Online Estimation of the Koopman Operator Using Fourier Features

Tahiya Salam TSALAM@SEAS.UPENN.EDU
Alice Kate Li ALICEKL@SEAS.UPENN.EDU
M. Ani Hsieh MYA@SEAS.UPENN.EDU
GRASP Lab, University of Pennsylvania, Philadelphia, PA

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

Transfer operators offer linear representations and global, physically meaningful features of nonlinear dynamical systems. Discovering transfer operators, such as the Koopman operator, require careful crafted dictionaries of observables, acting on states of the dynamical system. This is ad hoc and requires the full dataset for evaluation. In this paper, we offer an optimization scheme to allow joint learning of the observables and Koopman operator with online data. Our results show we are able to reconstruct the evolution and represent the global features of complex dynamical systems.

Keywords: Koopman Operators, Learning Nonlinear Dynamics, Nonlinear Identification

1. Introduction

In dynamical systems theory, a fundamental goal is data-driven state prediction and feature representation. Given collected sensor measurements over time, we often want to characterize the current nonlinear system states and predict the future states. Furthermore, for applications such as robotics, sensing data is collected in real-time and contains noise. In this work, we will focus on online methods for extracting global dynamics of the underlying system with the Koopman operator.

Many state estimation techniques for systems whose dynamics are governed by differential equations rely on extracting dynamics from data. A powerful lens for estimating system dynamics and their representations is by characterizing the global dynamics of an ensemble of sensor observations or measurements (Mezić, 2005; Klus et al., 2020). A useful mathematical tool for understanding the time evolution of states and their global dynamics is transfer operator theory, through which we can study the action of a dynamical system on mass densities of initial conditions. The transfer operator is defined on some infinite dimensional linear functional space and describes a linear time evolution of the transformed densities. This is desirable as the original, nonlinear dynamics are lifted to a space such that the transformed densities evolve linearly. For example, the global feature tracking of trajectories in flow-like environments, analyzing pair-wise relationships in text documents, or observation of conformational changes in metastable walking may be suitable for study under transfer operator theory (Salam et al., 2022; Klus et al., 2020; Costa et al., 2021). There are many methods for extracting the Koopman operator, a transfer operator in which the lifted densities are referred to as observables, from data (Schmid, 2010; Li et al., 2017; Klus et al., 2020). Many of these data-driven Koopman construction techniques have been used with real sensing data (Korda and Mezić, 2018; Abraham and Murphey, 2019; Folkestad et al., 2020a; Bruder et al., 2019; Salam et al., 2022). While these works are promising in their application of the Koopman operator to different domains, they are limited in that they derive the operator as a closed-form, rigid solution and require data across the entire time horizon.

This paper presents two fundamental advances in the estimation of the Koopman operator. First, we devise a way of jointly learning the observables and the Koopman operator from data collected by robots in a reproducing kernel Hilbert space using random Fourier features. Second, the joint learning is formulated such that construction of the observables and Koopman operator can be done completely online. We test proposed framework on dynamical systems with varying degrees of complexity, and show that it is desirable for the representation of complex dynamical systems; future state prediction; and computing physically meaningful eigenfunctions.

2. Related Works

We review the literature related to the construction of the Koopman operator and emphasize some shortcomings of the existing methods.

Closed-form solutions. One of the most common data-driven methods for estimating the Koopman operator relies on Dynamic Mode Decomposition (DMD) (Schmid, 2010). DMD is a spectral decomposition of the Koopman operator, for which the dynamics are assumed to be linear (Mezić, 2005; Rowley et al., 2009). Extended Dynamic Mode Decomposition (EDMD) is a nonlinear generalization of DMD, where dynamics are assumed to be described using a nonlinear invertible transformation characterized by a dictionary acting on input data (Williams et al., 2015a). In this work, instead of the Koopman operator acting directly on observables (in this case, the data from the system), the operator is acting on the dictionary applied to observables which results in an expanded set of observables. In EDMD, the user determines the choice of dictionary, tailored to the dataset.

Recent works have explored the connections between the Koopman operator and reproducing kernel Hilbert spaces (RKHS) (Williams et al., 2015b; Klus et al., 2020; Kawahara, 2016; Das and Giannakis, 2018). Representations of the Koopman operator in RKHS allows for the analysis of the operator in any domain where there is a similarity measure given by a kernel. As the Koopman operator construction relies on an inner product computation, these methods allows for the computation of inner products implicitly through the use of a defined kernel function. Another advantage to these algorithms is that they give an approximation of the Koopman operator with a set of nonlinear basis functions due to the expressiveness of kernel functions.

Dictionary learning approaches. A more flexible approach formulates a trainable dictionary represented by an artificial neural network (NN) within the EDMD framework (Li et al., 2017; Yeung et al., 2019; Mardt et al., 2020). Rather than the user defining a dictionary of observables, a dictionary can be learned using data from the system of interest. In many applications, it is not necessary to explicitly construct the Koopman operator but instead construct its eigenfunctions, a useful decomposition for understanding the stability of the process of interest (Mauroy and Mezić, 2016). There are many techniques focused on data-driven learning of Koopman eigenfunctions (Korda and Mezić, 2018; Folkestad et al., 2020a,b; Haseli and Cortés, 2021; Kaiser et al., 2021; Sznaier, 2021; Shi and Karydis, 2021; Leask et al., 2021).

In this work, we leverage insights from dictionary learning to automate the procedure for learning the transfer operators online. Here, we focus on learning the kernel transfer operators. Instead of black-box optimization to construct the dictionary of observables, we apply insights from the kernel methods literature to construct the dictionary. This method provides structure and meaning to the dictionary, but it is still an automated procedure similar to NN methods.

