

Research Article

A Convex Relaxation Approach for Learning the Robust Koopman Operator

Yong Su ¹, Jiawei Jin ¹, Weilong Peng ², Keke Tang ³, Asad Khan ², Simin An ¹, and Meng Xing ⁴

¹Tianjin Normal University, Tianjin 300387, China

²School of Computer Science and Cyber Engineering, Guangzhou University, Guangdong, China

³Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong, China

⁴Tianjin University College of Intelligence and Computing, Tianjin, China

Correspondence should be addressed to Keke Tang; tangbohutbh@gmail.com, Asad Khan; asad@gzhu.edu.cn, and Simin An; ansm@tju.edu.cn

Received 31 March 2022; Accepted 12 May 2022; Published 28 June 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Yong Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Although data-driven models, especially deep learning, have achieved astonishing results on many prediction tasks for nonlinear sequences, challenges still remain in finding an appropriate way to embed prior knowledge of physical dynamics in these models. In this work, we introduce a convex relaxation approach for learning robust Koopman operators of nonlinear dynamical systems, which are intended to construct approximate space spanned by eigenfunctions of the Koopman operator. Different from the classical dynamic mode decomposition, we use the layer weights of neural networks as eigenfunctions of the Koopman operator, providing intrinsic coordinates that globally linearize the dynamics. We find that the approximation of space can be regarded as an orthogonal Procrustes problem on the Stiefel manifold, which is highly sensitive to noise. The key contribution of this paper is to demonstrate that strict orthogonal constraint can be replaced by its convex relaxation, and the performance of the model can be improved without increasing the complexity when dealing with both clean and noisy data. After that, the overall model can be optimized via backpropagation in an end-to-end manner. The comparisons of the proposed method against several state-of-the-art competitors are shown on nonlinear oscillators and the lid-driven cavity flow.

1. Introduction

The linear dynamical system, which can be described as a function that captures the implicit evolution rule of states in a geometrical space, has achieved huge success in a wide spectrum of applications from machine learning to automatic control. However, effectively modeling states present in highly nonlinear dynamical systems while also accurately quantifying uncertainty is still a challenging task [1]. Research on nonlinear dynamic systems has a long history, while the parameters of the complex system may not be known precisely; thus, we can only obtain approximate results and require stability analysis (such as the Lyapunov stability or structural stability) [2]. In recent years, fully data-driven machine learning methods, in particular deep neural networks, appear to be a viable alterna-

tive, which have a remarkable ability to learn complex patterns from training samples. Specifically, recurrent neural networks (RNNs) [3] are a form of neural network architecture which is mainly used to detect patterns in sequential data. RNNs pass information through the network that transmits information back into themselves, which enable them to take into account both previous states and the current state. However, as in most neural networks, vanishing or exploding gradients is a fundamental problem of RNNs. Although long short-term memory units (LSTMs) [4] can alleviate the vanishing gradient problem, it is still limited in maintaining long-term memory.

Another limitation of the deep learning method, which is also the main concern of this article, is the lack of intuitive causal interpretability. In some cases involving specific physical phenomena that need the interpretability of the model,

good performance alone is not enough to satisfy the need for practical application.

The complexity of deep neural networks makes it hard to understand and reason the predictions, which hinders its further application and improvement [5]. Instead, most engineers prefer classical off-the-shelf models as it ensures physical plausibility while the learned models only achieve good performance in the vicinity of the training data. Fortunately, motivated by this issue, researchers are becoming more and more interested in how to bridge this gap. The combination of deep learning and physics seems natural, as the compositional structure of deep networks enables the efficient computation of the derivatives, thus, can encode a differential equation describing physical processes [6–8]. The physics-based deep model not only helps to improve the generalization and interpretability of the deep model but also eliminates counterfactual components in representation learning.

Nonlinear dynamical systems are very common in engineering, but most of them are task-specific, and there exists no general framework for solving them. Therefore, representing nonlinear dynamics in a linear framework is particularly attractive, because techniques from linear dynamical systems, linear algebra, geometric, and spectral theory can be directly applied. The Koopman operator theory [9] developed in 1931 has recently become the main research focus of linear representations of nonlinear systems. Based on the Koopman operator, the analysis of nonlinear dynamical systems can be lifted to an infinite-dimensional linear space. Most algorithms handle the Koopman operator by converting an infinite-dimensional space into a finite-dimensional space. In this way, it naturally combines the neural network with flexible expression ability for subspace and the physical regularization based on the Koopman operator to achieve the global linear expression of the nonlinear system. Typically, by embedding the sequence onto a low-dimensional latent space in an unsupervised manner, the Koopman operator can be approximated by a linear layer of a neural network. This deep Koopman model retains the physical interpretability and can predict the future hidden state and the hidden state of the current state [10–15].

However, most of the current methods impose strict orthogonal constraints to the Koopman eigenvectors spanned by the linear layer, which makes them sensitive to noise [16]. Drawing inspiration from the Koopman theory and convex optimization, in this paper, we propose a convex relaxation approach for learning the robust Koopman operator. Specifically, we find that solving the approximation of the Koopman operator can be regarded as an orthogonal Procrustes problem on the Stiefel manifold. In order to improve model robustness against input noise, we prove that the geometric constraint of the Koopman operator can be replaced with its convex counterpart. Without requiring extra computation cost, the proposed model, which can be trained in an end-to-end manner, shows considerable accuracy and robustness gains by incorporating this convex counterpart regular term into the deep unsupervised model. The proposed method overcomes the overconstraint problem of the Koopman operator in the previous methods [10, 11].

The numerical experiment results on two complex nonlinear dynamic systems also prove the effectiveness of the proposed algorithm.

2. Review of Related Work

Most real-world phenomena can be described by dynamic systems; hence, we can implement state prediction, estimation, and control by constructing and analyzing dynamic systems. Linear dynamical systems can be solved exactly, which can be traced back to Kalman [17]. In reality, more systems are usually complex nonlinear systems [18]. Occasionally, the solutions of some nonlinear systems can be well approximated by an equivalent linear system near its fixed points, such as EKF [19]. While nonlinearity remains a grand challenge in analyzing dynamical systems. Different from linear systems which can be completely characterized in terms of spectral decomposition, there is no similar general mathematical framework to analyze nonlinear systems.

