Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator

Qianxiao Li, Felix Dietrich, Erik M. Bollt, and Ioannis G. Kevrekidis

Citation: Chaos **27**, 103111 (2017);
View online: https://doi.org/10.1063/1.4993854
View Table of Contents: http://aip.scitation.org/toc/cha/27/10
Published by the American Institute of Physics

Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator

Qianxiao Li,[1,a)] Felix Dietrich,[2] Erik M. Bollt,[3] and Ioannis G. Kevrekidis[4,b),c)]
[1]*Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore, Singapore 138632*
[2]*Faculty of Mathematics, Technical University of Munich, Munich 80333, Germany*
[3]*Department of Mathematics and Department of Electrical and Computer Engineering, Clarkson University, Potsdam, New York 13699, USA*
[4]*Department of Chemical and Biological Engineering and The Program in Applied and Computational Mathematics, Princeton University, Princeton, New Jersey 08544, USA*

(Received 1 July 2017; accepted 25 September 2017; published online 9 October 2017)

Numerical approximation methods for the Koopman operator have advanced considerably in the last few years. In particular, data-driven approaches such as dynamic mode decomposition (DMD)[51] and its generalization, the extended-DMD (EDMD), are becoming increasingly popular in practical applications. The EDMD improves upon the classical DMD by the inclusion of a flexible choice of dictionary of observables which spans a finite dimensional subspace on which the Koopman operator can be approximated. This enhances the accuracy of the solution reconstruction and broadens the applicability of the Koopman formalism. Although the convergence of the EDMD has been established, applying the method in practice requires a careful choice of the observables to improve convergence with just a finite number of terms. This is especially difficult for high dimensional and highly nonlinear systems. In this paper, we employ ideas from machine learning to improve upon the EDMD method. We develop an iterative approximation algorithm which couples the EDMD with a *trainable* dictionary represented by an artificial neural network. Using the Duffing oscillator and the Kuramoto Sivashinsky partical differential equation as examples, we show that our algorithm can effectively and efficiently adapt the trainable dictionary to the problem at hand to achieve good reconstruction accuracy without the need to choose a fixed dictionary *a priori*. Furthermore, to obtain a given accuracy, we require fewer dictionary terms than EDMD with fixed dictionaries. This alleviates an important shortcoming of the EDMD algorithm and enhances the applicability of the Koopman framework to practical problems. *Published by AIP Publishing.* https://doi.org/10.1063/1.4993854

Every dynamical system has an associated Koopman operator, which encodes many important properties of the system. Most notably, it characterizes the temporal evolution of observables by linear, albeit infinite-dimensional, dynamics even when the underlying dynamical system is non-linear. In recent years, the growing availability of data and novel numerical techniques have enabled us to study this operator computationally. Extended Dynamic Mode Decomposition (EDMD) is one such technique for approximating the spectral properties of the operator. While effective for some problems, a clear drawback of EDMD is the requirement to select *a priori* a suitably efficient collection of basis functions, called a *dictionary*. In this paper, we use ideas from machine learning to optimally adapt the dictionary to data. This enables us to obtain improved numerical approximations without resorting to large dictionary sizes. We demonstrate the efficiency of our algorithm by approximating the Koopman operator for the Duffing oscillator system and also the Kuramoto-Sivashinsky partial differential equation (PDE).

I. INTRODUCTION

In the analysis of dynamical systems, a primary object of study is the state and its evolution. In this traditional setting, powerful tools from differential geometry can characterize dynamical systems by their trajectories and invariant manifolds in phase space. In recent years, however, advances in numerical techniques and the broader availability of data have sparked renewed interest in an alternative, *operator* view on dynamical systems:[1,2] the Koopman operator framework.[3] In this framework, the central objects of study are *observables*, which are functions of the state of the dynamical system. The Koopman operator then describes the temporal evolution of these functions driven by the underlying dynamics.

The Koopman formalism is useful in several ways. First, the Koopman dynamics is linear, albeit infinite-dimensional, and hence amenable to powerful methods from the operator theory, such as spectral analysis.[4–6] Second, it is especially suited for studying high-dimensional systems, where the

a)Electronic mail: liqix@ihpc.a-star.edu.sg
b)Electronic mail: yannis@princeton.edu
c)Present address: Departments of Chemical and Biomolecular Engineering and Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218, USA and Department of Urology, Johns Hopkins School of Medicine, Baltimore, MD 21218, USA.

phase space is so large that little can be said from the differential geometry point of view. The Koopman approach allows one to focus on the evolution of a lower number of observables. In fact, in applications, this is often the case: the evolution of a small number of measurements (observables) of an otherwise high dimensional system is recorded. Lastly, from a numerical point of view, it allows one to employ traditional techniques in numerical linear algebra to perform linearization and "normal mode analysis" for non-linear systems. This is an important advantage of the Koopman framework for current challenges in model reduction, prediction, data fusion, and system control.[7–12] Applications range from fluid dynamics,[13,14] energy modeling in buildings,[15] and oceanography[10] to molecular kinetics[16] and beyond.

We note that, given appropriate function spaces, the adjoint of the Koopman operator is the Perron-Frobenius operator. It operates on phase-space density functions and advances them in time according to the underlying dynamics. The duality of these two operators can be described as "dynamics of observables" for the Koopman operator in contrast to "dynamics of densities" for the Perron-Frobenius operator.[8] Both are valid descriptions of the underlying system through the perspective of linear operators.

In effect, the Koopman framework converts a finite-dimensional, possibly non-linear dynamical system to an infinite-dimensional linear system. In practice, this amounts to a simplification only when one can handle the latter numerically. Several numerical techniques have been developed in this regard. Many investigations focus on particular dynamical systems (linear systems, nonlinear systems with analytically known linearizations, and ergodic systems) and their associated Koopman operator. Numerical methods for (generalized) Fourier and Laplace analysis perform linearization of nonlinear systems close to steady states and limit cycles,[13,17] with a particular focus on the relation of isochrons and isostables to the eigenfunctions of the Koopman operator. These methods are useful in finding specific eigenfunctions of the dynamical system with desired properties but are less suited for obtaining a general spectral decomposition of the Koopman operator. Giannakis[18] describes how to estimate Koopman eigenfunctions with diffusion maps for ergodic systems. Klus *et al.*[19] discuss several data driven methods approximating transfer operators, including the *variational approach of conformation dynamics* (VAC) and *extended dynamic mode decomposition* (EDMD). Sparse identification of nonlinear dynamics (SINDy) searches for an optimal, sparse representation of the dynamics,[16,20] requiring a large dictionary of simple building blocks.