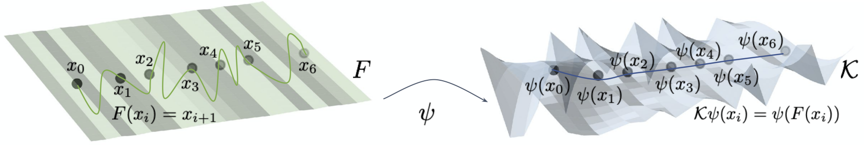


Figure 1: The state $\mathbf{x}_i$ evolves nonlinearly according to a dynamical system function F. Using the map ψ, the states can be lifted to an alternative space, where the lifted mappings of the states $\psi(\mathbf{x}_i)$ evolves linearly according to the Koopman operator $\mathcal{K}$.

3. Preliminaries

Before proposing our strategy for learning the Koopman operator with Fourier features, we introduce the definition of the Koopman operator and connections between kernels and Fourier features.

3.1. Koopman Operator

The basic form of a discrete-time dynamical system in state-space form is

$$\mathbf{x}(t+1) = f(\mathbf{x}(t)), \tag{1}$$

where the function f maps the state space in $\mathbb{R}^d$ to itself. Given a class of scalar, complex valued functions $\mathcal{F}$, an observable is any $\psi(\cdot) : \mathbb{R}^d \to \mathbb{C}$, where ψ maps the state of the system into a scalar. The value of the observable is determined by the state of the system and its evolution over time is described by the composition of the observable with the state dynamics $f(\cdot)$ as

$$\psi(\mathbf{x}(t+1)) = \psi(f(\mathbf{x}(t))). \tag{2}$$

The Koopman operator, $\mathcal{K}$, is applied to the observable and describe the evolution of observables, as in Eq. (2), under the state evolution. As such, the evolution of observable ψ can be written as $(\mathcal{K}\psi)(\mathbf{x}) = \psi(f(\mathbf{x}))$. The relationship between the dynamical systems representation and transfer operator can be shown in Fig. 1. The observables can be thought of as a lifting operation that maps state dynamics into a different space, where observables can be propagated forward in time by $\mathcal{K}$.

3.2. Kernels

Kernel functions $k : X \times X \to \mathbb{R}$ are applied to elements of some space, X, to measure the similarity between any pairs of elements. Let ϕ be the feature map associated with kernel k defined on X. Then, define feature matrices as

$$\Phi = [\phi(x_1) \ldots \phi(x_n)]. \tag{3}$$

The similarity metric in the form of a kernel function can be found by computing the inner product of features $\phi(x)$ acting on elements of the input space in some high-dimensional, possibly infinite, feature space in $\mathbb{R}^M$. Thus, kernels allow us to compare objects based on their features. Formally, for inputs $x, x' \in X$, a feature map $\phi : X \to \mathbb{R}^M$, and some valid inner product $\langle \cdot,\cdot \rangle_{\mathcal{V}}$, a kernel k is defined as $k(x, x') = \langle\, \phi(x), \phi(x') \rangle_{\mathcal{V}}$, where $\mathcal{V}$ is an inner product space. Kernels on features spaces that are positive definite can be used to define a function f on X. The space of such functions f, such that the evaluation of f at x can be represented as the inner product $\langle\, \phi(x), \phi(x') \rangle$ in feature space, is referred to as the Reproducing Kernel Hilbert Space (Gretton, 2013).

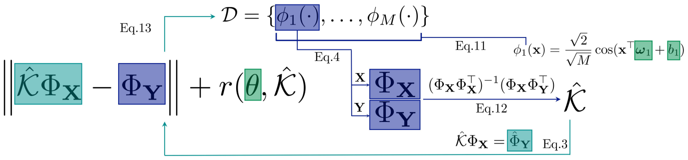

Eq.13
$\mathcal{D} = \{\phi_1(\cdot), \ldots, \phi_M(\cdot)\}$
Eq.11
$\phi_1(\mathbf{x}) = \frac{\sqrt{2}}{\sqrt{M}} \cos(\mathbf{x}^\top \boldsymbol{\omega}_1 + b_1)$
Eq.4
$\left\|\hat{\mathcal{K}}\Phi_{\mathbf{X}} - \Phi_{\mathbf{Y}}\right\| + r(\theta, \hat{\mathcal{K}})$
X
Y
$\Phi_{\mathbf{X}}$
$\Phi_{\mathbf{Y}}$
$(\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^\top)^{-1}(\Phi_{\mathbf{X}}\Phi_{\mathbf{Y}}^\top)$
Eq.12
$\hat{\mathcal{K}}$
$\hat{\mathcal{K}}\Phi_{\mathbf{X}} = \hat{\Phi}_{\mathbf{Y}}$
Eq.3

Figure 2: Overview of optimization framework for joint learning of dictionary of observables and Koopman operator. Random Fourier Features ϕ are initialized with weights ω, b and used to create dictionaries of observables $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ on the training data $\{\mathbf{X}, \mathbf{Y}\}$. The Koopman operator is constructed using $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ and estimates $\hat{\Phi}_{\mathbf{Y}}$. The loss is computed and the weights are updated in the dictionary. Dark blue values are calculated using the current weights in green, while light blue are the Koopman computed estimates.

3.3. Random Fourier Features

Random Fourier Features (RFFs) are spectral domain representations of kernels (Rahimi and Recht, 2009). These techniques rely on representing the stationary covariance function as the Fourier transform of a positive finite measure, formalized by Bochner's Theorem and its corollary stated below for the sake of completeness.

