Featurizing Koopman Mode Decomposition

David Aristoff,[1] Jeremy Copperman,[2] Nathan Mankovich,[3] and Alexander Davies[2, 4]
[1]*Colorado State University, Fort Collins, CO, 80523, USA*[a]
[2]*Oregon Health & Science University, Cancer Early Detection Advanced Research Center, Knight Cancer Institute, Portland, OR, 97201, USA*
[3]*University of Valencia, València, 46010, Spain*
[4]*Oregon Health & Science University, Division of Oncological Science, Knight Cancer Institute, Portland, OR, 97201, USA*

(Dated: 26 March 2024)

This article introduces an advanced Koopman mode decomposition (KMD) technique – coined *Featurized Koopman Mode Decomposition* (FKMD) – that uses time embedding and Mahalanobis scaling to enhance analysis and prediction of high dimensional dynamical systems. The time embedding expands the observation space to better capture underlying manifold structure, while the Mahalanobis scaling, applied to kernel or random Fourier features, adjusts observations based on the system's dynamics. This aids in *featurizing* KMD in cases where good features are not a priori known. We find that the Mahalanobis scaling from FKMD can be used for effective dimensionality reduction of alanine dipeptide data. We also show that FKMD improves predictions for a high-dimensional Lorenz attractor and a cell signaling problem from cancer research.

I. INTRODUCTION

Koopman mode decomposition[1,2] (KMD) has emerged as a powerful tool for analyzing nonlinear dynamical systems. The power of KMD comes from lifting the nonlinear dynamics into a vector space of observation functions; the evolution on this space is described the the linear *Koopman operator*[3,4]. Through this trick, KMD can identify patterns and coherent structures that evolve linearly in time.

KMD enables both quantitative predictions and qualitative analysis of a system's dynamics[5,6]. The basic framework for nonlinear features was introduced by Williams et al[5]. Since then, KMD has been kernelized[7], integrated with control theory[8], sped up with random Fourier features[9], used on time delay embedded data[10], viewed from the perspective of Gaussian processes[11], and imposed with physical constraints[12]. KMD has been widely applied, including in infectious disease control[4], video[13], neuroscience[14], fluid dynamics[15–17], molecular dynamics[18,19], and climate science[20]. For further reading on recent advances, challenges, and open problems in data-driven Koopman-based analyses see[21,22].

Kernel KMD, which uses kernel functions as features, is a natural choice when good feature functions are unknown[7,23]. The choice of kernel can have a large effect on the quality of KMD. For example, the most commonly used kernels (e.g., Gaussian) are isotropic, leading to uninformative measures of distance in high dimension. Artificial neural networks are natural competitors to KMD that can overcome this curse of dimensionality, but they cannot identify linearly evolving structures and require tuning over many hyperparameters.

We propose a novel method called *Featurized Koopman Mode Decomposition* (FKMD). Our method featurizes KMD by learning a Mahalanobis distance-based kernel[24]. This kernel prioritizes the most dynamically important directions in data, enforcing isotropic changes in space and time and potentially mitigating the curse of dimensionality. This leads to improvements over standard Gaussian kernel KMD.

FKMD includes three key ingredients: (i) a learned Mahalanobis distance-based kernel; (ii) a nonstandard time-delay embedding; and (iii) an efficient implementation with random Fourier features. Time delay embeddings[10] and random Fourier features[9] have previously been used within the KMD framework. To our knowledge, our featurization through the learned Mahalanobis matrix is new.

Through experiments, we find that both the Mahalanobis matrix and time delay embedding can be essential for robust predictions. We find that our *double* time embedding – where both the sample points and the features are embedded – is more effective than typical embeddings in KMD. The Mahalanobis matrix integrates nicely with this embedding, finding appropriate time correlation structure.

In sum, the **contributions** of this work are:

- *We introduce a new method, FKMD,* that encodes the structure of time-embedded data in a Mahalanobis matrix, leading to more effective KMD analysis and inference. We also show how to scale up to large datasets with random Fourier features[25,26].

- *We illustrate the power of FKMD* in high-dimensional experiments. The first illustrates that FKMD improves clustering for an alanine dipeptide trajectory. The second shows that FKMD allows for accurate prediction of a high-dimensional Lorenz attractor[27] when the observations are low-dimensional and noisy. Our last experiment uses cancer cell imaging to predict cell-signaling patterns hours into the future.

OVERVIEW OF KOOPMAN MODE DECOMPOSITION

We consider a dynamical system in real Euclidean space, with evolution map $\mathcal{F}_\tau$. Given the current state, $\boldsymbol{x}(t)$, the state

[a] Author to whom correspondence should be addressed: aristoff@colostate.edu

TABLE I: Definitions of symbols used in this work.

Symbol	Definition
$\boldsymbol{x}(t)$	system state at time t
$\boldsymbol{x}$, $\boldsymbol{x}'$	time embedded states, or sample points
$\boldsymbol{g}(\boldsymbol{x})$	$1 \times L$ real observation function
τ	evolution time step, or lag
$\mathcal{F}_\tau(\boldsymbol{x})$	evolution map at lag τ
$\mathcal{K}_\tau(\boldsymbol{g})$	Koopman operator at lag τ
N	number of samples
R	number of features ($R = N$ for kernel features)
$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$	time-embedded input sample sequence
$\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$	time-embedded output sequence; $\boldsymbol{y}_n = \mathcal{F}_\tau(\boldsymbol{x}_n)$
$\psi_1(\boldsymbol{x}), \ldots, \psi_R(\boldsymbol{x})$	scalar-valued feature functions
$\boldsymbol{\psi} = \begin{bmatrix}\psi_1 & \ldots & \psi_R\end{bmatrix}$	$1 \times R$ vector of feature functions
$\boldsymbol{\Psi}_{\boldsymbol{x}}$	$N \times R$ input samples × features matrix
$\boldsymbol{\Psi}_{\boldsymbol{y}}$	$N \times R$ output samples × features matrix
$\boldsymbol{K}$	$R \times R$ Koopman matrix in feature space
$\boldsymbol{B}$	$R \times L$ observation matrix in feature space
$\phi_m(\boldsymbol{x})$	scalar-valued Koopman eigenfunctions
$\boldsymbol{v}_m^*$	$1 \times L$ Koopman modes
μ_m	Koopman eigenvalues
$\lambda_m = \tau^{-1} \log \mu_m$	continuous-time Koopman eigenvalues
$k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}')$	kernel function
$\boldsymbol{M}$	Mahalanobis matrix
$\boldsymbol{I}$	identity matrix

at time τ into the future is $\mathcal{F}_\tau(\boldsymbol{x}(t))$. That is,

$$\boldsymbol{x}(t+\tau) = \mathcal{F}_\tau(\boldsymbol{x}(t)). \tag{1}$$

In realistic application problems, $\mathcal{F}_\tau$ is typically a complicated nonlinear function. However, there is a dual interpretation to equation (1) which is linear. For an observation function $\boldsymbol{g}(\boldsymbol{x})$ on the system states, the *Koopman* operator[15,21,28] determines the observations at time τ in the future:

$$\mathcal{K}_\tau(\boldsymbol{g})(\boldsymbol{x}) := \boldsymbol{g}(\mathcal{F}_\tau(\boldsymbol{x})). \tag{2}$$

While the linear framework does not remove the complexity inherent in $\mathcal{F}_\tau$, it provides a starting point for globally linear techniques: we can do linear analysis in (2) without resorting to local linearization of (1). From this point of view, we can construct finite dimensional approximations of $\mathcal{K}_\tau$ by choosing a collection of *feature functions* [29] that are evaluated at *sample points*.