The EDMD algorithm is an extension of Dynamic Mode Decomposition (DMD)[7,21] and was developed by Williams *et al.*[22,23] The main improvement over DMD is the possibility to choose a set of observables, called a *dictionary*. One can then approximate the Koopman operator as a linear map on the span of the finite set of dictionary elements. The spectral decomposition of this finite-dimensional linear map is numerically tractable, and its spectral properties can approximate those of the Koopman operator. The original DMD algorithm can be interpreted as choosing only the system state as the observation of the system. By a "careful" choice of dictionary containing elements, beyond the system state observation functions, the EDMD algorithm is seen to have improved performance over DMD.[23]

A clear drawback of the EDMD algorithm is the need to make an *a priori* choice of dictionary. It is well-known that the choice significantly impacts the approximation quality of the spectral properties of the system.[22–24] For high-dimensional and highly non-linear systems, it is often not easy to make a judicious selection without prior information of the dynamics. In this paper, we aim to alleviate this issue by borrowing ideas from machine learning. We develop an iterative approximation algorithm which couples the EDMD with a *trainable* dictionary represented here by an artificial neural network, acting as a universal function approximator. The dictionary can be trained to adapt to the data it is presented with, and this effectively reduces the need to specify a problem-dependent dictionary. To demonstrate the efficacy of our algorithm, we use it to perform approximate spectral decompositions of the Koopman operator for the autonomous Duffing oscillator and the Kuramoto-Sivashinsky PDE on a two-dimensional quasiperiodic and attracting invariant manifold.

In Sec. II, we describe briefly the background for the Koopman operator viewpoint of dynamical systems. We also introduce the notation used throughout this paper. Section III provides a summary of the EDMD algorithm, followed by the introduction of our extension of it to adapt the dictionary elements to the data. In Sec. IV, we use two examples, namely, the Duffing oscillator and the Kuramoto-Sivashinsky PDE, to demonstrate the efficacy of our approach. Section V discusses the results with respect to accuracy and possible extensions, and Sec. VI concludes the paper.

II. PRELIMINARIES

A. The Koopman operator

Consider a measure space $(\mathcal{M}, \mathcal{F}, \rho)$ on which we define a dynamical system

$$x(n+1) = f(x(n)), \quad x(n) \in \mathcal{M}, \quad n \geq 0, \tag{1}$$

for some $\mathcal{F}$-measurable mapping $f : \mathcal{M} \to \mathcal{M}$. The Koopman formalism focuses on the evolution of *observables*, which are represented by functions on $\mathcal{M}$ in a suitable function space. Usually, we consider the Hilbert space

$$L^2(\mathcal{M}, \rho) = \{\phi : \mathcal{M} \to \mathbb{C} : ||\phi||_{L^2(\mathcal{M},\rho)} < \infty\},$$

with the usual inner product and norm

$$(\phi, \psi) = \int_{\mathcal{M}} \phi(x)\overline{\psi(x)}\rho(dx),$$

$$||\phi||_{L^2(\mathcal{M},\rho)} = \left(\int_{\mathcal{M}} |\phi(x)|^2 \rho(dx)\right)^{\frac{1}{2}}.$$

The Koopman operator, $\mathcal{K}$, is defined as an operator acting on $L^2(\mathcal{M}, \rho)$, so that for an observable $\phi \in L^2(\mathcal{M}, \rho)$, we have

$$\mathcal{K}\phi = \phi \circ f. \tag{2}$$

Intuitively, $\mathcal{K}$ describes the evolution of each observable as driven by the dynamical system. For the map $\mathcal{K}$ to be well-defined, f must be *non-singular*, i.e., for every $S \in \mathcal{F}$, $\rho(f^{-1}(S)) = 0$ whenever $\rho(S) = 0$. If we assume in addition that $C > 0$ such that $\rho(f^{-1}(S)) \leq C\rho(S)$ for all $S \in \mathcal{F}$, then the Koopman operator is a bounded linear operator on $L^2(\mathcal{M}, \rho)$[25] and is amenable to spectral analysis. The so-called *Koopman mode decomposition* can be described as follows: Given a vector of observables, $O : \mathcal{M} \to \mathbb{C}^d$, we can write

$$O(n) = O(x(n)) = \sum_k \mu_k^n \phi_k(x(0)) m_k^O, \tag{3}$$

where $\phi_k \in L^2(\mathcal{M}, \rho)$ are the eigenfunctions of $\mathcal{K}$ with eigenvalues $\mu_k \in \mathbb{C}$. The vectors $m_k^O \in \mathbb{C}^d$ are known as the Koopman modes associated with the observable O.

In applications, we are often interested in the full-state observable $O(x) = x$. Then, the Koopman mode decomposition can be viewed as a nonlinear counterpart of normal mode analysis. In this case, we interpret a finite dimensional nonlinear system (1) as an infinite dimensional linear system (2), whose spectral evolution follows (3).

B. Continuous-time systems

The Koopman formalism can be similarly applied to continuous-time systems by considering infinitesimal generators. Alternatively, we can interpret continuous-time systems as discrete-time ones by using the flow map. Consider the dynamical system

$$\dot{x}(t) = g(x(t)).$$

Let $\tau > 0$ and define the flow map

$$\Phi_\tau(x_0) := x(\tau)$$

where $x(t)$ follows the dynamical system above with $x(0) = x_0$. Then, by defining $\tilde{x}(n) := x(n\tau)$, we obtain a discrete-time dynamical system as in (1), with $f \equiv \Phi_\tau$.

III. NUMERICAL METHODS

Although the linear Koopman dynamics is theoretically easier to analyze, in practice, it is often a challenge to compute its spectral properties due to its infinite-dimensionality. In this section, we briefly review the classical EDMD method for computing Koopman mode decompositions. We then introduce our numerical method that incorporates machine learning to address the important shortcoming of traditional methods—the need of selecting a fixed dictionary that may be generally "inefficient."

A. The EDMD algorithm

Since our dictionary learning algorithm is built upon the extended dynamic mode decomposition (EDMD),[23] we begin by describing briefly the EDMD algorithm. The main idea is to estimate a finite-dimensional representation of the Koopman operator $\mathcal{K}$ in the form of a finite-dimensional linear map K, whose spectral properties will then approximate those of $\mathcal{K}$. To do this, pick a dictionary $\Psi = \{\psi_1, \psi_2, \ldots, \psi_M\}$ where

$$\psi_i : \mathcal{M} \to \mathbb{R} \quad \text{for} \quad i = 1, 2, \ldots, M.$$

Now, consider the span $U(\Psi) = \text{span}\{\psi_1, \ldots, \psi_M\} = \{a^T\Psi : a \in \mathbb{C}^M\}$, which is a linear subspace of $L^2(\mathcal{M}, \rho)$. For any $\phi = a^T\Psi \in U(\Psi)$, we have $\mathcal{K}\phi = a^T\mathcal{K}\Psi = a^T\Psi \circ f$. If $\mathcal{K}(U(\Psi)) = U(\Psi)$, then we also have $\mathcal{K}\phi = b^T\Psi$ for some $b \in \mathcal{C}^M$. Hence, a finite dimensional representation of $\mathcal{K}$ is realized as the matrix $K \in \mathbb{R}^{M\times M}$ with $b = K^T a$. Thus, we have the equality $a^T\Psi \circ f = a^T K\Psi$. For this, to hold for all a, we must have $\Psi \circ f = K\Psi$. To find K, we use pairs of data points $\{x(n), y(n)\}_{n=1}^N$ with $y(n) = f(x(n))$ and solve the minimization problem

$$K = \underset{\tilde{K}\in\mathbb{R}^{M\times M}}{\text{argmin}}\, J(\tilde{K}) = \sum_{n=1}^N ||\Psi(y(n)) - \tilde{K}\Psi(x(n))||^2. \tag{4}$$

