Hamiltonian Neural Koopman Operator

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recently, physics-informed learning, a class of deep learning framework that incorporates the physics priors and the observational noise-perturbed data into the neural network models, has shown outstanding performances in learning physical principles with higher accuracy, faster training speed, and better generalization ability. Here, for the Hamiltonian mechanics and using the Koopman operator theory, we propose a typical physics-informed learning framework, named as **H**amiltonian **N**eural **K**oopman **O**perator (HNKO) to learn the corresponding Koopman operator automatically satisfying the conservation laws. We analytically investigate the dimension of the manifold induced by the orthogonal transformation, and use a modified auto-encoder to identify the nonlinear coordinate transformation that is required for approximating the Koopman operator. Taking the Kepler problem as an example, we demonstrate that the proposed HNKO in robustly learning the Hamiltonian dynamics outperforms the representative methods developed in the literature. Our results suggest that feeding the prior knowledge of the underlying system and the mathematical theory appropriately to the learning framework can reinforce the capability of the deep learning.

1 Introduction

Reconstructing nonlinear dynamical systems solely using the observational noise-perturbed data is a focal challenge in various fields. The neural networks embedded with the induced biases have remarkable abilities in learning and generalizing the intrinsic kinetics of the underlying systems from the noisy data, such as the Hamiltonian neural networks [7, 8], the Lagrangian neural networks [5], and the physics-informed neural networks [13, 9, 6]. These methods are successfully applied to a variety of tasks, such as the generative tasks [17] and the dynamics reconstruction [20, 15], sharing the similar design idea where utilisation of an appropriate loss function enforces the model to nearly obey the physical principles. Progress has been achieved; however, these methods have poor generalization abilities, either enlarging the network complexity or over-fitting the noisy data during the training stage for decreasing the loss. To overcome these drawbacks, endowing the neural networks with natural physical priors becomes one of the mainstream approaches to improving the sample efficiency, the robustness, and the generalization ability [11, 10].

Recent advances in the Koopman operator theory provide a new way to identify intrinsic kinetics using linear representations for strongly nonlinear systems [3, 12, 14]. There are several algorithms for approximating such an infinite-dimensional operator in a data-driven manner, including the dynamic mode decomposition (DMD) [16, 18, 14] and the extending dynamic mode decomposition (EDMD) [19, 2]. In these works, they all try to obtain parsimonious models, while maintaining an accurate reconstruction of the unknown system in a data-driven manner.

In this article, inspired by the advances of physics-informed learning and the Koopman operator theory, we aim to develop a typical framework to efficiently and robustly learn the Hamiltonian

Submitted to the DLDE Workshop in the 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

dynamics solely using the observational noisy data. Specifically, based on the fact that the Koopman operator of the Hamiltonian dynamics is unitary, we use an orthogonal neural network to learn the approximative Koopman opertaor that naturally keeps the conservation laws, and also use an auto-encoder to identify the nonlinear coordinate transformation, mapping the original data to a low-dimensional manifold on a high-dimensional sphere. Taking the Kepler problem as an example, we demonstrate the efficacy of the proposed framework.

2 Methodology

We consider a dynamical system whose state vector $\boldsymbol{x} = (x_1, \cdots, x_n)^\top \in \mathcal{M} \subset \mathbb{R}^n$ evolves along some smooth symplectic vector field $f(\boldsymbol{x})$. Actually, the flow mappings of this autonomous dynamical system form an operator group, denoted by $\{F_t : F_t(\boldsymbol{x}) = \boldsymbol{x} + \int_0^t f(\boldsymbol{x}(s))\mathrm{d}s, \boldsymbol{x} \in \mathcal{M}, t > 0\}$. The Koopman operator $\mathcal{K}_t$ with regard to the flow F_t is an infinite-dimensional linear operator, acting on the function space $\mathcal{F} = \{g : \mathcal{M} \to \mathbb{R}\}$, and satisfying $\mathcal{K}_t g = g \circ F_t$ for $g \in \mathcal{F}$. Specifically, if we fix the time interval Δt and given an initial state $\boldsymbol{x}_0 \in \mathcal{M}$, one can accordingly generate the state trajectory by this flow, i.e., $\{\boldsymbol{x}_k : \boldsymbol{x}_k = F_{k\Delta t}\boldsymbol{x}_0\}_{k=0}^m$. Thus, we have $\mathcal{K}_{\Delta t} g(\boldsymbol{x}_k) = g(F_{\Delta t}(\boldsymbol{x}_k)) = g(\boldsymbol{x}_{k+1})$. In practice, the data $\{\boldsymbol{x}_i\}_{i=0}^m$ often contains noise perturbations due to the nature of data measurement, which therefore motivates us to robustly approximate the Koopman operator, and preserve the energy-like quantity in the original Hamiltonian dynamics. For a sake of simplicity, we denote by $\mathcal{K}_{\Delta t}$ for $\mathcal{K}$ when Δt is given.

