

Efficient Approximation of Molecular Kinetics using Random Fourier Features

Feliks Nüske¹ and Stefan Klus²

¹*Max-Planck-Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany*

²*Heriot-Watt University, Edinburgh, United Kingdom*

(*Electronic mail: nueske@mpi-magdeburg.mpg.de, S.Klus@hw.ac.uk)

(Dated: 16 June 2023)

Slow kinetic processes of molecular systems can be analyzed by computing dominant eigenpairs of the Koopman operator or its generator. In this context, the Variational Approach to Markov Processes (VAMP) provides a rigorous way of discerning the quality of different approximate models. Kernel methods have been shown to provide accurate and robust estimates for slow kinetic processes, but are sensitive to hyper-parameter selection, and require the solution of large-scale generalized eigenvalue problems, which can easily become computationally demanding for large data sizes. In this contribution, we employ a stochastic approximation of the kernel based on random Fourier features (RFFs), to derive a small-scale dual eigenvalue problem which can easily be solved. We provide an interpretation of this procedure in terms of a finite randomly generated basis set. By combining the RFF approach and model selection by means of the VAMP score, we show that kernel parameters can be efficiently tuned, and accurate estimates of slow molecular kinetics can be obtained for several benchmarking systems, such as deca alanine and the NTL9 protein.

I. INTRODUCTION

The automated extraction of essential information about thermodynamic and kinetic properties of molecular systems from large-scale computer simulations remains one of the major open problems in computational physics and chemistry to this day. The challenge consists in processing very long time series of high-dimensional data in such a way that *relevant* features of the molecular system are determined automatically, with limited user input or intervention. The precise definition of what is regarded as relevant depends on the context, but in many applications, the determination of *slow* relaxation processes and their associated metastable sets is among the most important features to be detected. This applies, for example, to protein folding, ligand binding, or protein–protein association^{1–3}.

Let us highlight two milestones that have shaped the field in the last two decades: The first was the inception of Markov State Models (MSMs) in the late 1990s and early 2000s^{4–6}, which allowed for the construction of easily interpretable discrete kinetic models, from which a wealth of information could be extracted. The second milestone was the variational approach to conformational dynamics (VAC) for reversible systems^{7,8}, and its extension called *variational approach to Markov processes* (VAMP)⁹, which provided a rigorously defined scalar objective function (called VAMP score) to assess the quality of kinetic models. Combined, these developments have led to a standard workflow for the construction of MSMs, usually consisting of a linear dimensionality reduction using *time-lagged independent component analysis* (TICA)¹⁰, geometric clustering, as well as MSM estimation and validation^{11,12}. It was also shown that MSMs and the VAC are closely related to the Koopman operator framework^{13,14}, and in particular the *extended dynamic mode decomposition* (EDMD) algorithm^{15,16}, which have primarily emerged in the analysis of fluid dynamics and related fields. The Koopman framework is therefore a unifying theme for the construction of kinetic models. Extensions of the EDMD algorithm can also be used to approximate the generator of dynamical systems (e.g., the Kolmogorov backward operator for stochastic differential equations) as shown in¹⁷.

Despite the remarkable success of MSMs, the construction process remains somewhat unsatisfactory, at least from a theoretical perspective. The initial TICA step presupposes that there is a kinetically informative linear subspace of the initial set of descriptors used for MSM building. In order to move beyond this requirement, algorithms based on deep neural networks (DNNs) were introduced in recent years^{18,19}, allowing to learn in one step the complete non-linear transfor-

mation from raw atomic coordinates to low-dimensional features that optimize the VAMP score. Both the MSM construction process (including state definition) and neural networks require a non-linear optimization step, which is typically solved by means of stochastic algorithms. A significant number of hyper-parameters need to be tuned in both cases, such as network architecture, network depth and height for DNNs, or number of cluster centers, number of iterations, convergence threshold, etc. for clustering-based MSMs.

A different avenue that has been considered is to employ reproducing kernels, which have long been known in machine learning^{20,21} as a powerful non-linear model class with rich theoretical properties. Kernel methods have also been demonstrated in many applications to be fairly robust if the available data is only small- to medium-sized. Kernel-based versions of EDMD were suggested in^{22–24}, and applied to high-dimensional molecular simulations in²⁵. Extracting slow timescales from kernel representations requires the solution of a large-scale generalized eigenvalue problem, where the size of the matrix depends on the number of snapshots. Even for moderate data size, the solution of this problem can be challenging. Moreover, although kernel methods usually require only a few hyper-parameters (such as the bandwidth of the hugely popular Gaussian kernel), these need to be tuned accurately, requiring to solve the above-mentioned eigenvalue problem many times, which can be prohibitively time-consuming.

In this work, we show that kernel methods can be made tractable for the analysis of molecular kinetics by employing *Random Fourier Features* (RFFs)²⁶, a well-known low-rank approximation technique for kernel matrices, to convert the original large-scale eigenvalue problem into a much smaller one, which can be easily solved. We derive the associated small-scale matrix problem both for the Koopman operator and its generator, and show that this approach can be interpreted as applying EDMD with a specific randomly chosen basis set. We demonstrate that the VAMP score can be effectively used for hyper-parameter tuning of the kernel method. The combination of these ideas leads to an efficient approximation algorithm, which is robust with respect to the data size and the stochastic Fourier feature selection. Therefore, our method presents a promising new direction for the automated analysis of molecular kinetics.

The remainder of this paper is structured as follows: In Section II, we review elements of the Koopman formalism, the EDMD approximation algorithm, reproducing kernel Hilbert spaces (RKHSs), and Koopman modeling using RKHSs. Then, in Section III, we present our main contribution, an approximation algorithm for the Koopman operator or generator based on Random Fourier Features. Further computational details, especially regarding model selection, are dis-

cussed in Section IV, while applications to benchmarking problems in molecular dynamics are shown in Section V.

II. THEORETICAL BASICS

We will briefly introduce the stochastic Koopman operator and its generator as well as numerical methods to estimate these operators and their spectral properties from data.

A. Stochastic Dynamical Systems

Computer simulations are routinely used to explore the thermodynamics and kinetics of molecular systems. Many available implementations of molecular dynamics (MD) can be described as a stochastic process $\{X_t\}_{t \geq 0}$, where t is the time and $X_t \in \mathbb{X} \subset \mathbb{R}^d$ the state of the system at time t . In particular, a widely used model in this context is given by *stochastic differential equations* (SDEs) of the form

$$dX_t = F(X_t) dt + G(X_t) dW_t. \tag{1}$$

where $F: \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $G: \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ are vector- and matrix-valued fields, respectively, called the *drift* and *diffusion* of the SDE, and W_t is d -dimensional Brownian motion. The covariance matrix of the diffusion is denoted by

$$a(x) := G(x)G(x)^\top.$$

As far as the thermodynamics is concerned, the primary goal of molecular simulations is to sample the *Boltzmann distribution*

$$d\mu(x) \sim \exp(-\beta V(x)) dx,$$

where V is the potential energy at position x and $\beta^{-1} = k_B T$ is the inverse temperature. Most simulation protocols are built to achieve this goal, i.e., the Boltzmann distribution is invariant, and long time averages converge to spatial averages with respect to μ (ergodicity). A standard example is *Brownian* or *overdamped Langevin dynamics*

$$dX_t = -\frac{1}{\gamma} \nabla V(X_t) dt + \sqrt{2\gamma^{-1}\beta^{-1}} dW_t, \tag{2}$$

with friction parameter γ .

B. Koopman Operator and Generator

When it comes to inferring kinetic information about X_t , significant progress has been made in recent years by considering an operator-based approach to the statistics of the stochastic process X_t . For a fixed lag time $t \geq 0$, the *Koopman operator*¹³ \mathcal{K}^t acts on a functions $\phi: \mathbb{X} \rightarrow \mathbb{R}$ by taking the conditional expectation:

$$\mathcal{K}^t \phi(x) = \mathbb{E}^x[\phi(X_t)] = \mathbb{E}[\phi(X_t) | X_0 = x].$$

In other words, evaluation of the function $\mathcal{K}^t \phi$ at x yields the expectation of ϕ after starting the dynamics at x , and evolving until time t . The linear operators \mathcal{K}^t satisfy the semigroup equation $\mathcal{K}^{s+t} = \mathcal{K}^s \mathcal{K}^t$, which implies the existence of another linear operator \mathcal{L} , called *Koopman generator*, such that the following linear differential equation holds for the function $\phi_t = \mathcal{K}^t \phi$:

$$\frac{\partial}{\partial t} \mathcal{K}^t \phi(x) = \mathcal{L} \mathcal{K}^t \phi(x).$$

For SDE dynamics of the form (1), stochastic calculus shows that \mathcal{L} is a second-order differential operator, and we obtain the *Kolmogorov equation*

$$\begin{aligned} \frac{\partial}{\partial t} \phi_t(x) &= \sum_{i=1}^d F(i)(x) \frac{\partial}{\partial x^i} \phi_t(x) + \frac{1}{2} \sum_{i,j=1}^d a(i,j)(x) \frac{\partial^2}{\partial x^i \partial x^j} \phi_t(x) \\ &= F(x) \cdot \nabla \phi_t(x) + \frac{1}{2} a(x) : \nabla^2 \phi_t(x) \\ &= \mathcal{L} \phi_t(x), \end{aligned}$$

where we used the colon notation for the Frobenius inner product between matrices, i.e., $A : B = \sum_{i,j} A(i,j)B(i,j)$. For overdamped Langevin dynamics, the generator is given by

$$\mathcal{L}_{\text{OD}} = -\frac{1}{\gamma} \nabla V \cdot \nabla + \gamma^{-1} \beta^{-1} \Delta.$$

We will summarily refer to the Koopman operators or the generator as *dynamical operators* for the dynamical system X_t . If the dynamics are reversible with respect to the invariant distribution μ , then the generator \mathcal{L} is symmetric with respect to the inner product

$$\langle \phi_1, \phi_2 \rangle_\mu = \int_{\mathbb{X}} \phi_1(x) \phi_2(x) d\mu(x).$$

As a consequence, all eigenvalues of the generator \mathcal{L} are real-valued and non-positive, i.e., for all solutions of the equation

$$\mathcal{L} \psi_i = \kappa_i \psi_i,$$

we have $\kappa_i \leq 0$. An object of prime interest for molecular kinetics are low-lying eigenvalues $0 \leq -\kappa_i \ll 1$, as these encode slow motions corresponding to metastable behavior encountered in many examples of molecular systems^{6,27}. The connection between close-to-zero eigenvalues and metastability is also apparent in the fact that eigenfunctions of the generator are also eigenfunctions of the Koopman operator at any t , with eigenvalue $\lambda_i(t) = e^{-\kappa_i t}$. If $-\kappa_i \ll 1$, then the corresponding eigenvalues $\lambda_i(t)$ decay slowly, as expected for metastable dynamics.

