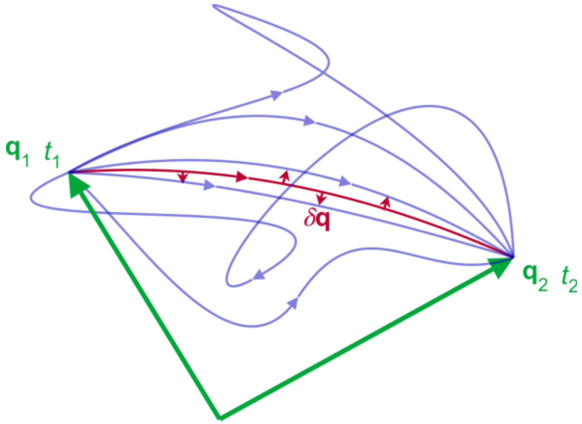
Bayesian Identification of Hamiltonian Dynamics from Symplectic Data

Nicholas Galioto (ngalioto@umich.edu) and Alex Arkady Gorodetsky
Aerospace Engineering Department, University of Michigan – Ann Arbor

Conference on Decision and Control 2020
December 14, 2020

Motivation

- We want to learn dynamical models of systems from data
  - Predict behavior
  - Control and plan
- We have a breadth of knowledge on physical systems from physics
  - Conservation of energy
  - Principle of least action
  - Stability
- In this work, we seek to enforce physical phenomena to learn Hamiltonian systems
  - Conservation
  - Reversibility
  - Symplecticness



$$\mathcal{H}(q,p) = T(q,p) + U(q,p)$$

$$\dot{q} = \frac{\partial \mathcal{H}}{\partial p} \quad \dot{p} = -\frac{\partial \mathcal{H}}{\partial q}$$

ngalioto@umich.edu

Existing Approaches

Optimal management of uncertainty can enhance performance

- Hamiltonian neural network (HNN) (Greydanus et al., 2019)
  - Parameterize the Hamiltonian
  - Minimize the objective

$$J(\theta) = \sum_{i=1}^{n} \left\| q_i - \int_{t_{i-1}}^{t_i} \frac{\partial \mathcal{H}_\theta}{\partial q} dt - q_{i-1} \right\|^2 + \left\| p_i + \int_{t_{i-1}}^{t_i} \frac{\partial \mathcal{H}_\theta}{\partial p} dt - p_{i-1} \right\|^2$$

    - Leapfrog (Toth et al., 2019; Chen et al., 2019)
- In (Galioto, Gorodetsky, 2020) we showed that objectives of this form do not account for all sources of uncertainty

S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” in *Advances in Neural Information Processing Systems*, 2019, pp. 15 353–15 363.

P. Toth, D. J. Rezende, A. Jaegle, S. Racaniere, A. Botev, ` and I. Higgins, “Hamiltonian generative networks,” *arXiv preprint arXiv:1909.13789*, 2019.

Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, “Symplectic recurrent neural networks,” *arXiv preprint arXiv:1909.13334*, 2019.

N. Galioto and A. A. Gorodetsky, “Bayesian system id: optimal management of parameter, model, and measurement uncertainty,” *Nonlinear Dynamics*, vol 102, no 1, pp. 241-267, 2020.

