Featurizing Koopman Mode Decomposition

David Aristoff,1 Jeremy Copperman,2 Nathan Mankovich,3 and Alexander Davies2
1)Colorado State University, Fort Collins, CO, 80523, USA
2)Oregon Health & Science University Knight Cancer Institute, Portland, OR, 97201, USA
3)University of Valencia, València, 46010, Spain

(Dated: 21 December 2023)

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 show that our method improves KMD predictions for a high dimensional Lorenz attractor and for a cell signaling problem from cancer research.

I. INTRODUCTION

Koopman mode decomposition (KMD) has emerged as a powerful tool for analyzing nonlinear dynamical systems by identifying patterns and coherent structures that evolve linearly in time. The power of KMD comes from lifting the nonlinear dynamics into a space of observation functions where the evolution of the system can be described by a linear operator, called the Koopman operator. KMD estimates the evolution of a system in a finite dimensional space of feature functions, enabling both quantitative predictions and qualitative analysis of the system’s dynamics (KBM). The basic framework for nonlinear features was introduced by Williams et al. Since then, KMD has been kernelized, integrated with control theory, viewed from the perspective of Gaussian processes, and imposed with physical constraints. KMD has been also been widely applied, including in infectious disease control, fluid dynamics, molecular dynamics, and climate science.

Kernalized KMD, which uses kernel functions as features, is a natural choice when good feature functions are unknown or difficult to find. However, the most commonly used kernels are isotropic, leading to uninformed measures of distance in high dimension. We show how to learn a Mahalanobis matrix mapping of time delay embeddings to prioritize the most dynamically important directions. This results in predictions which are more accurate and robust to noise.

We illustrate our approach, which we call Featurized Koopman Mode Decomposition (FKMD), on a high dimensional Lorenz system with added nuisance variables, and on a cancer cell signaling problem. The Lorenz example illustrates how the curse of dimensionality dooms isotropic kernels, and how our methods can rescue the situation to achieve good results; the cancer cell signaling problem shows the promise of our methods on real-world complex systems. We observe that both of our key ingredients – time embedding and Mahalanobis scaling – are critical for accurate predictions.

The contributions of this work are:

• We introduce Featurized Koopman Mode Decomposition (FKMD), a new version of Koopman mode decomposition that featurizes KMD by automatically finding dynamical structure in time embedded data. This featurization, based on a Mahalanobis matrix, enforces uniform changes in space and time from the data, allowing for more effective analysis and inference. We also show how to scale up to larger data sets, using random Fourier features for deeper learning.

• We illustrate the power of FKMD on two challenging examples. The first is a high-dimensional Lorenz attractor, where the observations are low dimensional and noisy. The second is a cell-signaling problem, where data comes from a video of a cancer cell. There, we use FKMD to infer signaling patterns hours into the future. We show that both of the key ingredients in FKMD – time embedding and Mahalanobis scaling – aid in deep learning of the dynamics.

OVERVIEW OF KOOPMAN MODE DECOMPOSITION

We are interested in analyzing and predicting the outputs of a dynamical system in real Euclidean space, with evolution map \( F_\tau \). Given the current state of our system, \( x(t) \), the system state at time \( \tau \) into the future is \( F_\tau(x(t)) \). That is,

\[
x(t + \tau) = F_\tau(x(t)).
\]

In realistic application problems, \( F_\tau \) is typically a complicated nonlinear function. However, there is a dual interpretation to equation which is linear. For an observation function \( g(x) \) on the system states, the Koopman operator determines the observations at time \( \tau \) in the future:

\[
K_\tau(g)(x) := g(F_\tau(x)).
\]

While the linear framework does not magically remove the complexity inherent in \( F_\tau \), it provides a starting point for globally linear techniques: we can do linear analysis in without

\[\text{arXiv:2312.09146v2 [math.DS]}\]
We use Gaussian kernel functions with a Koopman operator at lag \( \tau \). If we have found can give remarkably good results on complex systems. These choices are:

(i) We use a double time embedded structure to construct feature space. That is, the sample points \( x_n \) themselves are time embeddings, and the sequence \( x_1, \ldots, x_N \) is a time series of these embeddings.