Connections with DMD and EDMD The DMD [16, 18, 14] constructs two data matrices $\boldsymbol{X}$ and $\boldsymbol{X}'$ from the observational sequential data as $\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{m-1})$, $\boldsymbol{X}' = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)$, $\boldsymbol{X}, \boldsymbol{X}' \in \mathbb{R}^{n\times m}$. Then, obtain the optimal linear operator $\boldsymbol{K} = \boldsymbol{X}'\boldsymbol{X}^+ \in \mathbb{R}^{n\times n}$ [1] for $\boldsymbol{K}\boldsymbol{X} \approx \boldsymbol{X}'$ as the approximation of $\mathcal{K}$ [16]. Particularly, the Koopman operator for the conservative dynamical systems governed by the Hamiltonian dynamics is unitary. This property implies that we should restrict the candidate $\boldsymbol{K}$ in the orthogonal group $\mathbf{SO}(n) = \{\boldsymbol{B} \in \mathbb{R}^{n\times n} | \boldsymbol{B}\boldsymbol{B}^\top = I, \det(\boldsymbol{B}) = 1\}$, then we formally have the following vanilla optimal problem

$$\arg \min_{\boldsymbol{K}\in\mathbf{SO}(n)} \left\| \boldsymbol{K}\boldsymbol{X} - \boldsymbol{X}' \right\|_{\mathrm{F}},$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. The vanilla surrogate of the DMD has two major weaknesses:

1. The size of $\boldsymbol{K}$ is limited by the dimension n of the original system, which is not sufficiently large enough to approximate the intrinsic infinite-dimenisonal operator $\mathcal{K}$;
2. The orthogonal transformation preserves the norm of the state and hence induces its dynamics $\{\boldsymbol{K}^i\boldsymbol{x}_0\}_{i=0}^m$ embedded on a sphere, while the conserved orbit of the Hamiltonian dynamics may not be on some n-dimensional sphere.

For the first weakness, the EDMD [19, 2] lifts the dimension of $\boldsymbol{K}$ by introducing a dictionary of nonlinear observation functions $\{g_i\}_{i=1}^p$ to obtain the augmented state $\boldsymbol{y} = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \cdots, g_p(\boldsymbol{x}))^\top \in \mathbb{R}^p,\ p > n$. Analogous to the DMD, the two data matrices are constructed as $\boldsymbol{Y} = (\boldsymbol{y}_0, \cdots, \boldsymbol{y}_{m-1})$ and $\boldsymbol{Y}' = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m)$, yielding an approximated Koopman operator as $\boldsymbol{K} = \boldsymbol{Y}'\boldsymbol{Y}^+ \in \mathbb{R}^{p\times p}$. For the second weakness, in this work, we embed the augmented state $\boldsymbol{y}$ to some higher-dimensional sphere, denoted by $S^p(r) = \{\boldsymbol{y} \in \mathbb{R}^p : \|\boldsymbol{y}\|^2 = r^2\}$. Then, we obtain the following constrained optimization problem:

$$\begin{aligned} \arg\min_{\boldsymbol{K}} \left\| \boldsymbol{K}\boldsymbol{Y} - \boldsymbol{Y}' \right\|_{\mathrm{F}},\ \text{s.t. } \boldsymbol{K} \in \mathbf{SO}(p),\ r^2 \geq \max_{0\leq i\leq m} \|\boldsymbol{x}_i\|^2, \\ \text{and } \boldsymbol{y}_i = (g_1(\boldsymbol{x}_i), \cdots, g_p(\boldsymbol{x}_i))^\top \in S^p(r),\ i = 1, \cdots, m. \end{aligned} \tag{1}$$

In general, it is difficult to solve the optimization problem (1) with the classical optimization methods due to the highly nonconvex property. Hence, we transform the above problem into a neural network framework to efficiently address it.

