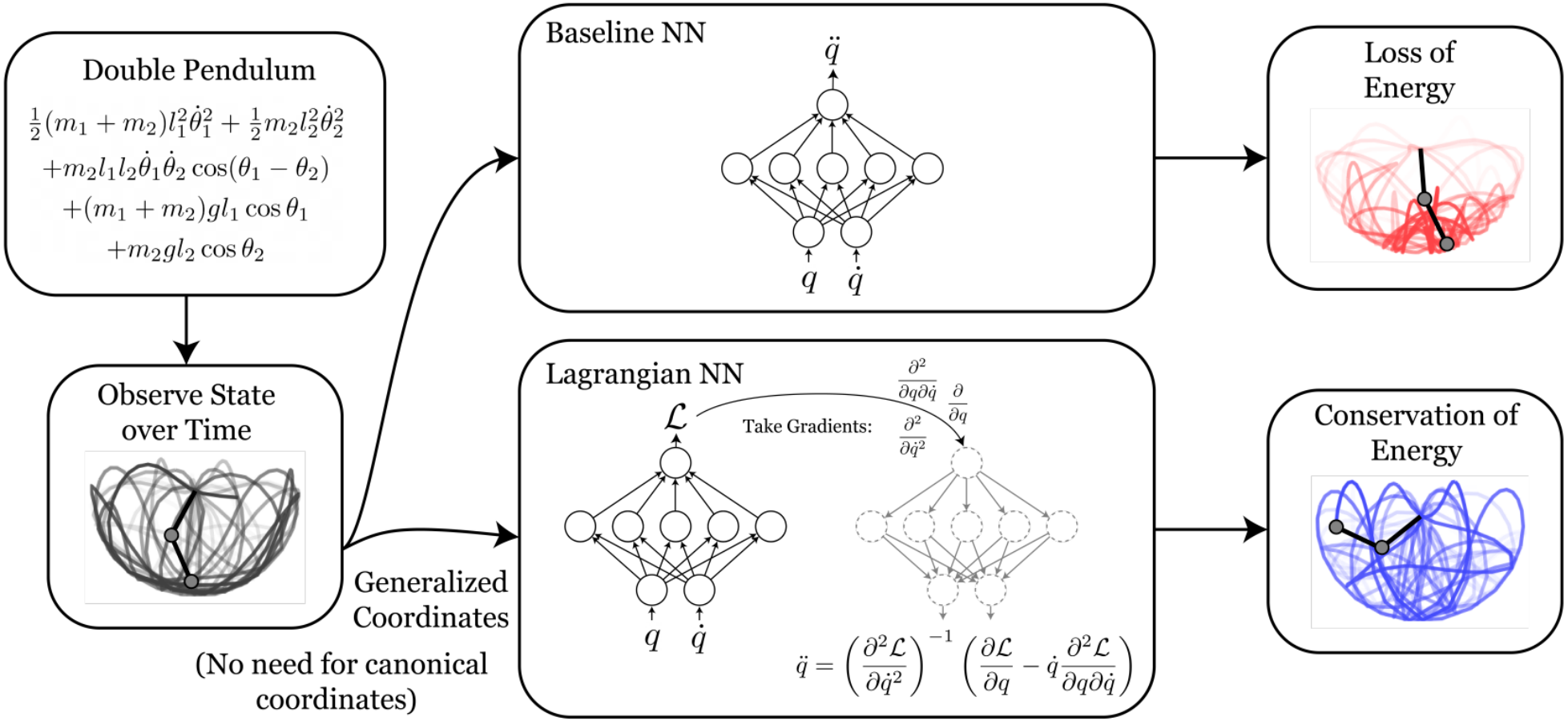
$$\left(\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}^{\mathrm{T}} L\right)\ddot{\mathbf{q}} + \left(\nabla_{\dot{\mathbf{q}}}\nabla_{\mathbf{q}}^{\mathrm{T}} L\right)\dot{\mathbf{q}} = \nabla_{\mathbf{q}} L$$