If $U(\Psi)$ is indeed invariant under $\mathcal{K}$, then $J(K) = 0$. Otherwise, $J(K) > 0$, and the procedure above seeks to find K that minimizes the residual J. The solution to (4) is

$$K = G^+ A,$$

with

$$G = \frac{1}{N}\sum_{n=1}^N \Psi(x(n))^T \Psi(x(n)),$$
$$A = \frac{1}{N}\sum_{n=1}^N \Psi(x(n))^T \Psi(y(n)), \tag{5}$$

and G^+ denotes the pseudo-inverse of G. With K derived, it is straightforward to find eigenfunctions and eigenvalues of K and likewise modes associated with an observable O. For example, one can check that for each left eigenvector ξ_j of K with eigenvalue μ_j, the function

$$\phi_j = \xi_j^T \Psi,$$

is an approximation of an eigenfunction of $\mathcal{K}$ with the same eigenvalue μ_j. Also, for any vector of observables $O = B\Psi$, the jth Koopman mode associated with u is given by

$$m_j = B\zeta_j,$$

where ζ_j is the jth right eigenvector of K.

The matrix K found in this way is shown to converge to $\mathcal{K}_\Psi$, the L^2 orthogonal projection of the Koopman operator $\mathcal{K}$ onto $U(\Psi)$, as $N \to \infty$.[23,26] It has been further established that if Ψ consists of linearly independent basis functions, then as $M \to \infty$ one has $\mathcal{K}_\Psi \to \mathcal{K}$ in the strong operator topology.[24]

In practice, however, both N and M are finite. Therefore, one primary assumption underlying EDMD's application is that the finite dimensional subspace $U(\Psi)$ is approximately invariant under $\mathcal{K}$. This is true if either M is very large or more practically, if the dictionary set Ψ is judiciously

chosen.[23] The choice of dictionary is especially difficult for highly nonlinear or high dimensional systems, for which even enumerating a standard basis (e.g., orthogonal polynomials) becomes prohibitively expensive. Although there exist kernel methods[22] to alleviate such problems, the choice of dictionary sets (including kernel functions) remains a central challenge for the general applicability of EDMD.

In Sec. III B, we use ideas from machine learning to show how one can alleviate the problem of having to choose a fixed dictionary. Most importantly, this holds the promise of allowing high-quality representation with relatively fewer dictionary terms.

B. EDMD with dictionary learning (EDMD-DL)

Dictionary learning (or sparse coding) is a classical problem in signal processing and machine learning.[27] The problem statement is as follows: given a set of input data, $X = (x(1) \quad x(2) \quad \dots \quad x(N)) \in \mathbb{R}^{d\times N}$, we wish to find a sparse representation of it in the form of $X = DK$, where $D \in \mathbb{R}^{d\times M}$ is a size-M set of dictionary vectors and $K \in \mathbb{R}^{M\times N}$ is a sparse representation. For any fixed D, it is difficult to fulfill accuracy ($X \approx DK$) *and* sparsity ($||K||_0$ small) at the same time. A better approach is to make D adapted to data and solve

$$(K, D) = \underset{(\tilde{K},\tilde{D})}{\operatorname{argmin}} J(\tilde{K}, \tilde{D}) = ||X - \tilde{D}\tilde{K}\ ||_F^2 + \lambda||\tilde{K}||_1, \quad (6)$$

where $||\cdot||_F$ is the Frobenius norm. The ℓ_1 penalty induces sparsity without turning it into a combinatorial optimization problem, as in the case for ℓ_0 penalty. To remove degeneracies, one may impose further conditions such as $\{D : ||D||_F = 1\}$.

From the above, the key idea we would like to adapt to the EDMD framework is allowing the dictionary to be trainable, which then enables one to find an "efficient" representation of the data with a smaller number of adaptive basis elements. In this sense, the procedure is similar to the Karhunen-Loève decomposition (KLD),[28,29] whose sampled versions are also known as principal component analysis (PCA) and proper orthogonal decomposition (POD). The goal of KLD is to obtain an expansion of stochastic processes in terms of adaptive basis functions for which the truncation error is optimal, in the mean-squared sense.

In our case of EDMD decompositions, our goal is to make the dictionary set Ψ adaptive so that we can minimize the norm of the residual $\Psi \circ f - K\Psi$ resulting from the finite-dimensional projection [see (4)]. Hence, instead of (4), we can consider the extended minimization problem

$$\begin{aligned}(K, \Psi) &= \underset{(\tilde{K},\tilde{\Psi})}{\operatorname{argmin}} J(\tilde{K}, \tilde{\Psi}) \\ &= \sum_{n=1}^{N} ||\tilde{\Psi}(y(n)) - \tilde{K}\tilde{\Psi}(x(n))||^2 + \lambda(\tilde{K}, \tilde{\Psi}), \end{aligned} \quad (7)$$

where $\lambda(K, \Psi)$ is a suitable regularizer. Unlike (6), the dictionary functions in Ψ are not assumed to be linear functions, and hence, nonlinear optimization methods must be used. Nevertheless, provided we can solve (7), this formulation provides us with a means to find an adaptive set of dictionary elements that give optimal truncation errors, in a similar vein to sparse coding or the Karhunen-Loève decomposition.

We have outlined the primary idea underlying our adaptive EDMD algorithm. Next, we present a computational algorithm to solve (7) by parameterizing it with neural networks.

C. A practical algorithm

To solve (7), we parameterize Ψ by a universal function approximator, i.e., $\Psi(x) = \Psi(x; \theta)$ for some $\theta \in \Theta$ to be varied. Here, we choose a simple feed-forward, 3-layer neural network as the approximator for Ψ (see Fig. 1). Concretely, we choose a hidden dimension ℓ and set

$$\begin{aligned}\Psi(x) &= W_{\text{out}}h_3 + b_{\text{out}}, \\ h_{k+1} &= \tanh(W_k h_k + b_k), \quad k = 0, 1, 2, \end{aligned} \quad (8)$$

where $h_0 = x$ and $W_0 \in \mathbb{R}^{\ell\times d}, b_0 \in \mathbb{R}^d$, $W_{\text{out}} \in \mathbb{R}^{M\times\ell}$, $b_{\text{out}} \in \mathbb{R}^M$, and $W_k \in \mathbb{R}^{\ell\times\ell}, b_k \in \mathbb{R}^\ell$ for $k = 1$ and 2. The set of all trainable parameters is $\theta = \{W_{\text{out}}, b_{\text{out}}, \{W_k, b_k\}_{k=0}^2\}$, which contains a total of $d(l+1) + l(2l+M+3)$ scalar variables.