To this end, we choose scalar-valued features

$$\boldsymbol{\psi}(\boldsymbol{x}) = \begin{bmatrix}\psi_1(\boldsymbol{x}) & \ldots & \psi_R(\boldsymbol{x})\end{bmatrix},$$

and obtain a set of input and output sample points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$, where $\boldsymbol{y}_n = \mathcal{F}_\tau(\boldsymbol{x}_n)$. From these we form $N \times R$ matrices $\boldsymbol{\Psi}_{\boldsymbol{x}}$ and $\boldsymbol{\Psi}_{\boldsymbol{y}}$ whose rows are samples and columns are features,

$$\boldsymbol{\Psi}_{\boldsymbol{x}} = \begin{bmatrix}\psi_1(\boldsymbol{x}_1) & \ldots & \psi_R(\boldsymbol{x}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{x}_N) & \ldots & \psi_R(\boldsymbol{x}_N)\end{bmatrix} \tag{3}$$

and

$$\boldsymbol{\Psi}_{\boldsymbol{y}} = \begin{bmatrix}\psi_1(\boldsymbol{y}_1) & \ldots & \psi_R(\boldsymbol{y}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{y}_N) & \ldots & \psi_R(\boldsymbol{y}_N)\end{bmatrix}. \tag{4}$$

A finite dimensional approximation, $\boldsymbol{K}$, of the Koopman operator should, up to estimation errors, satisfy

$$\boldsymbol{\Psi}_{\boldsymbol{x}} \boldsymbol{K} = \boldsymbol{\Psi}_{\boldsymbol{y}}. \tag{5}$$

Here $\boldsymbol{K}$ is a $R \times R$ matrix, and this is a linear system that can be solved with standard methods like ridge regression. We think of $\boldsymbol{K}$ as acting in the *feature space*.

If $\boldsymbol{g}$ is a $1 \times L$ vector-valued function, we also express $\boldsymbol{g}$ in feature space coordinates as a $R \times L$ matrix $\boldsymbol{B}$:

$$\boldsymbol{\Psi}_{\boldsymbol{x}} \boldsymbol{B} = \begin{bmatrix}\boldsymbol{g}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{x}_N)\end{bmatrix}. \tag{6}$$

Note that (6) can be solved in the same manner as (5).

Koopman mode decomposition converts an eigendecomposition of $\boldsymbol{K}$ back to the *sample space*, in order to interpret and/or predict the dynamics defined by $\mathcal{F}_\tau$. To this end, write the eigendecomposition of $\boldsymbol{K}$ as

$$\boldsymbol{K} = \sum_{m=1}^{R} \mu_m \boldsymbol{\xi}_m \boldsymbol{w}_m^*, \tag{7}$$

where μ_m are the eigenvalues of $\boldsymbol{K}$, and $\boldsymbol{\xi}_m$, $\boldsymbol{w}_m$ are the right and left eigenvectors, respectively, scaled so that $\boldsymbol{w}_m^* \boldsymbol{\xi}_m = 1$. That is, $\boldsymbol{K}\boldsymbol{\xi}_m = \mu_m \boldsymbol{\xi}_m$ and $\boldsymbol{w}_m^* \boldsymbol{K} = \mu_m \boldsymbol{w}_m^*$. By converting this to sample space, it can be shown that[5]

$$\mathcal{K}_\tau(\boldsymbol{g})(\boldsymbol{x}) \approx \sum_{m=1}^{R} e^{\tau\lambda_m} \phi_m(\boldsymbol{x}) \boldsymbol{v}_m^*, \tag{8}$$

with $e^{\tau\lambda_m} = \mu_m$ the *Koopman eigenvalues*, $\phi_m(\boldsymbol{x}) = \boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\xi}_m$ the *Koopman eigenfunctions*, and $\boldsymbol{v}_m^* = \boldsymbol{w}_m^* \boldsymbol{B}$ the *Koopman modes*. The right hand side of (8) is a finite dimensional approximation of the Koopman operator. See Appendix A for a derivation of equation (8).

With the Koopman eigenvalues, Koopman eigenfunctions, and Koopman modes in hand, equation (8) can be used to predict observations of the system at future times, as well as analyze qualitative behavior. There has been much work in this direction; we will not give a complete review, but refer to[5,30] for the basic ideas and to[18–20,31–33] for recent applications and extensions. Of course, the quality of the approximation in (8) is sensitive to the choice of features and sample space. With enough features and samples, actual equality in (8) can be approached[30]. In realistic applications, samples and features are limited by computational constraints.

II. METHODS

A. Overview

We make two data-driven choices which can give remarkably good results on complex systems. These choices are:

(i) We learn a linear map $\boldsymbol{x} \to \boldsymbol{M}^{1/2}\boldsymbol{x}$ to help define features. The *Mahalanobis matrix*, $\boldsymbol{M}$, is updated iteratively and reflects the underlying system's dynamics.

(ii) We use a a *double time embedded* structure to construct feature space. Both the sample points $\boldsymbol{x}_n$ and the features ψ_m are time embedded – that is, both are defined using a time sequence.

Our features are based on kernels[20,33]. Kernel features are a common choice when good feature functions are not *a priori* known. The kernels are centered around the sample points,

$$\psi_m(\boldsymbol{x}) = k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}_m), \qquad m = 1, \ldots, R. \tag{9}$$

Here, $R = N$ and $k_{\boldsymbol{M}}$ is the kernel function

$$k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left[-(\boldsymbol{x}-\boldsymbol{x}')^* \boldsymbol{M}(\boldsymbol{x}-\boldsymbol{x}')\right]. \tag{10}$$

We also use random Fourier features that estimate these kernels, allowing for larger sample size N; see Section II F.

B. Mahalanobis matrix

Inspired by the recent work[24] on understanding neural networks and improving kernel methods, we target a matrix $\boldsymbol{M}$ using a gradient outer product structure[24,34,35]. Up to a scalar bandwidth factor, we use

$$\boldsymbol{M} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{J}(\boldsymbol{x}_n)\boldsymbol{J}(\boldsymbol{x}_n)^*, \tag{11}$$

with the ideal $\boldsymbol{J}$ given by

$$\boldsymbol{J}(\boldsymbol{x}) = \lim_{\tau\to 0} \tau^{-1}[\nabla(\boldsymbol{g}\circ\mathcal{F}_\tau)(\boldsymbol{x}) - \nabla\boldsymbol{g}(\boldsymbol{x})]. \tag{12}$$

Compared to a standard Gaussian kernel, (10) comes from the change of variables $\boldsymbol{x} \mapsto \tilde{\boldsymbol{x}} = \boldsymbol{M}^{1/2}\boldsymbol{x}$. Correspondingly, let $\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}) = \boldsymbol{g}(\boldsymbol{x})$ and $\tilde{\mathcal{F}}_\tau(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{y}}$, where $\boldsymbol{y} = \mathcal{F}_\tau(\boldsymbol{x})$, and

$$\tilde{\boldsymbol{J}}(\boldsymbol{x}) = \lim_{\tau\to 0} \tau^{-1}[\nabla(\tilde{\boldsymbol{g}}\circ\tilde{\mathcal{F}}_\tau)(\boldsymbol{x}) - \nabla\tilde{\boldsymbol{g}}(\boldsymbol{x})]. \tag{13}$$

This *featurization* mapping, *i.e.*, $\boldsymbol{x} \mapsto \tilde{\boldsymbol{x}}$, $\boldsymbol{g} \mapsto \tilde{\boldsymbol{g}}$, and $\mathcal{F}_\tau \mapsto \tilde{\mathcal{F}}_\tau$, assumes that the input/output pairs are mapped by $\boldsymbol{M}^{1/2}$, but that the observations do not change. The matrices $\boldsymbol{J}$ and $\tilde{\boldsymbol{J}}$ measure infinitesimal changes in space and time of the original and transformed variables respectively. Theorem II.1 below shows that these changes are isotropic in the transformed variables (proof is in Appendix B).

Theorem II.1. *With $\boldsymbol{M}$ defined by* (11)-(12)*,*

$$\frac{1}{N}\sum_{n=1}^{N} \left|\boldsymbol{u}^*\tilde{\boldsymbol{J}}(\tilde{\boldsymbol{x}}_n)\right|^2 \equiv 1, \quad \text{for all unit } \boldsymbol{u}. \tag{14}$$

In particular, if $\boldsymbol{x}(t)$ satisfies a linear ODE driven by a real invertible matrix $\boldsymbol{A}$, then $\boldsymbol{J} = \boldsymbol{A}$ while $\tilde{\boldsymbol{J}} = (\boldsymbol{A}\boldsymbol{A}^*)^{-1/2}\boldsymbol{A}$ is an orthogonal matrix.

In practice, we compute $\boldsymbol{J}$ using

$$\boldsymbol{J}(\boldsymbol{x}) \approx \sum_{m=1}^{R} \lambda_m \nabla\phi_m(\boldsymbol{x})\boldsymbol{v}_m^*, \tag{15}$$

and iteratively improve both $\boldsymbol{J}$ and $\boldsymbol{M}$ using Algorithm II.2. Though $\boldsymbol{J}$ here may not be real, we can just replace $\boldsymbol{M}$ by its real part without changing $k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}')$.