Solving for $\ddot{\mathbf{q}}$:

$$\ddot{\mathbf{q}} = \left(\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}^{\mathrm{T}} L\right)^{-1}\left[\nabla_{\mathbf{q}} L - \left(\nabla_{\dot{\mathbf{q}}}\nabla_{\mathbf{q}}^{\mathrm{T}} L\right)\dot{\mathbf{q}}\right]$$

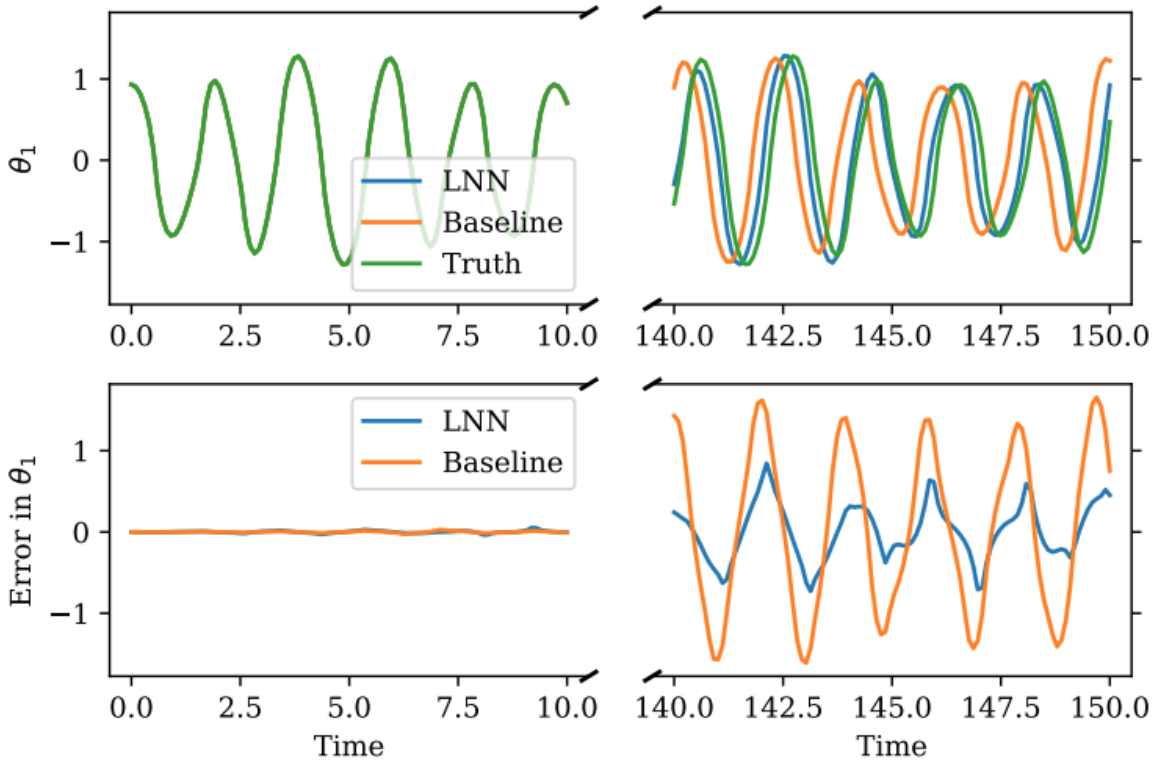
M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Lagrangian Neural Networks