With Ψ parameterized, we can then solve (7)

$$\begin{aligned}(K, \theta) &= \underset{(\tilde{K},\tilde{\theta})}{\operatorname{argmin}} J(\tilde{K}, \tilde{\theta}) \\ &= \sum_{n=1}^{N} ||\Psi(y(n); \tilde{\theta}) - \tilde{K}\Psi(x(n); \tilde{\theta})||^2 + \lambda||\tilde{K}||_F^2. \end{aligned} \quad (9)$$

We picked the Tikhonov regularization[30,31] with the identity matrix for $\tilde{K}$ to improve the stability of the algorithm. Notice that if there exists $\tilde{\theta}$ with $\Psi(x; \tilde{\theta}) \equiv 0$, the right hand side identically vanishes and the minimum is trivially attained. Thus, to obtain meaningful approximations, we need further restrictions. A natural one is to include in $\Psi = \{\psi_1, \dots, \psi_M\}$ some fixed (non-trainable) functions, such as the constant and the projection maps. The presence of the latter is important for reconstructing trajectories. This is because to find the Koopman modes, we require the identity map $O(x) = x$, whose components are projection maps, to be in the linear span $U(\Psi)$. The inclusion of these non-trainable dictionary functions then removes the possibility that $\Psi(x; \tilde{\theta}) \equiv 0$. In

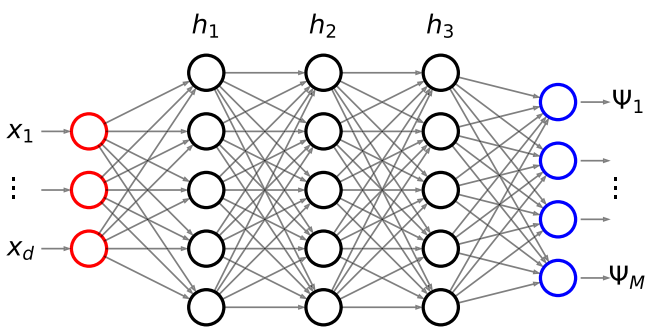

FIG. 1. Neural network function approximator for the trainable dictionary $\Psi(x; \theta)$. The network is fully connected and consists of 3 hidden layers h_1, h_2, h_3. Arrows connecting layers correspond to affine transformations followed by tanh activations. See Eq. (8).

other words, the optimization problem (9) seeks an invariant subspace of L^2 with respect to $\mathcal{K}$. Without any requirements, the trivial subspace $\{0\}$ suffices. Hence, it is natural to require that the subspace contains enough elements to reconstruct our observables of interest.

We solve (9) by iterating the following two steps: (a) Fix θ, optimize K; then (b), fix K, optimize θ.

- **Fix θ, optimize K.** For fixed θ (hence fixed Ψ), (9) is almost the same problem as (4) but with the addition of the Tikhonov regularizer. The solution is[32]

$$\tilde{K} = (G(\tilde{\theta}) + \lambda I)^{+} A(\tilde{\theta}), \tag{10}$$

  where G and A are defined in (5) and I is the d-dimensional identity matrix.
- **Fix K, optimize θ.** This is a standard machine learning problem. As there is no linear structure in the problem, we cannot write down its exact solution. Instead, we proceed by iterative updates in the form of gradient descent, i.e., we set

$$\tilde{\theta} \leftarrow \tilde{\theta} - \delta \nabla_{\theta} J(\tilde{K}, \tilde{\theta}). \tag{11}$$

  If both the dimension d and the sample size N is large, $\nabla_{\theta} J$ will be expensive to evaluate. We can then employ stochastic gradient descent and its variants,[33] where the gradient $\nabla_{\theta} J$ is replaced by randomly sampled unbiased estimators.

The above two steps are iterated until convergence. We have observed (empirically!) that the algorithm performed stably and converged for general initializations. A rigorous proof of the convergence of the algorithm will be left as future work. The algorithm is summarized in Algorithm 1, and we hereafter refer to it as EDMD with dictionary learning (EDMD-DL).

Algorithm 1. EDMD with dictionary learning (EDMD-DL).

Initialize K, θ.
Set learning rate $\delta > 0$, tolerance $\epsilon > 0$, regularizer $0 < \lambda \ll 1$
while $J(K, \theta) > \epsilon$ **do**:
  $K \leftarrow (G(\theta) + \lambda I)^{-1} A(\theta)$
  $\theta \leftarrow \theta - \delta \nabla_{\theta} J(K, \theta)$

IV. APPLICATIONS OF EDMD-DL

In this section, we compare the results from the EDMD-DL algorithm with the classical EDMD results on various example problems to illustrate the advantages of an adaptive, trainable dictionary. For each example, we evaluate the performance of various methods by two quantitative metrics:

- **Accuracy of trajectory reconstruction**. We reconstruct trajectories using the Koopman mode decomposition formula (3) with $O(x) = x$. We then monitor the reconstruction error as

$$\text{Error} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} |x(n) - \tilde{x}(n)|^2}, \tag{12}$$

  where x is the true trajectory data [according to (1)] and $\tilde{x}$ is the reconstructed trajectory [according to (3)].
- **Accuracy of eigenfunction approximation**. For each $j = 1, 2, \ldots, M$, we define the eigenfunction approximation error

$$E_j = ||\phi_j \circ f - \mu_j \phi_j||_{L^2(\mathcal{M},\rho)}, \tag{13}$$

where ϕ_j and μ_j are the jth eigenfunction and eigenvalue found by the algorithm, respectively. The above quantity can be approximated by Monte-Carlo integration

$$E_j \approx \sqrt{\frac{1}{I}\sum_{i=1}^{I} |\phi_j \circ f(x(i)) - \mu_j \phi_j(x(i))|^2},$$

where $x(i) \sim \rho$ are identically and independently distributed for all i.

A. Duffing oscillator

We start by applying EDMD-DL to the Duffing oscillator, which describes the evolution of $x = (x_1, x_2)$ governed by

$$\begin{aligned} \dot{x_1} &= x_2, \\ \dot{x_2} &= -\delta x_2 - x_1(\beta + \alpha x_1^2). \end{aligned} \tag{14}$$

We take $\alpha = 1$, $\beta = -1$, and $\delta = 0.5$ so that there are two stable steady states at $(\pm 1, 0)$ separated by a saddle point at (0, 0). We convert the continuous dynamical system to a discrete one by defining flow maps as discussed in II B, with the choice $\tau = 0.25$. We draw 1000 random initial conditions uniformly in the region $[-2, 2]^2$. Each initial condition is evolved up to $n = 10$ steps with the flow-map so that we have a total of 10^5 data points to form the training set.

Now, we apply the EDMD-DL algorithm with 22 trainable dictionary outputs (plus 3 non-trainable ones, i.e., one constant map and two coordinate projection maps) and compare its performance to EDMD with two choices of dictionary sets (1) using 25, two-dimensional Hermite polynomials and (2) 100 thin-plate radial basis functions (RBF) with centers placed on the training data using k-means clustering (`scipy.cluster.vq.kmeans`, thin-plates $r^2 \ln(r + \delta)$, regularized with $\delta = 10^{-4}$). In Fig. 2, we show the eigenvalues found by the three methods. To quantitatively compare the performance, we plot the trajectories reconstructed by the Koopman decomposition against the exact trajectories obtained by integrating the evolution equation (14). The results are shown in Fig. 3. We see that although EDMD-DL uses a small set of trainable dictionary outputs, it out-performs both EDMD with Hermite polynomials and RBFs, despite the fact that the latter is carefully chosen to be effective for the Koopman decomposition of the Duffing equation.[23] To confirm that EDMD-DL requires a smaller dictionary size, we plot in Fig. 4(a) the reconstruction error averaged over 50 random initial conditions vs the dictionary size for EDMD-DL and EDMD with RBF dictionaries. We see that EDMD-DL achieves lower reconstruction error at smaller dictionary sizes.