[1] $\boldsymbol{X}^+ = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$ is the Moore-Penrose pseudo-inverse.

Auto-encoder structure of dictionary functions In the EDMD, the dictionary $\{g_i\}_{i=1}^p$ is regarded as a basis of $\mathcal{F}$ and the coordinate function is reconstructed as the optimal linear combination of these functions. In practice, it always require a large dictionary to approximate a set of basis functions. To reduce the computational cost and sustain the representation ability, we adopt an auto-encoder structure to encode $\boldsymbol{y} = (g_1(\boldsymbol{x}), \cdots, g_p(\boldsymbol{x}))^\top$ as $\boldsymbol{y} = \phi_{\boldsymbol{\theta}_1}(\boldsymbol{x})$, and decode the coordinates as $\boldsymbol{x} = \phi_{\boldsymbol{\theta}_2}^{-1}(\boldsymbol{y})$. We train the parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ of this auto-encoder using the loss function as

$$\mathcal{L}_{\text{dict}}(\boldsymbol{\theta}) = \sum_{i=0}^{m} \left\| \boldsymbol{x}_i - \phi_{\boldsymbol{\theta}_2}^{-1}(\phi_{\boldsymbol{\theta}_1}(\boldsymbol{x}_i)) \right\|^2.$$

Introduction of the neural orthogonal method The Lie exponent map $\boldsymbol{A} \to \exp(\boldsymbol{A}) := \boldsymbol{I} + \boldsymbol{A} + \frac{1}{2}\boldsymbol{A}^2 + \cdots$ from the skew-symmetric group $\mathfrak{so}(n) = \{\boldsymbol{A} \in \mathbb{R}^{n\times n} : \boldsymbol{A} + \boldsymbol{A}^\top = \boldsymbol{0}\}$ to $\mathbf{SO}(n)$ is surjective [11], and there exists an isomorphism α from $\mathbb{R}^{\frac{n(n-1)}{2}}$ to $\mathfrak{so}(n)$ as $\alpha(\boldsymbol{A}) = \boldsymbol{A} - \boldsymbol{A}^\top$, where $\boldsymbol{A} \in \mathbb{R}^{\frac{n(n-1)}{2}}$ is identified as an upper triangular matrix with the zero diagonal elements. Thus, we can represent the approximative orthogonal Koopman operator as a parameterized form $\boldsymbol{K} = \exp(\alpha(\boldsymbol{A}))$, where $\boldsymbol{A}$ owns $\frac{n(n-1)}{2}$ learnable parameters. Then we train the matrix K using the loss function as

$$\mathcal{L}_{\text{koop}}(\boldsymbol{K}) = \sum_{i=0}^{m-1} \|\boldsymbol{K}\boldsymbol{y}_i - \boldsymbol{y}_{i+1}\|^2 = \sum_{i=0}^{m-1} \|\exp(\alpha(\boldsymbol{A}))\phi_{\boldsymbol{\theta}_1}(\boldsymbol{x}_i) - \phi_{\boldsymbol{\theta}_1}(\boldsymbol{x}_{i+1})\|^2.$$

Notice that, in our framework, the orthogonality of the operator $\boldsymbol{K}$ is automatically guaranteed by its construction procedure. Indeed, this framework does not require an injection of the regularization terms into the loss function as those usually required in the literature [7, 5].

Constrain the dimension of the embedded manifold To ensure the augmented states $\{\boldsymbol{y}_i\}_{i=0}^m$ on some p-dimensional sphere, we set the radius r of the embedded sphere as a learnable parameter and simply set the distance to the origin as the loss function

$$\mathcal{L}_{\text{sphere}}(\boldsymbol{\theta}, r) = \sum_{i=0}^{m} \left(\|\boldsymbol{y}_i\|^2 - r^2\right)^2 = \sum_{i=0}^{m} \left(\|\phi_{\boldsymbol{\theta}_1}(\boldsymbol{x}_i)\|^2 - r^2\right)^2.$$