Over the years, researchers considered analyzing the geometry of subspaces of local linearization around fixed points and global heteroclinic being effective ways to solve nonlinear dynamic systems [20]. The geometric theory provides us with a new perspective for modeling complex systems, such as the Hartman-Grobman theorem [21], which determines the conditional constraints of linear dynamics approaching nonlinear systems. Although the geometric perspective provides a quantitative local linear model, the global analysis still requires qualitative calculation, which limits the nonlinear prediction, estimation, and control theory far away from the fixed point [22]. In many fields, the lack of known physical laws makes it impossible to construct control and motion equations. For these complex nonlinear systems, which is also the main concern of modern researchers, the linearization method is more challenging, and the linear approximations often become suspect.

With increasingly powerful data-driven techniques, learning the approximate nonlinear dynamic that emerged in the data is the main focus of renewed interest. Operator theory [23] and deep learning models [24] seem to be two effective ways to automate the discovery of the underlying physical mechanisms of nonlinear dynamic systems. Deep learning methods, especially RNN and its derivative models (LSTM [4] and GRU [25]), can use nonlinear activation functions and recursive units to express nonlinear systems implicitly. However, the black-box nature of DNNs has become one of the primary obstacles to their wide acceptance in mission-critical fields, and ignoring necessary physical bias may lead to counterfactual results.

Discovering underlying dynamics from data that enable the linear representation of nonlinear systems is the main goal of operator-theoretic-based approaches. The operator theory has proved that infinite-dimensional linear operators can be used to represent nonlinear dynamic systems, such as Koopman operators [9, 26]. Although the Koopman operator is appealing, it requires infinite degrees of freedom to describe the space of measurement functions, which poses issues for representation and computation. The Koopman

operator can be approximated via the dynamic mode decomposition (DMD) algorithm [27–30].

Recently, many physics-based deep models have been proposed [10–13, 31, 32], which could encode the physical equations into their network to obtain physical plausibility and improve the generalization of deep models. Bethany et al. [12] first proposed to identify nonlinear coordinates on which the dynamics are globally linear using a modified autoencoder. Yeung et al. [11] introduced a method that automatically selects efficient deep dictionaries to approximate the Koopman operator. Samuel [31] leveraged neural networks to learn sets of adaptive, low-dimensional dictionary observables to avoid the over-fitting problem while retaining enough capacity to represent Koopman eigenfunctions. Takeishi et al. [13] combined the Koopman spectral analysis and autoencoder to learn Koopman invariant subspaces from observed data. Azencot et al. [10] considered the forward and backward dynamics of Koopman operators by adding consistency constraints to the loss function. Most recently, Li et al. [32] proposed to use graph neural networks to learn compositional Koopman operators and use a block-wise linear transition matrix to regularize the shared structure across objects. Although the motivations of these methods are different, most of them are based on the deep generative model and approximate the Koopman eigenvectors by the linear layer of the encoder. However, analysis of the Koopman operator in the above methods is based on the DMD method, and the linear operator constructed in this way may be very sensitive to noise.

3. Method

3.1. Koopman Operator. A dynamical system $\dot{x} = f(x)$ is a function that describes the time-dependence of a point in a geometrical space. Given any discrete time, a nonlinear dynamical system has a state given by a vector that can be represented by a point $x \in \mathcal{M}$ in an appropriate geometrical manifold. The evolution rule of the nonlinear discrete-time dynamical system describes what future states follow from the current state $t \in \mathbb{T}$:

$$x_{t+1} = f(x_t). \quad (1)$$

A function $g : x \rightarrow \mathbb{R}$ can be seen as an observable of the system, and the set of all observables forms an infinite-dimensional Hilbert space \mathcal{H} . The Koopman operator \mathcal{K} is defined as an infinite-dimensional linear operator that acts on observables g :

$$\mathcal{K}g(x) = g(f(x)). \quad (2)$$

Intuitively, the Koopman operator can be regarded as the state transition matrix of the evolution trajectory in an infinite-dimensional space. The key property of the Koopman operator \mathcal{K} is its linearity. For any two observables g and h and scalar values α and β , the linearity is easy to be proved:

$$\mathcal{K}(\alpha g + \beta h) = \alpha \mathcal{K}g + \beta \mathcal{K}h. \quad (3)$$

It would seem natural to characterize the state space dynamics by the spectral properties of the Koopman operator.

The spectrum $\gamma(\mathcal{K})$ of the Koopman operator \mathcal{K} on an infinite-dimensional vector space is defined as

$$\mathcal{K}\varphi = \gamma\varphi, \quad (4)$$

where $\mathcal{K} - \gamma I$ is not invertible. In the infinite-dimensional Hilbert space, the Koopman operator \mathcal{K} can be denoted as

$$\mathcal{K}g = \sum_{i=1}^{\infty} \sigma_i \varphi_i \gamma_i, \quad (5)$$

where σ is called the Koopman mode associated with the Koopman eigenvalue-eigenfunction pair, and it is given by the projection of the observable g onto φ .

3.2. Convex Relaxation Constraint. The purpose of most DMD and deep learning approaches is to find eigenfunctions of the Koopman operator directly, satisfying

$$\varphi(x_{k+1}) = \mathcal{K}\varphi(x_k) = \lambda\varphi(x_k). \quad (6)$$

The eigenfunctions of Koopman operator can span an invariant linear subspace. In practice, the Koopman operator can be solved as an orthogonal Procrustes problem (OPP) in the following form:

$$\arg \min_{\mathcal{K}} \|\mathcal{K}\varphi(x_k) - \varphi(x_{k+1})\|_F \quad (7)$$

subject to $\mathcal{K}^T \mathcal{K} = I$.

Although the deep learning method can learn these eigenfunctions from data, the eigenfunctions are very sensitive to the noise in the training data. Strict orthogonal constraint would result in the noisy input to be merged into the representation of the hidden features $\varphi(x)$. Thus, the continuity and smoothness of the Koopman spectrum evolution may be destroyed.