ngalioto@umich.edu

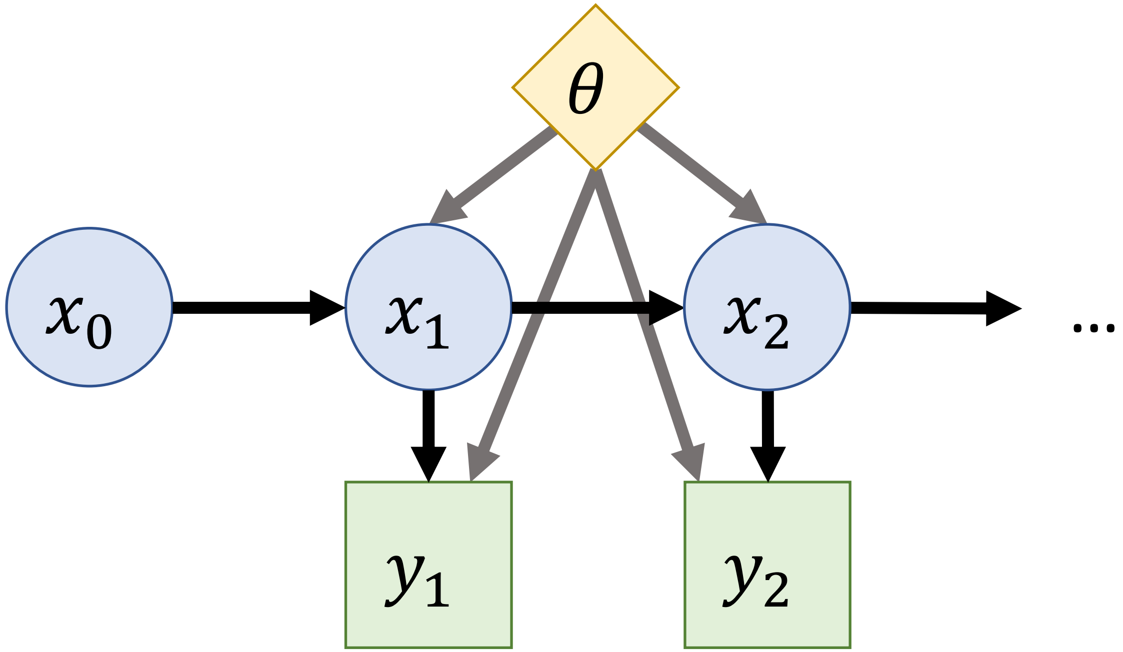
Probabilistic Formulation

Joint parameter-state estimation with stochastic dynamics

$$X_k \in \mathbb{R}^{d_x}, \qquad Y_k \in \mathbb{R}^{d_y}, \qquad \theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma) \in \mathbb{R}^{d_\theta}$$

$$X_k = \Psi(X_{k-1}, \theta_\Psi) + \xi_k; \quad \xi_k \sim \mathcal{N}(0, \Sigma(\theta_\Sigma))$$

$$Y_k = h(X_k, \theta_h) + \eta_k; \quad \eta_k \sim \mathcal{N}(0, \Gamma(\theta_\Gamma))$$



1. Parameter Uncertainty
2. Model Uncertainty
3. Measurement Uncertainty

ngalioto@umich.edu

Posterior Flow Chart

Log Joint Likelihood

$$\log \mathcal{L}(\theta;\, \mathcal{X}_n, \mathcal{Y}_n) \propto -\frac{1}{2}\sum_{k=1}^{n} \|y_k - h(x_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)} - \frac{1}{2}\sum_{k=1}^{n} \|x_k - \Psi(x_{k-1}, \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}$$

Deterministic dynamics:
$x_k = \Psi(x_{k-1})$

$$\log \mathcal{L}(\theta;\, \mathcal{Y}_n) \propto -\frac{1}{2}\sum_{k=1}^{n} \left\|x_k - h\left(\Psi^k(x_0, \theta_\Psi), \theta_h\right)\right\|^2$$

- Ayed et al., 2019
- Long et al., 2018
- Zhong et al., 2019

Identity observations:
$y_k = x_k$

$$\log \mathcal{L}(\theta;\, \mathcal{Y}_n) \propto -\frac{1}{2}\sum_{k=2}^{n} \|y_k - \Psi(y_{k-1}, \theta_\Psi)\|^2$$