From Theorem 3.1, it follows that the trajectory generated by $\boldsymbol{K} \in \mathbf{SO}(p)$ belongs to an at most $\lfloor p/2 \rfloor$-dimensional manifold. Hence, the freedom degree of the augmented state trajectory $\{\boldsymbol{y}_i\}_{i=0}^m$ should be lower than $\lfloor p/2 \rfloor$. For any nonzero vector $\boldsymbol{v}$, the hyperplane equation $\langle \boldsymbol{v}, \boldsymbol{y} \rangle = 0$ minus one degree of the freedom for $\boldsymbol{y}$. Thus, to restrict the freedom degree of the augmented states, we introduce the loss as follows:

$$\mathcal{L}_{\text{deg}}(q) = \sum_{k=1}^{q} \sum_{i=0}^{m} \left\langle \frac{\boldsymbol{v}_k}{\|\boldsymbol{v}_k\|}, \boldsymbol{y}_i \right\rangle^2 = \sum_{k=1}^{q} \sum_{i=0}^{m} \left\langle \frac{\boldsymbol{v}_k}{\|\boldsymbol{v}_k\|}, \phi_{\boldsymbol{\theta}_1}(\boldsymbol{x}_i) \right\rangle^2,$$

where $\{\boldsymbol{v}_k \in \mathbb{R}^p,\ k = 1, \cdots, q\}$ are learnable parameters and $q \in \mathbb{Z}^+$ holds $q \le p - 2$. To guarantee that $\{\boldsymbol{v}_k\}_{k=1}^q$ are linearly independent, we design an orthogonal regularization term as $\mathcal{L}_{\text{ind}} = \sum_{k\neq j} \langle \boldsymbol{v}_k, \boldsymbol{v}_j \rangle^2$. Finally, we train the parameters $\{\boldsymbol{\theta}, K, r, V = (\boldsymbol{v}_1, ..., \boldsymbol{v}_q)\}$ using the following integrated loss function

$$\mathcal{L}(K, \boldsymbol{\theta}, r, V) = \mathcal{L}_{\text{dict}}(\boldsymbol{\theta}) + \mathcal{L}_{\text{koop}}(K) + \mathcal{L}_{\text{sphere}}(\boldsymbol{\theta}, r) + \mathcal{L}_{\text{deg}}(q, \boldsymbol{\theta}, V) + \mathcal{L}_{\text{ind}}(V), \tag{2}$$

where q is an adjustable hyperparameter. We use $\mathcal{L}_{\text{sphere}}$ and $\mathcal{L}_{\text{deg}}$ to embed the original n-dimensional data to the $(p - q - 1)$-dimensional manifold on $S^p(r)$. Here we select q such that $p - \lfloor p/2 \rfloor - 1 \le q \le p - 2$ with $p - q - 1 \le \lfloor p/2 \rfloor$.

Robustness learning from noisy data The proposed framework can be directly applied to the noise perturbed case, i.e., $\{\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \xi_i\}_{i=0}^m$, where $\{\xi_i\}$ are the samples of independent random variables. Generally, the existing models, such as the DMD, the EDMD and the other learning-based methods, likely to overfit the noise perturbed data due to a lack of prior knowledge. For example, it may break the conservative law in the Hamiltonian dynamics as the noise perturbed data act as an effect of a diffusion dynamics. On the contrary, our framework can identify the noisy samples effectively, because the conservation is naturally rooted in the orthogonal transform. We numerically validate the natural conservation in Appendix A.2.2.

3 Theoretical Results

In this section, we focus on the lowest dimension of manifold $\mathcal{M} \subset \mathbb{R}^n$ that covers the trajectory $\Gamma = \{\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}^2\boldsymbol{x}, \cdots\}$ generated by the orthogonal transform $\boldsymbol{A} \in \mathbf{SO}(n)$. Although the orthogonal transform restricts the trajectory Γ to some n-dimensional sphere, this constraint is too rough to describe the manifold $\mathcal{M}$. In particular, Γ may have degrees of freedom lower than $n-1$. The following theorem precisely provides the admitted dimension for $\mathcal{M}$. The proof is presented in Appendix A.1.1.

Theorem 3.1 *For each $\boldsymbol{A} \in \boldsymbol{SO}(n)$, if the orthogonal normal form of $\boldsymbol{A}$ has s diagonal blocks, denoted by $A_i, i = 1, \cdots, s$ with $A_i = \pm 1$, $((a_i, -b_i)^\top, (b_i, a_i)^\top)$, and $b_i \neq 0$. Then, for any initial state $\boldsymbol{x}$, the trajectory $\Gamma = \{\boldsymbol{A}^j\boldsymbol{x}\}_{j=0}^{\infty}$ belongs to an $(n-s)$-dimensional manifold. Morever, Γ belongs to an at most $\lfloor \frac{n}{2} \rfloor$-dimensional manifold, where $\lfloor \cdot \rfloor$ represents the ceil function.*

Using the result in Theorem 3.1, we can determine all the admitted dimensions for $\mathcal{M}$ as $1, 2, \cdots, \lfloor n/2 \rfloor$ according to the number of diagonal blocks in A's Jordan normal form. This theorem suggests that the degrees of freedom in the embedding procedure should be restricted such that the augmented states are distributed on some at most $\lfloor p/2 \rfloor$-dimensional manifold.