C. Time embeddings

Time embeddings are useful for high dimensional systems that are only partially observed, and have a theoretical basis in Taken's theorem[10,36]. Recent work[10] has used time embeddings to construct larger matrices in (3)-(4) with Hankel structure[30]; our setup is different because time embedded data goes directly into the features, leading to smaller matrices in (3)-(4) while improving distance measurements.

Here, we define samples as time embeddings of length ℓ,

$$\begin{aligned} \boldsymbol{x}_{n+1} &= \left[\boldsymbol{x}(n\tau) \ \ldots \ \boldsymbol{x}((n+\ell-1)\tau)\right] \\ \boldsymbol{y}_{n+1} &= \left[\boldsymbol{x}((n+1)\tau) \ \ldots \ \boldsymbol{x}((n+\ell)\tau)\right], \end{aligned} \tag{16}$$

where $\boldsymbol{x}(0)$ is some initial state. The evolution map $\mathcal{F}_\tau$ extends to such states in a natural way, and the associated Koopman operator is then defined on functions of time embedded states. From here on, we abuse notation by writing $\boldsymbol{x}$ or $\boldsymbol{x}'$ for a time embedding (or sample) of the form (16).

Note that this gives both the samples (16) and features (9) a time-embedded structure. Increasing the embedding length of the samples enables recovery of the underlying manifold[10,36]. Moreover, it improves distance measurements when the input/output pairs are corrupted by additive noise, as is easily seen from the law of large numbers (assuming noise correlations decay sufficiently fast in time).

D. The FKMD Algorithm

We summarize our algorithm below, which we call Featurized Koopman Mode Decomposition (FKMD).

Algorithm II.2 (FKMD). *Generate samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ according to* (16)*, and choose bandwidth $h > 0$ and initial Mahalanobis matrix $\boldsymbol{M} = \boldsymbol{I}$. Then, iterate the following steps until approximate convergence:*

1. *Let σ = standard deviation of the pairwise distances between $\boldsymbol{M}^{1/2}\boldsymbol{x}_1, \ldots, \boldsymbol{M}^{1/2}\boldsymbol{x}_N$. Scale $\boldsymbol{M} \leftarrow \boldsymbol{M}/(h\sigma)^2$.*

2. *Construct $\boldsymbol{\Psi}_{\boldsymbol{x}}$ and $\boldsymbol{\Psi}_{\boldsymbol{y}}$ defined in* (3)-(4)*, using features defined either by* (9)- (10) *or by* (17)- (18).

3. *Solve for $\boldsymbol{K}$ and $\boldsymbol{B}$ in* (5)-(6), e.g. *via ridge regression.*

4. *Eigendecompose $\boldsymbol{K}$ according to* (7)*. That is, compute right and left eigenvectors $\boldsymbol{\xi}_m$ and $\boldsymbol{w}_m$ of $\boldsymbol{K}$, along with eigenvalues μ_m. Scale them so that $\boldsymbol{w}_m^*\boldsymbol{\xi}_m = 1$.*

5. *Compute continuous time Koopman eigenvalues, Koopman eigenfunction, and Koopman modes, using*

$$\lambda_m = \tau^{-1} \log \mu_m, \quad \phi_m(\boldsymbol{x}) = \psi(\boldsymbol{x})\boldsymbol{\xi}_m, \quad \boldsymbol{v}_m^* = \boldsymbol{w}_m^* \boldsymbol{B}.$$

6. *Update $\boldsymbol{M}$ using* (11) *and* (15). *Then return to Step 1.*

Note that observations can be predicted using (8) at any iteration of Algorithm II.2. We find empirically that convergence of $\boldsymbol{M}$ and the predictions occurs after 3-6 iterations.

Algorithm II.2 requires only a few user chosen parameters: an embedding length ℓ, a bandwidth h, and a number of features R (not counting regularization parameters or possible cutoffs and subsampling parameters, discussed below). This is a significant advantage over artificial neural network methods, which often require searching over a much larger set of hyperparameters[37], and a training procedure that is not guaranteed to converge to an optimal parameter set[38–40].

E. Tuning

The initial $\boldsymbol{M}$ could be chosen using information about the system. In the absence of that, we use a scalar multiple of $\boldsymbol{I}$ as described in Algorithm II.2. Subsampling may be used to estimate $\boldsymbol{M}$ and σ. If results degrade with iterations, we recommend adding a small ridge regularization to $\boldsymbol{M}$ by updating $\boldsymbol{M} \leftarrow \boldsymbol{M} + \delta \boldsymbol{I}$ after Step 6, where $\delta > 0$ is a small parameter. Note that $\boldsymbol{M}$ could be replaced by its real part after Step 6 without changing $k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}')$; this makes $\boldsymbol{M}$ into a symmetric positive semidefinite matrix, a necessary step when using random Fourier features (Section II F).

Note that equation (8) naturally allows for mode selection. For example, modes that lead to predictions that are known to be unphysical, *e.g.* diverging modes associated to eigenvalues with $\text{Re}(\lambda_m) \gg 0$, can simply be omitted from the sum in (8). Similar selection can be done to remove modes that oscillate too fast, *i.e.*, $|\text{Im}(\lambda_m)| \gg 0$. We have done this when applying equation (A1) in the experiments in Section III C.

We have also noticed empirically that estimates of $\boldsymbol{M}$ can suffer from noise effects if too many modes are used. We find good results by using only top modes according to some cutoff in (15); *e.g.*, removing modes with $\text{Re}(\lambda_m) < -\gamma$, where $\gamma > 0$ is some threshold. Intuitively, this means eliminating effects from the shortest timescales. In the experiments in Sections III A-III C below, we choose cutoffs using cross-validation.

F. Scaling up to larger sample size

Kernel methods have historically been limited by the computational complexity of large linear solves[41] (usually limiting sample size to $N \le 10^5$), as well as the difficulty of choosing good features to mitigate the curse of dimensionality. Here, we show how to scale FKMD to large sample size N.

To this end, we use random Fourier features[9,25,26,42,43]

$$\psi_m^{RFF}(\boldsymbol{x}) = \exp(i\boldsymbol{\omega}_m^T \boldsymbol{M}^{1/2}\boldsymbol{x}), \qquad m = 1, \ldots, R. \tag{17}$$

Here, $\boldsymbol{\omega}_m$ are iid Gaussians with mean $\boldsymbol{0}$ and covariance $\boldsymbol{I}$,

$$\boldsymbol{\omega}_m \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{18}$$

and $\boldsymbol{M}$ is symmetric positive semidefinite.

We expect good results with $R \ll N$; this leads to much more efficient linear solves and eigendecompositions. The features (17)-(18) essentially target the same linear system (5) as the kernel features, but they do it more efficiently by sampling. See Appendix C for details.

III. EXPERIMENTS

A. Alanine dipeptide

We illustrate Algorithm II.2 on alanine dipeptide[44], a small protein. Our input/output sequences are taken from the time series data in[45]. The sample points represent the 30 dimensional positions of heavy atoms, which are known to be a sufficient description of the system. We take $N = 10^6$, $R = 8000$, $\ell = 1$, and we compute $\boldsymbol{M}$ from the top 20 modes. We estimate σ and $\boldsymbol{M}$ using a random subsample of 5000 points. The algorithm converges in ≈ 4 iterations.

Figure 1 show that $\boldsymbol{M}^{1/2}$ featurizes the data in a way that is superior to a basic technique like PCA. Figure 1(a) shows the Mahalanobis matrix after convergence of FKMD. Figure 1(b) indicates that $\boldsymbol{M}^{1/2}$ maps into a 6-dimensional subspace of the 30 dimensional space of heavy atom positions. The (square root) of the data covariance, $\boldsymbol{C}^{1/2}$, does not map into low dimension quite as nicely. Figure 1(c) shows eigenvalues of a Markov model built using 100 k-means clusters from data mapped by $\boldsymbol{M}^{1/2}$, compared to the same number of clusters from PCA-projected data with 95% of the variance retained. The cluster of 4 eigenvalues near 1 suggests 3 long timescales. Figure 1(d) shows the corresponding implied timescales; using $\boldsymbol{M}^{1/2}$ leads to a better Markov model (capturing shorter timescales) due to faster flattening of these timescales[46].