Theorem 1 *(Bochner's Theorem) (Salomon Bochner, 1932) Every positive definite function* $\hat{\mu} : \mathbb{R}^D \to \mathbb{C}$ *for all* $\mathbf{x} \in \mathbb{R}^D$ *is the Fourier transform of a non-negative finite Borel measure* μ *on* $\mathbb{R}^D$*. That is, for any* $\hat{\mu}(\mathbf{x})$ *there exists a measure* μ *such that* $\hat{\mu}(\mathbf{x}) = \int_{\mathbb{R}^D} e^{-i\mathbf{x}^\top \boldsymbol{\omega}} d\mu(\boldsymbol{\omega})$.

Corollary 2 *If* $\hat{\mu}(0) = 1$ *and the measure* μ *is a probability measure with probability density function (pdf)* f_Ω *on random variable* Ω *with realizations* $\boldsymbol{\omega} \in \mathbb{R}^D$*, then* $\hat{\mu}(\mathbf{x} - \mathbf{x}') =: k(\mathbf{x}, \mathbf{x}')$ *is a continuous stationary positive-definite covariance function* $k(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{R}^D} e^{-i(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\omega}} f_\Omega(\boldsymbol{\omega}) d\boldsymbol{\omega}$.

While the dot product can be easily computed as the kernel evaluation, this means that the data can only be accessed through the evaluation of the kernel function or through a matrix containing all of the evaluations across all pairs of data points. However, this might be computationally expensive for large datasets. To alleviate this, Rahimi and Recht (2009) proposed explicitly mapping the data to a lower-dimensional Euclidean space using a randomized feature map $\hat{\phi}$. A known pdf can be represented using a finite number of Monte Carlo (MC) samples, where $\{\boldsymbol{\omega}_m\}_{m=1}^M \overset{i.i.d}{\sim} f_\Omega(\boldsymbol{\omega})$ lends to a finite dimensional approximation of the feature map $\hat{\phi}(\mathbf{x}) \in \mathbb{C}^M$ that is

$$k(\mathbf{x}, \mathbf{x}') \approx \frac{1}{M} \sum_{m=1}^{M} e^{-i(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\omega}_m} = \langle \hat{\phi}(\mathbf{x}), \hat{\phi}(\mathbf{x}') \rangle_{\mathbb{C}}. \tag{4}$$

Then, the feature map can be decomposed such that $\hat{\phi}(\mathbf{x}) = 1/\sqrt{M}[e^{-i\mathbf{x}^\top \boldsymbol{\omega}_1}, \ldots, e^{-i\mathbf{x}^\top \boldsymbol{\omega}_M}] \in \mathbb{C}$. Since the kernels we will use are real valued, we can re-write the feature map to be $\hat{\phi}(\mathbf{x}) \in \mathbb{R}^M$

$$\hat{\phi}(\mathbf{x}) = \frac{\sqrt{2}}{\sqrt{M}}[\cos(\mathbf{x}^\top \boldsymbol{\omega}_1 + b_1), \ldots, \cos(\mathbf{x}^\top \boldsymbol{\omega}_M + b_M)], \quad b_i \sim \text{Uniform}(0, 2\pi). \tag{5}$$

4. Methodology

In this section, we present our main algorithmic contributions: the online joint learning of Koopman operator and observable dictionary. We first present a construction of the Koopman operator using kernels (Klus et al., 2020) and demonstrate how we can bypass the need to explicitly specify a kernel by leveraging RFFs (Rahimi and Recht, 2009). In doing so, we can jointly learn the underlying Koopman operator and observable dictionary within an RKHS framework. Using RFFs as the feature map on large datasets is more efficient than existing kernel EDMD methods, since RFFs involve mapping to a lower-dimensional space, instead of a higher dimension that kernels require. RFFs also provide the necessary kernel structure, with $(d+1)$ learnable parameters per RFF, and so our framework should require less training and tuning than an NN. Furthermore, new observations can sequentially be appended to training data, lending to an online learning schema.

Let us re-write Eq. (1) as $\mathbf{y}_i = f(\mathbf{x}_i)$. Suppose we have N pairs of training data $(\mathbf{x}_i, \mathbf{y}_i)$ collected over the system of interest, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. We herein refer to a data sample as a particle. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N]^\top$ be a matrix of the N particles at a specific time instance and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_N]^\top$ be the position of the particles after time has elapsed. $\mathbf{Y}$ is then the time shifted version of $\mathbf{X}$. If the positions of the N pairs of training data are tracked over time, we can index the N particles at time t by $\mathbf{X}(t)$ and the time shifted particles as $\mathbf{Y}(t)$. The training data can then be summarized as $\{(\mathbf{X}(t), \mathbf{Y}(t))\}_{t=1}^{T}$, where these can be thought of as the T-length trajectories of the particles.

Define feature maps $\Phi_\mathbf{X} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_N)]^\top$ and $\Phi_\mathbf{Y} = [\phi(\mathbf{y}_1), \phi(\mathbf{y}_2), \ldots, \phi(\mathbf{y}_N)]^\top$ on the particles using the feature maps in Eq. (3). The feature map $\Phi_\mathbf{Y}$ can be thought of as a time shifted version of $\Phi_\mathbf{X}$. We can also construct the feature maps $\Phi_{\mathbf{X}(t)}$ and $\Phi_{\mathbf{Y}(t)}$ using the N particles from a specific time instance. From (Klus et al., 2020), the Koopman operator associated with the dynamical system of interest can be estimated using the relationship

$$\hat{\mathcal{K}} = (\Phi_\mathbf{X}\Phi_\mathbf{X}^\top)^{-1}(\Phi_\mathbf{X}\Phi_\mathbf{Y}). \tag{6}$$

This estimation of the Koopman operator leverages the kernel functional evaluation instead of the explicit feature map computation (Klus et al., 2020; Williams et al., 2015b). These approaches assume that the dependence in the training data can be appropriately described by a kernel function, a computationally efficient operation. In the standard formulation of the RFFs, the distribution $f_\Omega(\boldsymbol{\omega})$ is known and corresponds to specific kernel forms. For example, sampling from a Gaussian distribution corresponds to the RBF kernel.