C. Finite-dimensional Approximation

The basic data-driven approximation algorithm for the Koopman operator is called *extended dynamic mode decomposition* (EDMD)²⁸, its adaptation to the generator is then called *gEDMD*¹⁷. Both methods require choosing a finite *dictionary* of observable functions $\{\phi_1, \dots, \phi_n\}$. The projections of the dynamical operators onto the linear span of these functions can be represented by matrices:

$$\mathbf{K}^t = \mathbf{G}^{-1} \mathbf{A}^t, \quad \mathbf{L} = \mathbf{G}^{-1} \mathbf{A}^L,$$

with mass and stiffness matrices given by

$$\mathbf{G}(i, j) = \langle \phi_i, \phi_j \rangle_\mu, \quad \mathbf{A}^t(i, j) = \langle \phi_i, \mathcal{K}^t \phi_j \rangle_\mu, \quad \mathbf{A}^L(i, j) = \langle \phi_i, \mathcal{L} \phi_j \rangle_\mu.$$

Given a long trajectory $\{x_1, x_2, \dots, x_{m+1}\}$ of the process X_t , the data-based estimators for these matrices are given by

$$\begin{aligned} \hat{\mathbf{G}}(i, j) &= \frac{1}{m} \sum_{l=1}^m \phi_i(x_l) \phi_j(x_l), & \hat{\mathbf{A}}^t(i, j) &= \frac{1}{m} \sum_{l=1}^m \phi_i(x_l) \phi_j(x_{l+1}), \\ \hat{\mathbf{A}}^L(i, j) &= \frac{1}{m} \sum_{l=1}^m \phi_i(x_l) \mathcal{L} \phi_j(x_l), \end{aligned}$$

where we assumed the time step between samples to equal t in the formula for $\hat{\mathbf{A}}^t$. This is not required in practice if a sliding-window estimator is used, see^{12,29} for details.

For reversible dynamics, the stiffness matrix \mathbf{A}^L and its estimator $\hat{\mathbf{A}}^L$ above can be replaced by the following energy-like expressions¹⁷:

$$\begin{aligned} \mathbf{A}^L(i, j) &= -\frac{1}{2} \int_{\mathbb{X}} \nabla^\top \phi_i(x) a(x) \nabla \phi_j(x) d\mu(x), \\ \hat{\mathbf{A}}^L(i, j) &= -\frac{1}{2m} \sum_{l=1}^m \nabla^\top \phi_i(x_l) a(x_l) \nabla \phi_j(x_l). \end{aligned} \tag{3}$$

The latter estimator only requires first-order derivatives and retains the symmetry of \mathbf{A}^L even for finite data.

Eigenvalues of the dynamical operators can be approximated by diagonalizing the data-driven estimators $\hat{\mathbf{K}}^t = \hat{\mathbf{G}}^{-1}\hat{\mathbf{A}}^t$ and $\hat{\mathbf{L}} = \hat{\mathbf{G}}^{-1}\hat{\mathbf{A}}^L$, or equivalently, by solving one of the generalized eigenvalue problems

$$\hat{\mathbf{A}}^t \mathbf{v}_i = \hat{\lambda}_i(t) \hat{\mathbf{G}} \mathbf{v}_i, \quad \hat{\mathbf{A}}^L \mathbf{v}_i = \hat{\kappa}_i \hat{\mathbf{G}} \mathbf{v}_i. \quad (4)$$

D. Collective Variables

For large molecular systems, the dictionary is typically defined in terms of a set of lower-dimensional descriptors $z \in \mathbb{R}^N$, $N \leq d$, often called *collective variables* (CVs), such as intramolecular distances or angles. Formally, CVs are given by a smooth mapping $\xi: \mathbb{X} \rightarrow \mathbb{R}^N$. Choosing a dictionary $\{\tilde{\phi}_i\}_{i=1}^n$ of functions depending only on CV space, that is $\tilde{\phi}_i(x) = \phi_i(\xi(x))$, does not lead to any conceptual changes to the EDMD approach. When approximating the generator, the gEDMD method is straightforwardly adapted by computing all x -derivatives in the definition of the generator using the chain rule. This leads to a particularly elegant reformulation of the symmetric estimator (3) in terms of an effective diffusion matrix

$$a^\xi(x) = \nabla_x^\top \xi(x) a(x) \nabla_x \xi(x), \quad (5)$$

where $\nabla_x \xi$ is the $d \times N$ -dimensional Jacobian of the mapping ξ . The symmetric estimator (3) then becomes

$$\hat{\mathbf{A}}^L(i, j) = -\frac{1}{2m} \sum_{l=1}^m \nabla_z^\top \phi_i(z_l) a^\xi(x_l) \nabla_z \phi_j(z_l)$$

as shown in¹⁷. Note that once the effective diffusion matrices have been computed, only the basis functions and their derivatives in CV-space are required.

E. Reproducing Kernel Hilbert Space

A critical modeling decision for the (g)EDMD approach is the choice of the dictionary $\{\phi_1, \dots, \phi_n\}$. Recent works^{22–25} have shown that accurate results, even for large systems, can be obtained using methods based on *reproducing kernel Hilbert spaces* (RKHS) generated by a kernel function k . These approaches lead to powerful approximation spaces, yet only depend on few model parameters.

A reproducing kernel Hilbert space (RKHS)^{21,30} is a Hilbert space \mathbb{H} , with inner product $\langle \cdot, \cdot \rangle_{\mathbb{H}}$, of continuous functions on \mathbb{X} , defined by a symmetric, positive-definite two-argument function $k(x, y)$, called the *kernel*. Each such kernel corresponds to a unique RKHS³⁰. Given k , one can define a map from \mathbb{X} into \mathbb{H} , called the *feature map*, by

$$\Phi(x)(y) = k(x, y).$$

Functions f in the RKHS then satisfy the reproducing property

$$\langle f, \Phi(x) \rangle_{\mathbb{H}} = f(x) \quad \forall f \in \mathbb{H}.$$

Example II.1 *Arguably the most widely used kernel is the Gaussian radial basis function (RBF) kernel*

$$k_{\sigma}(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right),$$

with bandwidth parameter $\sigma > 0$. On a periodic domain $[-L, L]^d$, the closely related periodic Gaussian kernel

$$k_{\sigma}^p(x, y) = \exp\left(-\frac{2}{\sigma^2} \sum_{i=1}^d \sin^2(0.5(x^i - y^i))\right),$$

analogously generates an RKHS of periodic functions³¹. △

For many popular kernels, including the Gaussian RBF kernel, the associated RKHS is infinite-dimensional, and densely embedded into most relevant function spaces. Kernels therefore provide powerful approximation spaces, reducing the problem of dictionary selection to the choice of a single function, the kernel, and its hyper-parameters, such as the bandwidth for the Gaussian RBF kernel.