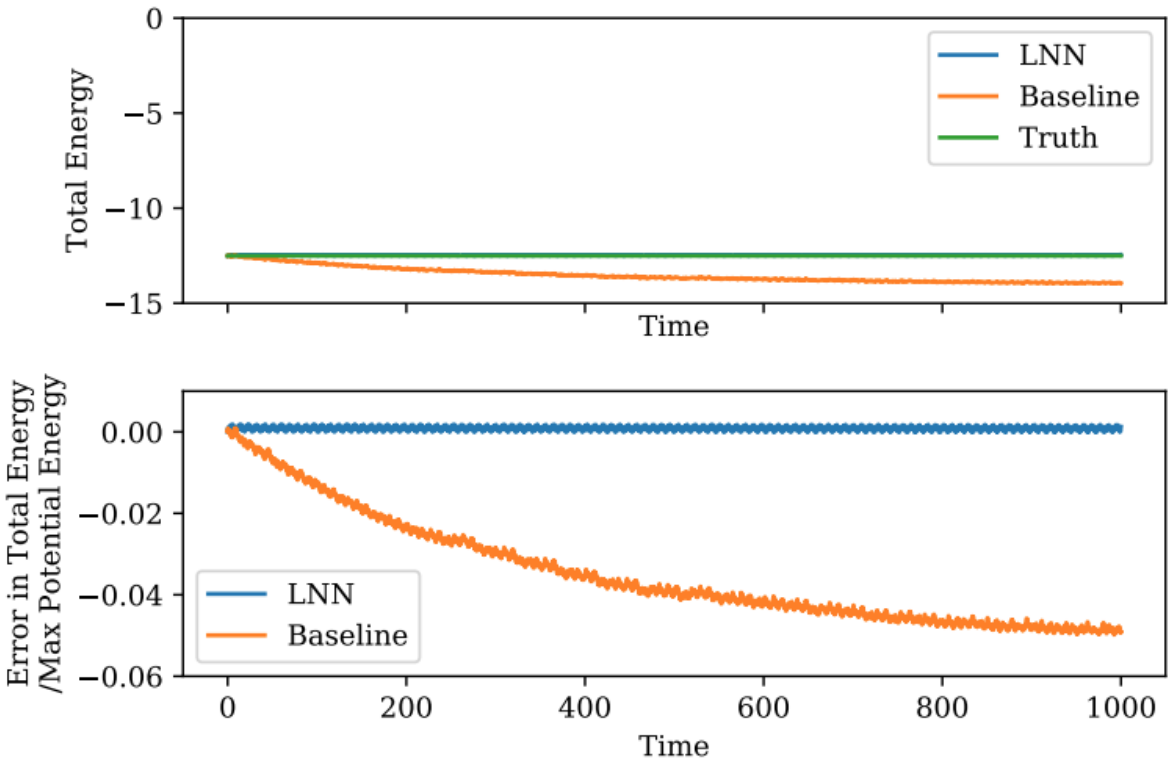


M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Lagrangian Neural Networks



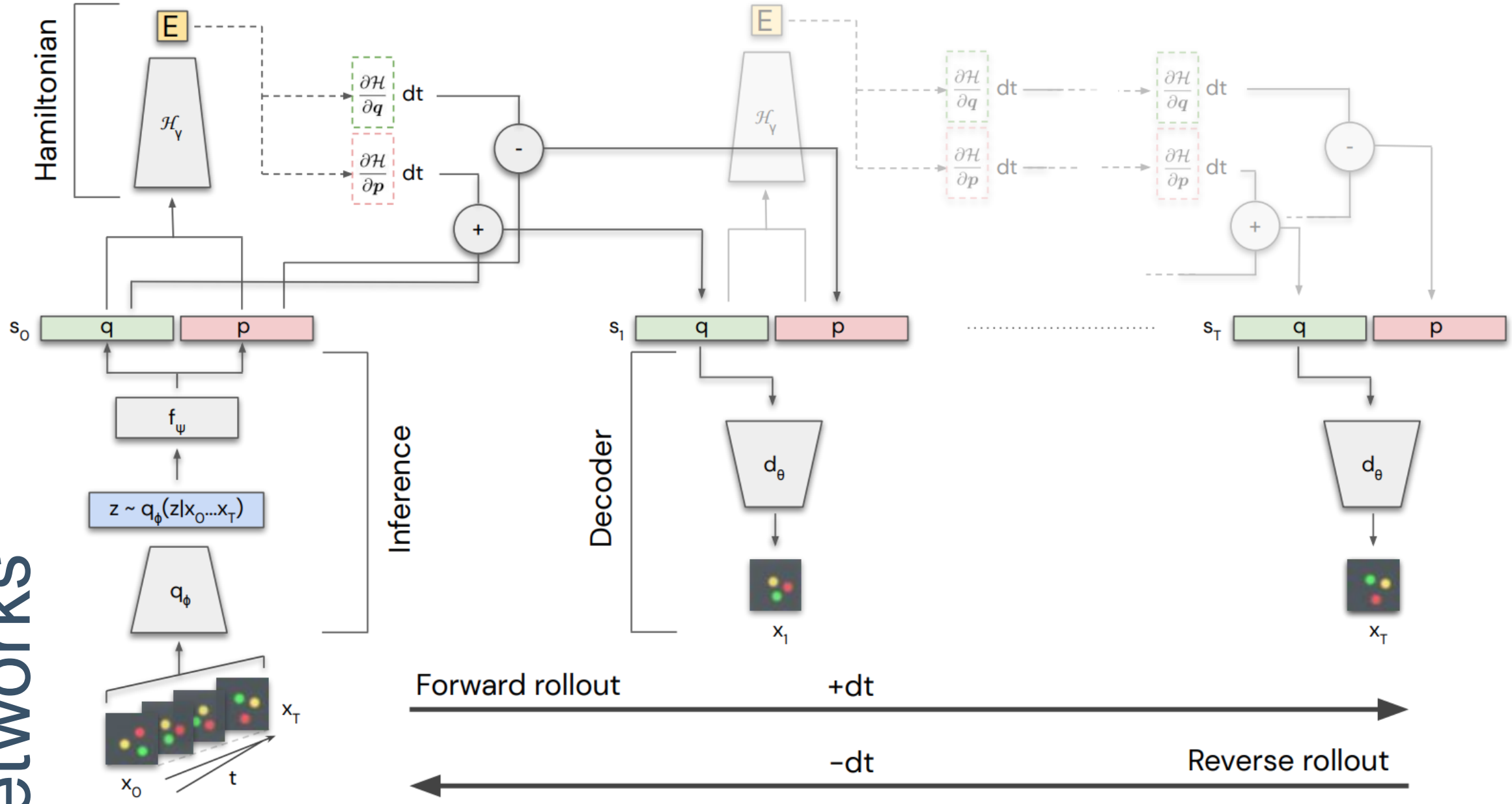
(a) Error in angle



(b) Error in energy

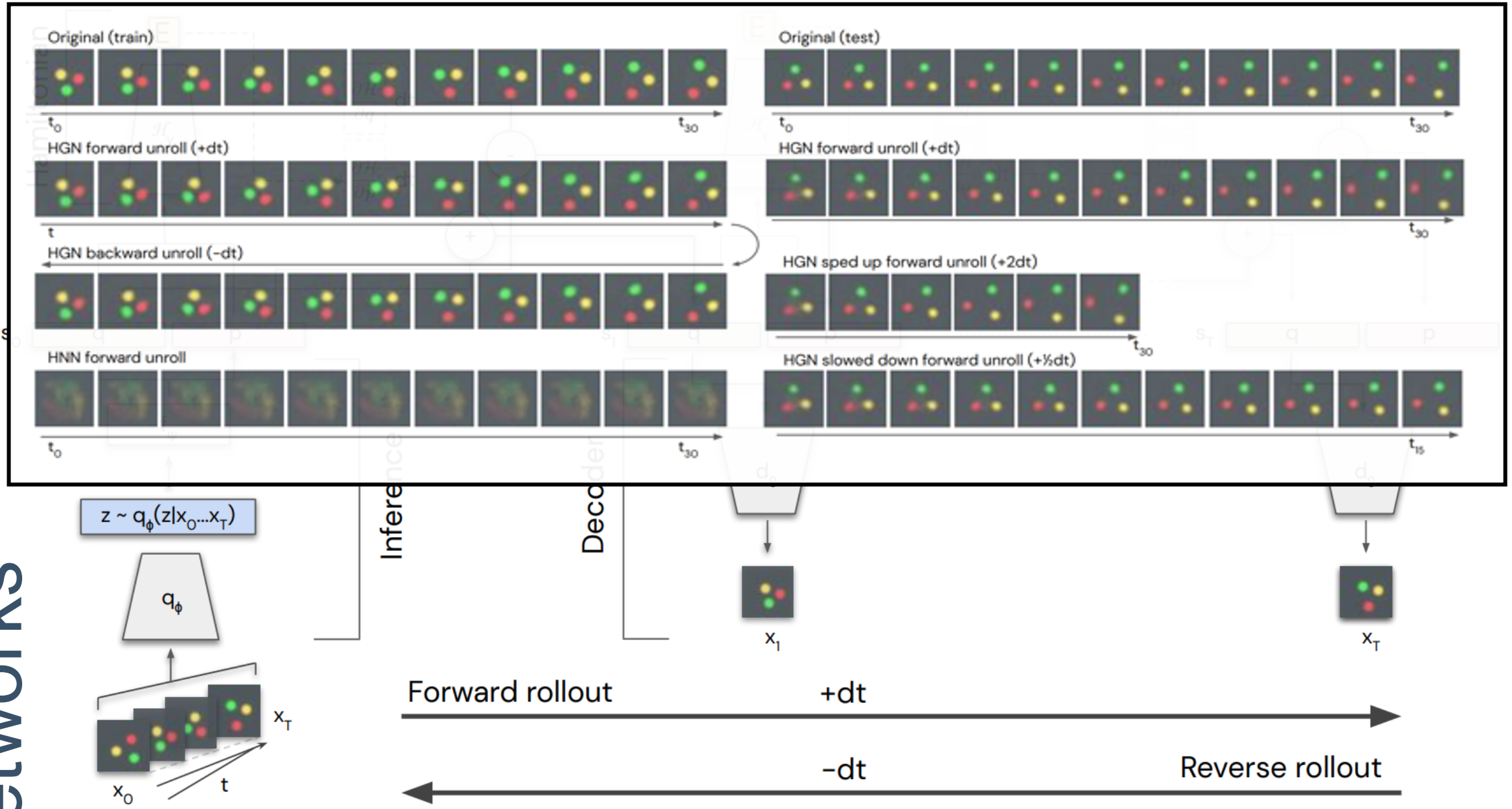
M. Cranmer, S. Greydanus, S. Hoyer, et. al. Lagrangian Neural Networks. ICLR 2020 Workshop