In this work, instead of assuming a known form of the kernel function $k(\mathbf{x}, \mathbf{x}')$, such as the RBF, we instead approximate the kernel function using the RFFs, as in Eq. 3, described in Section 3.3. Thus, instead of assuming some known form of the dependence within the data, we jointly the learn the structure of the feature maps and the corresponding Koopman operator. Let θ be the collection of weights $\boldsymbol{\omega}_i$ and b_i associated with the RFFs that we would like to learn in order to determine the structure of the underlying feature maps that best describe the data. Previous work by Li et al. (2017) and Yeung et al. (2019) show this is achieved through the following minimization scheme

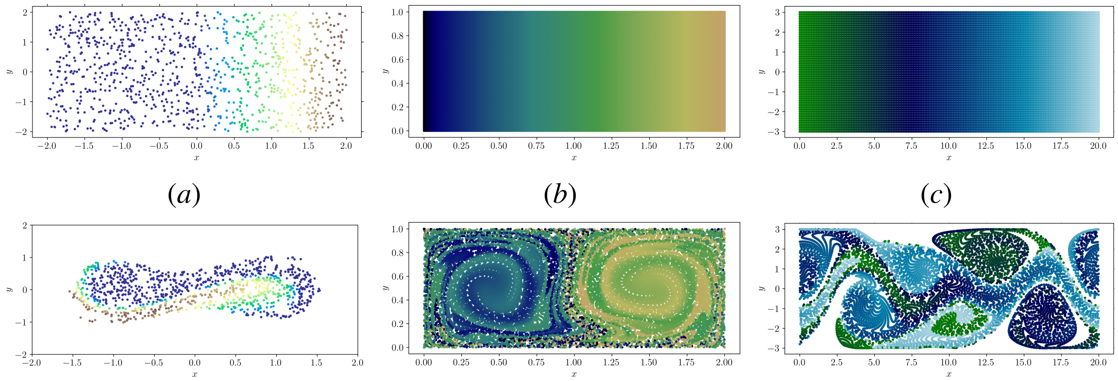
$$\min_{\hat{\mathcal{K}},\theta} \sum_{t=1}^{T} \left\| \Phi_{\mathbf{Y}(t)} - \hat{\mathcal{K}}\Phi_{\mathbf{X}(t)} \right\|_2 + \lambda_1 \left\| \hat{\mathcal{K}} \right\|_2 + \lambda_2 \left\| \theta \right\|_1 . \tag{7}$$

We learn the dictionary of observables, where observables are expressed through the feature maps. This connection between dictionaries of observables and features maps was made when kernel EDMD was first proposed (Williams et al., 2015b). In the work by Li et al. (2017), the dictionary of observables is unknown and an NN is used to estimate them. We use the kernel expression of the Koopman operator and leverage RFFs in capturing the dictionary of observables. The learned

RFF feature maps and kernel Koopman operator are learnt via minimizing Eq. (7) via stochastic gradient descent and enforcing the kernel Koopman operator structure using Eq. (6). We note that any optimization scheme can be used. The loss schematic is summarized in Fig. 2.

5. Experimental Results

We describe the details for simulating benchmark prototypical dynamical systems with varying degrees of complexity used to study transfer operators, our quantitative results evaluating the reconstruction of trajectories and eigenfunction approximation, and qualitative results depicting physically meaningful global features within these systems.



(*a*) (*b*) (*c*)

(*d*) (*e*) (*f*)

Figure 3: Prototypical dynamical systems for studying transfer operators. (*a*) 1000 particles, colored by their initial x-coordinate to visualize movement, shown in a $[-2, 2] \times [-2, 2]$ grid. (*d*) Positions of particles are shown at time $t = 2.5$ under the Duffing oscillator, which describes the dynamics of a point mass in a double well potential. (*b*) 20000 particles are initialized in a $[0, 2] \times [0, 1]$ grid. (*e*) Particles' positions are tracked in a time-dependent double gyre flow at time $t = 12.1$. Two dynamically distinct regions form in this flow, a representative feature of the system. (*c*) 9900 particles are initialized in a $[0, 20] \times [-3, 3]$ grid. (*e*) Movement of the particles at $t = 10.1$ under a Bickley jet model, an idealized model for fluid flows.

5.1. Simulated Dynamical Systems

In this subsection, we describe three prototypical dynamical systems used in the study of transfer operators, the Duffing oscillator, the double gyre, and Bickley jet. In the subsequent subsections, we demonstrate the use of the proposed algorithms for the kernel Koopman operator in Section 4.

5.1.1. DUFFING OSCILLATOR

The Duffing oscillator is a nonlinear second order differential equation used to model damped and driven oscillators. The evolution of the particles whose positions are described by (x, y) is governed by the $\dot{x} = y$ and $\dot{y} = -\delta y - x(\beta + \alpha x^2)$. We use the parameters $\delta = 0.5$, $\beta = -1$, and $\alpha = 1$. We simulate 1000 points uniformly sampled on grid $[-2, 2] \times [0, 1]$, shown in Fig. 3(*a*). From time $t \in [0, 2.75]$ with step size 0.25, we use a differential equations solver based on an explicit Runge-Kutta (4,5) formula on the velocities to solve an initial value problem for the system of ordinary differential equations with the sampled points. The visualization of the movement of particles in a Duffing oscillator is shown in Fig. 3(*d*).

5.1.2. DOUBLE GYRE MODEL

A simple model of the wind-driven, time-dependent double gyre flow is described by $f(x,t) = \epsilon * \sin(\omega t) * x^2 + (1 - 2\epsilon \sin(\omega t)) * x$, $\frac{\partial f}{\partial x} = 2\alpha * \sin(\omega t) * x + (1 - 2\alpha \sin(\omega t))$, $\dot{x} = -\pi A \sin(\pi f(x,t)) \cos(\pi y)$, $\dot{y} = \pi A \cos(\pi f(x,t)) \sin(\pi y) * \frac{\partial f}{\partial x}$.