There are several standard featurization techniques for constructing Markov models, including time-lagged independent component analysis (TICA)[47,48], the variational approach for conformational dynamics (VAC)[49,50], and VAMPnets[51]. We are not suggesting that FKMD should replace any of these methods. Our goal here is simply to use a well-known metric – implied timescales – to illustrate that the $\boldsymbol{M}^{1/2}$ mapping can reliably featurize high dimensional data.

B. Lorenz attractor

Here, we illustrate Algorithm II.2 on data from the Lorenz 96 model[27], a high-dimensional ODE exhibiting chaotic behavior. This model (and its 3-dimensional predecessor[52]) are often used to interpret atmospheric convection and to test tools in climate analysis[53]. The model is

$$\frac{d\theta_j}{dt} = (\theta_{j+1} - \theta_{j-2})\theta_{j-1} - \theta_j + F,$$

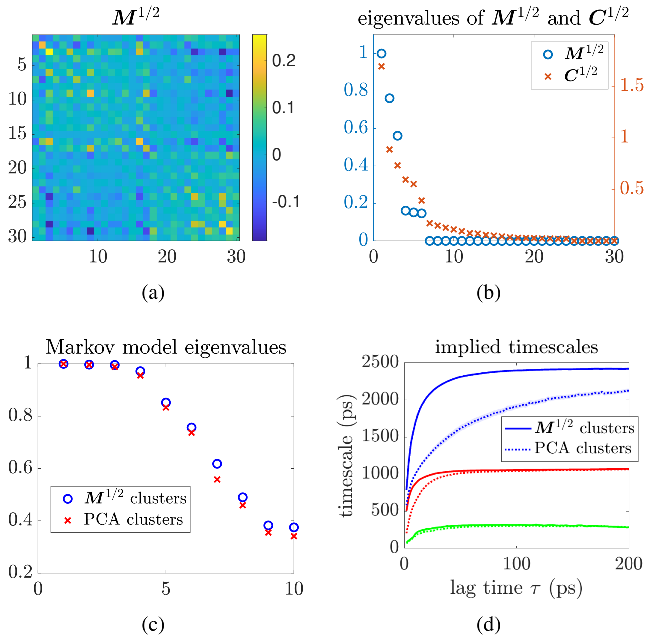


FIG. 1: For alanine dipeptide, mapping by $M^{1/2}$ improves clustering compared to PCA projection. (a) $M^{1/2}$ after convergence of FKMD. (b) Eigenvalues of $M^{1/2}$ after convergence of Algorithm II.2 compared to eigenvalues of the data covariance matrix. (c) Eigenvalues of the Markov model at the smallest lag time. (d) Largest implied timescales when using $M^{1/2}$ (solid lines) and PCA (dotted lines). Confidence regions from 25 independent simulations are mostly too small to see.

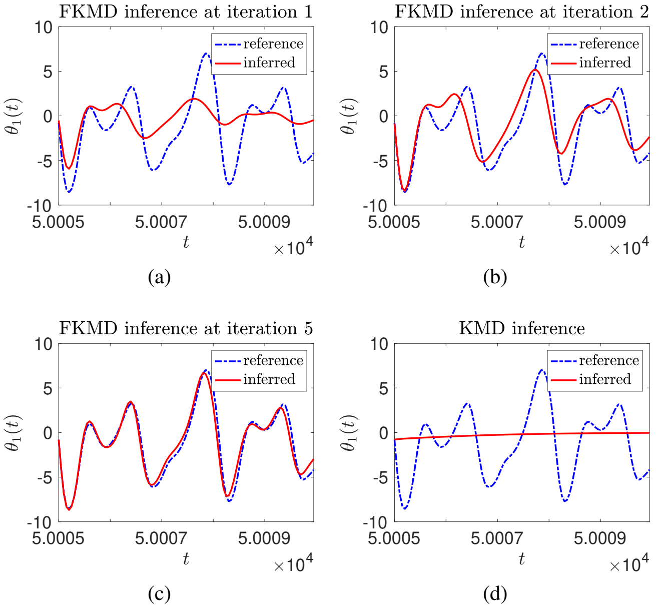


FIG. 2: Illustration of FKMD for predicting a single observed coordinate of a noisy high-dimensional Lorenz system. (a)-(c): FKMD at the 1st, 2nd and 5th iterations, respectively. Results do not change much after 5 iterations. (d): Ordinary KMD does not correlate with the data, even when nuisance coordinates are not included.

with $j = 1,\ldots,40$ periodic coordinates ($j \equiv j \bmod 40$). We set $F = 8$, and integrate using 4th order Runge-Kutta[54] with integrator time step 10^{-2}. The initial condition is[53]

$$\theta_j(0) = \begin{cases} F+1, & j \bmod 5 = 0 \\ F, & \text{else} \end{cases}.$$

To illustrate the power of FKMD, we observe just 2.5% of the system, namely the first coordinate θ_1, and we add nuisance or "noise" variables. Specifically, we use the time embedding (16) with $\tau = 0.05$ and

$$\boldsymbol{x}(n\tau) = \begin{bmatrix} \theta_1(n\tau) & \text{noise}(n\tau) \end{bmatrix}, \qquad (19)$$

where noise$(n\tau)$ for $n = 0,1,2,\ldots$ are independent standard Gaussian random variables. To infer $\theta_1(t)$ from training data, we use Algorithm II.2 with random Fourier features defined in (17)-(18). Inference begins at the end of the training set and consists of 100 discrete steps of time length τ. The code for this experiment is available here[55].

For the FKMD parameters, we use $N = 10^6$ sample points, $R = 8000$ features, a time embedding of length $\ell = 100$, and a constant bandwidth factor $h = 1$. The observation $\boldsymbol{g}(\boldsymbol{x})$ is a 1×200 vector associated with time embeddings of (19) as defined in (16). Results are plotted in Figure 2. If any one of N, R, or ℓ is decreased, results degrade noticeably on the time horizon we use for inference. We use the top 20 modes to define $\boldsymbol{M}$, and we estimate $\boldsymbol{\sigma}$ and $\boldsymbol{M}$ using a random subsample of 5000 points.

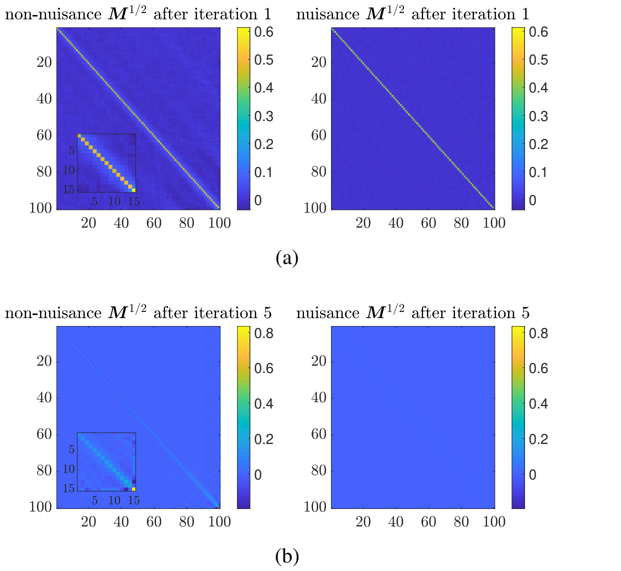
Figure 2(a)-(c) shows inference using equation (8). FKMD converges in about 5 iterations, providing a very close match to the actual data. Figure 2(d) shows ordinary KMD. (Ordinary KMD corresponds to Algorithm II.2 with all the same parameters except $\ell = 1$, $\boldsymbol{M}$ is a scalar, iteration is unnecessary, and nuisance variables are not included.) In this experiment, ordinary KMD is not able to make good predictions.

The Mahalanobis matrices after the 1st and 5th iterations are shown in Figure 3(a)-(b). We split the mapping $\boldsymbol{M}^{1/2}$ into nuisance and non-nuisance parts based on (19). The $\boldsymbol{M}^{1/2}$ mapping eliminates the nuisance coordinates, while preserving the structure of the underlying signal.

C. Cell signaling dynamics

In a real-world data-driven setting, complex and potentially noisy temporal outputs derived from measurement may not obey a simple underlying ODE or live on a low-dimensional dynamical attractor. Information contained by internal signaling pathways within living cells is one such example, being complex and subject to noisy temporal outputs arising from properties of the system itself and experimental sources.