F. RKHS Representation of Dynamical Operators

Given a kernel k with RKHS \mathbb{H} , the analogues of the matrices \mathbf{G} , \mathbf{A}^t and \mathbf{A}^L in Section II C are given by linear operators on \mathbb{H} , see^{23,24} for a derivation:

$$\mathcal{G}f = \int_{\mathbb{X}} f(x)\Phi(x) d\mu(x), \quad \mathcal{A}^t f = \int_{\mathbb{X}} \mathcal{K}^t f(x)\Phi(x) d\mu(x), \quad \mathcal{A}^L f = \int_{\mathbb{X}} \mathcal{L} f(x)\Phi(x) d\mu(x). \quad (6)$$

Here, Φ is again the feature map. The eigenvalue problem (4) then turns into an infinite-dimensional generalized eigenvalue equation for eigenfunctions $\psi_i \in \mathbb{H}$:

$$\mathcal{A}^t \psi_i = \lambda_i(t) \mathcal{G} \psi_i, \quad \mathcal{A}^L \psi_i = \kappa_i \mathcal{G} \psi_i. \quad (7)$$

Natural empirical estimators for the RKHS operators in (6) are given by

$$\hat{\mathcal{G}}f = \frac{1}{m} \sum_{l=1}^m f(x_l) \Phi(x_l), \quad \hat{\mathcal{A}}^t f = \frac{1}{m} \sum_{l=1}^m f(x_{l+1}) \Phi(x_l), \quad \hat{\mathcal{A}}^L f = \frac{1}{m} \sum_{l=1}^m \mathcal{L}f(x_l) \Phi(x_l).$$

The range of these operators is in the linear span of the feature map at the data sites, $\mathbb{H}_m = \text{span}\{\Phi(x_l)\}_{l=1}^m$. Therefore, it makes sense to restrict these operators to the m -dimensional space \mathbb{H}_m , which leads to the matrix representations

$$\begin{aligned} [\mathbf{K}_X](r, s) &= k(x_r, x_s), \\ [\mathbf{K}_X^t](r, s) &= k(x_{r+1}, x_s), \\ [\mathbf{K}_X^L](r, s) &= \frac{1}{2} \sum_{i,j=1}^d a(i, j)(x_r) \frac{\partial^2}{\partial x^i \partial x^j} k(x_r, x_s) + \sum_{i=1}^d F_i(x_r) \frac{\partial}{\partial x^i} k(x_r, x_s), \end{aligned} \tag{8}$$

and (7) can be replaced by a matrix generalized eigenvalue problem

$$\mathbf{K}_X^t \mathbf{w}_i = \hat{\lambda}_i(t) \mathbf{K}_X \mathbf{w}_i, \quad \mathbf{K}_X^L \mathbf{w}_i = \hat{\kappa}_i \mathbf{K}_X \mathbf{w}_i. \tag{9}$$

The derivatives in the definition of \mathbf{K}_X^L are taken with respect to first argument of the kernel function. It is important to note that, as expected for a kernel method, assembling the matrices (8) requires only kernel evaluations at the data sites, along with kernel derivatives and coefficients of the SDE in the generator setting. For reversible dynamics, there is again an alternative symmetric formulation, which leads to the following Hermitian generalized eigenvalue problem instead of (9):

$$\begin{aligned} \mathbf{K}_X^{\text{rev}} \mathbf{w}_i &= \hat{\kappa}_i \mathbf{K}_X \mathbf{K}_X \mathbf{w}_i, \\ \mathbf{K}_X^{\text{rev}}(r, s) &= -\frac{1}{2m} \sum_{l=1}^m \nabla^\top k(x_l, x_r) a(x_l) \nabla k(x_l, x_s). \end{aligned} \tag{10}$$

The derivations of (8), (9) and (10) were shown in^{23,24}, we repeat them for the reader's convenience in Appendix A.

The linear problems (9) and (10) represent large-scale generalized eigenvalue problems. Solving these problems efficiently for large data sets is one of the central challenges common to most applications of kernel methods. In the following section, we present an approach based on a randomized low-rank representation obtained from Fourier transformation of the kernel function.

III. LOW-RANK REPRESENTATIONS OF KERNEL EIGENVALUE PROBLEMS

In this section, we will present our main contribution: an approach to solving the generalized eigenvalue problems (9) and (10) using a randomized low-rank decomposition.

A. Random Fourier Features

Random Fourier Features (RFFs) have been introduced in²⁶ as a low-rank approximation framework for large kernel matrices. The basis for this approach is Bochner’s theorem, stating that a translation-invariant kernel k , which is normalized to $k(x, x) = 1$ for all x , possesses the following stochastic representation³²:

$$k(x, y) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}^d} e^{-i(x-y)^\top \omega} d\rho(\omega) = \mathbb{E}^{\omega \sim \rho} \left[e^{-ix^\top \omega} \overline{e^{-iy^\top \omega}} \right].$$

Here, ρ is a probability measure in frequency space, called the spectral measure. This representation extends to the derivatives of the kernel function, namely

$$D^\alpha k(x, y) = \mathbb{E}^{\omega \sim \rho} \left[(-i\omega)^\alpha e^{-ix^\top \omega} \overline{e^{-iy^\top \omega}} \right], \quad (11)$$

where $\alpha \in \mathbb{N}^d$ is a multi-index and $\omega^\alpha = (\omega^1)^{\alpha(1)} \dots (\omega^d)^{\alpha(d)}$. The derivatives in the above expression are again taken with respect to the first argument. For the Gaussian RBF kernel with bandwidth σ , the spectral measure is also Gaussian with bandwidth σ^{-1} . For the periodic Gaussian kernel, the spectral measure is discrete, supported on all wave vectors $\frac{\pi}{L}\mathbf{k}$ for $\mathbf{k} \in \mathbb{Z}^d$, with probabilities

$$\rho\left(\frac{\pi}{L}\mathbf{k}\right) = \prod_{j=1}^d I_{\mathbf{k}(j)}(\sigma^{-2}) e^{-\sigma^{-2}},$$

where I_k is the modified Bessel function of the first kind, see³³. Thus, drawing independent samples from the spectral measure is easily accomplished for many popular kernel functions.

B. Representation of Kernel Matrices

We apply the RFF representation to the generalized eigenvalue problems (9) and (10) by substituting it into the kernel matrices $\mathbf{K}_X, \mathbf{K}_X^t, \mathbf{K}_X^L$ given by (8). We again denote samples in real space (in the form of a long trajectory¹) by $\{x_l\}_{l=1}^{m+1}$. Additionally, we assume there are p samples $\{\omega_u\}_{u=1}^p$ in frequency space, sampled from the spectral measure ρ . Replacing expectation values by finite-sample averages, we obtain

$$\begin{aligned} \mathbf{K}_X &= [k(x_r, x_s)]_{r,s} = \left[\mathbb{E}^{\omega \sim \rho} \left[e^{-ix_r^\top \omega} \overline{e^{-ix_s^\top \omega}} \right] \right]_{r,s} \approx \frac{1}{p} [\mathbf{M}\mathbf{M}^H]_{r,s}, \\ \mathbf{K}_X^t &= [k(x_{r+1}, x_s)]_{r,s} \approx \frac{1}{p} [\mathbf{M}^t \mathbf{M}^H]_{r,s}, \end{aligned} \quad (12)$$

¹ Instead of using one long trajectory, it would also be possible to extract a training data set $\{(x_l, y_l)\}_{l=1}^m$ from many short trajectories, see³⁴ for details.

with RFF feature matrices

$$\mathbf{M} = \left[e^{-ix_r^\top \boldsymbol{\omega}_u} \right]_{r,u}, \quad \mathbf{M}^t = \left[e^{-ix_{r+1}^\top \boldsymbol{\omega}_u} \right]_{r,u}. \quad (13)$$

That is, the rows of \mathbf{M} correspond to the data points and the columns of \mathbf{M} to the randomly sampled features. When approximating the generator, we use the representation of derivatives given in (11), to find

$$\begin{aligned} \mathbf{K}_X^L &= \left[\frac{1}{2} \sum_{i,j=1}^d a(i,j)(x_r) \frac{\partial^2}{\partial x^i \partial x^j} k(x_r, x_s) + \sum_{i=1}^d F(i)(x_r) \frac{\partial}{\partial x^i} k(x_r, x_s) \right]_{r,s} \\ &\approx \frac{1}{p} \sum_{u=1}^p \left[\left(-\frac{1}{2} \sum_{i,j=1}^d a(i,j)(x_r) \boldsymbol{\omega}_u(i) \boldsymbol{\omega}_u(j) - i \sum_{i=1}^d F(i)(x_r) \boldsymbol{\omega}_u(i) \right) e^{-ix_r^\top \boldsymbol{\omega}_u} \overline{e^{-ix_s^\top \boldsymbol{\omega}_u}} \right]_{r,s} \\ &= \frac{1}{p} [\mathbf{M}^L \mathbf{M}^H]_{r,s}, \end{aligned} \quad (14)$$

with generator feature matrix \mathbf{M}^L , which can be written compactly using the Frobenius inner product as

$$\mathbf{M}^L = \left[\left(-\frac{1}{2} a(x_r) : (\boldsymbol{\omega}_u \otimes \boldsymbol{\omega}_u) - i F(x_r) \cdot \boldsymbol{\omega}_u \right) e^{-ix_r^\top \boldsymbol{\omega}_u} \right]_{r,u}.$$

We will address the reversible case further below after analyzing the RFF-based approximation in more detail. Together, (12) and (14) provide low-rank approximations of the kernel matrices which can be easily assembled by evaluation of complex exponentials.

C. Solution of the Generalized Eigenvalue Problems

Let us now use the approximations (12) and (14) to solve the generalized eigenvalue problems (9), which read (note that the normalization $\frac{1}{p}$ can be omitted):

$$\mathbf{M}^t \mathbf{M}^H \mathbf{w}_i = \hat{\lambda}_i(t) \mathbf{M} \mathbf{M}^H \mathbf{w}_i, \quad \mathbf{M}^L \mathbf{M}^H \mathbf{w}_i = \hat{\kappa}_i \mathbf{M} \mathbf{M}^H \mathbf{w}_i. \quad (15)$$

If $p \leq m$, we can employ a standard trick to express this equivalently as a lower-dimensional eigenvalue problem. Defining $\mathbf{v}_i = \mathbf{M}^H \mathbf{w}_i$, we can verify directly that

$$\mathbf{M}^H \mathbf{M}^t \mathbf{v}_i = \mathbf{M}^H \mathbf{M}^t \mathbf{M}^H \mathbf{w}_i = \hat{\lambda}_i(t) \mathbf{M}^H \mathbf{M} \mathbf{M}^H \mathbf{w}_i = \hat{\lambda}_i(t) \mathbf{M}^H \mathbf{M} \mathbf{v}_i.$$

The last equality shows that all non-zero eigenvalues of (15) can be equivalently computed by the lower-dimensional *dual* problem:

$$\mathbf{M}^H \mathbf{M}^t \mathbf{v}_i = \hat{\lambda}_i(t) \mathbf{M}^H \mathbf{M} \mathbf{v}_i, \quad \mathbf{M}^H \mathbf{M}^L \mathbf{v}_i = \hat{\kappa}_i \mathbf{M}^H \mathbf{M} \mathbf{v}_i. \quad (16)$$

This p -dimensional generalized eigenvalue equation is the central problem to be solved within the context of RFF-based kernel approximation. If p is much smaller than m , (16) already leads to computational savings compared to (9).