The parameters are set to $\epsilon = 0.25, \alpha = 0.25, A = 0.25$, and $\omega = 2\pi$ (Forgoston et al., 2011). We simulate 20000 points, as in Fig. 3(*b*), uniformly sampled on grid $[0,2] \times [0,1]$. From time $t \in [0,20]$ with step size 0.1, we use a differential equations solver based on an explicit Runge-Kutta (4,5) formula on the velocities to calculate the trajectories of the points, shown in Fig. 3(*e*).

5.1.3. BICKLEY JET

The Bickley jet model is a prototypical model in the study of coherence that is a meandering zonal jet, flanked both above and below by counter rotating vertices. The Bickley jet is used as an idealized model for the Gulf Stream in the ocean and polar night jets in the atmosphere (Del-Castillo-Negrete and Morrison, 1992; Beron-Vera et al., 2010). The stream function for the Bickley jet model is $\psi(x,y,t) = \psi_0(y) + \psi_1(x,y,t)$, $\psi_0(y) = -U_0 L_0 \tanh\left(\frac{y}{L_0}\right)$, $\psi_1(x,y,t) = U_0 L_0 \operatorname{sech}^2\left(\frac{y}{L_0}\right) \Re\left(\sum_{n=1}^{3} f_n(t) \exp(ik_n x)\right)$, with $f_n(t) = \epsilon_n \exp(-ik_n c_n t)$. The velocities can be computed as $\dot{x} = \partial\psi/\partial x$ and $\dot{y} = \partial\psi/\partial y$. We use scaled parameters $U_0 = 5.4138, L_0 = 1.77, c_1 = 0.1446U_0, c_2 = 0.2053U_0, c_3 = 0.4561U_0, \epsilon_1 = 0.075, \epsilon_2 = 0.4, \epsilon_3 = 0.3, r_0 = 6.371, k_1 = 2/r_0, k_2 = 4/r_0, k_3 = 6/r_0$ (Hadjighasem et al., 2017). We sample 9900 points uniformly on a grid $[0,20] \times [-3,3]$, seen in Fig. 3(*c*). From time $t \in [0,40]$ with step size 0.1, we calculate the trajectories of the points using a variable-step, variable-order Adams-Bashforth-Moulton solver of orders 1 to 13 on the velocity $[\dot{x}, \dot{y}]$, as shown in Fig. 3(*f*).

5.2. Evaluation Metrics

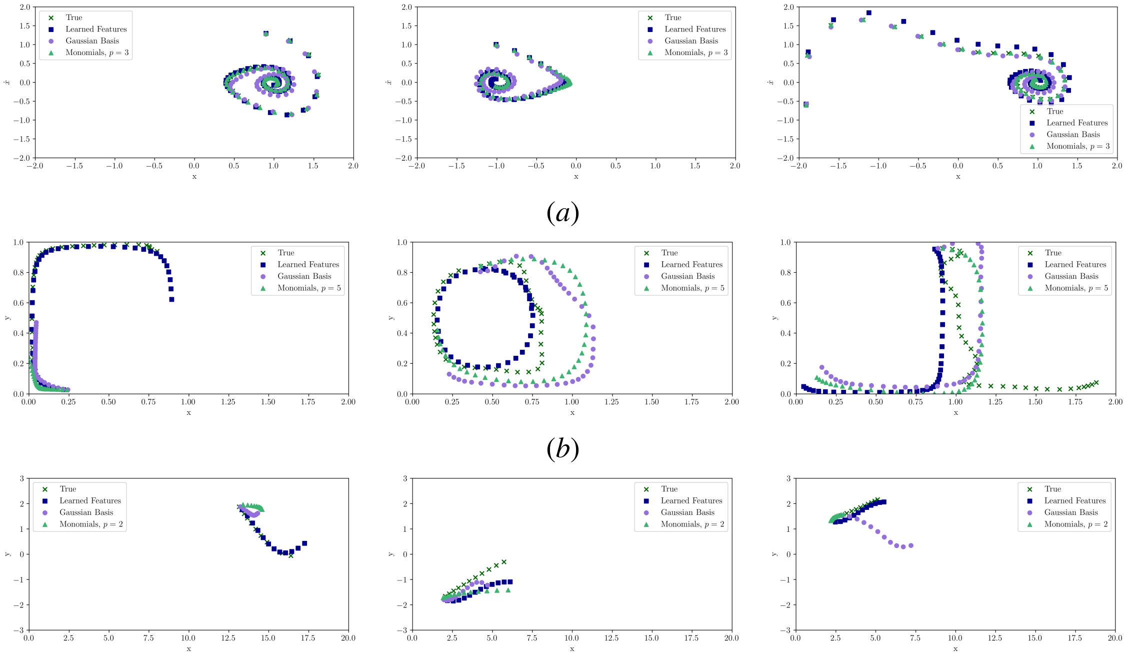
In this section, we discuss two evaluation metrics used to compare the kernel Koopman operator with RFFs against EDMD algorithms with dictionaries selected based on the known properties of the simulated environments. We first evaluate the reconstruction error associated with the known trajectory $\mathbf{x}_i$ of a particle i over multiple time steps. We reconstruct trajectories of the system using the Koopman mode decomposition formula. This is described in the presentation of the EDMD algorithm and other works related to dictionary learning (Williams et al., 2015a; Li et al., 2017). We summarize the procedure here for completeness.