4 Experiments

We consider the planar Kepler problem with the Hamiltonian quantity $H = \frac{m}{2}(\dot{x}^2 + \dot{y}^2) - \frac{gm^2}{\sqrt{x^2+y^2}}$, which is a special case for the classical n-body Hamiltonian system with $n = 1$ [1]. Here, (x, y) and $(\dot{x}, \dot{y})$ represent the coordinate and the velocity, respectively. Here, we set $m = g = 1$ and sample the trajectory of this 4-D equation with the `odeint` package [4] and add the Gaussian noise $\mathcal{N}(0, 0.03)$ to each data point. We compare our HNKO with the currently mainstream methods, the HNN [7] and the EDMD [19], in terms of the trajectory prediction and the energy conservation. As shown in Figure 1, our HNKO outperforms the other methods in terms of robustness and accuarcy.

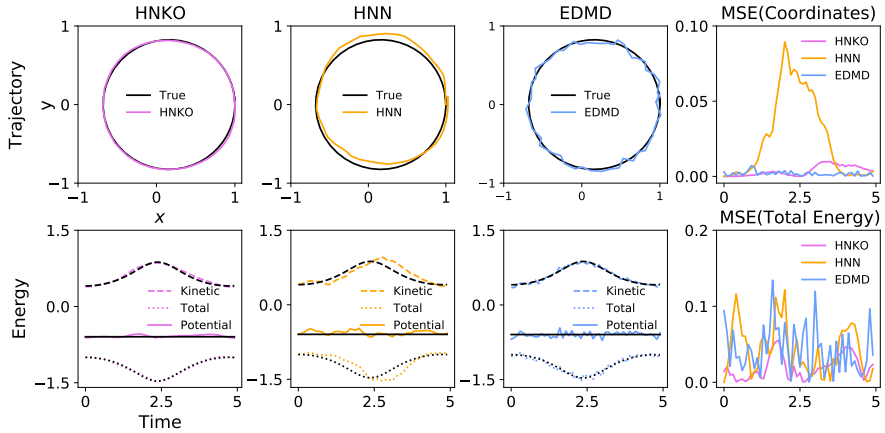

Figure 1: Results of Kepler problem. The first row and the second row show the qualitative comparison of trajectories and different kinds of energy, respectively. The right column displays the mean square error (MSE) for trajectories and energy prediction.

5 Conclusion

We propose a learning-based framework to approximate the Koopman operator for the Hamiltonian dynamics. We provide theoretical analysis for the dimension of manifold induced by the orthogonal transform and using an auto-encoder to encode the original data to some low-dimensional manifold restricted in some high-dimensional sphere, then decode it back to the original data space. We show the proposed framework outperforms the existing methods through the numerical comparison on the Kepler problem.

References

[1] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

[2] Steven L Brunton. Notes on koopman operator theory. *Universität von Washington, Department of Mechanical Engineering, Zugriff*, 30, 2019.

[3] Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

[4] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[5] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

[6] Filipe de Avila Belbute-Peres, Yi-fan Chen, and Fei Sha. Hyperpinn: Learning parameterized differential equations with physics-informed hypernetworks. In *The Symbiosis of Deep Learning and Differential Equations*, 2021.

[7] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

[8] Chen-Di Han, Bryan Glaz, Mulugeta Haile, and Ying-Cheng Lai. Adaptable hamiltonian neural networks. *Physical Review Research*, 3(2):023156, 2021.

[9] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[10] Mario Lezcano-Casado. Trivializations for gradient-based optimization on manifolds. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 9154–9164, 2019.

[11] Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning*, pages 3794–3803. PMLR, 2019.

[12] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

[13] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[14] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

[15] Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ode integrators. *arXiv preprint arXiv:1909.12790*, 2019.

[16] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[17] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

[18] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

[19] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[20] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.