(ii) We use Gaussian kernel functions with a Mahalanobis scaling for our features. The Mahalanobis scaling is updated iteratively and reflects the underlying dynamical structure of the system.

A finite dimensional approximation, \( K \), of the Koopman operator should, as close as possible, satisfy

\[
\Psi_x K = \Psi_y.
\]

Here \( K \) 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 \( K \) as acting in feature space. If \( g \) is a \( 1 \times L \) vector-valued function, we also express \( g \) in feature space coordinates as a \( R \times L \) matrix \( B \):

\[
\Psi_x B = \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{bmatrix}.
\]

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

What Koopman mode decomposition does is convert an eigendecomposition of \( K \) back to sample space, in order to interpret and/or predict the dynamics defined by \( \mathcal{F}_\tau \). To this end, we write the eigendecomposition of \( K \) as

\[
K = \sum_{m=1}^{R} \mu_m \xi_m w_m^*,
\]

where \( \mu_m \) are the eigenvalues of \( K \), and \( \xi_m, w_m \) are the right and left eigenvectors, respectively, scaled so that \( w_m^* \xi_m = 1 \). That is, \( K \xi_m = \mu_m \xi_m \) and \( w_m^* K = \mu_m w_m^* \). It is straightforward to check that, by converting this expression back into sample space, we get the following formula:

\[
\mathcal{K}_\tau(g)(x) \approx \sum_{m=1}^{R} e^{\tau \mu_m} \phi_m(x) v_m^*.
\]
Featurizing Koopman Mode Decomposition

Time embeddings are useful for high dimensional systems that are only partially observed, and have a theoretical basis in Taken’s theorem.22 Other recent work23 uses a double time embedding to construct larger matrices in (3)–(4) with Hankel structure.24 Our setup is different in that the time embedded data goes directly into the kernel functions, allowing us to retain smaller matrices in (3)–(4).

Here, we define samples as time embeddings of length $\ell$,

\[
\begin{align*}
    x_{n+1} &= [x(n\tau) \ldots x((n+\ell-1)\tau)] \\
    y_{n+1} &= [x((n+1)\tau) \ldots x((n+\ell)\tau)],
\end{align*}
\]

(9)

where $x(0)$ is some initial state. The evolution map $F_\tau$ 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 $x$ or $x'$ for a time embedding (or sample) of the form (9).

Our features are based on kernels.26 Kernel features are a common choice when good feature functions are not a priori known. With high dimensional data, especially with the special time-embedded form of our samples, it is important to have distance measurements that adequately capture any underlying dynamical structure. The kernels are centered around the sample points,

\[
\psi_m(x) = k_M(x, x_m), \quad m = 1, \ldots, N.
\]

(10)

Here, $k_M$ is the Mahalanobis kernel

\[
k_M(x, x') = \exp \left[ -((x-x')^\dagger M (x-x')) \right].
\]

(11)

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

Inspired by the recent work23 on interpreting neural networks and improving kernel methods, we propose a novel choice of $M$ using a gradient outerproduct structure.

\[
M = \sum_{n=1}^{N} J(x_n) J(x_n)^\dagger
\]

(12)

up to a scalar factor (defined below), with

\[
J(x) = \sum_{m=1}^{M} \lambda_m \phi_m(x) \phi_m^*. 
\]

(13)

The matrix $M$ is positive (semi)definite Hermitian; replacing $M$ by its real part does not change $k_M(x, x')$.

The scaling by $M$ in (11) is essentially a change of variables which maps samples $x$ to $M^{1/2}x$. The Mahalanobis matrix $M$ is chosen to reflect the system’s underlying dynamical properties. In particular, $M$ enforces uniform changes in space and time of dynamical data; this is explained in more detail in Appendix B. The Mahalanobis matrix $M$ is a priori unknown but estimated iteratively, starting with $M = I$ (up to a scalar standard deviation). Note that the initial kernel is then a standard Gaussian kernel.