Obviously, in Equation (7) the linear operator \mathcal{K} lies on the Stiefel manifold.

$$V_d(\mathbb{R}^n) := \left\{ \mathcal{K} \in \mathbb{R}^{n \times d} : \mathcal{K}^T \mathcal{K} = I_d \right\}. \quad (8)$$

\mathcal{K}^T denotes the transpose of \mathcal{K} , and I_d denotes the $d \times d$ identity matrix. From the above description, we can find that if \mathcal{K} is a linear operator, then the following propositions are equivalent: (1) \mathcal{K} is an isometry. (2) $\mathcal{K}^T \mathcal{K} = I_d$. (3) $\langle \mathcal{K}x, \mathcal{K}y \rangle = \langle x, y \rangle$. The above propositions can be proved by the following formula:

$$\langle \zeta_i, \mathcal{K}^T \mathcal{K} \zeta_j \rangle_{\mathcal{H}_\zeta} = \int_{\mathcal{H}_\zeta} (\zeta_i(x), \mathcal{K}^T \mathcal{K} \zeta_j(x)) dx = \langle \zeta_i, \zeta_j \rangle_{\mathcal{H}_\zeta}. \quad (9)$$

Each of the Stiefel manifolds $V_d(\mathbb{R}^n)$ can be viewed as a homogeneous space for the action of a classical group in a

natural manner. The orthogonal group $O(n)$ acts transitively on $V_d(\mathbb{R}^n)$. When $d = n$, the corresponding action is free so that the Stiefel manifold $V_d(\mathbb{R}^d)$ is a principal homogeneous space for the corresponding classical group.

In order to enhance the robustness of the model, in this work, we try to enforce the manifold constraint on \mathcal{K} by its convex counterpart. Specifically, we generalize it by seeking the closest Koopman eigenfunctions in which the columns may be orthogonal, but not necessarily orthonormal.

Lemma 1. *The convex hull of the Stiefel manifold $V_d(\mathbb{R}^n) := \{\mathcal{K} \in \mathbb{R}^{n \times d} : \mathcal{K}^T \mathcal{K} = I_d\}$ equals the unit spectral-norm ball $\text{conv}(V_d) = \{\mathcal{K} \in \mathbb{R}^{n \times d} \mid \|\mathcal{K}\|_2 \leq 1\}$. $\|\mathcal{K}\|_2$ denotes the spectral norm of a matrix \mathcal{K} , which is defined as the largest singular value of \mathcal{K} .*

Proof. Let $U\Sigma V^T$ be the singular value decomposition of \mathcal{K} . U is $n \times n$ orthonormal, V is $d \times d$ orthonormal, and Σ is $n \times d$ diagonal with values $\sigma_i(\mathcal{K})$ on the diagonal. Then

$$\begin{aligned} \max_{\mathcal{K} \in V_d} \langle \mathcal{K}, \mathcal{K} \rangle &= \max_{\mathcal{K} \in V_d} \langle U\Sigma V^T, \mathcal{K} \rangle \\ &= \max_{\mathcal{K} \in V_d} \text{Tr} \Sigma \langle V^T \mathcal{K}^T U \rangle \\ &= \max_{P \in V_d} \text{Tr} \Sigma P^T \\ &= \max_{P \in V_d} \sum_{i=1}^n \sigma_i(\mathcal{K}) p_{ii} \leq \sum_{i=1}^n \sigma_i(\mathcal{K}). \end{aligned} \quad (10)$$

□

The above lemma demonstrates that the manifold constraint $\{\mathcal{K} \in \mathbb{R}^{n \times d} : \mathcal{K}^T \mathcal{K} = I_d\}$ can be conveniently imposed by minimizing a convex regularizer.

We can use the above lemma directly to get the convex relaxation constraint term. In particular, the above conditions take the form

$$L_{\text{Stiefel}} = \left\| \|\mathcal{K}\|_2 - 1 \right\|_2^2. \quad (11)$$

By combining with an autoencoder, we propose a model capable of processing time-series data. Our model is trained by minimizing the following loss function:

$$L = \lambda_{re} L_{re} + \lambda_{pre} L_{pre} + \lambda_{\text{Stiefel}} L_{\text{Stiefel}}, \quad (12)$$

where λ_{re} , λ_{pre} , λ_{Stiefel} are user-defined hyperparameters that balance between reconstruction, prediction, and convex relaxation constraint. L_{re} , L_{pre} , and L_{Stiefel} are defined in Equations (11), (13), and (14) respectively.

$$L_{re} = \frac{1}{2} \sum_{k=1}^n \|x_k - \varphi^{-1}(\varphi(x_k))\|_2^2, \quad (13)$$

$$L_{pre} = \frac{1}{2} \sum_{k=1}^n \|\mathcal{K} \varphi(x_k) - \varphi(x_{k+1})\|_2^2. \quad (14)$$

4. Experiments

To evaluate our proposed approach (named Stiefel Koopman), we perform a comprehensive study using two datasets and compare to state-of-the-art Koopman-based approaches. The proposed Stiefel Koopman network minimizes Equation (12) with a decaying learning rate initially set to 0.01. We fix the loss weights to $\lambda_{re} = \lambda_{pre} = 1$ and $\lambda_{\text{Stiefel}} = 5 \times e^{-4}$, for the AE, prediction, and convex relaxation regularizing penalty, respectively. We use $ls = 8$ prediction steps forward in time.

4.1. Baselines. We compare our approach with two different Koopman-based baselines detailed next. For a fair comparison, unless said otherwise, we always use the same parameters in all methods.

4.1.1. Koopman AE [11]. The Koopman AE simulates the evolution process of the Koopman operator by using spectral decomposition in the AE training process. Many methods such as those in [13, 32] are its derivatives; without loss of generality, we will take it as a representative method for comparison. The Koopman AE also can be viewed as a special case of our method without manifold regularizing penalty.

4.1.2. Consistent Koopman AE [10]. Consistent Koopman AE leverages the forward and backward dynamics, and the consistency penalty is constructed by the relationship between the forward and reverse Koopman operators. The idea of this method is similar to the smooth filtering in Bayesian filtering and is also related to the consistent GAN. We will show that our method can degenerate to consistent Koopman AE and has lower computational complexity and higher accuracy.