A Appendix

A.1 Proofs

A.1.1 Proof of Theorem 3.1

First, denote the orthogonal normal form of $\boldsymbol{A}$ as $J\text{diag}(A_1, \cdots, A_s)$, and there exists transform matrix $\boldsymbol{P} \in \mathbf{SO}(n)$ such that $\boldsymbol{A} = \boldsymbol{P}^\top \boldsymbol{J}\boldsymbol{P}$. Then the transform $\boldsymbol{P}$ induce an isomorphism between $\Gamma = \{\boldsymbol{A}^j \boldsymbol{x}\}_{j=0}^{\infty}$ and $\tilde{\Gamma} = \{\boldsymbol{J}^j \boldsymbol{y}\}_{j=0}^{\infty}$, where $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$. Hence, we only need to identify the degree of freedom of $\tilde{\Gamma}$.

Next, split the state $\boldsymbol{y}$ as $\boldsymbol{y} = (y_1^\top, \cdots, y_s^\top)^\top$ such that the number of variables of y_s is the same as the order of A_s. Then we have $\boldsymbol{J}\boldsymbol{y} = ((A_1 y_1)^\top, \cdots, (A_s y_s)^\top)$. Since $A_i, i = 1, \cdots, s$ are also orthogonal matrix, then preserve the norm of the corresponding vector, i.e.

$$\|A_i y_i\| = \|y_i\|,\ A_i \in \mathbf{SO}(2),\ \text{ or }\ |A_i y_i| = |y_i|,\ A_i \in \mathbf{SO}(1).$$

Each of the above equality constraint will minus one degree from $\tilde{\Gamma}$. Then these s equations will restrict $\tilde{\Gamma}$ on some $(n-s)$-dimensional manifold, which completes the proof.

A.2 Experimental Configurations

Numerical Implementation For parameters learning of the constructed orthogonal Koopman operator K, we use the `GeoTorch` package for constraints on the manifold [10]. All the necessary models can be trained within several minutes on a computing device with a single i7-10870 CPU with 16GB memory.

A.2.1 Kepler Problem

Mathematically, the kepler problem is written as

$$\begin{aligned} \ddot{x} &= gm^2 \frac{x}{(x^2+y^2)^{\frac{3}{2}}}, \\ \ddot{y} &= gm^2 \frac{y}{(x^2+y^2)^{\frac{3}{2}}}. \end{aligned} \tag{3}$$

The corresponding physical quantities are

$$\begin{aligned} &\text{Kinetic energy: } E_k = \frac{m}{2}(\dot{x}^2 + \dot{y}^2), \\ &\text{Potential energy: } E_p = -gm^2 \frac{1}{\sqrt{x^2+y^2}}, \\ &\text{Total energy: } E = E_k + E_p \end{aligned} \tag{4}$$

We uniformly sample 50 observables $\{\boldsymbol{x}_i\}_{i=0}^{49}$ from the initial point $(1.0, 0.0, 0.0, 0.9)^\top$ on time interval $[0, 5]$, then we add Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.03$ to the samples.

In the learning procedure, we use a $4 \times 40 \times 40 \times 40 \times 13$ fully connected neural network (FNN) with **Tanh** activation to encode the original 4-D data to 13-D sphere and use a $13 \times 40 \times 40 \times 4$ FNN as decoder. The Koopman operator K is constructed as a linear layer in FNN without bias under constraint `geotorch.orthogonal`. We train the parameters simultaneously with the loss function (2).

In the predicting procedure, we select any point $\boldsymbol{x}$ on the original continuous trajectory as initial point and encode it to $\boldsymbol{y} = \phi_{\boldsymbol{\theta}_1}(\boldsymbol{x})$, then we use K to generate the high-dimensional predictive trajectory $\{K^j \boldsymbol{y}\}_{j=0}^{l}$. Finally, we use decoder to obtain the predictive trajectory in original space $\{\hat{\boldsymbol{x}}_i = \phi_{\boldsymbol{\theta}_2}^{-1}(K^j \boldsymbol{y})\}_{j=0}^{l}$. Specifically, we choose $\boldsymbol{x} = \boldsymbol{x}_0$ and $l = 50$ to test the performance of long term prediction of learned K.

For HNN, we use $4 \times 100 \times 100 \times 1$ FNN with **Tanh** activation and without the last layer bias as proposed by [7]. For EDMD, we use Hermite polynomial functions up to order 3 as dictionary functions, and this dictionary corresponds to a 256 order Koopman matrix, which is far greater than our HNKO. The predictive procedure is the same as above.