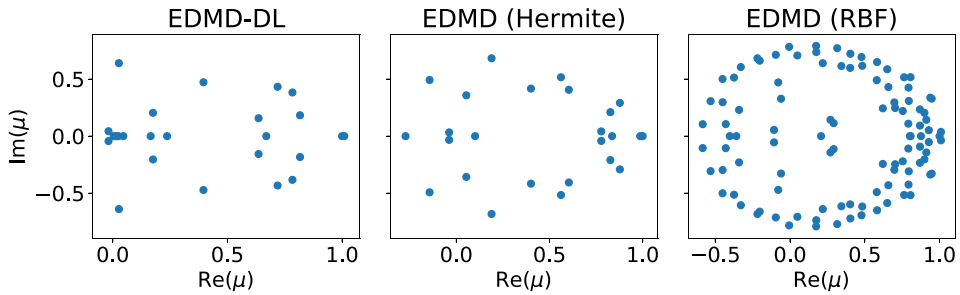

FIG. 2. Eigenvalues of the Koopman operator for the Duffing oscillator estimated from each algorithm. For EDMD-DL and EDMD with Hermite basis, 25 estimated eigenvalues are shown (since 25 dictionary functions are used). For EDMD with the RBF dictionary, 100 RBF functions are used, and so, 100 estimated eigenvalues are shown. We see that EDMD with Hermite polynomials has found many more eigenvalues with large magnitudes but does not improve accuracy significantly over EDMD-DL (see Fig. 3). In this sense, we see that EDMD-DL has found a more efficient representation.

As a further quantitative comparison, we evaluate the quality of the eigenfunctions by calculating for each j the eigenfunction error E_j [See definition (13)] with $\rho = 1_{[-2,2]^2}$. The value of E_j for the first 8 leading eigenfunctions is shown in Fig. 5. Again, we can see that EDMD-DL achieves comparable performance with a well-picked dictionary (RBF) and outperforms poorly picked ones (Hermite).

The Duffing oscillator is a low dimensional dynamical system, and hence, enumerating polynomial basis functions are still reasonably tractable. Provided that enough of them

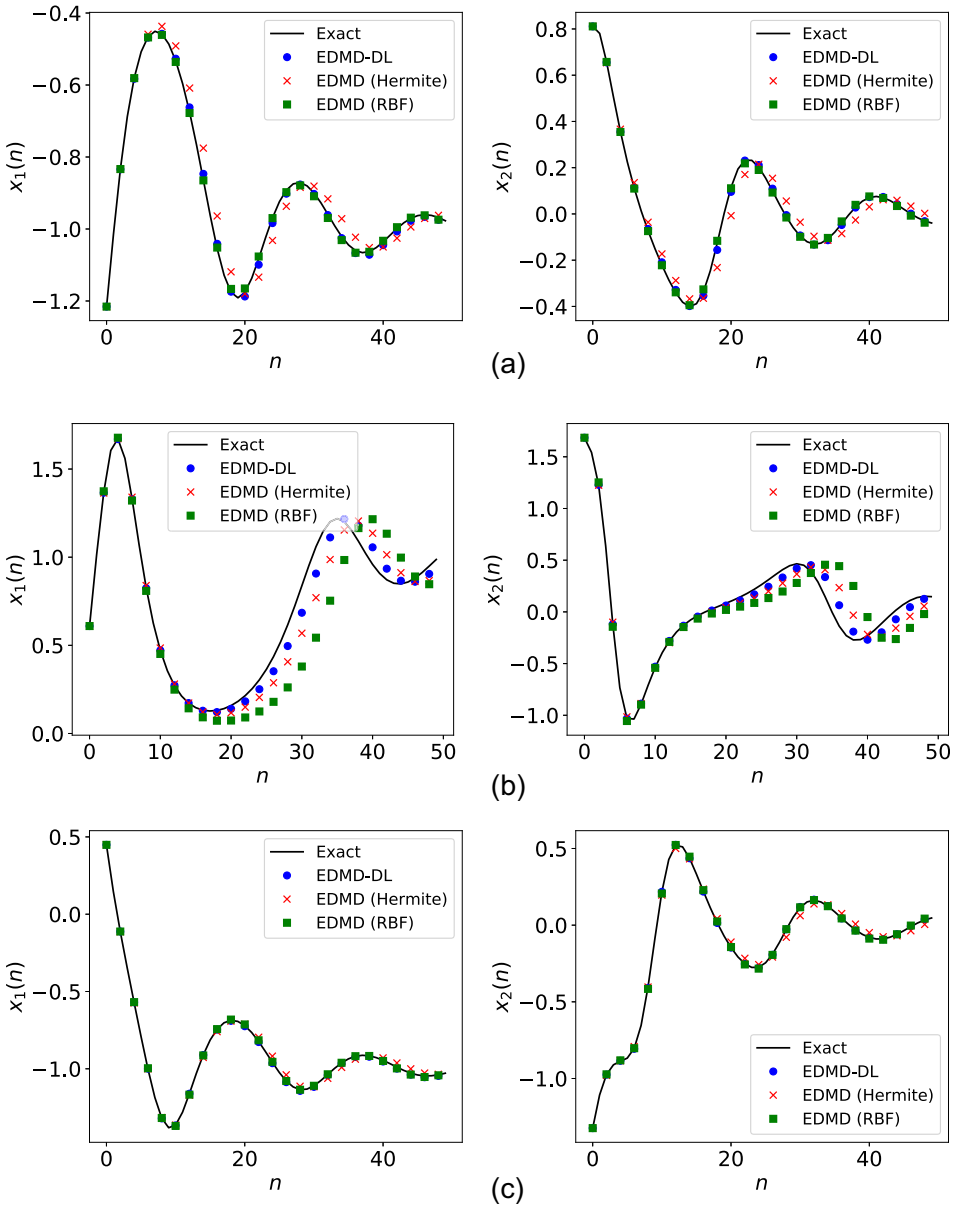

FIG. 3. Trajectories of the Duffing oscillator reconstructed from Koopman decomposition using various algorithms. Three different initial conditions in $[-2,2]^2$ are selected. We observe that EDMD-DL (with 25 dictionary elements) has better reconstruction accuracy than classical EDMD with Hermite polynomials with the same number of dictionary elements. We also see that EDMD-DL performs approximately on par with EDMD with the RBF dictionary (100 dictionary elements), which is known to be especially suited for this problem.[23] A quantitative comparison of reconstruction errors vs dictionary size is given in Fig. 4(a).