4.2. Nonlinear Pendulum. A simple gravity pendulum without friction is an idealized mathematical model and a dynamic system which swings back and forth only under the influence of gravity. When a pendulum is displaced sideways from its equilibrium position, it is subject to a restoring force due to gravity that will accelerate it back toward the equilibrium position. The differential equation of an idealized pendulum can be represented as a second-order ODE as follows:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0, \quad (15)$$

where $g = 9.81$ is acceleration due to gravity, $l = 1$ is the length of the pendulum, and θ is the angular displacement. Following the same protocol as that in [10], the initial oscillation angles conditions $\theta_0 = 0.8$ and $\theta_0 = 2.4$ are considered in our experiment, and the time interval and sampling interval are set to $t = [0, 51]$ and $\Delta t = 0.03$, respectively. After obtaining the 2D sampling sequence, we use a random orthogonal matrix to map the samples to a high-dimensional space. This is the finite-dimensional approximation of the Koopman

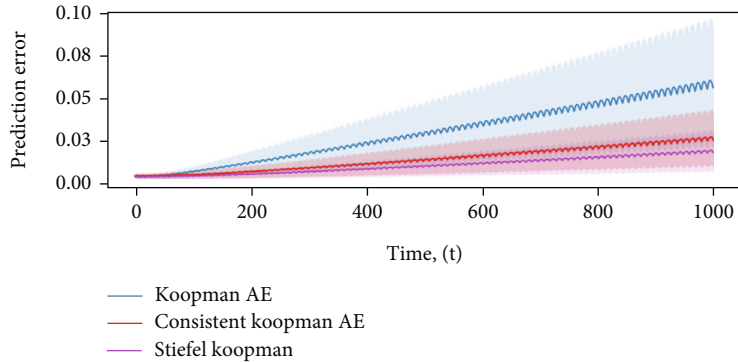
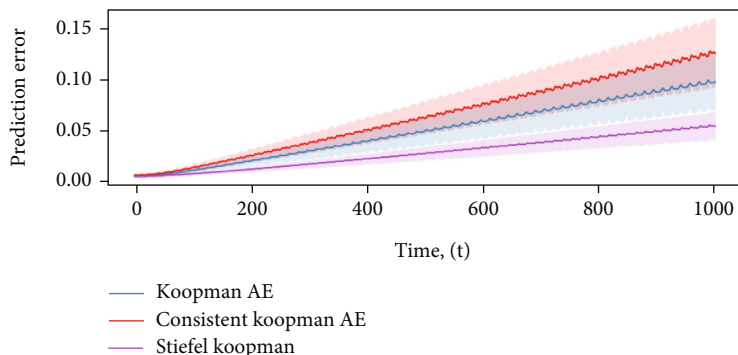
FIGURE 1: The comparison of our approach vs. SOTA methods by setting $\theta_0 = 0.8$.

TABLE 1: Average run time for 600 epochs training cases with different bottlenecks. In general, our model is almost two times faster than consistent Koopman AE during training and similar to Koopman AE.

Scenario	$bn = 6$	$bn = 8$	$bn = 10$
Koopman AE	61.18 s	62.38 s	63.49 s
Stiefel Koopman	61.34 s	62.41 s	63.34 s
Consistent Koopman AE	127.44 s	136.27 s	150.85 s

FIGURE 2: The comparison of our approach vs. SOTA methods by setting $\theta_0 = 2.4$.

operator in the infinite-dimensional Hilbert space. The training set contains 600 snapshots and 1600 testing snapshots.

As there are many network parameters, the source code will be available at <https://github.com/usodoney/Hci-groupgithub>. For detailed parameter settings of the model, please refer to the above link. For the comparative evaluation, the performance of the proposed method is tested against other two SOTA approaches listed above for clean pendulum sequences. Figure 1 shows the pendulum results for initial conditions $\theta_0 = 0.8$ and bottleneck = 6, which illustrates that our model can achieve the best results in all the cases we explored. The relative prediction error is computed at each time step via $(\|\varphi(x_k) - \mathcal{K}\varphi(x_{k-1})\|_2) / (\|\varphi(x_k)\|_2)$. $\mathcal{K}\varphi(x_{k-1})$ is the high-dimensional estimated prediction. We make predictions over a time frame of 1000 steps and average the error across 30 different initial observations $\varphi(x_k)$, where the shaded areas represent the standard deviations.

It is worth noting that, as shown in Table 1, we do not need to pay more computational costs to obtain the above

results in Figure 1. The running time of our model is equivalent to that of Koopman AE, less than half of that of consistent Koopman AE. As shown in Figure 2, our algorithm can still get the best results when the initial conditions are set to $\theta_0 = 2.4$.

In the theoretical part, we have mentioned that the convex relaxation term could improve the robustness of the model to noise. Therefore, we added two levels (0.05 and 0.1) of the Gaussian white noise to the input. The results are shown in Figures 3 and 4. Obviously, noise has varying degrees of influence on the prediction accuracy of all models. However, compared with other methods, it is not difficult to find that our method exhibits stronger robustness to noise and has a smaller standard deviation.

In addition, we can also add the backward prediction and consistency regular items in consistent Koopman AE to our model. The entire training process is consistent with the paper [10]. As shown in Figure 5, we find that additional constraints can further improve the prediction accuracy of

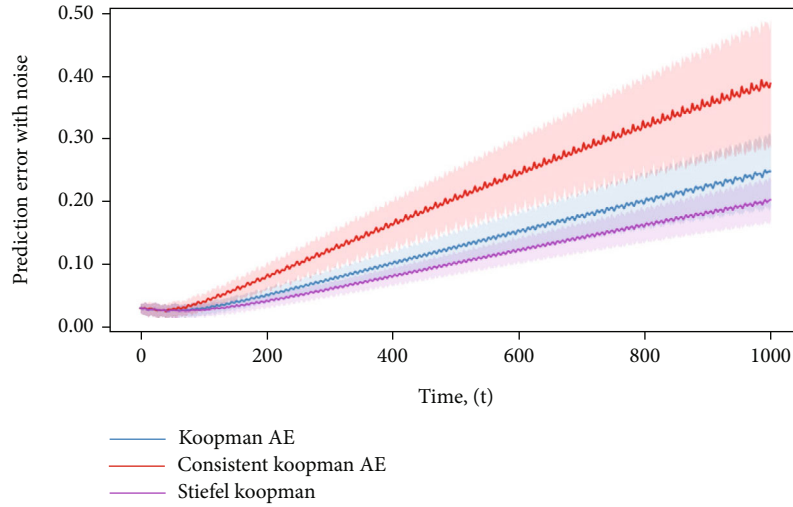


FIGURE 3: The comparison of our approach vs. SOTA methods on the pendulum's trajectories on noisy inputs. The noise level is set to 0.05.