A.2.2 Planar Rotation

We compare the vanilla version of our HNKO and the classic DMD method in an planar rotation model to demonstrate the robustness of our HNKO.

First, we define a rotation matrix as $A = ((\cos(a), \sin(a))^\top, (-\sin(a), \cos(a))^\top)$. Then we obtain m observables as $\{A^j \boldsymbol{x}\}_{j=0}^m$. We add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to this dataset. Now we test the performance of HNKO and DMD. We select $a = 0.2,\ m = 99,\ \sigma^2 = 0.0, 0.1$. Notice that A itself is the unique solution of the following optimal problem

$$\begin{aligned} &\arg\min_{K \in \mathbb{R}^{2\times 2}} \|KX - X^{'}\|, \\ &X = (\boldsymbol{x}, A\boldsymbol{x}, \cdots, A^{m-1}\boldsymbol{x}), \\ &X^{'} = (A\boldsymbol{x}, A^2\boldsymbol{x}, \cdots, A^{m}\boldsymbol{x}). \end{aligned} \tag{5}$$

Hence DMD can exactly learn the dynamic A in noise-free case. However, DMD solution $K^* = X^{'}X^{+}$ generally cannot equal to A in the noisy case because A cannot minimize the objective function of the following oprimal problem

$$\begin{aligned} &\arg\min_{K \in \mathbb{R}^{2\times 2}} \|K\tilde{X} - \tilde{X}^{'}\|, \\ &\tilde{X} = (\boldsymbol{x} + \xi_0, A\boldsymbol{x} + \xi_1, \cdots, A^{m-1}\boldsymbol{x} + \xi_{m-1}), \\ &\tilde{X}^{'} = (A\boldsymbol{x} + \xi_1, A^2\boldsymbol{x} + \xi_2, \cdots, A^{m}\boldsymbol{x} + \xi_{m}), \\ &\xi_i \sim \mathcal{N}(0, \sigma^2),\ i = 1, \cdots, m. \end{aligned} \tag{6}$$

We demonstrate the above analysis in Figure 2. To guarantee the predictive stability of HNKO, we modify $\mathcal{L}_{\text{koop}}$ as

$$\mathcal{L}_{\text{koop}} = \sum_{i=1}^{m-2} \|K\boldsymbol{y}_i - \boldsymbol{y}_{i+1}\|^2 + \sum_{i=1}^{m-2} \|K^2\boldsymbol{y}_i - \boldsymbol{y}_{i+2}\|^2.$$

We can see that HNKO learn the exact transform A in both noisy and noise-free case, while DMD fail to learn the A in the noisy case due to overfit on the noisy data.

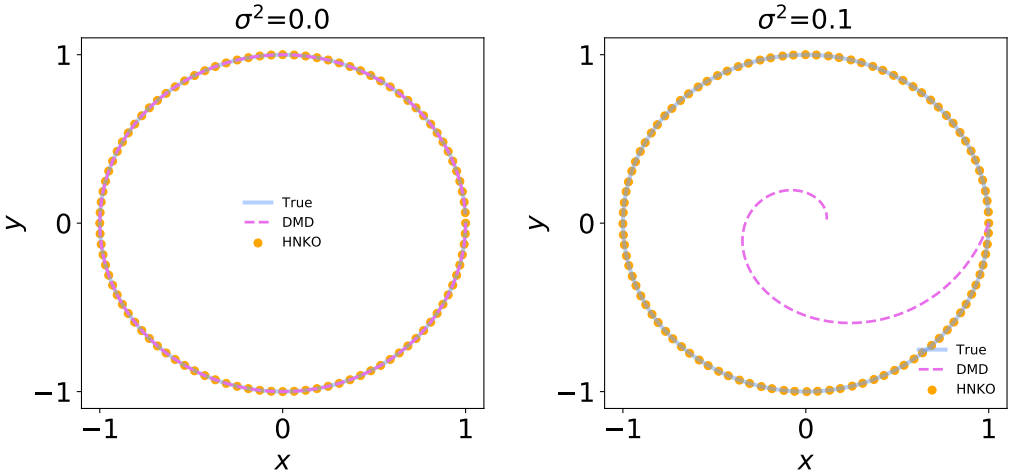

Figure 2: Results of planar rotation. The prediction performance of HNKO and DMD in noise-free case (left) and noisy case (right).