In fact, it is not even necessary to assemble the matrices in (16), because the equation can be solved by operating only on the RFF feature matrices $\mathbf{M}, \mathbf{M}^t, \mathbf{M}^L$. To this end, we apply a whitening transformation, which is widely used within the context of (g)EDMD. Using a truncated singular value decomposition of \mathbf{M} ,

$$\mathbf{M} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{W}^H,$$

and retaining $r \leq p$ components of the SVD, we obtain a linear transformation $\mathbf{T} = \mathbf{W}\mathbf{\Sigma}^{-1}$, which eliminates the matrix on the right-hand sides of (16):

$$\mathbf{T}^H \mathbf{M}^H \mathbf{M} \mathbf{T} = \mathbf{\Sigma}^{-1} \mathbf{W}^H \mathbf{W} \mathbf{\Sigma} \mathbf{U}^H \mathbf{U} \mathbf{\Sigma} \mathbf{W}^H \mathbf{W} \mathbf{\Sigma}^{-1} = \text{Id}_{r \times r}.$$

Applying the same transformation to the left-hand sides of (16) leads to the reduced matrices

$$\mathbf{R}^t = \mathbf{T}^H \mathbf{M}^H \mathbf{M}^t \mathbf{T} = \mathbf{U}^H \mathbf{M}^t \mathbf{W} \mathbf{\Sigma}^{-1}, \quad \mathbf{R}^L = \mathbf{T}^H \mathbf{M}^H \mathbf{M}^L \mathbf{T} = \mathbf{U}^H \mathbf{M}^L \mathbf{W} \mathbf{\Sigma}^{-1}. \quad (17)$$

It is then sufficient to diagonalize these reduced matrices in order to solve (16). If the truncation rank of \mathbf{M} is significantly smaller than p , additional computational savings can be harnessed this way. Algorithm 1 summarizes our solution method in compact form.

D. Interpretation as (g)EDMD Problem

The transformation of the dual problem (16) to its reduced form is completely analogous to what is called the whitening transformation for (g)EDMD²⁹. This is not a coincidence: consider a finite dictionary defined by

$$\phi_{\text{RFF}}(x) := \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_p(x) \end{bmatrix} = \begin{bmatrix} e^{ix^\top \omega_1} \\ \vdots \\ e^{ix^\top \omega_p} \end{bmatrix}.$$

Using the notation of Section II C, the empirical mass and stiffness matrices for this basis set

are given by

$$\begin{aligned}\hat{\mathbf{G}}(u, v) &= \frac{1}{m} \sum_{l=1}^m \phi_u(x_l) \overline{\phi_v(x_l)} = \frac{1}{m} \sum_{l=1}^m e^{ix_l^\top \omega_u} e^{-ix_l^\top \omega_v} = \frac{1}{m} [\mathbf{M}^H \mathbf{M}] (u, v), \\ \hat{\mathbf{A}}^t(u, v) &= \frac{1}{m} [\mathbf{M}^H \mathbf{M}^t] (u, v), \\ \hat{\mathbf{A}}^L(u, v) &= \frac{1}{m} [\mathbf{M}^H \mathbf{M}^L] (u, v),\end{aligned}$$

which are precisely the constituent matrices in (16)². We conclude that the solution of the transposed problem is equivalent to the eigenvalue estimation based on (g)EDMD using a random dictionary of complex plane waves. With this interpretation at hand, we can also derive a convenient symmetric formulation for reversible dynamics, upon replacing the estimator for $\hat{\mathbf{A}}^L$ by its symmetric version (3):

$$\begin{aligned}\hat{\mathbf{A}}^L(u, v) &= -\frac{1}{2m} \sum_{l=1}^m \nabla^\top \phi_u(x_l) a(x_l) \nabla \phi_v(x_l) \\ &= -\frac{1}{2m} \sum_{l=1}^m \omega_u^\top a(x_l) \omega_v e^{ix_l^\top \omega_u} e^{-ix_l^\top \omega_v}.\end{aligned}$$

To compute the reduced matrix, we just have to apply the transformation \mathbf{T} from Section III C from left and right, i.e.,

$$\mathbf{R}^L = \mathbf{T}^H \hat{\mathbf{A}}^L \mathbf{T} = \Sigma^{-1} \mathbf{W}^H \hat{\mathbf{A}}^L \mathbf{W} \Sigma^{-1}, \quad (18)$$

resulting again in a symmetric estimator.

² We need to consider complex-valued observables at this point, leading to complex conjugation of the second argument in inner products.

Algorithm 1 RFF-based Spectral Approximation of the Operator or Generator

Input: data matrix $\mathbf{X} = [x_1, \dots, x_{m+1}] \in \mathbb{R}^{d \times (m+1)}$,
kernel function k with spectral measure ρ ,
number of features p , truncation rule for singular values.

Output: Approximate eigenpairs $(\hat{\lambda}_i(t), \hat{\psi}_i)$ or $(\hat{\kappa}_i, \hat{\psi}_i)$ of the dynamical operator.

- 1: Draw p samples $\{\omega_u\}_{u=1}^p$ from the spectral measure ρ .
 - 2: Form matrix \mathbf{M} given by (13).
 - 3: Non-reversible case: form matrix \mathbf{M}^t or \mathbf{M}^L as in (13) or (14).
 - 4: Compute SVD of \mathbf{M} , choose rank r according to truncation rule: $\mathbf{M} \approx \mathbf{U}\Sigma\mathbf{W}^H$.
 - 5: Form reduced matrix \mathbf{R}^t or \mathbf{R}^L according to (17) or (18).
 - 6: Compute eigenpairs of reduced problem $\mathbf{R}^t \mathbf{u}_i = \hat{\lambda}_i(t) \mathbf{u}_i$ or $\mathbf{R}^L \mathbf{u}_i = \hat{\kappa}_i \mathbf{u}_i$.
 - 7: Transform to original RFF basis: $\mathbf{v}_i = \mathbf{T} \mathbf{u}_i$, $\hat{\psi}_i(x) = \mathbf{v}_i^H \phi_{\text{RFF}}(x)$.
-

E. Computational Effort

Let us break down the computational effort required for Algorithm 1: The SVD of \mathbf{M} requires $\mathcal{O}(mp^2)$ operations. Assembling the reduced matrix costs $\mathcal{O}(rpm)$ operations for the Koopman operator. In the case of generator approximation, we obtain an additional dependence on the dimension, due to the required contraction of derivatives of the kernel function. We can upper-bound the resulting effort as $\mathcal{O}(mp(d^2 + r))$ in the non-reversible case and by $\mathcal{O}(m(rpd + r^2d^2))$ in the reversible case. Note that this bound assumes a dense and state-dependent diffusion field a , it reduces to linear dependence on d for diagonal diffusions, such as overdamped Langevin dynamics. Finally, the diagonalization of the reduced matrix amounts to $\mathcal{O}(r^3)$ floating point operations. Comparing this to full kernel method, we see that the most drastic computational savings will result from being able to choose p small compared to m , as it reduces a factor $\mathcal{O}(m^3)$ to $\mathcal{O}(mp^2)$.

IV. METHODS

In this section, we will discuss aspects of the practical application of our spectral RFF method and also introduce the systems studied in the results section below. For the numerical examples,

we will use the Gaussian RBF kernel or its periodic counterpart for periodic domains.

A. Model Selection

Before we can apply Algorithm 1, we have to choose the number of features p , the truncation threshold for singular values, and as many hyper-parameters as required by the kernel function (e.g., the bandwidth for Gaussian RFF kernels). We select these parameters by cross-validating the VAMP score metric on randomly selected subsets of the data, as explained below. The only exception is the truncation threshold for singular values of \mathbf{M} . Here, we adopt the rule that all singular values smaller than 10^{-4} times the largest singular value are discarded. This choice is based on prior experience and works well in all considered examples.

The VAMP score metric is based on the *Rayleigh variational principle*, which states that for reversible systems, the exact leading K eigenfunctions are optimizers of the Rayleigh trace

$$\sum_{i=0}^{K-1} \lambda_i(t) = \max_{\phi_0, \dots, \phi_{K-1}} \sum_{i=0}^{K-1} \langle \phi_i, \mathcal{K}^t \phi_i \rangle_{\mu}, \quad \text{s.t. } \langle \phi_i, \phi_j \rangle_{\mu} = \delta_{ij}.$$

The sum to be maximized on the right-hand side is a special case of the slightly more general *VAMP score* introduced for Koopman operators in^{7,9} and for the Koopman generator in³⁵. When considering the generator, the maximum becomes a minimum instead. The variational principle also generalizes to non-reversible systems by considering singular functions instead of eigenfunctions⁹. However, we only analyze reversible systems in what follows. The VAMP score can be easily estimated upon replacing inner products by their empirical estimators, as shown in Section II C. The maximal VAMP score within a finite-dimensional model class can also be straightforwardly computed, see again⁷. In this way, we can simply compare the maximal VAMP score obtained for each choice of hyper-parameters, e.g., choice of kernel bandwidth σ and number of Fourier features p , and choose the best.