DeepDiffEq. https://openreview.net/forum?id=iE8tFa4Nq, arXiv:2003.04630

Hamiltonian Generative Networks



P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Hamiltonian Generative Networks



P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Neural Hamiltonian Flow

- Let's examine the transformation induced by the Hamiltonian:

$$\begin{pmatrix}\mathbf{q}'\\ \mathbf{p}'\end{pmatrix} = \begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} + dt \cdot \begin{pmatrix}\nabla_{\mathbf{p}} H\\ -\nabla_{\mathbf{q}} H\end{pmatrix}$$

- One can prove that this transformation is volume-preserving:

$$\det J(dt) = 1 + O(dt^2)$$

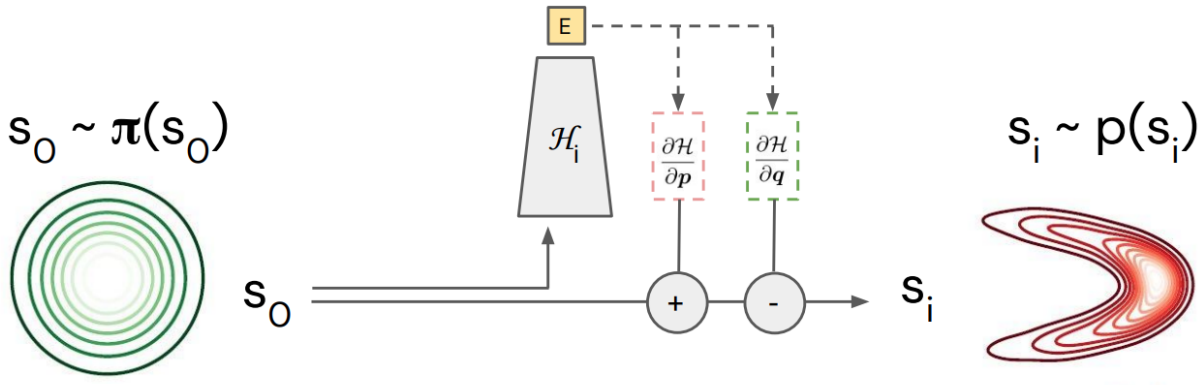
- It is also easily invertible ($dt \to -dt$)