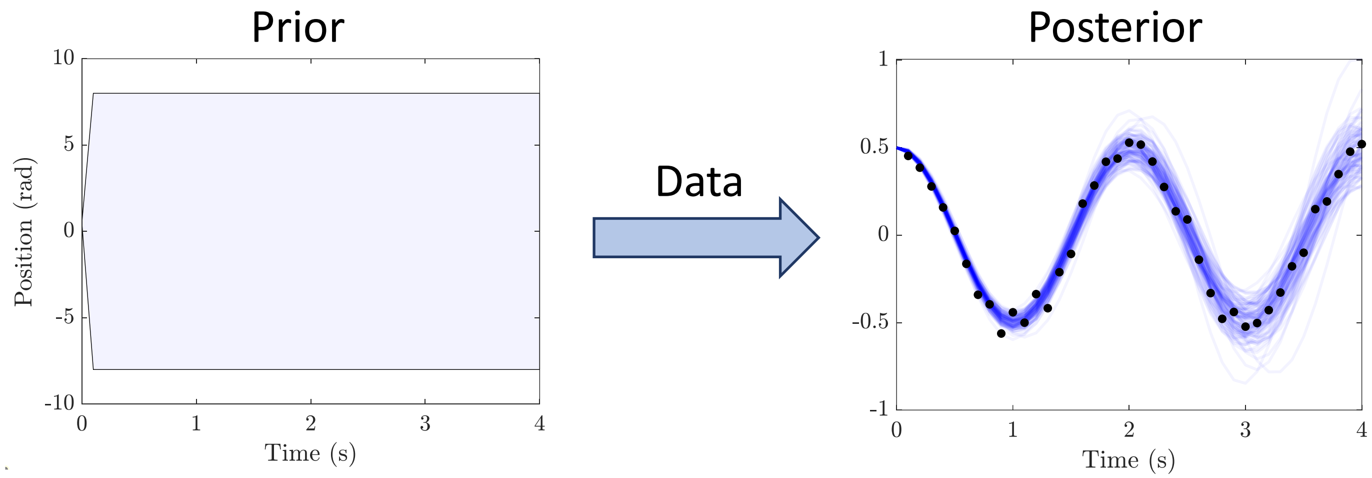
- Hamiltonian neural network
- Hills et al., 2015
- Qin et al., 2019
- Raissi, 2018

Ayed, I., de Bézenac, E., Pajot, A., Brajard, J., & Gallinari, P. (2019). Learning dynamical systems from partial observations. *arXiv preprint arXiv:1902.11136*.
Long, Z., Lu, Y., Ma, X., & Dong, B. (2018, July). Pde-net: Learning pdes from data. In *International Conference on Machine Learning* (pp. 3208-3216).
Zhong, Y. D., Dey, B., & Chakraborty, A. (2019). Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*.
Hills, D. J., Grütter, A. M., & Hudson, J. J. (2015). An algorithm for discovering Lagrangians automatically from data. *PeerJ Computer Science*, *1*, e31.
Qin, T., Wu, K., & Xiu, D. (2019). Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, *395*, 620-635.
Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, *19*(1), 932-955.

ngalioto@umich.edu

Bayesian Inference

- Goal: learn $p(\theta|\mathcal{Y}_n)$ where $\mathcal{Y}_n = (y_1, y_2, \ldots, y_n)$
- Bayes' rule: $p(\theta|\mathcal{Y}_n) = \frac{\mathcal{L}(\theta;\mathcal{Y}_n)p(\theta)}{p(\mathcal{Y}_n)}$



- Due to uncertainty in the states, we can only access the joint likelihood: $\mathcal{L}(\theta;\, \mathcal{X}_n, \mathcal{Y}_n)$
- To get the marginal likelihood, we must evaluate the integral

$$\mathcal{L}(\theta;\, \mathcal{Y}_n) = \int \mathcal{L}(\theta;\, \mathcal{X}_n, \mathcal{Y}_n) d\mathcal{X}_n$$

ngalioto@umich.edu

Approximate Marginal Markov Chain Monte Carlo (MCMC) Algorithm (Särkkä, 2013)