To account for the estimation error stemming from the use of finite data, and also to avoid overfitting, we employ a standard cross-validation approach, and first compute the optimal functions $\phi_0, \dots, \phi_{K-1}$ for each model class on a random subsample of the data (training data). We then recompute the VAMP score for the K -dimensional space spanned by $\phi_0, \dots, \phi_{K-1}$ on the remaining data (test data). This process is repeated n_{test} times, where n_{test} is either 10 or 20 below, and we choose the model class optimizing the test score on average. The ratio of training data to test data is always chosen as a 75/25 percent split.

B. Spectral Analysis and Clustering

Solution of the eigenvalue problem for the reduced matrix \mathbf{R} provides estimates of the leading eigenvalues $\hat{\lambda}_i(t)$ or $\hat{\kappa}_i$, respectively, and their associated eigenfunctions $\hat{\psi}_i$. The first point to note is that (16) is a complex eigenvalue problem, so the eigenvectors can be scaled by arbitrary complex numbers. Before analyzing the eigenvectors, we simply apply a grid search over complex phase factors to determine the one that minimizes the imaginary part of all eigenvector entries and use the rescaled eigenvectors for further analysis.

Second, we transform the eigenvalues of the Koopman operator into *implied timescales* defined as

$$t_i = -\frac{t}{\log(\hat{\lambda}_i(t))},$$

which have units of time, and can be interpreted as physical relaxation timescales for the i -th eigenmode of the Koopman operator. Since timescales are independent of the lag time in the absence of projection and estimation error³⁶, we monitor convergence of implied timescales with increasing lag time t in order to select an optimal lag time¹¹. We also compute the timescales when approximating the generator, in this case they are simply given by

$$t_i = -\frac{1}{\hat{\kappa}_i}.$$

In order to analyze metastable sets, we evaluate the top eigenfunctions $\hat{\psi}_i$ at all data points, and apply the spectral clustering method PCCA³⁷ to assign each data point x_l to each metastable set q with a degree of membership $\chi_q(x_l)$, which is a number between zero and one. We then identify data point x_l as part of metastable set q if $\chi_q(x_l) \geq 0.6$, the remaining data points are treated as transition states.

C. Systems and Simulation Setups

We study four different examples in order to illustrate different aspects of the performance of the proposed method. The first is overdamped Langevin dynamics (2) in a two-dimensional model potential, which is a minor modification of the *Lemon-Slice potential*³⁸, given in polar coordinates as

$$V(r, \varphi) = \cos(4\varphi) + 10(r-1)^2 + 1 + \frac{1}{r} + \frac{1}{\cos \varphi}, \quad (19)$$

see Figure 1 for a visualization. We can easily generate large amounts of simulation data by applying the standard Euler–Maruyama integrator, the elementary simulation time step is set to $\Delta_t = 10^{-3}$. Friction γ and inverse temperature β are both set to one. Reference implied timescales are obtained from a Markov state model based on a k -means discretization of the two-dimensional state space, using 50 discrete states and $m = 10^5$ data points.

The second system is molecular dynamics simulation data of the alanine dipeptide in explicit water, at temperature $T = 300$ K, employing the Langevin thermostat. We generated a total of one million data points at 1 ps time spacing. Reference implied timescales were computed by a Markov state model in the well-known two-dimensional space of backbone dihedral angles ϕ and ψ , using a 30×30 box discretization and all one million data points.

The third example is molecular dynamics simulation data of the deca alanine peptide in explicit water at temperature $T = 300$ K, see³⁹ for a description of the simulation setup. The data comprises a total of four million time steps at 1 ps time spacing. As reference values, we refer to the MSM analysis presented in⁴⁰, which is built on a 500-state k -means discretization after applying time-lagged independent component analysis (TICA)¹⁰ in the space of the peptide’s sixteen backbone dihedral angles.

Finally, we re-analyze the 39-residue protein NTL9, which was simulated on the Anton supercomputer by D.E. Shaw Research, see⁴¹ for details. The data comprises around 1.2 million frames corresponding to a time spacing of $\Delta_t = 2$ ns. As a reference, we apply linear TICA on the set of 666 minimal heavy atom inter-residue distances. We also rank these distance according to the fraction of simulation time where a contact is formed, that is, the distance does not exceed 0.35 nm, see⁴⁰, and calculate TICA models using only the first [20, 100, 200, 300, 400, 500, 666] of these distances.

V. RESULTS

A. Lemon-Slice Potential

We start by analyzing the Lemon-Slice potential (19), shown in Figure 1 A, which features four metastable states corresponding to the wells of the potential energy. We use this system as a first test case for approximating the generator \mathcal{L} , due to the availability of all relevant quantities in analytical form.

Figure 1 B shows the VAMP score as a function of the bandwidth for a selection of data sizes m and feature numbers p . We see that while $m = 1000$ data points are not sufficient, for $m = 5000$ data points, the score stabilizes for a range of bandwidths $\sigma \in [0.3, 0.8]$, and already attains optimal values for very small p . We confirm this finding by plotting the first three non-trivial eigenvalues of the generator, for $m = 5000$ and $p = 50$, as a function of the bandwidth in Figure 1 C. Indeed, accurate estimates for eigenvalues can be obtained in the suggested regime of bandwidths. After calculating the eigenvectors of the reduced matrix for $\sigma = 0.4$, and applying the PCCA analysis, we see that the metastable structure is perfectly recovered, see Figure 1 D. Thus, the VAMP score reliably guides the hyper-parameter search towards a very efficient approximation of the leading spectral components of this system.

B. Alanine Dipeptide

For alanine dipeptide, we approximate the Koopman operator with lag time t , using the periodic Gaussian kernel on the well-known two-dimensional reaction coordinate space given by the dihedral angles ϕ and ψ . Figure 2 A shows the familiar free energy landscape of the system in the reaction coordinate space. For all RFF-based approximations, we downsample the data set to $m = 20000$ points, corresponding to a time spacing of 50 ps. We apply the same model selection protocol as before, and find that a range of kernel bandwidths $\sigma \in [0.4, 1.0]$ can be stably identified as optimal in terms of the VAMP score. We observe in Figure 2 B that again a small number of Fourier features $p = 50$ is sufficient to arrive at an optimal VAMP score. We also confirm that the optimal parameter regime is stable across different lag times. Note that the VAMP score necessarily decreases with increasing lag time, as all eigenvalues but $\lambda_0(t)$ decay exponentially with t . As shown in Figure 2 C, the optimal range of bandwidths indeed allows for accurate estimation of the three leading implied timescales of the system. Further, we also verify that the metastable decomposition of the dihedral space into four states is correctly recovered using the RFF model, see Figure 2 D. The dramatic rank reduction for the kernel matrix achieved by using a small number of Fourier features allows us to easily compute eigenvalues and test scores for a broad range of parameters, using $m = 20000$ data points in real space. Solving the same problem by means of standard methods would be significantly more costly if the same data size was used.

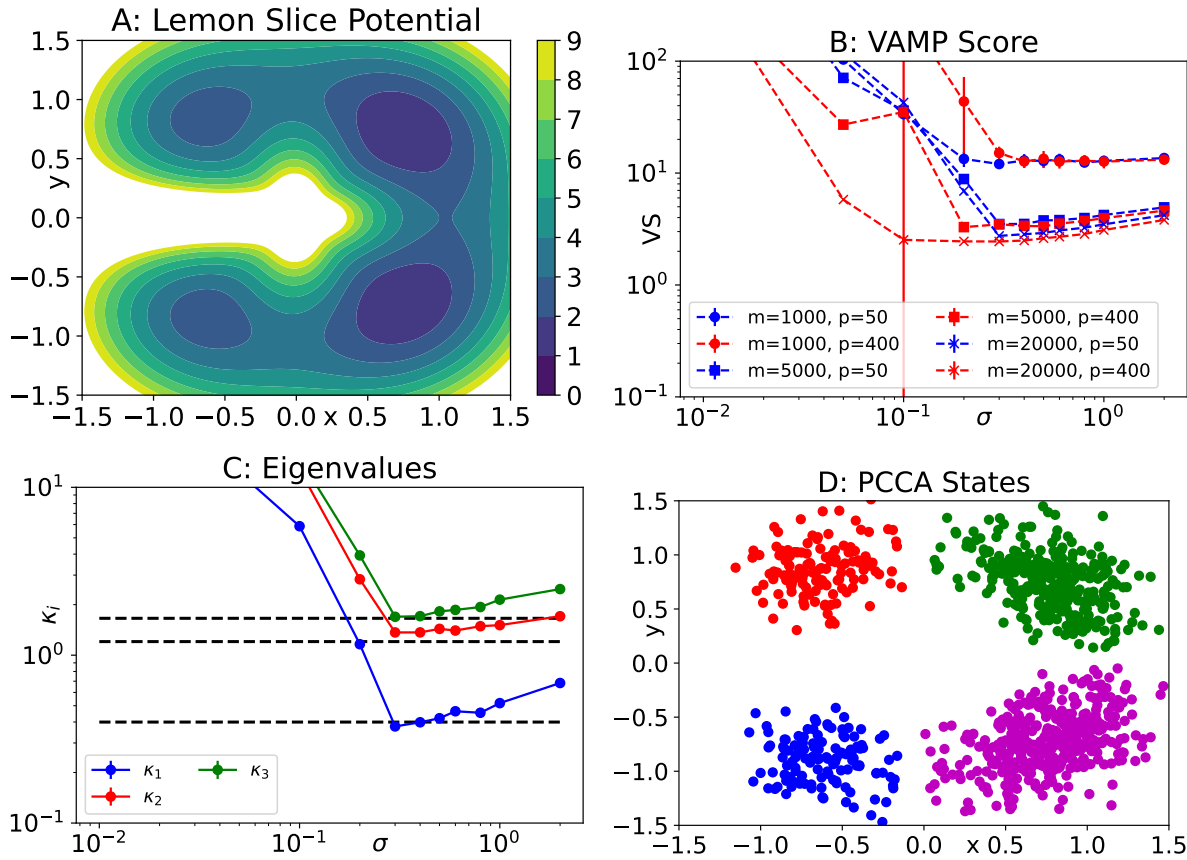


FIG. 1. Results for the Lemon-Slice potential. A: Contour of the potential (19). B: VAMP score as a function of the kernel bandwidth for different data sizes m and feature numbers p . C: Leading non-trivial eigenvalues of the generator for $m = 5000$ and $p = 50$, as a function of the bandwidth. Black lines indicate the Markov model reference values. D: Decomposition into four metastable states based on eigenvectors for $\sigma = 0.4$, $p = 50$, $m = 5000$. All error bars are based on twenty independent simulations.