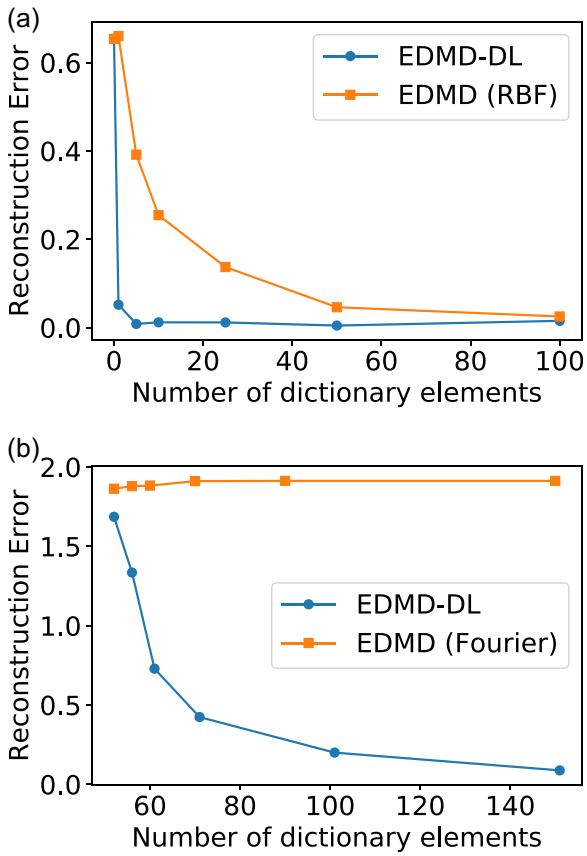

FIG. 4. Trajectory reconstruction error for EDMD-DL vs classical EDMD with a hand-picked dictionary, whose sizes are varied. The errors are averaged over 50(10) random and unseen initial conditions for the Duffing (Kuramoto-Sivashinsky) system. We see that EDMD-DL requires much smaller dictionary sizes in order to capture the system's dynamics.

are included in the dictionary, the finite-dimensional approximations for the Koopman operator become reasonably accurate. Moreover, *a priori* domain knowledge of the eigenfunctions can also allow us to pick better dictionaries, such as the RBFs.[23] Consequently, standard EDMD is still reasonable even though EDMD-DL still performs better. For general high dimensional systems, it is difficult to choose a dictionary in a systematic and efficient way. This situation is precisely where dictionary learning is most advantageous.

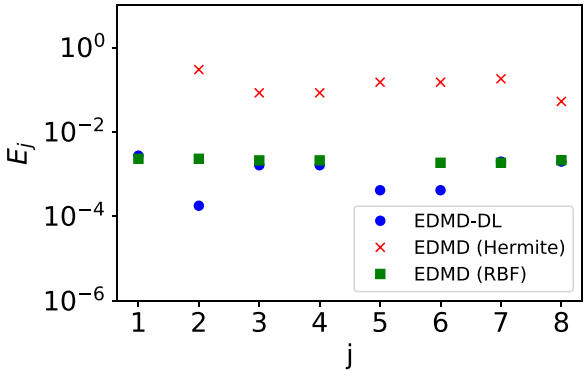

FIG. 5. Eigenfunction errors for the Duffing oscillator. Both EDMD-DL and EDMD with the Hermite dictionary have 25 dictionary functions. EDMD with the RBF dictionary has 100 dictionary elements. Again, we observe that dictionary learning has comparable performance to the well-chosen (and large) RBF dictionary and has better performance than the Hermite dictionary.

B. Kuramoto-Sivashinsky PDE

Consider the Kuramoto-Sivashinsky PDE

$$u_t + 4u_{zzzz} + \alpha(u_{zz} + uu_z) = 0, \quad z \in [0, 2\pi], \tag{15}$$

with $\alpha = 16$ and periodic boundary conditions on $[0, 2\pi)$. The initial condition is parameterized with $a \in [0.8, 1]$, $b \in [0.5, 1]$ and given as

$$u(z, 0) = a \sin(2\pi z) + b \exp(\cos(2\pi z)). \tag{16}$$

We sample a and b with 100 random, uniformly distributed points and compute the solution $u(z, t)$ at 50 equally distributed spatial points on $[0, 2\pi)$ and 100 points in time, in the interval $[0, 0.5]$.

As discussed before, it is difficult to pick a dictionary for the classical EDMD algorithm. Here, we use the following two choices:

1. A dictionary containing the state $u(z_i, t)$ and four of its spatial derivatives, all sampled at 50 equally spaced grid points $z_i \in [0, 2\pi)$. Thus, in total, this dictionary contains 250 elements.
2. A dictionary containing the state $u(z_i, t)$, sampled at 100 equally spaced grid points $z_i \in [0, 2\pi)$, and its Fourier coefficients, separated into real and imaginary parts. In total, this dictionary contains 150 elements.

These two dictionaries are compared to the results of EDMD-DL, where we pick 50 trainable dictionary outputs on top of the constant and projection maps (so that $M = 101$). In Fig. 6, we plot the eigenvalues found by each algorithm. We observe that although EDMD-DL uses a smaller number of dictionary outputs, the eigenvalue spectrum found is richer than those found by classical EDMD, where many computed eigenvalues are effectively 0. Next, we plot in Fig. 8 a reconstructed trajectory from a previously unseen initial condition. We see that classical EDMD with either choice of dictionaries cannot reproduce the fine-scale structures of the solution, whereas EDMD-DL achieves good reconstruction accuracy and manages to capture detailed behavior of the trajectory. Figure 4(b) again confirms that EDMD-DL achieves good performance with smaller dictionary sizes. In fact, in this PDE case, it is harder to pick a good dictionary, and hence, we see that the Fourier basis choice does not become better when more modes are included. This may be partially attributed to the fact that the Kuramoto-Sivashinsky PDE is known to possess an inertial manifold, so that the amplitudes of higher Fourier modes are effectively determined by those of the lower modes.[34,35]

Lastly, in Fig. 7, we observe that the eigenfunction errors E_j [defined in (13)] are much lower for EDMD-DL compared with classical EDMD. Here, instead of performing infinite-dimensional integration with some generic measure, we set ρ to be the sample distribution of u of the test trajectory used in Fig. 8.

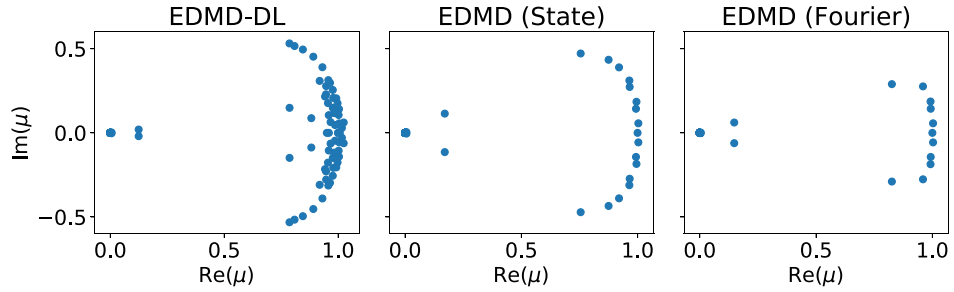

FIG. 6. Eigenvalues of the Koopman operator of the Kuramoto-Sivashinsky PDE estimated from each algorithm. The number of eigenvalues corresponds to dictionary sizes, which are 101 for EDMD-DL; 250 EDMD with the dictionary containing states and derivatives; and 150 for EDMD with the dictionary containing Fourier coefficients. Observe that although EDMD-DL produced less eigenvalues (because of a smaller dictionary), it produced more meaningful eigenvalues as compared to those of EDMD, where most are concentrated at 0. This is the opposite case to Fig. 2 because the PDE system necessarily requires a richer representation. Although both EDMD (state) and EDMD (Fourier) produced less eigenvalues with large magnitudes, they result in inaccurate representations of the dynamics (see Fig. 8) and hence cannot be considered sparse representations.