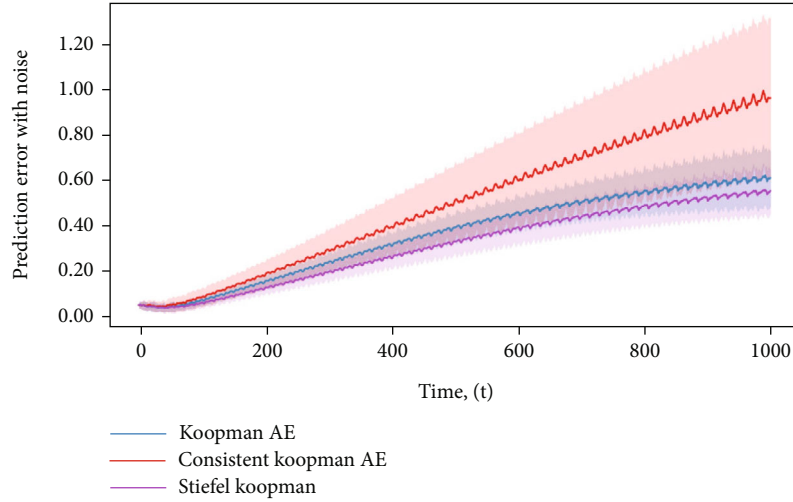


FIGURE 4: We compare the behavior of our approach vs. SOTA methods on the pendulum's trajectories on noisy inputs. The noise level is set to 0.1.

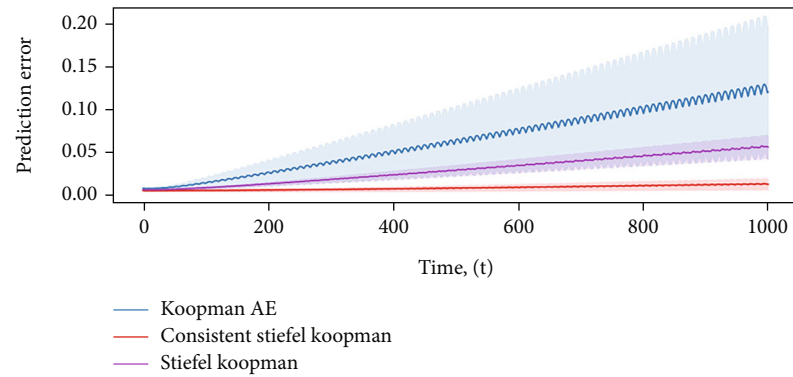


FIGURE 5: Prediction error after adding consistency constraint of consistent Koopman AE [10] to Stiefel Koopman.

the proposed method. But unfortunately, in this configuration, we need to pay nearly twice the cost of computing time. However, we believe that the proposed method obtains rel-

atively optimal results under the condition of limited computational complexity. Moreover, we have an interesting discovery in the experiment, and the proposed method

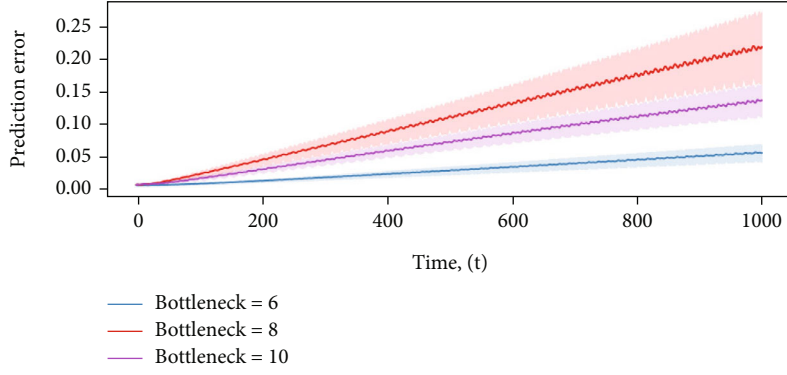
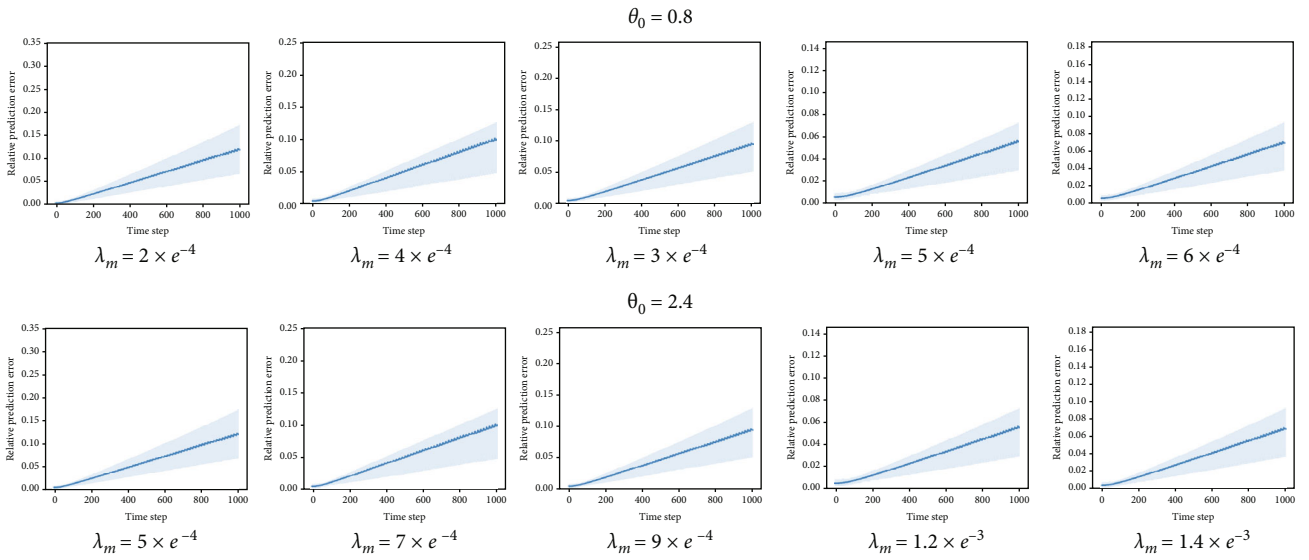


FIGURE 6: Prediction errors with different bottlenecks, over a time horizon of 1000 steps.