C. Deca Alanine

We use the deca alanine example to demonstrate that, by means of low-rank kernel methods, generator models can be efficiently learned on a reaction coordinate space that is more than just one- or two-dimensional, and that dynamical hypotheses can be tested this way. We project the system onto the space of its sixteen interior backbone torsion angles. The generator of the full molecular dynamics is not readily available in closed form. It is well-known, however, that if observed only on a subset of position space, many systems driven by thermostatted molecular dynamics behave like the reversible overdamped Langevin dynamics at long timescales⁴². For

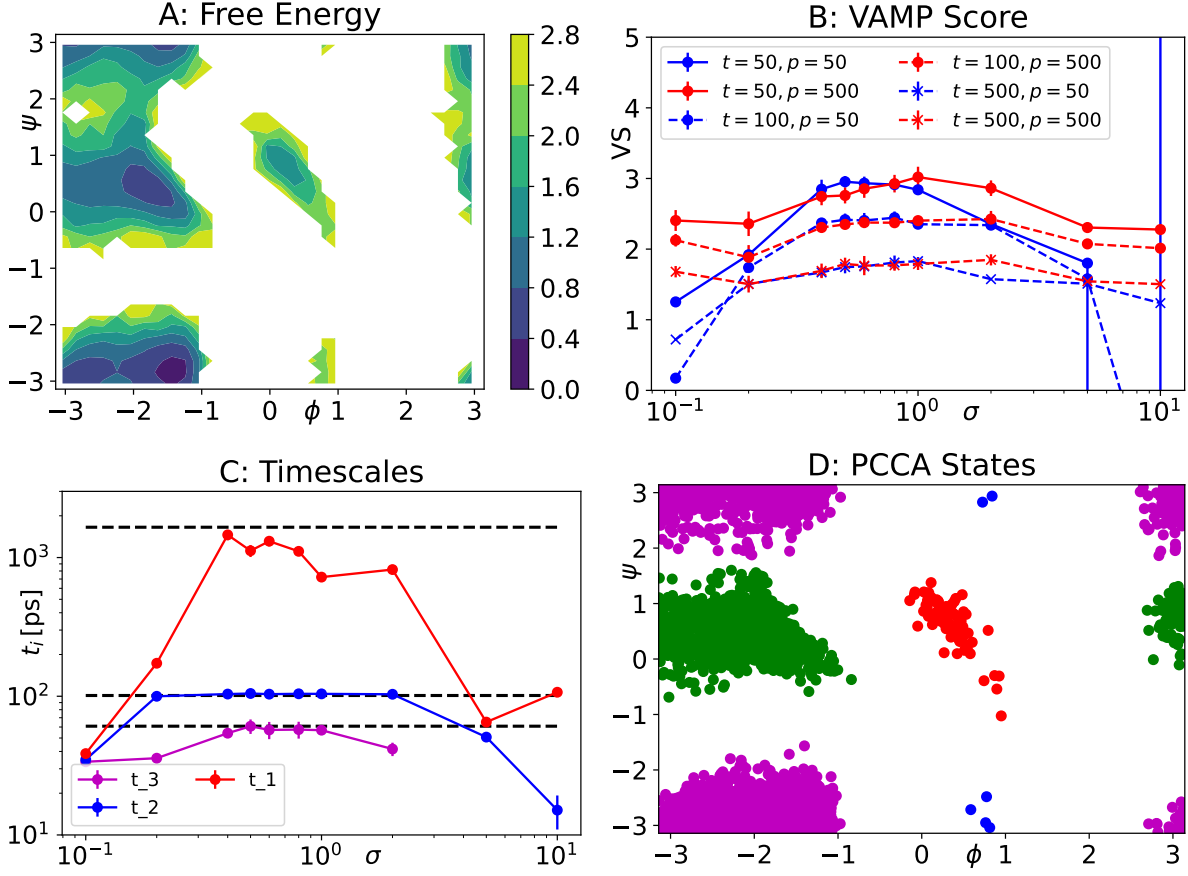


FIG. 2. Results for alanine dipeptide. A: Free Energy (in kJ/mol) in two-dimensional dihedral space. B: VAMP Score for selected feature sizes p and lag times t as a function of the kernel bandwidth. C: Implied timescales for $t = 100$ ps and $p = 50$ as a function of the bandwidth. D: Metastable decomposition obtained for $\sigma = 0.6$, $p = 50$ and $t = 100$ ps.

that reason, we set the full space diffusion matrix to $\frac{1}{\beta}$ times the identity and employ our reversible estimator. The effective diffusion (5) then turns into

$$a^\xi(z) = \frac{1}{\beta} \nabla \xi^\top(x) \nabla \xi(x),$$

where $\nabla \xi$ is the Jacobian of the mapping from Euclidean coordinates to the peptide's backbone dihedral angles, which can be evaluated analytically. Note that this approach can be seen as a dynamical hypothesis: we choose to model the position space dynamics as an overdamped Langevin process, and test if its projection onto the dihedral angle space can recover all relevant features of the full MD simulation.

We find in Figure 3 A that a kernel bandwidth $\sigma \in [2.0, 8.0]$ leads to an optimal VAMP score for

the projected generator. Using about $m = 1000$ data points in real space and $p = 300$ Fourier features is sufficient to obtain stable results, rendering the calculation highly efficient. The optimality of this regime of bandwidths is confirmed by plotting the first three non-trivial eigenvalues of the generator, as shown in Figure 3 B. We note that these eigenvalues differ from those of the reference MSM by a constant factor. This is due to the re-scaling of time inherent in assuming overdamped Langevin dynamics in position space, rendering the effective dynamics too fast. However, we also see in Figure 3 that all slow eigenvalues are indeed re-scaled by the same factor. Moreover, the PCCA analysis of the slowest eigenvectors obtained for the optimal RFF model also recovers the metastable states of the deca alanine system, corresponding to the folded and unfolded states as well as two intermediates. The effective overdamped Langevin model learned by the kernel method thus retains all major features of the original molecular dynamics, up to the re-scaling of time.

D. NTL9

We use the NTL9 data set to demonstrate once again the ability of the proposed method to efficiently and robustly analyze large data sets. We downsample the available trajectory data to around $m = 15.000$ data points at 200 ns time spacing. As mentioned before, $m = 15.000$ is still a significant data size for kernel methods. We project the MD simulations onto the 666-dimensional space of minimal heavy atom inter-residue distances. It is known from previous publications that these coordinates capture the folding process of NTL9 very well^{40,43}. Indeed, just using linear TICA on the top-ranked 300 distances allows for an accurate estimation of the slowest implied timescale t_1 , associated to the folding process, see the top green line in Figure 4 B.

Figure 4 A shows that using Gaussian bandwidth parameters $\sigma \in [10, 50]$ can be identified as optimal in terms of the VAMP score, again across different lag times. In Figure 4 B, we first compare the slowest timescale t_1 obtained for $\sigma = 15$ as a function of the lag time, for different feature numbers p (blue lines). We see that already for $p \geq 300$, satisfactory convergence of t_1 with t can be obtained, reducing the computational cost by about a factor two thousand compared to the full kernel eigenvalue problem. Interestingly, timescale convergence improves with p in about the same way as the performance of TICA improves as more and more of the ranked distances are added. However, we show that the kernel method is more robust with respect to the selection of input coordinates, by randomly selecting $d = 50$ out of the 666 distances, and recomputing both