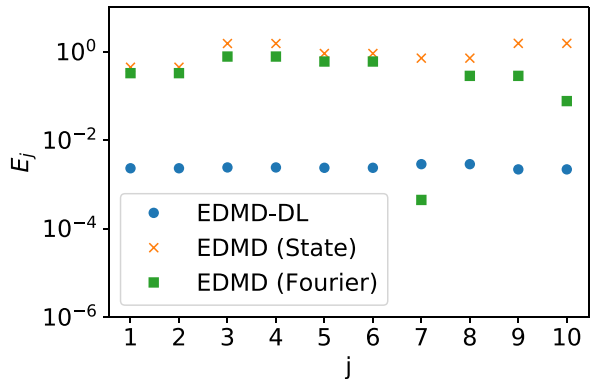

FIG. 7. Eigenfunction errors for various methods. Dictionary learning outperforms classical EDMD due to the data-adapted dictionary. Here, we used 25 dictionary elements for both EDMD-DL and EDMD with Hermite polynomial dictionaries and 100 RBFs for EDMD with the RBF dictionary.

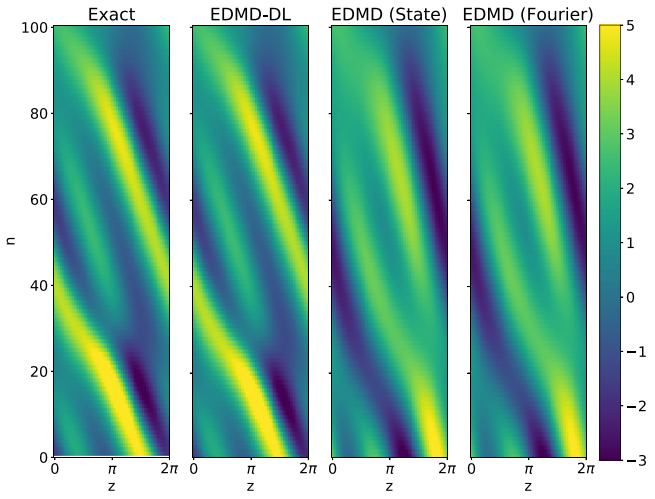

FIG. 8. Trajectory of the Kuramoto-Sivashinsky PDE reconstructed from Koopman decomposition using various algorithms. A random initial condition is picked from the same distribution as, but not from, the training data. We see that classical EDMD with both dictionaries cannot reproduce fine structures of the solution, whereas EDMD-DL performs well by adapting the dictionary to data. Also, see Fig. 4 for a quantitative comparison of the reconstruction error vs number of dictionary functions.

V. DISCUSSION

Extended Dynamic Mode Decomposition approximates the spectrum of the Koopman operator, its eigenvalues, eigenfunctions, and modes. Until now, a dictionary in the form of a set of *a priori* chosen observables of the system states was not only necessary, but carefully choosing these was crucial to the performance of the method. In highly nonlinear and high-dimensional systems, such choices are hard to make. Our main contribution is formulating a problem (7) to find an optimal (in terms of the norm of the residual) choice of dictionary given the data. This allows for a low number of optimized dictionary functions to span a linear subspace on which the Koopman operator can be accurately approximated. To realize this algorithmically, we introduced an iterative algorithm in combination with a general approximator, in the form of a neural network. This leads to much more accurate reconstructions of the Koopman operator spectrum with fewer (adapted) observation functions. Furthermore, we see that these adaptive descriptions usually have greater reconstruction accuracy over longer trajectory lengths even those exceeding the length of the training trajectories (see Fig. 3).

These desirable properties of the EDMD-DL method enable a greater range of applications of spectral analysis of nonlinear dynamical systems in general. For instance, since fewer dictionary elements are needed by EDMD-DL (see Fig. 4), it can be readily applied to obtain accurate reconstruction for high dimensional ordinary differential equation (ODE) systems or PDE systems. Moreover, this linearization technique is also useful in enabling the control theory of linear systems (which is a much studied subject) to be applied to nonlinear dynamics.[36]

The use of neural networks as dictionary approximators is also interesting on its own. Besides being a universal approximator, a neural network can also be built with certain invariance properties if so desired. For example, for applications involving spatially homogeneous PDEs, it is natural to seek eigenfunctions that are translation-invariant. Such symmetries can be built into the neural networks by considering convolution layers as their main building blocks instead of the fully connected layers considered in this paper. These convolution neural networks (CNNs) have been extensively

used in image processing and classification[37] and are likely to be highly effective in dealing with PDE systems with spatially homogeneous and local interaction terms. Moreover, CNNs are also useful in picking up multi-scale features, and hence, using CNNs as dictionary approximators is also expected to be useful in dealing with systems with dynamics that have multiple length scales.[37–40]

There are nevertheless some limitations of our proposed approach. The most important one is computational speed. In the classical EDMD algorithm, only one least-squares (or SVD) step is required to produce a decomposition. However, EDMD-DL requires the iterative solution of a sequence of least-squares problems and (stochastic) gradient descent steps. This incurs a notable computational overhead over the classical EDMD. Each (stochastic) gradient descent step (with back propagation to compute gradients) has computational complexity proportional to $d|\Theta|$, where $|\Theta|$ is the dimension of Θ, i.e., the number of trainable parameters. For moderately sized neural networks, this step is fast and the running time is dominated by the least-squares step. Hence, the computational overhead of Algorithm 1 is approximately k times that of EDMD where k is the number of least-squares evaluations to get to the prescribed error tolerance ϵ (for stochastic gradient descent (SGD) without least-squares projection, we have at best $k \sim \mathcal{O}(\epsilon^{-1})$. See, e.g., Ref. 41). For applications where it is required to perform fast Koopman decompositions (say in an online setting, where data are generated very quickly), the EDMD-DL algorithm is at a disadvantage. However, for many applications, the data-generation step is the bottle-neck, and hence, this drawback is not severe. Another potential issue is the need to choose an appropriate function approximator. Although not much fine-tuning on the neural networks was performed in this study, it is reasonable to believe that for certain specialized problems, we may obtain better performance by choosing a good network structure that reflects for example the symmetries of the underlying dynamics. Lastly, as EDMD-DL is essentially a non-linear approximation procedure[42] of an invariant subspace (as opposed to fixed EDMD with fixed dictionary elements, which can be regarded as linear approximations), it is in general difficult to derive rigorous error estimates for finite dictionary sizes.

VI. CONCLUSION AND OUTLOOK

In this paper, we combine modern machine learning approaches with the EDMD algorithm for estimating spectral decompositions of the Koopman operator. This allows us to address an important shortcoming of the EDMD algorithm, namely, the choice of a dictionary. In the EDMD-DL framework, we regard the dictionary itself as an additional optimization variable. Consequently, we can seek the optimal finite-dimensional approximation of the Koopman operator given the size of the dictionary. This allows the application of the Koopman operator framework to a broader range of problems with improved reconstruction accuracies.