First, we solve for a matrix $\mathbf{B} \in \mathbb{R}^{M \times N}$ such that $\mathbf{x} = (\Phi_{\mathbf{X}} \mathbf{B})^\top$. Let $\mathbf{V}$ be the matrix containing the set of right eigenvectors, $\mathbf{W}^*$ be the matrix contained the set of left eigenvectors associated with $\mathcal{K}$, and $\boldsymbol{\mu}$ be the vector of eigenvalues. Each of the i left eigenvectors of should be scaled such that $\mathbf{w}_i^* \mathbf{v}_i = 1$. The approximate eigenfunction corresponding to the eigenvalue μ_j of $\mathcal{K}$ is then $\psi_j = \mathbf{v}_j^\top \Phi$. Then, the full state can be reconstructed as $\hat{\mathbf{x}} = (\mathbf{W}^* \mathbf{B})^\top (\Phi_{\mathbf{X}} \mathbf{V})^\top$. To estimate the evolution of the trajectory after a time t has elapsed from the positions of the particles at $\mathbf{x}(0)$, we leverage the linear representation of the Koopman decomposition to construct the trajectory as

$$\hat{\mathbf{x}}(t) = \boldsymbol{\mu}^t (\mathbf{W}^* \mathbf{B})^\top (\Phi_{\mathbf{X}_0} \mathbf{V})^\top. \quad (8)$$

Table 1: Comparison of Error Reconstruction with Different Dictionaries for Various Environments

System		Duffing Oscillator		Double Gyre		Bickley Jet	
Prediction Horizon		NT	LT	NT	LT	NT	LT
Dict.	Learned	0.5929	0.3713	0.2307	0.7125	2.7584	13.856
	Gaussian	0.4668	0.3427	0.3610	1.0896	9.2036	20.441
	Monomial	0.5466	0.3523	0.3277	0.8891	5.2428	13.524



(a)

(b)

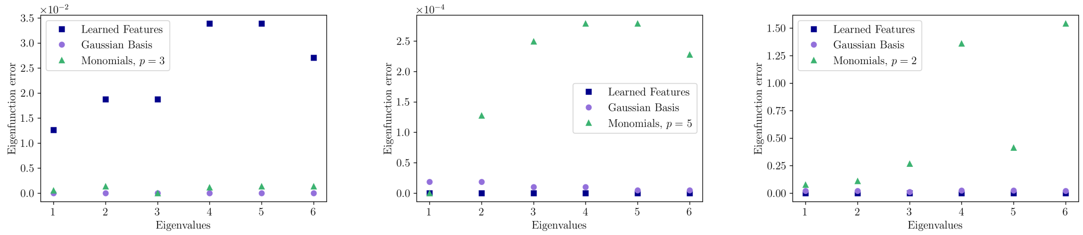
(c)

Figure 4: Reconstruction of trajectories in the (a) Duffing oscillator, (b) double gyre, and (c) Bickley jet using the Koopman operator constructed from learned features and known dictionaries.

Finally, we compute the error between predicted trajectories from Eq. (8) and true trajectories as

$$e_p = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\|_F^2}. \tag{9}$$

For all examples, we analyze near-term and long-term prediction errors in the reconstruction. For the learned features, we set up the optimization framework using Eq. (7). To compare against our framework, we construct the approximation of the Koopman operator through the EDMD algorithm (Williams et al., 2015a). Specifically, we use the insights from previously studied EDMD algorithms on the Duffing oscillator, double gyre, and Bickley jet to construct dictionaries (Li et al., 2017; Kaiser et al., 2021; Salam et al., 2022). For the Duffing oscillator, we compare our 100 RFFs against dictionaries created using the Gaussian basis function (GBF) on a 50 × 50 discretization of the system with $\sigma = 1e^{-}4$ and monomial basis functions of degree 3. For the double gyre, we use 100 RFFs, and the baseline has GBFs on a 10 × 5 discretization with $\sigma = 0.1$ and monomial basis



(a) (b) (c)

Figure 5: Eigenfunction error approximation for Koopman operator constructed with learned features and known dictionaries. Errors were computing using randomly samples points in the (a) Duffing oscillator, (b) double gyre, and (c) Bickley jet environments and averaging their corresponding eigenfunction error.

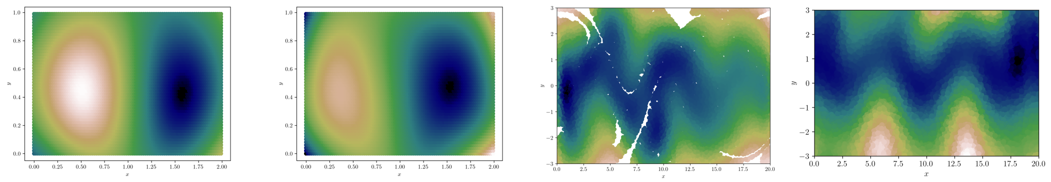
functions of degree 5. For the Bickley jet, we use 200 RFFs, while the dictionaries are GBFs with $\sigma = 1.1$ on a 10×10 discretization and monomial basis function of degree 2. While the learned features are computed online over the entire available dataset, the EDMD algorithm is computed against each snapshot pair of data $\mathbf{X}(t)$ and $\mathbf{X}(t+1)$. The estimated near-term (NT) trajectory is found using Eq. (8) for 10 time steps into the future, as in we reconstruct $[\hat{\mathbf{X}}(t+2), \ldots \hat{\mathbf{X}}(t+11)]$. For the long-term (LT) estimation, we compute 40 time steps into the future. Finally, the error is computing using the Eq. (9) for trajectory prediction errors.