B. The FKMD Algorithm

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

Algorithm II.1 (FKMD). Generate samples $x_1, \ldots, x_N$ and $y_1, \ldots, y_N$ according to (9), and choose $h > 0$ and initial Mahalanobis matrix $M = I$. Then iterate the following steps:

1. Let $\sigma$ = standard deviation of the pairwise distances between $M^{1/2}x_1, \ldots, M^{1/2}x_N$. Scale $M \leftarrow M/(h\sigma)^2$.
2. Construct $\Psi_x$ and $\Psi_y$ defined in (3)–(4), using features defined either by (10)–(11) or by (14)–(15).
3. Solve for $K$ and $B$ in (5)–(6), e.g. via ridge regression.
4. Eigendecompose $K$ according to (7). That is, compute right and left eigenvectors $\xi_m$ and $w_m$ of $K$, along with eigenvalues $\mu_m$. Scale them so that $w_m^*\xi_m = 1$.
5. Compute continuous time Koopman eigenvalues, Koopman eigenfunction, and Koopman modes, using

\[
\lambda_m = \tau^{-1} \log \mu_m, \quad \phi_m(x) = \psi(x)\xi_m, \quad v_m^* = w_m^*B.
\]

6. Update $M$ using (12)–(13), and return to Step 1.

Note that observations can be predicted using (8) at any iteration of Algorithm II.1. The initial $M$ could be chosen using information about the system, but in absence of that we use a scalar multiple of $I$ as described in Algorithm II.1. If results degrade with iterations, we recommend adding a small ridge regularization to $M$ by updating $M \leftarrow M + \delta I$ after Step 6, where $\delta > 0$ is a small parameter. If computation of the Mahalanobis matrix is too expensive, we suggest subsampling the samples in (12) and/or using cutoffs for modes in (13); see Appendix B. Subsampling can also be used to estimate $\sigma$.

Algorithm II.1 only requires a few user chosen parameters: an embedding length $\ell$, a bandwidth $h$, and a number of features $R$ (not counting regularization parameters or possible cutoffs and subsampling parameters). This is a significant advantage over artificial neural network methods, which often require tuning over a much larger set of hyperparameters, and a training procedure that is not guaranteed to converge to an optimal parameter set.

C. Scaling up to larger sample size

Kernel methods have historically been limited by the computational complexity of large linear solve (usually limiting sample size to $N \leq 10^5$), 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.

In detail, we use random Fourier features.

\[
\psi_m^{RFF}(x) = \exp(i\omega_m^* \text{Re}(M^{1/2}x)), \quad m = 1, \ldots, R.
\]

(14)

Here, $\omega_m$ are iid Gaussians with mean 0 and covariance $I$,

\[
\omega_m \sim \mathcal{N}(0, I).
\]

(15)
and $\text{Re}(M)$ denotes the real part of $M$. In practice we expect good results with $R \ll N$; this leads to much more efficient linear solves and eigendecompositions. The features (12)-(15) 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. Lorenz attractor

Here, we illustrate Algorithm [11] on data from the Lorenz 96 model [13] a high-dimensional ODE exhibiting chaotic behavior. This model (and its 3-dimensional predecessor [14]) are often used to interpret atmospheric convection and to test tools in climate analysis [15]. The model is

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

with $j = 1, \ldots, 40$ periodic coordinates ($j \equiv j \text{ mod } 40$). We set $F = 8$, and integrate using 4th order Runge-Kutta with integrator time step $10^{-2}$. The initial condition is [16]

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

To illustrate the power of our method, we assume that we only observe 2.5% of the system, namely the first coordinate $\theta_1$, and we add nuisance or “noise” variables. Specifically, we use the time embedding (9) with $\tau = 0.05$ and

$$x_n = [\theta_1(n\tau) \ \text{noisy}(n\tau)], \quad (16)$$

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 [11] with random Fourier features defined in [12)-(15]. Inference begins at the end of the training set, and consists of 100 discrete steps of time length $\tau$. The code for this experiment is available here [17].