1. **for** $i = 1, \dots, N$
2. Propose sample θ^*

   Evaluate posterior:
3. for $\mathrm{k} = 1, \dots, n$
4. Predict: $p(X_{k+1}|\mathcal{Y}_k, \theta) = \int p(X_{k+1}|X_k, \theta) p(X_k|\mathcal{Y}_k, \theta) dX_k$
5. Update: $p(X_{k+1}|\mathcal{Y}_{k+1}, \theta) = \dfrac{p(y_{k+1}|X_{k+1}, \theta) p(X_{k+1}|\mathcal{Y}_k, \theta)}{p(y_{k+1}|\mathcal{Y}_k, \theta)}$
6. Marginalize: $\mathcal{L}(\theta; \mathcal{Y}_{k+1}) = \int p(y_{k+1}|X_{k+1}, \theta) p(X_{k+1}|\mathcal{Y}_k, \theta) dX_{k+1}$
7. **end for**
8. Accept θ^* with Metropolis-Hastings probability; otherwise reject
9. **end for**

Unscented Kalman Filter (steps 4–6)

MCMC (steps 1–9)

S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

ngalioto@umich.edu

Dynamical Model Parameterization

Ensures the learned system is Hamiltonian

$$\mathcal{H}(q, p, \theta_\Psi) = \frac{1}{2} p^T p + U(q, \theta_\Psi)$$

Differentiation

$$\dot{q} = p, \qquad \dot{p} = -\frac{\partial U(q, \theta_\Psi)}{\partial q}$$

Leapfrog Method

Conserves Hamiltonian and preserves symplectic structure throughout evaluation

$$\Psi(q_k, p_k; \theta_\Psi) = \begin{bmatrix} q_k + \Delta t p_k - \frac{\Delta t^2}{2} \frac{\partial U(q, \theta_\Psi)}{\partial q}\Big|_{q_k} \\ p_k - \frac{\Delta t}{2}\left(\frac{\partial U(q, \theta_\Psi)}{\partial q}\Big|_{q_k} + \frac{\partial U(q, \theta_\Psi)}{\partial q}\Big|_{q_{k+1}} \right) \end{bmatrix}$$