The results of the error reconstructions are shown in Table 1. The errors shown in the table represent the difference between the true trajectory and trajectories from the Koopman operator constructed from dictionaries with the learned features, the Gaussian basis functions, and the monomial basis functions. We see for simpler systems, such as the Duffing oscillator, the learned features perform comparably, if not slightly worse, than models from handcrafted dictionaries. However, for more complicated systems, such as the double gyre and Bickley jet, reconstruction in the long-term regime using the learned features outperforms the methods using the handcrafted dictionaries. This is exemplified most clearly in Fig. 4 for trajectories of randomly sampled particles. For short-term predictions, the trajectories of the particles estimated using the learned features versus the Gaussian or monomial basis functions are similar. However, as more time elapses between the initial position of the particle and the estimate, the learned features are able to more closely match the true trajectory. The trajectories estimated using the Gaussian basis functions diverge after several time steps. While other basis functions capture some notion of coherence, the learned features are able to capture both the notion of coherence and the movement of individual particles within the space.

We next evaluate the accuracy of the eigenfunction approximation (Li et al., 2017). Let the approximate eigenfunction corresponding to the eigenvalue μ_j of $\mathcal{K}$ be $\psi_j = \mathbf{v}_j^\top \Phi$. Then, the accuracy of the eigenfunction approximation is

$$e_{f_j} = \sqrt{\frac{1}{I} \sum_{i=1}^{I} |\psi_j(f(x(i)) - \mu_j \psi_j(x(i))|^2}. \tag{10}$$

We show the eigenfunction approximation for the Duffing oscillator and double gyre comparisons for various features and dictionaries in Fig. 5. The value of e_j as in Eq. (10) are selected for the first few eigenvalues in all examples. We select 100 randomly sampled points for each dynamical system and calculate the eigenfunction approximation error for the eigenfunctions from the



(*a*) (*b*) (*c*) (*d*)

Figure 6: Eigenfunction comparison for kernel EDMD algorithm versus EDMD with learned features for time-dependent double gyre and Bickley jet. Dominant eigenfunction (*a*) from the Koopman operator constructed using kernel EDMD and (*b*) using learned features for double gyre. Dominant eigenfunction (*c*) from the Koopman operator constructed using kernel EDMD and (*d*) using learned features for Bickley jet.

learned model, the Gaussian basis functions, and the monomial basis functions. We can see the learned features perform comparably to the monomial basis functions, which also provided good reconstruction. The learned features also outperform the Gaussian basis functions. Note, that the low error in approximation is an indicator of the invariance of the eigenfunctions, as it shows that the application of eigenfunctions do not cause high divergence.

5.3. Quantitative features

Next, we quantitatively compare eigenfunction values derived from the EDMD using the learned RFFs with the kernel EDMD algorithm (Williams et al., 2015b). For the double gyre and Bickley jet, we know that the coherent sets, an important property of global dynamics, can be estimated using an RBF kernel with $\sigma = 0.75$ and $\sigma = 1.0$, respectively (Salam et al., 2022). We plot the eigenfunctions of the Koopman operator associated with the dominant eigenvalues of the decomposition. The eigenfunctions of the Koopman operator corresponding to the kernel EDMD formulation elucidate two distinct regions for the double gyre and waves for the Bickley jet, as seen in Fig. 6(*a*) and 6(*c*) respectively. The eigenfunctions of the Koopman operator learned using the proposed algorithm in Fig. 6(*b*) and 6(*d*) agree closely with verified notions of coherence within these systems.

6. Conclusion

In this paper, we develop an automated, online procedure for constructing operators to describe global representations of dynamical systems. This work can be generalized through the use of non-stationary kernels and extended to applications in controls. Non-stationary kernels capture input-dependent correlations. While non-stationary kernel learning gained attention in recent years (Samo and Roberts, 2015; Remes et al., 2017), the problem of automated non-stationary kernel is still an active area of research. While transfer operators are well-studied in controls, there are practical considerations in applying these learned models to control theory. The dictionary of observables may fail to satisfy the invariance property for observables and the derived evolution of the observable may leave the lifted space (Bruder et al., 2019). While several works focus on finding functions that are the basis of Koopman invariant subspaces (Kaiser et al., 2021; Kawahara, 2016), there are no mathematical guarantees for the identified functions to be Koopman eigenfunctions (Haseli and Cortés, 2021). Notably, many open questions surround the representation and evolution of control inputs in transfer operator methods (Proctor et al., 2018; Otto and Rowley, 2021).

7. Acknowledgements

We greatly acknowledge the support of the NSF IUCRC 1939132, NSF IIS 1910308, and the University of Pennsylvania's University Research Foundation Award.

References

Ian Abraham and Todd D. Murphey. Active Learning of Dynamics for Data-Driven Control Using Koopman Operators. *IEEE Transactions on Robotics*, 35(5):1071–1083, 10 2019. ISSN 19410468. doi: 10.1109/TRO.2019.2923880.

Francisco J. Beron-Vera, María J. Olascoaga, Michael G. Brown, Huseyin Koçak, and Irina I. Rypina. Invariant-tori-like Lagrangian coherent structures in geophysical flows. *Chaos*, 20(1), 2010. doi: 10.1063/1.3271342.

Daniel Bruder, Brent Gillespie, C David Remy, and Ram Vasudevan. Modeling and Control of Soft Robots Using the Koopman Operator and Model Predictive Control. In *Proceedings of Robotics: Science and Systems*, 2019. doi: 10.15607/rss.2019.xv.060.

Antonio C Costa, Tosif Ahamed, David Jordan, and Greg J Stephens. Maximally predictive ensemble dynamics from data. *bioRxiv*, page 2021.05.26.445816, 1 2021. doi: 10.1101/2021.05.26.445816. URL http://biorxiv.org/content/early/2021/10/20/2021.05.26.445816.abstract.

Suddhasattwa Das and Dimitrios Giannakis. Koopman spectra in reproducing kernel Hilbert spaces. 2018. URL http://arxiv.org/abs/1801.07799.