With this in mind, we next apply FKMD to dynamic signaling activity in cancer cells to assess its performance. We show



(a)

(b)

FIG. 3: The Mahalanobis mapping $\boldsymbol{M}^{1/2}$ after the 1st and 5th iterations. To aid in visualization, we have split the matrix into nuisance coordinates at the right and non-nuisance coordinates at the left. (The insets at left are the bottom right 15×15 submatrices of non-nuisance coordinates.) The matrix $\boldsymbol{M}^{1/2}$ maps away the nuisance coordinates, and finds appropriate structure in the non-nuisance coordinates.

that our methods enable the forward prediction of single-cell signaling activity from past knowledge in a system where signaling is highly variable from cell to cell and over time[56]. The extracellular signal-regulated kinases (ERK) signaling pathway is critical for the perception of cues outside of cells and for translation of these cues into cellular behaviors such as changes in cell shape, proliferation rate, and phenotype[57]. Dynamic ERK activity is monitored via the nuclear or cytoplasmic localization of the fluorescent reporter (Figure 4A). We track single-cells through time in the live-cell imaging yielding single-cell ERK activity time series (Figure 4C). The first 72 hours of single-cell trajectories serve as the training set to estimate the Koopman operator, and we withhold the final 18 hours of the single-cell trajectories to test the predictive capability of FKMD. The raw ERK activity trajectories on their own yield no predictive capability via standard Kernel DMD methods, but our iterative procedure to extract the Mahalanobis matrix leads to a coordinate rescaling which couples signaling activity across delay times (Figure 4B,D) and enables a forward prediction of ERK activity across the testing window (Figure 4D).

We use $N = 5202$ samples and kernel features with $R = N$, and we choose bandwidth $h = 1.05$ and a time embedding of length $\ell = 49$. The function $\boldsymbol{g}(\boldsymbol{x})$ is a 1×49 time embedding of the scalar ERK activity. For inference, we exclude modes where $\mathrm{Re}(\lambda_m) > 0.15$ and $|\mathrm{Im}(\lambda_m)| > \pi/3$. This amounts to excluding unstable modes and modes that oscillate quickly. We use the remaining modes to construct $\boldsymbol{M}$. Prediction quality is quantified by estimating the relative error and correlation between inferred and test set ERK activity trajectories.

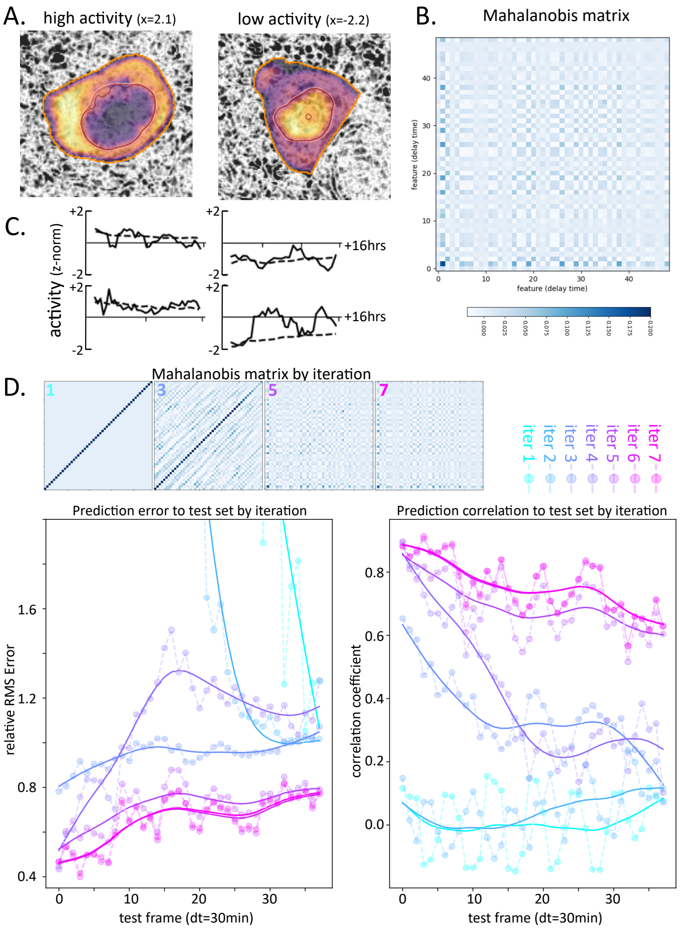


FIG. 4: FKMD enabled cell signaling state prediction. (A) Fluorescent ERK reporter expressing breast cancer cell embedded in a mammary tissue organoid, showing representative high activity (left, cytoplasmic localized) and low activity (right, nucleus localized). (B) Malahanobis matrix at iteration 8, where test set correlation is maximized. (C) Representative single-cell ERK activity traces, measured test set (solid lines), and FKMD-predicted (dashed lines). (D) Mahalanobis matrix by FKMD iteration (top), and FKMD prediction performance from 0-18hrs quantified by relative root mean squared (RMS) error and correlation to test set (bottom left and right) by iteration number (cyan to magenta circles), solid lines are spline fits added as guides to the eye.

IV. DISCUSSION AND FUTURE WORK

This article introduces FKMD, a method we propose that generates more accurate predictions than ordinary Gaussian kernel KMD. The method is based on time embeddings and a Mahalanobis featurization that mitigates the curse of dimensionality. Results in three separate application areas – molecular dynamics, climate, and cell signaling dynamics – illustrate the promise of the method for analyzing time series generated by complex systems.

Many theoretical and algorithmic questions remain. Empirically, we have found that a few iterations and modes lead to good results, but more empirical testing is needed, and our theoretical understanding of these issues is lacking. For example, we cannot yet describe a simple set of conditions that guarantees good behavior of Algorithm II.2, like convergence to a fixed point.

We would also like to explore alternative methods for scaling up FKMD to larger sample sizes. Here, we use random Fourier features, but there are other possibilities. Some come from modern advances in randomized numerical linear algebra, *e.g.* randomly pivoted Cholesky[41,58]. Such methods promise spectral efficiency for solving symmetric positive definite linear systems. Assuming fast spectral decay of $\boldsymbol{\Psi}_{\boldsymbol{x}}$, these techniques could help our methods scale to even larger sample sizes. We will explore the application of these cutting-edge methods in future works.

Finally, we would like to better understand *mechanisms*, *i.e.*, what makes a system go to B rather than A? Here, A and B could be associated with El Niño occurring or not, or with a pre-cancerous lesion remaining benign or becoming invasive. This can be formalized by the concept of the *committor function*[59], the probability that the system reaches state B before A from a given starting state. In this case, the committor could be represented as a Koopman eigenfunction, and $\boldsymbol{M}$ could be chosen to identify important mechanisms leading to A or B. We hope to explore this idea in future work.

ACKNOWLEDGMENTS

D. Aristoff gratefully acknowledges support from the National Science Foundation via Award No. DMS 2111277. N. Mankovich acknowledges the support of Generalitat Valenciana and the Conselleria d'Innovació, Universitats, Ciència i Societat Digital, through the project "AI4CS: Artificial Intelligence for complex systems: Brain, Earth, Climate, Society" (CIPROM/2021/56). This project was supported in part by funding (2023-1713) from the Cancer Early Detection Advanced Research Center at Oregon Health & Science University's Knight Cancer Institute (Jeremy Copperman and Alexander Davies). Alexander Davies gratefully acknowledges support from NIH/ORIP award K01OD031811.

Appendix A: Derivation of Koopman eigendecomposition

Here, we show how to arrive at the Koopman eigendecomposition (8). This has been shown already in[5], but we provide a streamlined derivation here for convenience.