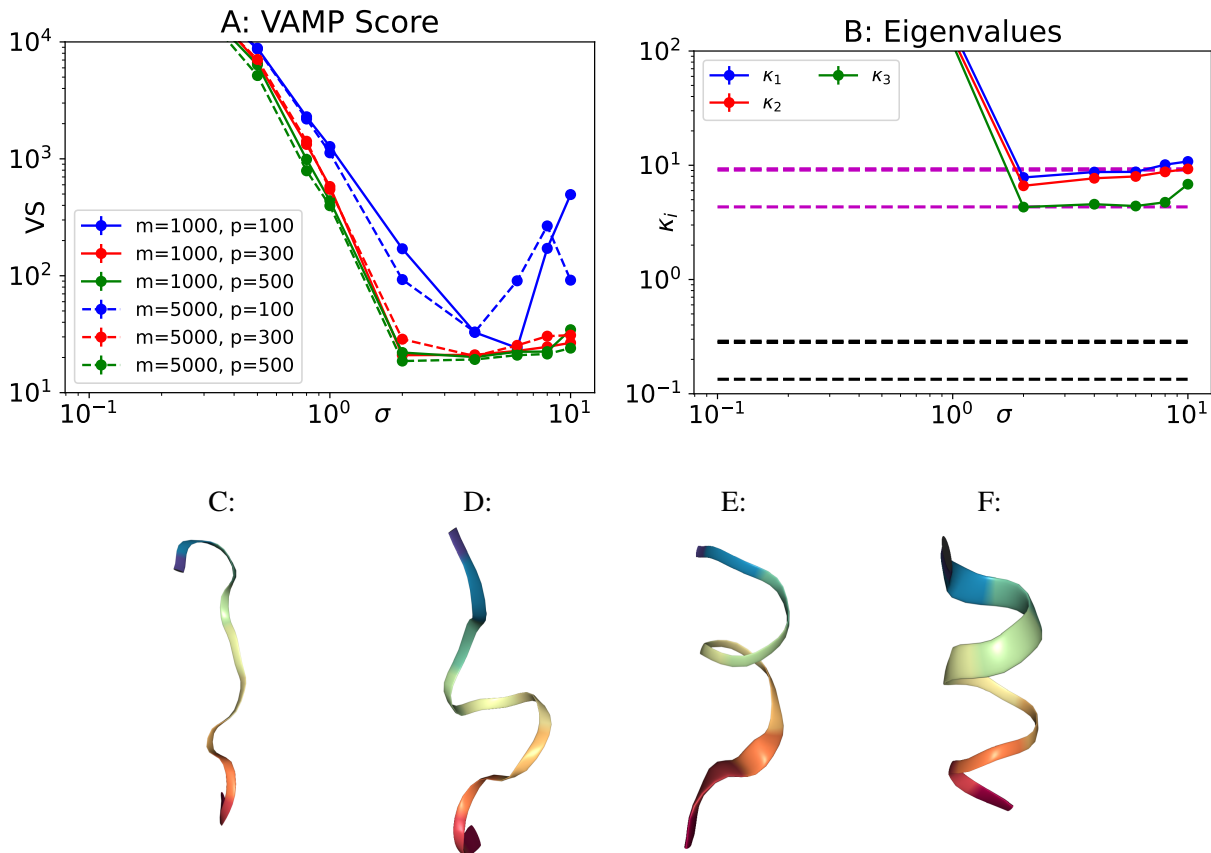


FIG. 3. Results for the Koopman generator on backbone dihedral angle space of the deca alanine peptide. A: VAMP score for selected data sizes m and feature sizes p as a function of the kernel bandwidth. B: Eigenvalues for $m = 1000$ and $p = 300$ as a function of the bandwidth. The reference MSM results are shown in black. Re-scaling MSM eigenvalues by the average ratio between optimal RFF and MSM eigenvalues leads to the magenta lines. C–F: Representative structures for each of the four PCCA states based on the RFF model at $m = 1000$, $p = 300$, $\sigma = 4.0$.

the VAMP score and implied timescales for different bandwidths, keeping the number of Fourier features fixed at $p = 300$. We see in Figures 4 A–B that both the optimal value of the VAMP score as well as the timescale estimates remain stable across random selections of the input distances (magenta lines in panel A, red lines in panel B). Convergence of timescales is therefore mainly dependent on the number of Fourier features p , which can be automatically tuned using the VAMP score, and not on the selection of input coordinates. Moreover, we also find that under the random distance selection, the optimal regime of kernel bandwidths changes as the dimension of the input coordinate space is significantly smaller. This is reliably reflected by the behavior of the VAMP score, see again Figure 4 A.

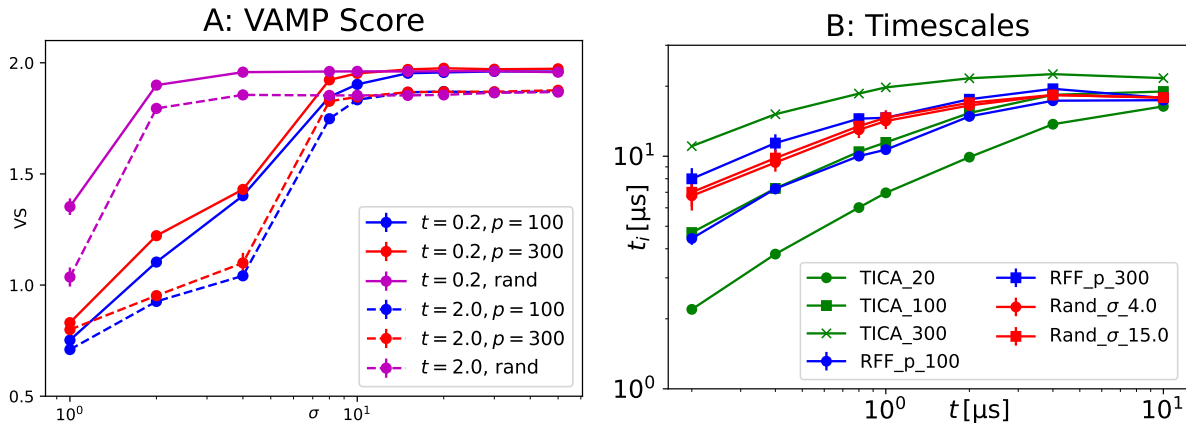


FIG. 4. Results for NTL9 protein. A: VAMP score for Gaussian RFF approximation as a function of the kernel bandwidth σ . Red and blue lines show the results using all 666 distances, for different lag times and different numbers of Fourier features p . The magenta lines show the average values over ten random selections of only 50 distances, with $p = 300$ fixed. B: Slowest implied timescale t_1 as a function of the lag time t . Green lines show estimates based on linear TICA using the top-ranked 20, 100, and 300 distances. Blue lines show RFF-based estimates on all distances using different values of p , using $\sigma = 15.0$. Red lines are averages over the above-mentioned random subsamples of the distance coordinates, where $p = 300$ is fixed.

VI. CONCLUSIONS

We have introduced a new approach to extract the dominant eigenvalues of dynamical operators associated with stochastic processes, corresponding to the slow timescale dynamics in molecular kinetics. Building on kernel representations of the Koopman operator and generator, our main novelty is a dual low-rank representation of the kernel matrices using random Fourier features. We have derived the corresponding reduced eigenvalue problem, and provided an interpretation of the method in terms of a random dictionary of plane waves. Using four examples of increasing complexity, we have shown that hyper-parameters of the method can be very effectively tuned using the VAMP score metric. Moreover, we have shown that only a few hundred Fourier features were sufficient to obtain accurate and robust results for all systems considered. This finding allowed us to rapidly scan kernel hyper-parameters for data sizes in the ten thousands, on coordinate spaces of dimension up to several hundred. The computational savings compared to the full eigenvalue problem, measured in floating point operations, were on the order of three to four orders of