Diego Del-Castillo-Negrete and P. J. Morrison. Chaotic transport by Rossby waves in shear flow. *Physics of Fluids A*, 5(4):948–965, 1992. doi: 10.1063/1.858639.

Carl Folkestad, Daniel Pastor, and Joel W. Burdick. Episodic Koopman Learning of Nonlinear Robot Dynamics with Application to Fast Multirotor Landing. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 9216–9222. Institute of Electrical and Electronics Engineers Inc., 5 2020a. ISBN 9781728173955. doi: 10.1109/ICRA40945.2020.9197510.

Carl Folkestad, Daniel Pastor, Igor Mezic, Ryan Mohr, Maria Fonoberova, and Joel Burdick. Extended Dynamic Mode Decomposition with Learned Koopman Eigenfunctions for Prediction and Control. In *Proceedings of the American Control Conference*, 2020b. ISBN 9781538682661. doi: 10.23919/ACC45564.2020.9147729.

Eric Forgoston, Lora Billings, Philip Yecko, and Ira B. Schwartz. Set-based corral control in stochastic dynamical systems: Making almost invariant sets more invariant. *Chaos*, 2011. ISSN 10541500. doi: 10.1063/1.3539836.

Arthur Gretton. Introduction to rkhs, and some simple kernel algorithms. *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 16:5–3, 2013.

Alireza Hadjighasem, Mohammad Farazmand, Daniel Blazevski, Gary Froyland, and George Haller. A critical comparison of Lagrangian methods for coherent structure detection. *Chaos*, 27(5):1–25, 2017. ISSN 10541500. doi: 10.1063/1.4982720.

Masih Haseli and Jorge Cortés. Learning Koopman eigenfunctions and invariant subspaces from data: Symmetric subspace decomposition. *IEEE Transactions on Automatic Control*, 2021. ISSN 0018-9286.

Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of Koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2(3):35023, 2021. ISSN 2632-2153.

Yoshinobu Kawahara. Dynamic Mode Decomposition with Reproducing Kernels for Koopman Spectral Analysis. *Advances in Neural Information Processing Systems*, 29(1):911–919, 2016.

Stefan Klus, Ingmar Schuster, and Krikamol Muandet. Eigendecompositions of Transfer Operators in Reproducing Kernel Hilbert Spaces. *Journal of Nonlinear Science*, 30(1):283–315, 2020. ISSN 14321467. doi: 10.1007/s00332-019-09574-z. URL http://arxiv.org/abs/1712.01572.

Milan Korda and Igor Mezić. Optimal construction of Koopman eigenfunctions for prediction and control. 2018. URL http://arxiv.org/abs/1810.08733.

SB Leask, VG McDonell, and S Samuelsen. Modal extraction of spatiotemporal atomization data using a deep convolutional koopman network. *Physics of Fluids*, 33(3):033323, 2021.

Qianxiao Li, Felix Dietrich, Erik M. Bollt, and Ioannis G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos*, 27(10):1–25, 2017. ISSN 10541500. doi: 10.1063/1.4993854.

Andreas Mardt, Luca Pasquali, Frank Noé, and Hao Wu. Deep learning Markov and Koopman models with physical constraints. In *Mathematical and Scientific Machine Learning*, pages 451–475. PMLR, 2020. ISBN 2640-3498.

Alexandre Mauroy and Igor Mezić. Global Stability Analysis Using the Eigenfunctions of the Koopman Operator. *IEEE Transactions on Automatic Control*, 2016. ISSN 00189286. doi: 10.1109/TAC.2016.2518918.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, pages 309–325, 2005. ISSN 0924090X. doi: 10.1007/s11071-005-2824-x.

Samuel E Otto and Clarence W Rowley. Koopman Operators for Estimation and Control of Dynamical Systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):59–87, 5 2021. ISSN 2573-5144. doi: 10.1146/annurev-control-071020-010108. URL https://doi.org/10.1146/annurev-control-071020-010108.

Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018. ISSN 15360040. doi: 10.1137/16M1062296.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 2009. ISBN 160560352X.

Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):4643–4652, 2017. ISSN 10495258.

Clarence W. Rowley, Igor Mezi, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009. ISSN 00221120. doi: 10.1017/S0022112009992059.

Tahiya Salam, Victoria Edwards, and M Ani Hsieh. Learning and Leveraging Features in Flow-Like Environments to Improve Situational Awareness. *IEEE Robotics Autom. Lett.*, 7(2):2071–2078, 2022. doi: 10.1109/LRA.2022.3141762. URL https://doi.org/10.1109/LRA.2022.3141762.

Salomon Bochner. *Vorlesungen über Fouriersche Integrale*. Leipzig, 1932. doi: 10.1007/bf01708925.

Yves-Laurent Kom Samo and Stephen Roberts. Generalized Spectral Kernels. *arXiv preprint arXiv:1506.02236*, 2015. URL http://arxiv.org/abs/1506.02236.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. ISSN 14697645. doi: 10.1017/S0022112010001217.

Lu Shi and Konstantinos Karydis. Acd-edmd: Analytical construction for dictionaries of lifting functions in koopman operator-based nonlinear robotic systems. *IEEE Robotics and Automation Letters*, 7(2):906–913, 2021.

Mario Sznaier. A Convex Optimization Approach to Learning Koopman Operators. 144:1–11, 2021. URL http://arxiv.org/abs/2102.03934.

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, 2015a. ISSN 1432-1467.

Matthew O. Williams, Clarence W. Rowley, and Ioannis G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2), 2015b. ISSN 21582505. doi: 10.3934/jcd.2015005.

E Yeung, S Kundu, and N Hodas. Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems. In *2019 American Control Conference (ACC)*, pages 4832–4839, 2019. ISBN 2378-5861 VO -. doi: 10.23919/ACC.2019.8815339.