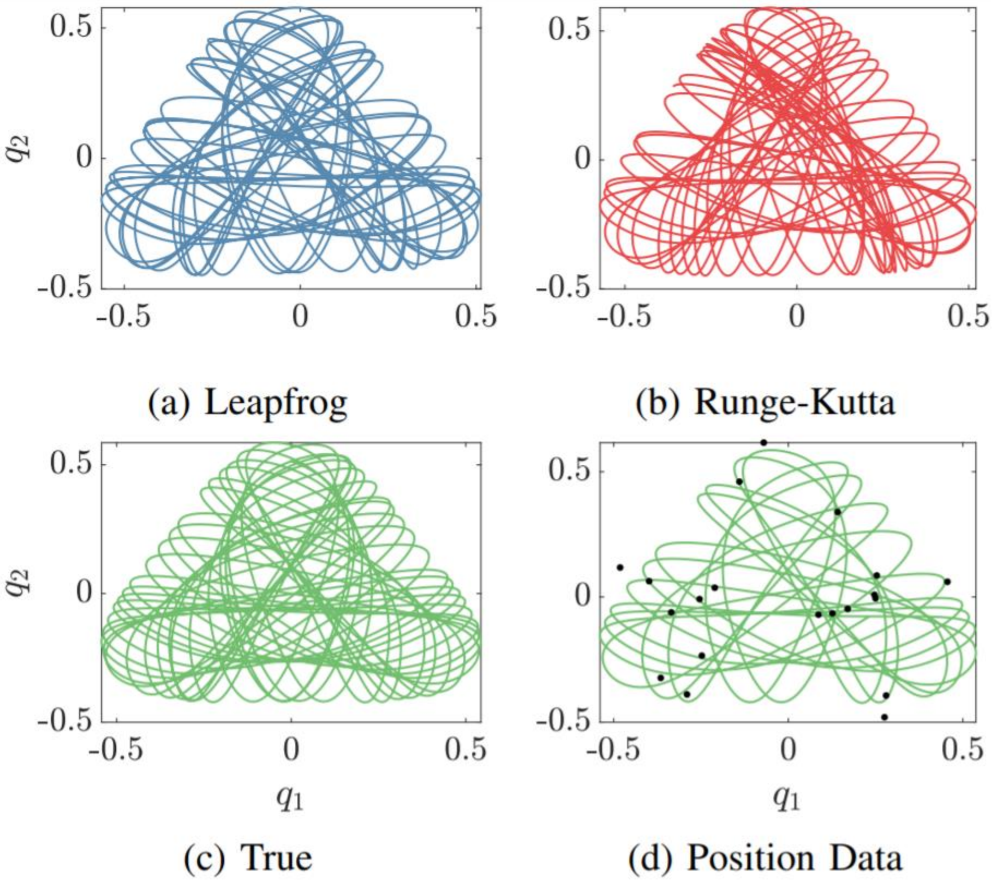
ngalioto@umich.edu

Results: Hénon-Heiles

The symplectic approach learns a more accurate Hamiltonian

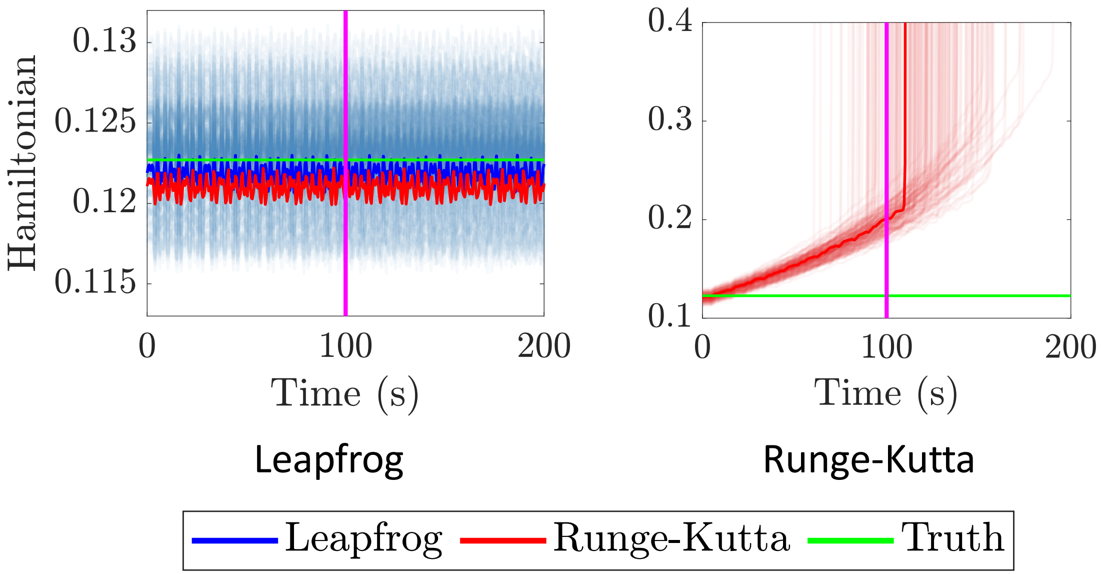
Truth: $U(q_1, q_2) = \frac{1}{2}q_1^2 + \frac{1}{2}q_2^2 + q_1^2 q_2 - \frac{1}{3}q_2^3$

Phase plots



Data Generation:
$n = 20, \quad \Delta t = 5, \quad \sigma = 0.05$

Hamiltonian over time



The method equipped with RK must learn a smaller Hamiltonian to compensate for being non-conservative