FIGURE 7: Effect of our manifold loss terms with different weights. Experiment results with oscillation angles $\theta_0 = 0.8$ (top row) and $\theta_0 = 2.4$ (bottom row).

can improve the robustness of Koopman AE by adding a convex relaxation term. However, the robustness of consistent Koopman AE cannot be enhanced by an additional convex relaxation term. This result also may suggest that the poor robustness of the algorithm is caused by the consistency constraint.

4.3. Ablation Study. As we all know, if we use a lot of nodes in the bottleneck layer when designing a neural network, high-dimensional coding will be generated. At this time, the network may overfit the input data by simply remembering the inputs. In this case, we will not be able to get the correct relationship in our coding. Similarly, if we use a bottleneck layer with a very small number of nodes, it will be difficult to capture all relationships. Therefore, we set the number of nodes in the bottleneck layer to 6, 8, and 10. As shown in Figure 6, our method can get the best result when the bottleneck is set to 6.

The weight hyperparameter of the convex relaxation term will also affect the performance of the model. If the

penalty hyperparameter is too large, our model will degenerate to the Koopman AE. If the penalty hyperparameter is too small, the convex relaxation term will be invalid. Therefore, we conduct hyperparameter selection experiments under different initial conditions. The experimental results are shown in Figure 7. When the initial condition is set to $\theta_0 = 0.8$, setting the penalty hyperparameter $\lambda_{Stiefel} = 5 \times 10^{-4}$ can obtain the best result, and when the initial condition is set to $\theta_0 = 2.4$, setting the penalty hyperparameter $\lambda_{Stiefel} = 1.2 \times 10^{-3}$ can obtain the best solution.

4.4. Cavity Flow. The lid-driven cavity flow is most probably one of the most studied fluid problems in the computational fluid dynamics field. Due to the simplicity of the cavity geometry, applying a numerical method to this flow problem in terms of coding is quite easy and straightforward. Despite its simple geometry, the driven cavity flow retains a rich fluid flow physics. The flow inside a driven cavity is governed by the 2D steady incompressible Navier–Stokes equations. These equations

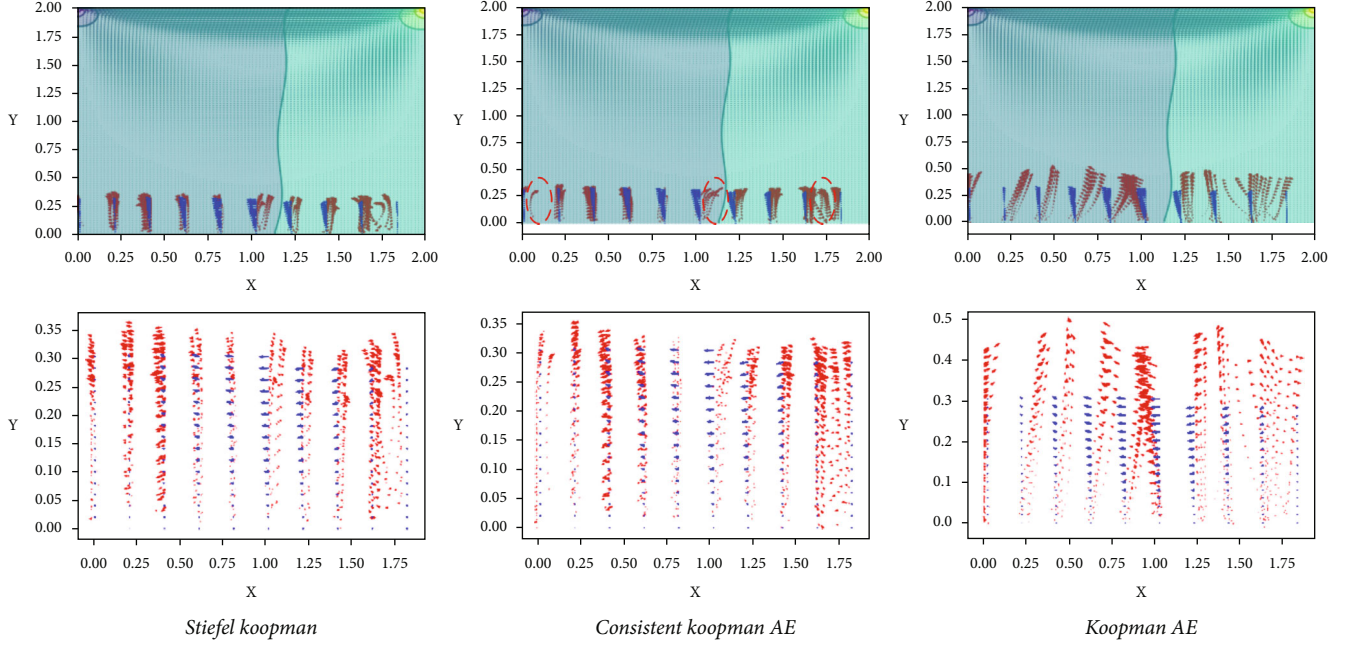


FIGURE 8: The predictions for the lid-driven cavity flow. We show the results of the predictions (top row) and its details (bottom row), where the blue arrow is ground truth.