Recall that the matrix $\boldsymbol{K}$ is a finite dimensional approximation to the Koopman operator. This approximation is obtained by applying a change of variables from sample space to feature space. The change of variables is given by the matrix $\boldsymbol{\Psi}_{\boldsymbol{x}}$. This leads to the following equation for inference:

$$\begin{bmatrix} \mathcal{K}_\tau(\boldsymbol{g})(\boldsymbol{x}_1) \\ \vdots \\ \mathcal{K}_\tau(\boldsymbol{g})(\boldsymbol{x}_N) \end{bmatrix} \approx \boldsymbol{\Psi}_{\boldsymbol{x}} \boldsymbol{K} \boldsymbol{\Psi}_{\boldsymbol{x}}^\dagger \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{x}_N) \end{bmatrix}, \tag{A1}$$

where † denotes the Moore-Penrose pseudoinverse. Similarly,

$$\boldsymbol{B} = \boldsymbol{\Psi}_{\boldsymbol{x}}^\dagger \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{x}_N) \end{bmatrix}.$$

The eigendecomposition of $\boldsymbol{K}$ can be written as

$$\boldsymbol{K} = \Xi \boldsymbol{D} \boldsymbol{W}^*, \tag{A2}$$

where $\boldsymbol{K}\Xi = \Xi\boldsymbol{D}$ and $\boldsymbol{W}^*\boldsymbol{K} = \boldsymbol{D}\boldsymbol{W}^*$, and we may assume that $\boldsymbol{W}^*\Xi = \boldsymbol{I}$. Here, $\boldsymbol{D}$ is the diagonal matrix of Koopman eigenvalues, μ_m; that is, $\boldsymbol{D} = \exp(\tau\boldsymbol{\Lambda})$ where $\boldsymbol{\Lambda}$ is the diagonal matrix of continuous time Koopman eigenvalues, λ_m.

Plugging (A2) into (A1),

$$\begin{bmatrix} \mathcal{K}_\tau(\boldsymbol{g})(\boldsymbol{x}_1) \\ \vdots \\ \mathcal{K}_\tau(\boldsymbol{g})(\boldsymbol{x}_N) \end{bmatrix} \approx \boldsymbol{\Psi}_{\boldsymbol{x}} \Xi \exp(\tau\boldsymbol{\Lambda}) \boldsymbol{W}^* \boldsymbol{B}. \tag{A3}$$

The definition of Koopman modes and Koopman eigenfunctions shows that the rows of $\boldsymbol{W}^*\boldsymbol{B}$ are the Koopman modes $\boldsymbol{v}_m^*$, while the Koopman eigenfunctions are sampled by the columns of

$$\boldsymbol{\Psi}_{\boldsymbol{x}}\Xi = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \dots & \phi_R(\boldsymbol{x}_1) \\ \vdots & & \vdots \\ \phi_1(\boldsymbol{x}_N) & \dots & \phi_R(\boldsymbol{x}_N) \end{bmatrix}. \tag{A4}$$

Substituting (A4) into (A3) and writing the matrix multiplication in terms of outer products yields equation (8), provided we substitute $\boldsymbol{x}_n$ for $\boldsymbol{x}$, using any sample point $\boldsymbol{x}_n$.

Appendix B: Choice of Mahalanobis matrix

Here, we explain the reasoning behind the choice of Mahalanobis matrix in more detail. Recall that the matrix defines a change of variables, $\tilde{\boldsymbol{x}} = \boldsymbol{M}^{1/2}\boldsymbol{x}$, where the tilde notation indicates the changed variables. Below, we assume that $\boldsymbol{M}$ is symmetric positive definite.

Write $\tilde{\boldsymbol{x}}_n = \boldsymbol{M}^{1/2}\boldsymbol{x}_n$, $\tilde{\boldsymbol{y}}_n = \boldsymbol{M}^{1/2}\boldsymbol{y}_n$, and

$$\tilde{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{M}^{-1/2}\boldsymbol{x}), \quad \tilde{\mathcal{F}}_\tau(\boldsymbol{x}) = \boldsymbol{M}^{1/2}\mathcal{F}_\tau(\boldsymbol{M}^{-1/2}\boldsymbol{x}).$$

Observe that then $\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}) = \boldsymbol{g}(\boldsymbol{x})$ and $\tilde{\mathcal{F}}_\tau(\tilde{\boldsymbol{x}}_n) = \tilde{\boldsymbol{y}}_n$. We summarize these notations and mappings in Figure 5. Define

$$\boldsymbol{J}(\boldsymbol{x}) = \lim_{\tau\to 0} \tau^{-1}[\nabla(\boldsymbol{g}\circ\mathcal{F}_\tau)(\boldsymbol{x}) - \nabla\boldsymbol{g}(\boldsymbol{x})],$$
$$\tilde{\boldsymbol{J}}(\boldsymbol{x}) = \lim_{\tau\to 0} \tau^{-1}[\nabla(\tilde{\boldsymbol{g}}\circ\tilde{\mathcal{F}}_\tau)(\boldsymbol{x}) - \nabla\tilde{\boldsymbol{g}}(\boldsymbol{x})].$$

$$\begin{tikzcd}
\boldsymbol{x}_n \arrow[r, "\mathcal{F}_\tau"] \arrow[d, "\boldsymbol{M}^{1/2}"] \arrow[dd, bend right=60, "\boldsymbol{g}"'] & \boldsymbol{y}_n \arrow[d, "\boldsymbol{M}^{1/2}"'] \arrow[dd, bend left=60, "\boldsymbol{g}"] \\
\tilde{\boldsymbol{x}}_n \arrow[r, "\tilde{\mathcal{F}}_\tau"] \arrow[d, "\tilde{\boldsymbol{g}}"] & \tilde{\boldsymbol{y}}_n \arrow[d, "\tilde{\boldsymbol{g}}"'] \\
\boldsymbol{g}(\boldsymbol{x}_n) & \boldsymbol{g}(\boldsymbol{y}_n)
\end{tikzcd}$$

FIG. 5: A figure summarizing the "tilde" notation for B.1.

The next result, Proposition B.1, justifies our choice of $\boldsymbol{M}$. It shows that the changes in space and time in the transformed variables, as measured by $\tilde{\boldsymbol{J}}$, are isotropic. This is a general result that is true regardless of kernel choice and other KMD hyperparameters. Below, *we implicitly assume appropriate smoothness so that all the calculations make sense.*

Proposition B.1. *With $\boldsymbol{M}$ defined by* (11)-(12)*,*

$$\frac{1}{N}\sum_{n=1}^{N}\left|\boldsymbol{u}^*\tilde{\boldsymbol{J}}(\tilde{\boldsymbol{x}}_n)\right|^2 \equiv 1, \quad \text{for unit } \boldsymbol{u}. \tag{B1}$$

Proof. By the chain rule,

$$\begin{aligned}
&\nabla(\tilde{\boldsymbol{g}}\circ\tilde{\mathcal{F}})(\tilde{\boldsymbol{x}}_n)\\
&= \nabla\tilde{\mathcal{F}}_\tau(\tilde{\boldsymbol{x}}_n)\nabla\tilde{\boldsymbol{g}}(\tilde{\mathcal{F}}_\tau(\tilde{\boldsymbol{x}}_n))\\
&= \boldsymbol{M}^{-1/2}\nabla\mathcal{F}_\tau(\boldsymbol{M}^{-1/2}\tilde{\boldsymbol{x}}_n)\boldsymbol{M}^{1/2}\boldsymbol{M}^{-1/2}\nabla\boldsymbol{g}(\boldsymbol{M}^{-1/2}\tilde{\boldsymbol{y}}_n)\\
&= \boldsymbol{M}^{-1/2}\nabla\mathcal{F}_\tau(\boldsymbol{x}_n)\nabla\boldsymbol{g}(\boldsymbol{y}_n)\\
&= \boldsymbol{M}^{-1/2}\nabla(\boldsymbol{g}\circ\mathcal{F}_\tau)(\boldsymbol{x}_n).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\nabla\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}_n) &= \boldsymbol{M}^{-1/2}\nabla\boldsymbol{g}(\boldsymbol{M}^{-1/2}\tilde{\boldsymbol{x}}_n)\\
&= \boldsymbol{M}^{-1/2}\nabla\boldsymbol{g}(\boldsymbol{x}_n).
\end{aligned}$$

As a result,

$$\tilde{\boldsymbol{J}}(\tilde{\boldsymbol{x}}_n) = \boldsymbol{M}^{-1/2}\boldsymbol{J}(\boldsymbol{x}_n).$$

It follows that

$$\begin{aligned}
&\frac{1}{N}\sum_{n=1}^{N}\left|\boldsymbol{u}^*\tilde{\boldsymbol{J}}(\tilde{\boldsymbol{x}}_n)\right|^2\\
&= \frac{1}{N}\sum_{n=1}^{N}\left|\boldsymbol{u}^*\boldsymbol{M}^{-1/2}\boldsymbol{J}(\boldsymbol{x}_n)\right|^2\\
&= \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{u}^*\boldsymbol{M}^{-1/2}\boldsymbol{J}(\boldsymbol{x}_n)\boldsymbol{J}(\boldsymbol{x}_n)^*\boldsymbol{M}^{-1/2}\boldsymbol{u} \equiv 1.
\end{aligned}$$

□

The following special case in Proposition B.2 is helpful for intuition. It states that this change of variables applied to a *linear* ODE makes that ODE appear to be driven by an orthogonal matrix.