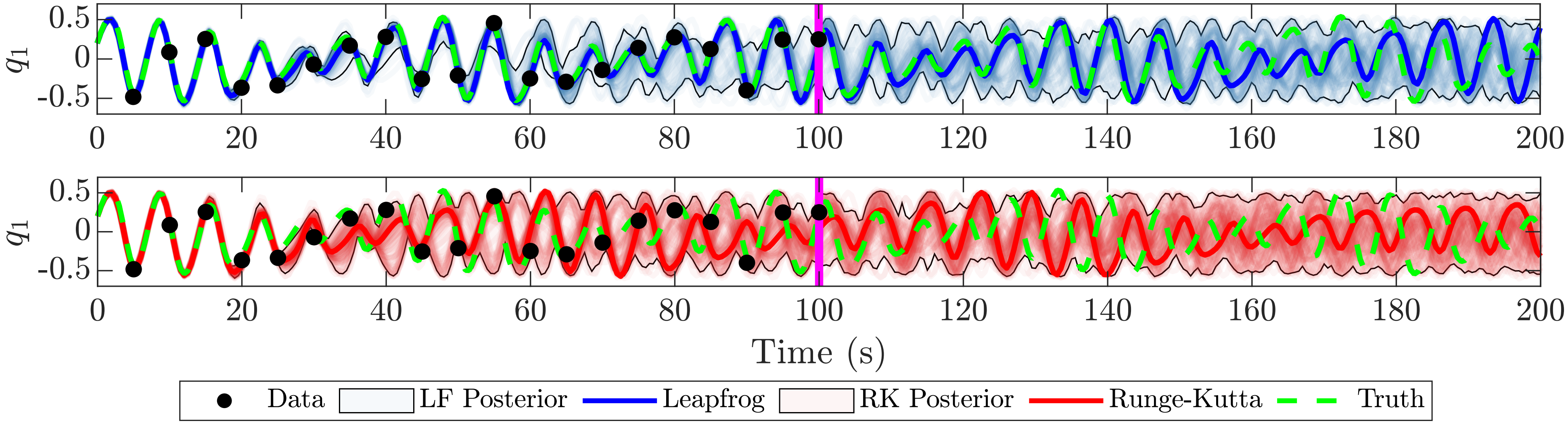
Relative mean error:
Leapfrog: 0.7%
Runge-Kutta: 1.3%

ngalioto@umich.edu

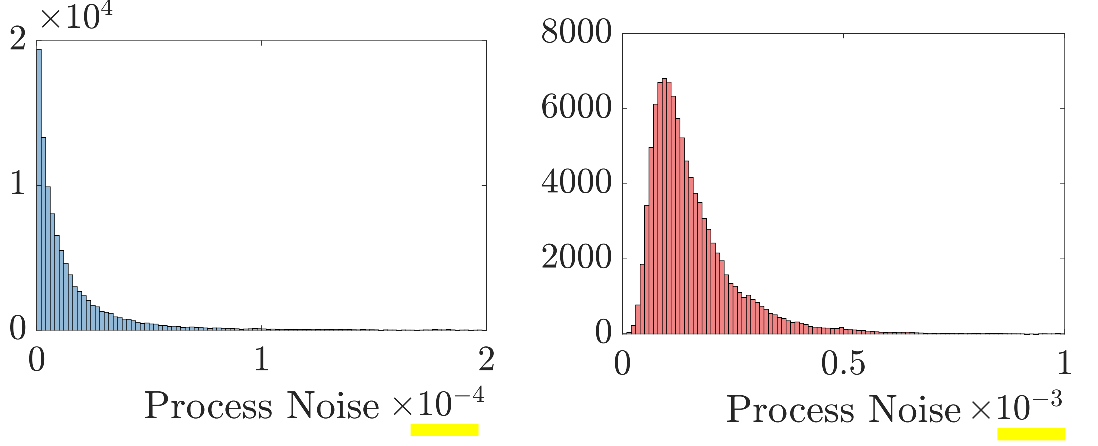
Results: Hénon-Heiles

The symplectic approach yields greater certainty

Posterior estimates of q_1 trajectory



Process noise marginal posteriors



Leapfrog

Runge-Kutta

Symplectic approach learns a model with an order of magnitude greater certainty

ngalioto@umich.edu

UNIVERSITY OF MICHIGAN

Main Takeaway

- Optimally accounting for different types of uncertainty can lead to robustness even when data are few and/or noisy
- Embedding the learning process with a symplectic integrator yields two main benefits
  - Greater accuracy
  - Greater certainty

Funding

- DARPA Physics of AI Program
  - “Physics Inspired Learning and Learning the Order and Structure of Physics.”
- AFOSR Program in Computational Mathematics

ngalioto@umich.edu