For the FKMD parameters, we use $N = 10^6$ sample points, $R = 7500$ features, a time embedding of length $\ell = 100$, and a constant bandwidth factor $h = 1$. The observation $g(x)$ is a $1 \times 200$ vector associated with time embeddings of (16) as defined in (9). Results are plotted in Figure 1. If any one of $N$, $R$ or $\ell$ is decreased, results degrade noticeably. We use the top 20 modes to define $M$, as discussed in Appendix [B] and we estimate $\Sigma$ and $M$ using a random subsample of 5000 points. See Figure 2 for a visualization of $M$.

Figure 1(a)-(c) shows inference using equation (8). On the 1st iteration, the predictions collapse, but by the 5th iteration, they are nearly a perfect match with the reference. The results do not change much after 5 iterations. Figure 1(d) shows ordinary KMD. (Ordinary KMD corresponds to Algorithm [11] with all the same parameters except the “second” time embedding $\ell = 1$ is trivial, $M$ is a scalar, iteration is unnecessary, and there are no nuisance variables.) 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 2(a)-(b). Here, we split $M$ into nuisance and non-nuisance parts based on (16). Intuitively, the Mahalanobis matrix corresponds to a change of variables that eliminates the nuisance coordinates, while preserving the structure of the underlying signal, which here is $\theta_1(t)$.

B. 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 likely one such example, as it is both complex and subject to noisy temporal outputs arising from properties of the system itself and experimental sources.

With these considerations in mind, we next apply our method to dynamic signaling activity in cancer cells to assess its performance. Here we show 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 [18]. The extracellular signal-regulated kinases (ERK) signaling pathway is critical for perception of cues outside of cells, and for translation of these cues into
Featurizing Koopman Mode Decomposition

A. high activity (x=2.1)
low activity (x=-2.2)

B. Mahalanobis matrix

C. activity (y=norm)
D. Mahalanobis matrix by iteration

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

FIG. 3: 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) Mahalanobis 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 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 have proposed to featurize Koopman Mode Decomposition that generates more accurate predictions than standard (e.g., Gaussian kernel) KMD. By modeling data assuming uniform changes in space and time, our Mahalanobis scaling helps mitigate the curse of dimensionality. Results in two separate application areas – climate and cell signaling dynamics – illustrate the promise of the method for inferring future states of time series generated by complex systems.

Many theoretical and algorithmic questions remain. We
would like to better understand parameter choices, particularly how to choose modes for the Mahalanobis matrix computation, as described in Appendix [B] and how many iterations to run. 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 [A1], e.g., convergence to a fixed point.

We would also like to explore alternative methods for scaling up our methods to larger sample sizes. Here, we use random Fourier features to scale up FKMD, but more analysis and comparison with alternative methods are still needed. Another possible trick to increase sample size may come from modern advances in randomized numerical linear algebra, e.g., randomly pivoted Cholesky [C5]. Such methods promise spectral efficiency for solving symmetric positive definite linear systems. Assuming fast spectral decay of $\Psi_x$ (which we have observed empirically), these techniques could help our methods scale to even larger sample sizes. We will explore applications of these cutting-edge methods in future works.

Finally, we would like to better understand mechanisms, i.e., what makes a system head towards a certain state $A$ rather than another state $B$? These states could be associated with El Niño occurring or not, or with cancer remaining in a cell or spreading. This can be formalized by the concept of the committor, the probability that the system reaches state $B$ before $A$, starting at point $x$. By imposing appropriate boundary conditions on the data, the committor can be represented as a Koopman eigenfunction associated to eigenvalue 1. This suggests a particular choice of $M$ based on using only this mode. In this case, $M$ could potentially identify what directions are associated with 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 support from the project "Artificial Intelligence for complex systems: Brain, Earth, Climate, Society," funded by the Department of Innovation, Universities, Science, and Digital Society, code: CIPROM/2021/56.