P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Neural Hamiltonian Flow

- Prior distribution $\pi(\cdot)$ at time $t = 0$
- Observed data distribution $p_{\text{data}}(\cdot)$ at time $t = T$
- Denote by H_i^{dt} the transformation

$$\begin{pmatrix} \mathbf{q}_{i+1} \\ \mathbf{p}_{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_i \\ \mathbf{p}_i \end{pmatrix} + dt \cdot \begin{pmatrix} \nabla_{\mathbf{p}} H_i \\ -\nabla_{\mathbf{q}} H_i \end{pmatrix}$$



- Then the flow is defined by:

$$s_T = H_T^{dt} \circ H_{T-1}^{dt} \circ \cdots \circ H_1^{dt}(s_0), \qquad s_i = (\mathbf{q}_i, \mathbf{p}_i), \qquad s_0 \sim \pi(\cdot), \qquad s_T \sim p_{\text{data}}(\cdot)$$

P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Neural Hamiltonian Flow

- How to interpret part of an object as $\mathbf{q}_T$ and another part as $\mathbf{p}_T$?
- Instead: observed object is $\mathbf{q}_T$, while $\mathbf{p}_T$ is a latent variable:

$$p(\mathbf{q}_T) = \int p(\mathbf{q}_T, \mathbf{p}_T)$$

- Approximate posterior (at time $t = T$):

$$p(\mathbf{p}_T|\mathbf{q}_T) \approx f_\psi(\mathbf{p}_T|\mathbf{q}_T)$$

P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Neural Hamiltonian Flow

$$\ln p(\mathbf{q}_T) = \ln \int p(\mathbf{q}_T, \mathbf{p}_T) d\mathbf{p}_T = \ln \int \frac{p(\mathbf{q}_T, \mathbf{p}_T)}{f_\psi(\mathbf{p}_T|\mathbf{q}_T)} f_\psi(\mathbf{p}_T|\mathbf{q}_T) d\mathbf{p}_T$$