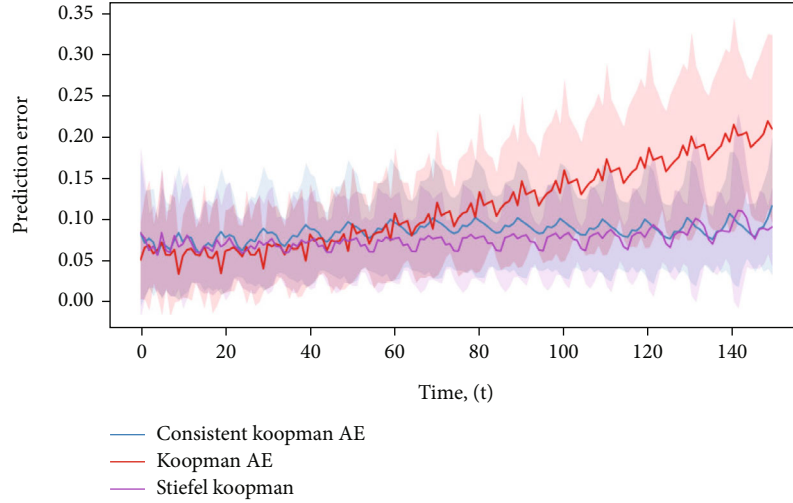


FIGURE 9: We compare the behavior of our approach vs. SOTA methods on the lid-driven cavity flow.

in stream function and vorticity formulation are given as follows [33]:

$$\begin{aligned} \nabla \cdot \vec{v} &= 0, \\ \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} &= -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{v}, \end{aligned} \quad (16)$$

where \vec{v} represents the velocity field. The first equation represents mass conservation at a constant density. The second equation is the conservation of momentum. In incompressible flow, the continuity equation $\nabla \cdot \vec{v} = 0$ provides a kinematic constraint that requires the pressure field to evolve so that the rate of expansion $\nabla \cdot \vec{v}$ should vanish everywhere. We use

the Poisson equation $((\partial^2 p)/(\partial x^2)) + ((\partial^2 p)/(\partial y^2)) = b$ to construct a pressure field that guarantees the continuity is satisfied; such a relation can be obtained by taking the divergence of the momentum equation. Finally, cavity flow with Navier-Stokes can be denoted by the following formulations:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \\ \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} &= -\rho \left(\frac{\partial u}{\partial x} \frac{\partial u}{\partial x} + 2 \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial v}{\partial y} \right), \end{aligned} \quad (17)$$

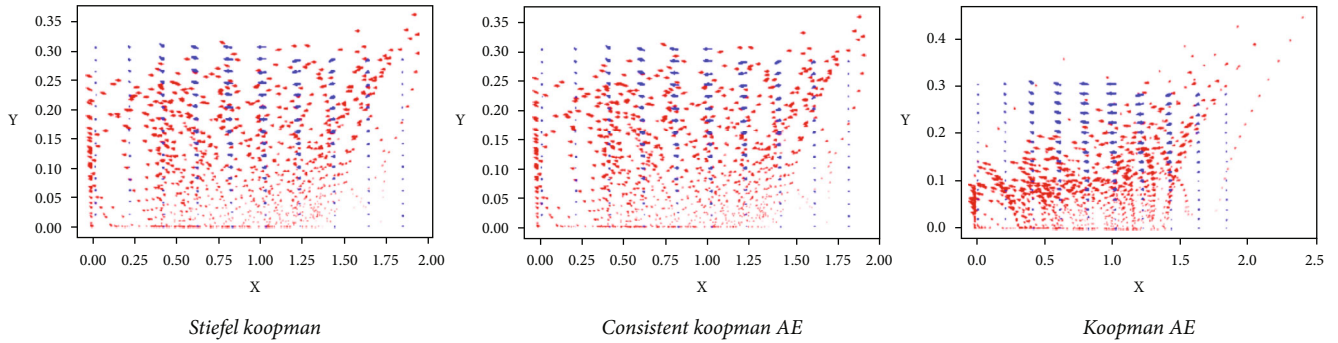


FIGURE 10: The comparison of our approach vs. SOTA methods on the lid-driven cavity flow with noisy inputs.

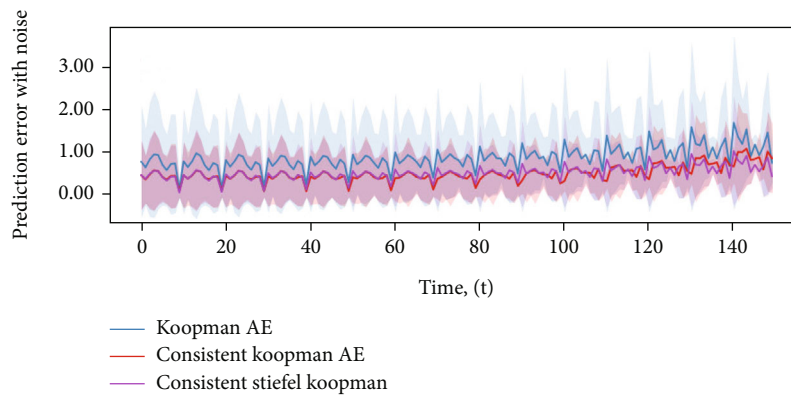


FIGURE 11: We compare the behavior of our approach vs. SOTA methods on the lid-driven cavity flow with noisy inputs.

where the terms $x, t, u(x, t)$, and v denote spatial coordinate, temporal coordinate, speed of fluid at the indicated spatial and temporal coordinates, and viscosity of fluid, respectively. The viscosity is a constant physical property of the fluid, and the other terms represent the dynamics contingent on that viscosity.

By setting initial condition $u, v, p = 0$ everywhere and following the boundary conditions, we can get 2000 samples which are uniformly sampled from 10,000 data by setting the model parameters ($nx = 100, ny = 100, nt = 500$, and $nit = 50$); half of the samples are used for training and half for testing.

Compared with the pendulum data, the cavity flow data has a higher dimensionality. Therefore, we set the bottleneck to 24 (the optimal empirical value). The qualitative prediction experiment results are shown in Figure 8. The blue arrow represents the ground truth, and the red is the predictions of different algorithms. We can find that our method is significantly better than Koopman AE. Compared with consistent Koopman AE, our results are also improved (we have marked the significant differences in red circles).

The corresponding prediction error curve on the lid-driven cavity flow is shown in Figure 9. It can be found that the prediction error of Koopman AE will increase greatly with the increase of time step.

As shown in Figures 10 and 11, when the input data contains noise, we analyze the prediction performance of our model and the competitive methods from the quantitative and qualitative perspectives, respectively. It is not difficult

to find that for complex nonlinear dynamic model models, noise has a very serious impact on the prediction results.