**Appendix A: Derivation of Koopman eigendecomposition**

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

Recall that the matrix $K$ 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 $\Psi_x$.

This leads to the following equation for inference:

$$
\begin{bmatrix}
\mathcal{K}_\tau(g)(x_1) \\
\vdots \\
\mathcal{K}_\tau(g)(x_N)
\end{bmatrix}
\approx
\Psi_x K \Psi_x^\dagger
\begin{bmatrix}
g(x_1) \\
\vdots \\
g(x_N)
\end{bmatrix},
$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse. Similarly,

$$
B = \Psi_x^\dagger
\begin{bmatrix}
g(x_1) \\
\vdots \\
g(x_N)
\end{bmatrix}.
$$

The eigendecomposition of $K$ can be written as

$$
K = \Xi D W^*,
$$

where $K \Xi = \Xi D$ and $W^* K = D W^*$, and we may assume that $W^* \Xi = I$. Here, $D$ is the diagonal matrix of Koopman eigenvalues, $\mu_m$; that is, $D = \exp(\tau A)$ where $A$ is the diagonal matrix of continuous time Koopman eigenvalues, $\lambda_m$.

Plugging (A2) into (A1),

$$
\begin{bmatrix}
\mathcal{K}_\tau(g)(x_1) \\
\vdots \\
\mathcal{K}_\tau(g)(x_N)
\end{bmatrix}
\approx
\Psi_x \Xi \exp(\tau A) W^* B.
$$

The definition of Koopman modes and Koopman eigenfunctions shows that the rows of $W^* B$ are the Koopman modes $v_m^\tau$, while the Koopman eigenfunctions are sampled by the columns of

$$
\Psi_x \Xi =
\begin{bmatrix}
\phi_1(x_1) & \cdots & \phi_R(x_1) \\
\vdots & & \vdots \\
\phi_1(x_N) & \cdots & \phi_R(x_N)
\end{bmatrix}.
$$

Substituting (A4) into (A3) and writing the matrix multiplication in terms of outerproducts yields equation [8], provided we substitute $x_n$ for $x$, using any sample point $x_n$.

Note that equation [8] naturally allows for mode selection. For example, modes that lead to predictions that are a priori known to be unphysical, e.g., diverging modes associated to eigenvalues with $\text{Re}(\lambda_m) > 0$, can simply be omitted from the sum in [8]. Similar selection can be done to remove modes that oscillate too fast, i.e., $|\text{Im}(\lambda_m)| > 0$. We have done this in the experiments in Section [II-B].

**Appendix B: Choice of Mahalanobis matrix**

Here, we explain the reasoning behind the choice of Mahalanobis matrix. Notice that our kernels have the form

$$
k_M(x, x') = \exp\left(-\|x - x'\|_M^2\right), \quad \|x\|_M^2 := x^T M x.
$$

In addition, $\|x\|_M = \|M^{1/2} x\|$, where the second norm is the usual Euclidean one. The Mahalanobis scaling can then be interpreted as just a change of variables, $\tilde{x} = M^{1/2} x$, where the tilde notation indicates the changed variables.
Below, we use tildes to denote the changed variables, \( \tilde{x} = M^{1/2}x \), and write \( \tilde{g}(\tilde{x}) = g(x) \) for \( g \) in these changed variables. We will implicitly assume appropriate smoothness so that all the derivatives calculations below make sense. Also, we will assume that \( M \) is invertible.

**Proposition B.1.** Define a function \( f \) on unit vectors \( u \) by

\[
f(u) = \frac{1}{N} \sum_{n=1}^{N} |u^* \tilde{J}(\tilde{x}_n)|^2.
\]  

(B1)

where \( \tilde{J} \) is given by

\[
\tilde{J}(\tilde{x}_n) = \lim_{\tau \to 0} \tau^{-1} [\nabla \tilde{g}(\tilde{x}_n) - \nabla \tilde{g}(\tilde{x}_n)].
\]

In the idealized case of zero Koopman estimation error, i.e., equality in \([6]\), the function \( f \) is constant valued.