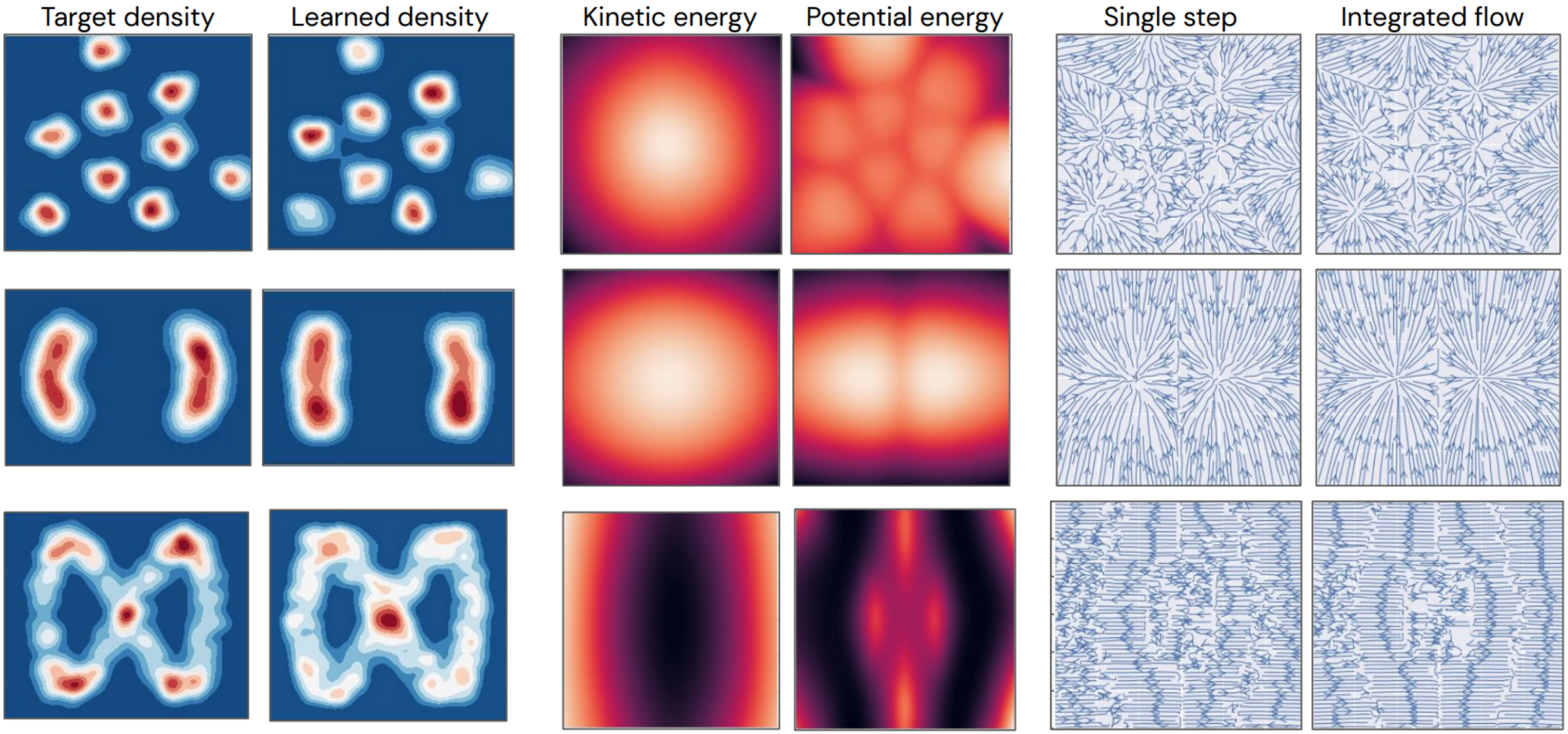
$$= \ln \mathop{\mathbb{E}}_{\mathbf{p}_T \sim f_\psi} \left[\frac{p(\mathbf{q}_T, \mathbf{p}_T)}{f_\psi(\mathbf{p}_T|\mathbf{q}_T)} \right] \geq \mathop{\mathbb{E}}_{\mathbf{p}_T \sim f_\psi} \left[\ln \frac{p(\mathbf{q}_T, \mathbf{p}_T)}{f_\psi(\mathbf{p}_T|\mathbf{q}_T)} \right]$$

using Jensen's inequality

$$= \mathop{\mathbb{E}}_{\mathbf{p}_T \sim f_\psi} \left[\ln \pi \left(H_1^{-dt} \circ \cdots \circ H_T^{-dt}(\mathbf{q}_T, \mathbf{p}_T) \right) - \ln f_\psi(\mathbf{p}_T|\mathbf{q}_T) \right]$$

P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Neural Hamiltonian Flow



P. Toth, D. J. Rezende, A. Jaegle, et. al. Hamiltonian Generative Networks. Conference paper at Eighth International Conference on Learning Representations (ICLR 2020)
https://iclr.cc/virtual_2020/poster_HJenn6VFvB.html, arXiv:1909.13789

Summary

- One may impose symmetries and conservation laws onto a modelled system by directly learning the Lagrangian or Hamiltonian
- Hamiltonian dynamics is simple to integrate, but it may be unclear how to extract the canonical momentum from the observables
- Lagrangian dynamics doesn’t need the canonical coordinates, though integration involves a matrix inversion
- Hamiltonian dynamics is driven by a volume-preserving and easily invertible transformation, allowing to define a Normalizing Flow