There are many directions for future research. From the algorithmic point of view, the conditions which guarantee the convergence of Algorithm 1 can be studied. One can also explore variants of the algorithm with, e.g., a different regularization term or a different function approximator that may be more suited for solving specific problems. If the data are known, or suspected, to live on a low-dimensional manifold, then the relation of the number of dictionary elements found with the number of "generic observables" suggested by the Whitney, Nash, and Takens embedding theorems[43] should be both interesting and informative to explore. The stochastic counter-part to the Koopman operator is the backward Kolmogorov operator[44,45] for stochastic dynamics. It will be also interesting to apply this method to obtain spectral analysis of the evolution of expected values of observables driven by stochastic dynamics.

ACKNOWLEDGMENTS

The work of I.G.K. was partially supported by DARPA-MoDyL (HR0011-16-C-0116) and by the U.S. National Science Foundation (ECCS-1462241). I.G.K. and F.D. are grateful for the hospitality and support of the IAS-TUM. F.D. is also grateful for the support from the TopMath Graduate Center of TUM Graduate School at the Technical University of Munich, Germany, and from the TopMath Program at the Elite Network of Bavaria. E.M.B. thanks the Army Research Office (N68164-EG) and the Office of Naval Research (N00014-15-1-2093). Q.L. is grateful for the support of the Agency for Science, Technology and Research, Singapore.

[1] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," Physica D **197**, 101 (2004).
[2] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," Nonlinear Dyn. **41**, 309–325 (2005).
[3] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," Proc. Natl. Acad. Sci. U.S.A. **17**, 315–318 (1931).
[4] J. v. Neumann, "Zur operatorenmethode in der klassischen mechanik," Ann. Math. **33**, 587–642 (1932).
[5] P. R. Halmos and J. von Neumann, "Operator methods in classical mechanics, II," Ann. Math. **43**, 332–350 (1942).
[6] P. R. Halmos, P. R. Halmos, P. R. Halmos, H. Mathématicien, P. R. Halmos, and H. Mathematician, *Introduction to Hilbert Space and the Theory of Spectral Multiplicity* (Chelsea Publishing Company, Chelsea New York, 1957).
[7] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," J. Fluid Mech. **641**, 115–127 (2009).
[8] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," Chaos **22**, 047510 (2012).
[9] M. O. Williams, C. W. Rowley, I. Mezić, and I. G. Kevrekidis, "Data fusion via intrinsic dynamic variables: An application of data-driven koopman spectral analysis," Europhys. Lett. **109**, 40007 (2015).
[10] D. Giannakis, J. Slawinska, and Z. Zhao, "Spatiotemporal feature extraction with data-driven Koopman operators," J. Mach. Learn. Res. Proc. **44**, 103–115 (2015).
[11] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representation of nonlinear dynamical systems for control," PloS One **11**, e0150171 (2016).
[12] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," e-print arXiv:1703.10112.
[13] I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator," Annu. Rev. Fluid Mech. **45**, 357–378 (2013).
[14] A. S. Sharma, I. Mezić, and B. J. McKeon, "Correspondence between koopman mode decomposition, resolvent mode decomposition, and invariant solutions of the Navier-Stokes equations," Phys. Rev. Fluids **1**, 032402 (2016).

[15]M. Georgescu and I. Mezić, "Building energy modeling: A systematic approach to zoning and model reduction using koopman mode analysis," Energy Build. **86**, 794–802 (2015).
[16]H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noć, "Variational koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations," J. Chem. Phys. **146**, 154104 (2017).
[17]A. Mauroy, I. Mezić, and J. Moehlis, "Isostables, isochrons, and koopman spectrum for the action–angle representation of stable fixed point dynamics," Physica D **261**, 19–30 (2013).
[18]D. Giannakis, "Data-driven spectral decomposition and forecasting of ergodic dynamical systems," Applied and Computational Harmonic Analysis (to be published).
[19]S. Klus, N. Peter, K. Peter, W. Hao, K. Ioannis, S. Christof, and N. Frank, "Data-driven model reduction and transfer operator approximation," J. Nonlinear Sci. (unpublished).
[20]S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proc. Natl. Acad. Sci. U.S.A. **113**, 3932–3937 (2016).
[21]P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," J. Fluid Mech. **656**, 5–28 (2010).
[22]M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, "A kernel approach to data-driven koopman spectral analysis," preprint arXiv:1411.2260 (2014).
[23]M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," J. Nonlinear Sci. **25**, 1307–1346 (2015).
[24]M. Korda and I. Mezić, "On convergence of extended dynamic mode decomposition to the koopman operator," e-print arXiv:1703.04680v1.
[25]R. K. Singh and J. S. Manhas, *Composition Operators on Function Spaces* (Elsevier, 1993), Vol. 179.
[26]C. W. Rowley, see http://online.kitp.ucsb.edu/online/transturb-c17/rowley/ for "Data-driven methods for identifying nonlinear models of fluid flows" (2017).
[27]D. L. Donoho, "Compressed sensing," IEEE Trans. Inf. Theory **52**, 1289–1306 (2006).
[28]K. Karhunen, *Über lineare Methoden in der Wahrscheinlichkeitsrechnung* (Universitat Helsinki, 1947), Vol. 37.
[29]M. Loeve, *Probability Theory, Vol. II*, Graduate Texts in Mathematics (Springer-Verlag New York, 1978), Vol. 46, p. 387.
[30]A. N. Tikhonov, A. V Goncharsky, V. V. Stepanov, and A. G. Yagola, *Numerical Methods for the Solution of Ill-Posted Problems* (Springer Science and Business Media, 2013), Vol. 328.
[31]A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning* (ACM, 2004), p. 78.
[32]G. H. Golub and C. F. van Loan, *Matrix Computations* (JHU Press, 2012), Vol. 3.
[33]S. Ruder, "An overview of gradient descent optimization algorithms," e-print arXiv:1609.04747.
[34]P. Constantin, C. Foias, B. Nicolaenko, and R. Temam, *Integral Manifolds and Inertial Manifolds for Dissipative Partial Differential Equations* (Springer Science & Business Media, 2012), Vol. 70.
[35]M. Jolly, I. Kevrekidis, and E. Titi, "Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: Analysis and computations," Physica D **44**, 38–60 (1990).
[36]H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems* (Wiley-Interscience New York, 1972), Vol. 1.
[37]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.
[38]Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks* (1995), Vol. 3361, p. 1995.
[39]Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE, 2010), pp. 253–256.
[40]Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).
[41]F. Bach and E. Moulines, "Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n)," in *Advances in Neural Information Processing Systems* (2013) pp. 773–781.
[42]R. A. DeVore, "Nonlinear approximation," Acta Numer. **7**, 51–150 (1998).
[43]T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," J. Stat. Phys. **65**, 579–616 (1991).
[44]A. Kolmogoroff, "Über die analytischen methoden in der wahrscheinlichkeitsrechnung," Math. Ann. **104**, 415–458 (1931).
[45]J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," J. Comput. Dyn. **1**, 391–421 (2014).