**Proof.** Assuming no estimation error, \([6]\) and \([13]\) imply

\[
J(x_n) = \nabla \left( \frac{d}{dt} K_t g(x_n) \bigg|_{t=0} \right)
= \lim_{\tau \to 0} \tau^{-1} [\nabla g(y_n) - \nabla g(x_n)].
\]

(B2)

By the chain rule, \( \nabla \tilde{g}(\tilde{x}_n) = M^{-1/2} \nabla g(x_n) \). Thus,

\[
f(u) = \frac{1}{N} \sum_{n=1}^{N} |u^* M^{-1/2} \tilde{J}(x_n)|^2
= \frac{1}{N} \sum_{n=1}^{N} u^* M^{-1/2} J(x_n) J(x_n)^* M^{-1/2} u
= \frac{1}{N} (h \sigma)^2,
\]

where \( M \) is defined by \([12]\) and Step 1 of Algorithm\([11]\).

We interpret \( J \) and \( \tilde{J} \) as a kind of dynamical curvature in the original and transformed variables. See Remark \([B.2]\) for intuition behind this term. We have shown that the average inner product between any test vector \( u \) and the dynamical curvature at each point is constant. This implies that we have constant average dynamical curvature in all directions. We have loosely referred to this property as “uniform changes in space and time.”

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

\[
\frac{dx(t)}{dt} = x(t)^* A,
\]

and that the observation function records the whole state, \( g(x) = x^* \). In this case, \( J = A \) and \( M = AA^* \), while

\[
\tilde{J} = (AA^*)^{-1/2} A
\]

is an orthogonal matrix. Here, \( J \) and \( \tilde{J} \) represent curvature in the original and changed variables, respectively; in particular, the variable change makes the evolution appear to be driven by \( \tilde{J} \) rather than \( J = A \).

This explains the choice of \( M \) except for the variable scalar bandwidth \( \sigma \). Computing \( \sigma \) from standard deviations of pairwise distances is standard, except that in Algorithm\([11]\), it is applied to the transformed samples, \( M^{1/2}x \), to appropriately reflect the change of variables. The additional constant scaling factor \( h \) can be chosen using standard techniques such as cross validation\([23]\).

We have noticed empirically that estimates of \( M \) can suffer from noise effects if too many modes are used. In particular, we find good results by using only top modes according to some cutoff in \([13]\); 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. We use this idea in the experiments in Section\([III.A]\) and Section\([III.B]\).

**Remark B.3.** It is intuitively reasonable to consider using

\[
J(x) := \left. \frac{d}{dt} K_t g(x) \right|_{t=0}^*,
\]

where \( g(x) = x^* \) observes the full sample. Empirically, we find that equation \([B.2]\) outperforms \([B.3]\). Another advantage of \([B.2]\) is that it is tailored to a particular observation function, while \([B.3]\) requires reconstructing the full sample.

Appendix C: Connection between kernel and random Fourier features

The connection between the kernel features \([10]\)-\([11]\) and random Fourier features \([13]\)-\([15]\) is the following.

**Proposition C.1.** We have

\[
k_M(x, x') = \mathbb{E} \left[ \psi_m^\text{RF}(x)^* \psi_m^\text{RF}(x') \right]
\]

where \( \mathbb{E} \) denotes expected value.