5. Conclusion

In this paper, we present a data-driven global linearization method for nonlinear dynamic systems. We found that the Koopman operator lies on the Stiefel manifold. The eigenvector of the Koopman operator directly calculated will cause the noise to be merged into the feature representation of the data. Therefore, we propose using the convex relaxation operator of the Stiefel manifold to constrain the eigenvectors of the Koopman operator. Compared to the recently proposed consistent Koopman AE, our algorithm performs better both on clean and noisy data, and the model complexity is much lower than consistent Koopman AE. In future work, we will study more complex nonlinear systems in the real world, such as human action sequences.

Data Availability

The source code will be available at <https://github.com/usodonkey/Hci-group>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62102105), the Guangdong Basic and Applied Basic Research Foundation (2020A1515110997, 2022A1515010138, and 2022A1515011501), the Science and Technology Program of Guangzhou (202002030263, 202102010419), and the Open Project Program of the State Key Lab of CAD&CG (Grant No. A2218), Zhejiang University.

References

- [1] H. U. Voss, J. Timmer, and J. Kurths, "Nonlinear dynamical system identification from uncertain and indirect measurements," *International Journal of Bifurcation and Chaos*, vol. 14, no. 6, pp. 1905–1933, 2004.
- [2] D. L. Debeljkovic, S. B. Stojanovic, N. S. Visnjic, and S. A. Milinkovic, "Singular time delayed system stability theory in the sense of lyapunov: A quite new approach," *American Control Conference, ACC*, pp. 4939–4944, 2007.
- [3] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [5] X. Liu, X. Wang, and S. Matwin, "Improving the interpretability of deep neural networks with knowledge distillation," *IEEE International Conference on Data Mining Workshops, ICDM Workshops*, pp. 905–912, 2018.
- [6] M. Raissi and G. E. Karniadakis, "Hidden physics models: machine learning of nonlinear partial differential equations," *Journal of Computational Physics*, vol. 357, pp. 125–141, 2018.
- [7] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian networks: using physics as model prior for deep learning," in *International Conference on Learning Representations, ICLR*, 2019.
- [8] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in Neural Information Processing Systems (NIPS)*, pp. 15 353–15 363, 2019.
- [9] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.
- [10] O. Azencot, N. B. Erichson, V. Lin, and M. W. Mahoney, "Forecasting sequential data using consistent koopman autoencoders," in *International Conference on Machine Learning*, vol. 119, pp. 475–485, ICML, 2020.
- [11] E. Yeung, S. Kundu, and N. O. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," *American Control Conference, ACC*, pp. 4832–4839, 2019.
- [12] L. Bethany, J. Nathan, L. S. Kutz, and Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, 2018.
- [13] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning koopman invariant subspaces for dynamic mode decomposition," *Advances in Neural Information Processing Systems, NIPS*, pp. 1130–1140, 2017.
- [14] J. Morton, A. Jameson, M. J. Kochenderfer, and F. D. Witherden, "Deep dynamical modeling and control of unsteady fluid flows," *Advances in Neural Information Processing Systems (NIPS)*, pp. 9278–9288, 2018.
- [15] E. Ling, L. Zheng, L. J. Ratliff, and S. Coogan, "Koopman operator applications in signalized traffic systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3214–3225, 2022.
- [16] T. Pumir, A. Singer, and N. Boumal, "The generalized orthogonal procrustes problem in the high noise regime," *Information and Inference: A Journal of the IMA*, vol. 10, no. 3, pp. 921–954, 2021.
- [17] D. Simon, *Optimal State Estimation: Kalman, H ∞ , and Non-linear Approaches*, Wiley, 2006.
- [18] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*, Academic Press, 1974.
- [19] H. E. Emara-Shabaik and C. T. Leondes, "A note on the extended kalman filter," *Automatica*, vol. 17, no. 2, pp. 411–412, 1981.
- [20] Y. Liu, F. Li, and P. Dang, "Bifurcation analysis in a class of piecewise nonlinear systems with a nonsmooth heteroclinic loop," *International Journal of Bifurcation and Chaos*, vol. 28, no. 2, p. 1850026, 2018.
- [21] D. S. Graça, N. Zhong, and H. S. Dumas, "The connection between computability of a nonlinear problem and its linearization: the Hartman–Grobman theorem revisited," *Theoretical Computer Science*, vol. 457, pp. 101–110, 2012.
- [22] I. Mezic, "On applications of the spectral theory of the Koopman operator in dynamical systems and control theory," in *IEEE Conference on Decision and Control*, pp. 7034–7041, CDC, 2015.
- [23] A. Surana, "Koopman operator based nonlinear dynamic textures," *American Control Conference, ACC*, pp. 1333–1338, 2015.
- [24] I. J. Goodfellow, Y. Bengio, and A. C. Courville, "Deep learning, ser," *Adaptive computation and machine learning*, 2016.
- [25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *International Conference on Machine Learning, ICML*, vol. 37, pp. 2067–2075, 2015.
- [26] A. Hervé, D. Sipp, P. J. Schmid, and M. Samuelides, "A physics-based approach to flow control using system identification," *Journal of Fluid Mechanics*, vol. 702, no. 1, pp. 26–58, 2012.
- [27] P. J. Schmid and J. Sesterhenn, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, no. 10, pp. 5–28, 2010.
- [28] H. Arbabi, M. Korda, and I. Mezic, "A data-driven koopman model predictive control framework for nonlinear flows," 2018, <https://arxiv.org/abs/1804.05291>.
- [29] H. Arbabi and I. Mezic, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 4, pp. 2096–2126, 2017.
- [30] M. Korda and I. Mezic, "On convergence of extended dynamic mode decomposition to the koopman operator," *Journal of Nonlinear Science*, vol. 28, no. 2, pp. 687–710, 2018.
- [31] S. E. Otto and C. W. Rowley, "Linearly recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 558–593, 2019.

- [32] Y. Li, H. He, J. Wu, D. Katabi, and A. Torralba, "Learning compositional koopman operators for model-based control," in *International Conference on Learning Representations*, ICLR, 2020.
- [33] L. Barba and G. Forsyth, "Cfd python: the 12 steps to Navier-Stokes equations," *Journal of Open Source Education*, vol. 1, no. 9, p. 21, 2018.