Proposition B.2. *Suppose that $\mathcal{F}_\tau$ is the evolution map of a linear ODE driven by a real invertible matrix $\boldsymbol{A}$,*

$$\frac{d\boldsymbol{x}(t)^*}{dt} = \boldsymbol{x}(t)^*\boldsymbol{A},$$

and that the observation is the whole state, $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}^*$. Then $\boldsymbol{J} = \boldsymbol{A}$, $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{A}^*$, and $\tilde{\boldsymbol{J}} = (\boldsymbol{A}\boldsymbol{A}^*)^{-1/2}\boldsymbol{A}$ is an orthogonal matrix.

Proof. Since $\mathcal{F}_\tau(\boldsymbol{x}) = e^{\tau\boldsymbol{A}^*}\boldsymbol{x}$,

$$\begin{aligned}
\boldsymbol{J}(\boldsymbol{x}) &= \lim_{\tau\to 0}\tau^{-1}[\nabla(\boldsymbol{g}\circ\mathcal{F}_\tau)(\boldsymbol{x}) - \nabla\boldsymbol{g}(\boldsymbol{x})]\\
&= \lim_{\tau\to 0}\tau^{-1}(e^{\tau\boldsymbol{A}} - \boldsymbol{I}) = \boldsymbol{A}.
\end{aligned}$$

By (11), $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{A}^*$. Similarly,

$$\begin{aligned}
\tilde{\boldsymbol{J}}(\boldsymbol{x}) &= \lim_{\tau\to 0}\tau^{-1}[\nabla(\tilde{\boldsymbol{g}}\circ\tilde{\mathcal{F}}_\tau)(\boldsymbol{x}) - \nabla\tilde{\boldsymbol{g}}(\boldsymbol{x})]\\
&= \lim_{\tau\to 0}\tau^{-1}\boldsymbol{M}^{-1/2}(e^{\tau\boldsymbol{A}} - \boldsymbol{I}) = \boldsymbol{M}^{-1/2}\boldsymbol{A}.
\end{aligned}$$

Finally, $\tilde{\boldsymbol{J}}^*\tilde{\boldsymbol{J}} = \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}\boldsymbol{A} = \boldsymbol{I}$. □

Propositions B.1-B.2 explain the choice of $\boldsymbol{M}$ except for the variable scalar bandwidth σ. Computing σ from standard deviations of pairwise distances is standard, except that in Algorithm II.2 it is applied to the transformed samples, $\boldsymbol{M}^{1/2}\boldsymbol{x}$, to appropriately reflect the change of variables. The additional constant scaling factor h can be chosen using standard techniques such as cross validation[32].

Remark B.3. *We could have used another kernel that incorporates our change of variables,* e.g.*, the Laplace kernel*

$$k_{\boldsymbol{M}}^{laplace}(\boldsymbol{x},\boldsymbol{x}') = \exp\left[-\left[(\boldsymbol{x}-\boldsymbol{x}')^*\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{x}')\right]^{1/2}\right].$$

Propositions B.1 and C.1 are independent of kernel choice.

Remark B.4. *It is intuitively reasonable to consider using*

$$\boldsymbol{J}(\boldsymbol{x}) \approx \sum_{m=1}^{R}\lambda_m\phi_m(\boldsymbol{x})\boldsymbol{v}_m, \tag{B2}$$

where $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}^*$ observes the full sample. We observed better results, though, with our method based on Theorem II.1.

Appendix C: Connecting kernels with random Fourier features

Below, we assume that $\boldsymbol{M}$ is symmetric positive definite. The connection between the kernel features (9)- (10) and random Fourier features (17)- (18) is the following.

Proposition C.1. *We have*

$$k_{\boldsymbol{M}}(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}\left[\psi_m^{RFF}(\boldsymbol{x})^* \psi_m^{RFF}(\boldsymbol{x}')\right]$$

where $\mathbb{E}$ *denotes expected value.*

Proof. Let $\boldsymbol{\delta} = \boldsymbol{x}' - \boldsymbol{x}$, $\tilde{\boldsymbol{\delta}} = \boldsymbol{M}^{1/2}\boldsymbol{\delta}$. By completing the square,

$$-\frac{1}{2}|\boldsymbol{\omega}|^2 + i\boldsymbol{\omega}^T \boldsymbol{M}^{1/2}\boldsymbol{\delta} = -\frac{1}{2}\left[(\boldsymbol{\omega} - i\tilde{\boldsymbol{\delta}})^T(\boldsymbol{\omega} - i\tilde{\boldsymbol{\delta}})\right] - \frac{1}{2}|\tilde{\boldsymbol{\delta}}|^2,$$

so if samples live in d-dimensional (real) space $\mathbb{R}^d$, we get

$$\begin{aligned}&\mathbf{E}\left[\psi_m^{RFF}(\boldsymbol{x})^* \psi_m^{RFF}(\boldsymbol{x}')\right] \\ &= (2\pi)^{-d/2}\int \exp(-|\boldsymbol{\omega}|^2)\exp(i\boldsymbol{\omega}^T \boldsymbol{M}^{1/2}\boldsymbol{\delta})\,d\boldsymbol{\omega} \\ &= \exp(-|\tilde{\boldsymbol{\delta}}|^2/2) = k_{\boldsymbol{M}}(\boldsymbol{x},\boldsymbol{x}').\end{aligned}$$

□

Based on Proposition C.1, we now show the connection between FKMD procedures with kernel and random Fourier features. Let $\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF}$ and $\boldsymbol{\Psi}_{\boldsymbol{y}}^{RFF}$ be the $N \times R$ samples by features matrices associated to random Fourier features (17), and let $\boldsymbol{\Psi}_{\boldsymbol{x}}$ and $\boldsymbol{\Psi}_{\boldsymbol{y}}$ be the same matrices associated with kernel features (9). Using Proposition C.1, for large R,

$$\boldsymbol{\Psi}_{\boldsymbol{x}} \approx \boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF}(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*, \qquad \boldsymbol{\Psi}_{\boldsymbol{y}} \approx \boldsymbol{\Psi}_{\boldsymbol{y}}^{RFF}(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*. \tag{C1}$$

Assume the columns of $\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF}$ are linearly independent. Then

$$(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*[(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*]^\dagger = \boldsymbol{I} \tag{C2}$$

where † is the Moore-Penrose pseudoinverse. Define

$$\boldsymbol{K}^{RFF} = (\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*\boldsymbol{K}[(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*]^\dagger, \tag{C3}$$

where $\boldsymbol{K}$ satisfies

$$\boldsymbol{\Psi}_{\boldsymbol{x}}\boldsymbol{K} = \boldsymbol{\Psi}_{\boldsymbol{y}}. \tag{C4}$$

Multiplying (C4) by $(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*$ and $[(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*]^\dagger$ on the left and right respectively, and then using (C1)-(C3), leads to

$$(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF}\boldsymbol{K}^{RFF} \approx (\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^*\boldsymbol{\Psi}_{\boldsymbol{y}}^{RFF},$$

which is the least squares normal equation for

$$\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF}\boldsymbol{K}^{RFF} = \boldsymbol{\Psi}_{\boldsymbol{y}}^{RFF}. \tag{C5}$$

This directly connects the linear solves (C4) and (C5) for the Koopman matrix using kernel and random Fourier features, respectively. Moreover, from (C1)-(C3),

$$\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF}\boldsymbol{K}^{RFF}(\boldsymbol{\Psi}_{\boldsymbol{x}}^{RFF})^\dagger \approx \boldsymbol{\Psi}_{\boldsymbol{x}}\boldsymbol{K}\boldsymbol{\Psi}_{\boldsymbol{x}}^\dagger. \tag{C6}$$

In light of (A1), equation (C6) shows that Fourier features and kernel features give (nearly) the same equation for inference.