**Proof.** Since \( M \) is Hermitian,

\[
||x||_M = ||x||_{\text{Re}(M)}
\]

for real \( x \). Because of this we may, without loss of generality, assume that \( M \) is real. Let \( \delta = x' - x \) and \( \tilde{\delta} = M^{1/2} \delta \). By completing the square,

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

so if samples live in \( d \)-dimensional (real) space, we get

\[
\mathbb{E} \left[ \psi_m^\text{RF}(x)^* \psi_m^\text{RF}(x') \right] = (2\pi)^{-d/2} \int \exp(-||\omega||^2) \exp(i \omega^T M^{1/2} \delta) d\omega
= \exp(-||\tilde{\delta}||^2/2) = k_M(x, x').
\]

Based on Proposition\([C.1]\) we now show the connection between FKMD procedures with kernel and random Fourier features. Let \( \Psi_2^\text{RF} \) and \( \Psi_2^\text{RF} \) be the \( N \times R \) samples by features
matrices associated to random Fourier features \( \langle 14 \rangle \), and let \( \Psi_x \) and \( \Psi_y \) be the same matrices associated with kernel features \( \langle 10 \rangle \). Using Proposition \( \langle C.1 \rangle \) for large \( R \) we have
\[ \Psi_x \approx \Psi_x^{RFF}(\Psi_x^{RFF})^*, \quad \Psi_y \approx \Psi_y^{RFF}(\Psi_y^{RFF})^*. \] (C1)
Assume the columns of \( \Psi_x^{RFF} \) are linearly independent. Then
\[ (\Psi_x^{RFF})^*(\Psi_x^{RFF})^\dagger = I \] (C2)
where \( \dagger \) is the Moore-Penrose pseudoinverse. Define
\[ K^{RFF} = (\Psi_x^{RFF})^* K [(\Psi_x^{RFF})^*]^\dagger, \] (C3)
where \( K \) satisfies
\[ \Psi_x K = \Psi_y. \] (C4)
Multiplying \( \langle C.4 \rangle \) by \( (\Psi_x^{RFF})^* \) and \( [(\Psi_x^{RFF})^*]^\dagger \) on the left and right, and then using \( \langle C.1 \rangle - \langle C.3 \rangle \), leads to
\[ (\Psi_x^{RFF})^* \Psi_x^{RFF} K^{RFF} \approx (\Psi_x^{RFF})^* \Psi_y^{RFF}, \] which is the least squares normal equation for
\[ \Psi_x^{RFF} K^{RFF} = \Psi_y^{RFF}. \] (C5)
This directly connects the linear solves \( \langle C.4 \rangle \) and \( \langle C.5 \rangle \) for the Koopman matrix using kernel and random Fourier features, respectively. Moreover, from \( \langle C.1 \rangle - \langle C.6 \rangle \)
\[ \Psi_x^{RFF} K^{RFF} (\Psi_x^{RFF})^* \approx \Psi_x K \Psi_x^* \] (C6)
In light of \( \langle A.1 \rangle \), equation \( \langle C.6 \rangle \) shows that Fourier features and kernel features give (nearly) the same equation for inference. Due to Proposition \( \langle C.4 \rangle \) and the computations in \( \langle C.1 \rangle - \langle C.6 \rangle \) above, random Fourier features \( \langle 14 \rangle \) and kernel features \( \langle 10 \rangle \) 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.\(^{14} \). Here we utilize a dataset monitoring ERK activity in tissue-like 3D extracellular matrix. Images were collected every 30 minutes over a 90-hour window. Single-cells were segmented using Cellpose software\(^{34} \) and tracked through time by matching cells to their closest counterpart at the previous timepoint. 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.

\(^{11} \) I. Mezić, Nonlinear Dynamics 41, 309 (2005).
\(^{3} \) B. O. Koopman, Proceedings of the National Academy of Sciences 17, 315 (1931).

\(^{11} \) I. Mezić, Annual review of fluid mechanics 45, 357 (2013).
\(^{13} \) H. Arzabi and I. Mezić, Physical Review Fluids 2, 124402 (2017).
\(^{17} \) A. Rahimi and B. Recht, Advances in neural information processing systems 20 (2007).
\(^{21} \) A. Mauroy, Y. Susuki, and I. Mezić, Koopman operator in systems and control (Springer, 2020).