Due to Proposition C.1 and the computations in (C1)-(C6) above, random Fourier features (17) and kernel features (9) target essentially the same FKMD procedure whenever R is sufficiently large. In practice, this means random Fourier features can be a more efficient way of solving the same problem.

DATA GENERATION

ERK activity reporters, cell line generation, and live-cell imaging have been described in detail in Davies et al[56]. Here we utilize a dataset monitoring ERK activity in a tissue-like 3D extracellular matrix. Images were collected every 30 minutes over a 90-hour window. Single cells were segmented using Cellpose software[60] and tracked through time by matching cells to their closest counterpart at the previous time point. ERK reporter localization was monitored via the mean-centered and variance stabilized cross-correlation between the nuclear reporter and ERK activity reporter channels in the single-cell cytoplasmic mask. Single-cell trajectories up to 72 hours served as the training set to estimate the Koopman operator. Training and test set data is available and can be accessed via a Zenodo repository (https://doi.org/10.5281/zenodo.10849852).

[1] I. Mezić, Nonlinear Dynamics **41**, 309 (2005).

[2] I. Mezić, Not. Am. Math. Soc. **68**, 1087 (2021).

[3] B. O. Koopman, Proceedings of the National Academy of Sciences **17**, 315 (1931).

[4] J. Koopman, Annu. Rev. Public Health **25**, 303 (2004).

[5] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, Journal of Nonlinear Science **25**, 1307 (2015).

[6] J. H. Tu, *Dynamic mode decomposition: Theory and applications*, Ph.D. thesis, Princeton University (2013).

[7] I. Kevrekidis, C. W. Rowley, and M. Williams, Journal of Computational Dynamics **2**, 247 (2016).

[8] J. L. Proctor, S. L. Brunton, and J. N. Kutz, SIAM Journal on Applied Dynamical Systems **15**, 142 (2016).

[9] A. M. DeGennaro and N. M. Urban, SIAM Journal on Scientific Computing **41**, A1482 (2019).

[10] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, SIAM Journal on Applied Dynamical Systems **19**, 886 (2020).

[11] T. Kawashima and H. Hino, Neural Computation **35**, 82 (2022).

[12] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, Proceedings of the Royal Society A **479**, 20220576 (2023).

[13] N. B. Erichson, S. L. Brunton, and J. N. Kutz, Journal of Real-Time Image Processing **16**, 1479 (2019).

[14] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, Journal of neuroscience methods **258**, 1 (2016).

[15] I. Mezić, Annual review of fluid mechanics **45**, 357 (2013).

[16] S. Bagheri, Journal of Fluid Mechanics **726**, 596 (2013).

[17] H. Arbabi and I. Mezić, Physical Review Fluids **2**, 124402 (2017).

[18] H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé, The Journal of chemical physics **146** (2017).

[19] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, Physica D: Nonlinear Phenomena **406**, 132416 (2020).

[20] A. Navarra, J. Tribbia, and S. Klus, Journal of the Atmospheric Sciences **78**, 1227 (2021).

[21] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, arXiv preprint arXiv:2102.12086 (2021).

[22] P. Bevanda, S. Sosnowski, and S. Hirche, Annual Reviews in Control **52**, 197 (2021).

[23] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, arXiv preprint arXiv:1411.2260 (2014).

[24] A. Radhakrishnan, D. Beaglehole, P. Pandit, and M. Belkin, arXiv preprint arXiv:2212.13881 (2022).

[25] A. Rahimi and B. Recht, Advances in neural information processing systems **20** (2007).

[26] F. Nüske and S. Klus, arXiv preprint arXiv:2306.00849 (2023).

[27] E. N. Lorenz, in *Proc. Seminar on predictability*, Vol. 1 (Reading, 1996).

[28] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman operator in systems and control* (Springer, 2020).

[29] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, Vol. 4 (Springer, 2006).

[30] H. Arbabi and I. Mezic, SIAM Journal on Applied Dynamical Systems **16**, 2096 (2017).

[31] P. J. Baddoo, B. Herrmann, B. J. McKeon, and S. L. Brunton, Proceedings of the Royal Society A **478**, 20210830 (2022).

[32] F. Nüske, S. Peitz, F. Philipp, M. Schaller, and K. Worthmann, Journal of Nonlinear Science **33**, 14 (2023).

[33] S. Klus, F. Nüske, and B. Hamzi, Entropy **22**, 722 (2020).

[34] K.-C. Li, Journal of the American Statistical Association **86**, 316 (1991).

[35] S. Trivedi, J. Wang, S. Kpotufe, and G. Shakhnarovich, in *UAI* (2014) pp. 819–828.

[36] F. Takens, in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80* (Springer, 2006) pp. 366–381.

[37] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, Insights into imaging **9**, 611 (2018).

[38] T. Khanna, *Foundations of neural networks* (Addison-Wesley Longman Publishing Co., Inc., 1990).

[39] J. A. Freeman and D. M. Skapura, *Neural networks: algorithms, applications, and programming techniques* (Addison Wesley Longman Publishing Co., Inc., 1991).

[40] B. Cheng and D. M. Titterington, Statistical science , 2 (1994).

[41] J. A. Tropp and R. J. Webber, arXiv preprint arXiv:2306.12418 (2023).

[42] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, Advances in neural information processing systems **25** (2012).

[43] A. Kammonen, J. Kiessling, P. Plecháč, M. Sandberg, and A. Szepessy, arXiv preprint arXiv:2007.10683 (2020).

[44] P. E. Smith, The Journal of chemical physics **111**, 5568 (1999).

[45] A. Agarwal, S. Gnanakaran, N. Hengartner, A. F. Voter, and D. Perez, arXiv preprint arXiv:2008.11623 (2020).

[46] C. Wehmeyer, M. K. Scherer, T. Hempel, B. E. Husic, S. Olsson, and F. Noé, Living J. Comput. Mol. Sci **1** (2018).

[47] C. R. Schwantes and V. S. Pande, Journal of chemical theory and computation **9**, 2000 (2013).

[48] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, The Journal of chemical physics **139** (2013).

[49] J. McCarty and M. Parrinello, The Journal of chemical physics **147** (2017).

[50] F. Nuske, B. G. Keller, G. Pérez-Hernández, A. S. Mey, and F. Noé, Journal of chemical theory and computation **10**, 1739 (2014).

[51] A. Mardt, L. Pasquali, H. Wu, and F. Noé, Nature communications **9**, 5 (2018).

[52] E. N. Lorenz, Journal of atmospheric sciences **20**, 130 (1963).

[53] C.-C. Hu and P. J. Van Leeuwen, Quarterly Journal of the Royal Meteorological Society **147**, 2352 (2021).

[54] J. C. Butcher, Applied numerical mathematics **20**, 247 (1996).

[55] `https://github.com/davidaristoff/FKMD/tree/main`.

[56] A. E. Davies, M. Pargett, S. Siebert, T. E. Gillies, Y. Choi, S. J. Tobin, A. R. Ram, V. Murthy, C. Juliano, G. Quon, *et al.*, Cell systems **11**, 161 (2020).

[57] J. Copperman, S. M. Gross, Y. H. Chang, L. M. Heiser, and D. M. Zuckerman, Communications Biology **6**, 484 (2023).

[58] Y. Chen, E. N. Epperly, J. A. Tropp, and R. J. Webber, arXiv preprint arXiv:2207.06503 (2022).

[59] Y. Khoo, J. Lu, and L. Ying, Research in the Mathematical Sciences **6**, 1 (2019).

[60] M. Pachitariu and C. Stringer, Nature methods **19**, 1634 (2022).

[61] A. S. Christensen and O. A. Von Lilienfeld, Machine Learning: Science and Technology **1**, 045018 (2020).

[62] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, arXiv preprint arXiv:1807.02582 (2018).

[63] Y. T. Lin, Y. Tian, D. Perez, and D. Livescu, SIAM Journal on Applied Dynamical Systems **22**, 2890 (2023).

[64] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, The Journal of chemical physics **134** (2011).

[65] F. Noé and E. Rosta, The Journal of chemical physics **151** (2019).

[66] N. Rayner, D. E. Parker, E. Horton, C. K. Folland, L. V. Alexander, D. Rowell, E. C. Kent, and A. Kaplan, Journal of Geophysical Research: Atmospheres **108** (2003).