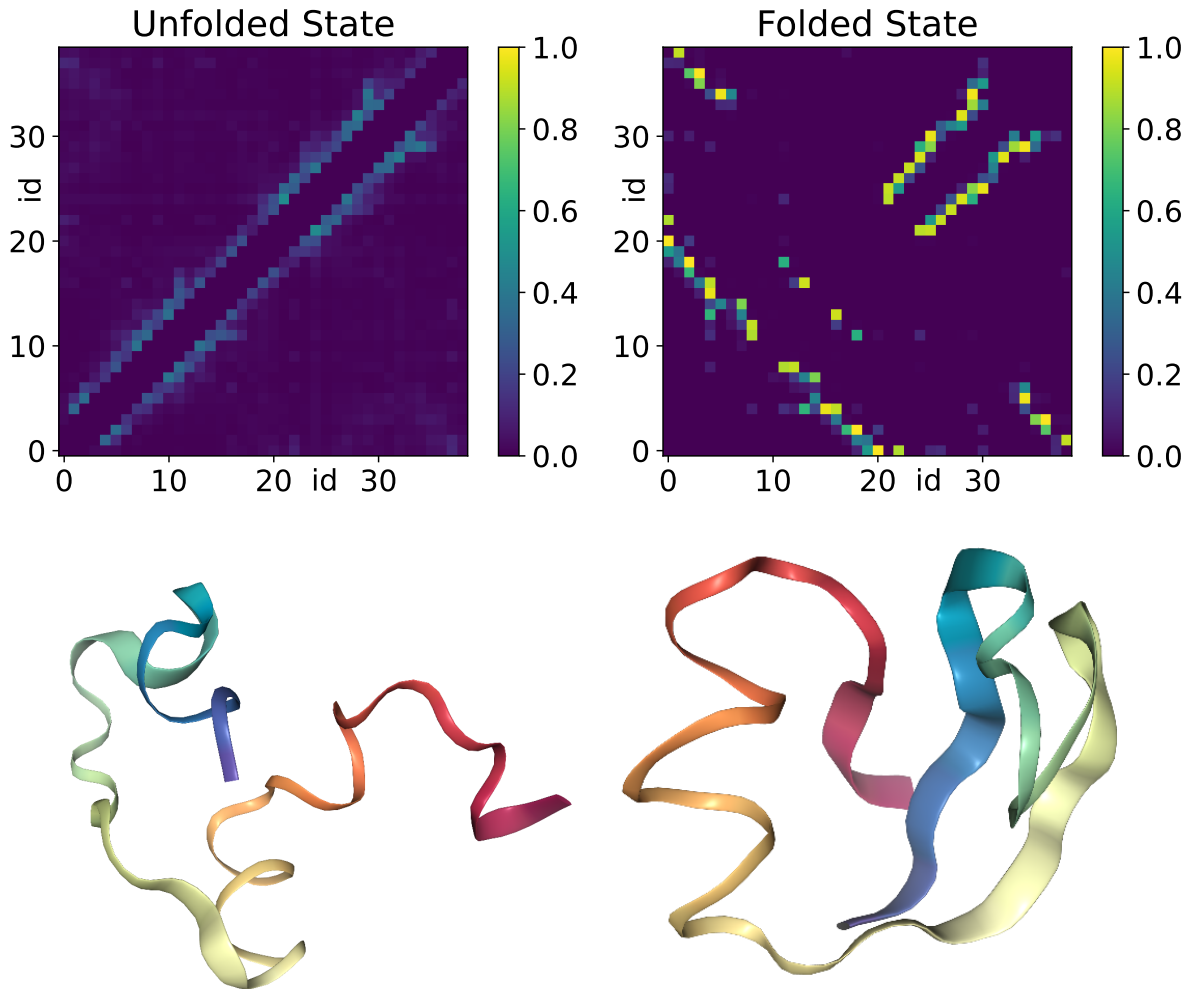


FIG. 5. Top Row: Representations of the two PCCA states obtained for $\sigma = 15$, $p = 300$, $t = 2 \mu s$. For each PCCA state, we show the fraction of simulation time during which each residue-residue pair forms a contact, see also⁴⁰. Bottoms Row: Representative protein structure for both PCCA states.

magnitude.

The main promise of our method is the ability to analyze large data sets using a very generic, and in fact randomly generated basis set. The success on the method depends on the number of required Fourier features. Future research will need to test if this number indeed remains small even for larger systems, which would make the method scalable beyond the benchmarking examples studied here. It will also be interesting to see if kernel functions with more specific properties can be used, for instance, kernels incorporating more detailed symmetries than only translational invariance, or kernels acting on mixed types of domain (periodic and non-periodic). Finally, combining our approach with re-weighting techniques would allow application of the method to en-

hanced sampling simulation data.

ACKNOWLEDGMENTS

The authors thank D.E. Shaw Research for providing the molecular dynamics simulation data of NTL9.

DATA AVAILABILITY STATEMENT

The methods presented in this paper are available as part of the KoopmanLib library³. The complete data and code to reproduce the examples shown above are available from a public repository⁴, with the exception of the raw molecular dynamics simulation data, which can be obtained directly from the authors or, in the case of NTL9, from the copyright owners.

REFERENCES

- ¹P. L. Freddolino, C. B. Harrison, Y. Liu, and K. Schulten, “Challenges in protein-folding simulations,” *Nat. Phys.* **6**, 751–758 (2010).
- ²N. Plattner, S. Doerr, G. De Fabritiis, and F. Noé, “Complete protein–protein association kinetics in atomic detail revealed by molecular dynamics simulations and markov modelling,” *Nat. Chem.* **9**, 1005–1011 (2017).
- ³F. Paul, C. Wehmeyer, E. T. Abualrous, H. Wu, M. D. Crabtree, J. Schöneberg, J. Clarke, C. Freund, T. R. Weikl, and F. Noé, “Protein-peptide association kinetics beyond the seconds timescale from atomistic simulations,” *Nat. Commun.* **8**, 1095 (2017).
- ⁴S. M. Ulam, *A collection of mathematical problems* (Interscience, 1960).
- ⁵M. Dellnitz and O. Junge, “On the approximation of complicated dynamical behavior,” *SIAM J. Numer. Anal.* **36**, 491–515 (1999).
- ⁶C. Schütte, A. Fischer, W. Huisinga, and P. Deuffhard, “A direct approach to conformational dynamics based on hybrid Monte Carlo,” *J. Comput. Phys.* **151**, 146–168 (1999).
- ⁷F. Noé and F. Nüske, “A variational approach to modeling slow processes in stochastic dynamical systems,” *Multiscale Model. Simul.* **11**, 635–655 (2013).

³ <https://github.com/fnueske/KoopmanLib>

⁴ <https://doi.org/10.5281/zenodo.8036525>

- ⁸F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, “Variational approach to molecular kinetics,” *J. Chem. Theory Comput.* **10**, 1739–1752 (2014).
- ⁹H. Wu and F. Noé, “Variational approach for learning Markov processes from time series data,” *J. Nonlinear Sci.* **30**, 23–66 (2020).
- ¹⁰G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, “Identification of slow molecular order parameters for Markov model construction,” *J. Chem. Phys.* **139**, 15102 (2013).
- ¹¹J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, “Markov models of molecular kinetics: Generation and Validation,” *J. Chem. Phys.* **134**, 174105 (2011).
- ¹²G. R. Bowman, V. S. Pande, and F. Noé, eds., *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation.*, Advances in Experimental Medicine and Biology, Vol. 797 (Springer Heidelberg, 2014).
- ¹³B. O. Koopman, “Hamiltonian systems and transformation in Hilbert space,” *Proc. Natl. Acad. Sci. U S A* **17**, 315 (1931).
- ¹⁴I. Mezić, “Spectral properties of dynamical systems, model reduction and decompositions,” *Nonlinear Dyn.* **41**, 309–325 (2005).
- ¹⁵M. 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).
- ¹⁶S. Klus, P. Koltai, and C. Schütte, “On the numerical approximation of the Perron-Frobenius and Koopman operator,” *J. Comput. Dyn.* **3**, 51–79 (2016).
- ¹⁷S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, “Data-driven approximation of the Koopman generator: Model reduction, system identification, and control,” *Physica D* **406**, 132416 (2020).
- ¹⁸A. Mardt, L. Pasquali, H. Wu, and F. Noé, “VAMPnets for deep learning of molecular kinetics,” *Nat. Commun.* **9**, 5 (2018).
- ¹⁹W. Chen, H. Sidky, and A. L. Ferguson, “Nonlinear discovery of slow molecular modes using state-free reversible vampnets,” *J. Chem. Phys.* **150**, 214114 (2019).
- ²⁰H. Wendland, *Scattered Data Approximation* (Cambridge University Press, 2004).
- ²¹I. Steinwart and A. Christmann, *Support Vector Machines*, 1st ed. (Springer, New York, 2008).
- ²²M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, “A kernel-based method for data-driven Koopman spectral analysis,” *J. Comput. Dyn.* **2**, 247–265 (2015).

- ²³S. Klus, I. Schuster, and K. Muandet, “Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces,” *J. Nonlinear Sci.* **30**, 283–315 (2020).
- ²⁴S. Klus, F. Nüske, and B. Hamzi, “Kernel-based approximation of the Koopman generator and Schrödinger operator,” *Entropy* **22**, 722 (2020).
- ²⁵S. Klus, A. Bittracher, I. Schuster, and C. Schütte, “A kernel-based approach to molecular conformation analysis,” *J. Chem. Phys.* **149**, 244109 (2018).
- ²⁶A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” *Advances in Neural Information Processing Systems* **20** (2007).
- ²⁷E. B. Davies, “Metastable states of symmetric Markov semigroups II,” *J. London Math. Soc.* **2**, 541–556 (1982).
- ²⁸M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition,” *Journal of Nonlinear Science* **25**, 1307–1346 (2015).
- ²⁹S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, “Data-Driven Model Reduction and Transfer Operator Approximation,” *J. Nonlinear Sci.* **28**, 985–1010 (2018).
- ³⁰N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American Mathematical Society* **68**, 337–404 (1950).
- ³¹D. MacKay, “Introduction to Gaussian processes,” *NATO ASI series F computer and systems sciences* **168**, 133–166 (1998).
- ³²S. Bochner, “Monotone Funktionen, Stieltjessche Integrale und Harmonische Analyse,” *Mathematische Annalen* **108**, 378–410 (1933).
- ³³A. Tompkins and F. Ramos, “Fourier feature approximations for periodic kernels in time-series modelling,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32 (2018).
- ³⁴S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, “Data-driven model reduction and transfer operator approximation,” *J. Nonlinear Sci.* **28**, 985–1010 (2018).
- ³⁵W. Zhang, T. Li, and C. Schütte, “Solving eigenvalue PDEs of metastable diffusion processes using artificial neural networks,” *J. Comput. Phys.* **465**, 111377 (2022).
- ³⁶M. Sarich, F. Noé, and C. Schütte, “On the approximation quality of Markov state models,” *Multiscale Model. Simul.* **8**, 1154–1177 (2010).
- ³⁷P. Deuffhard and M. Weber, “Robust Perron cluster analysis in conformation dynamics,” *Linear Algebra Appl* **398**, 161–184 (2005).

- ³⁸A. Bittracher, P. Koltai, S. Klus, R. Banisch, M. Dellnitz, and C. Schütte, “Transition manifolds of complex metastable systems,” *J. Nonlinear Sci.* **28**, 471–512 (2018).
- ³⁹F. Nüske, R. Schneider, F. Vitalini, and F. Noé, “Variational tensor approach for approximating the rare-event kinetics of macromolecular systems,” *J. Chem. Phys.* **144**, Article 054105 (2016).
- ⁴⁰F. Nüske, P. Gelß, S. Klus, and C. Clementi, “Tensor-based computation of metastable and coherent sets,” *Physica D* **427**, 133018 (2021).
- ⁴¹K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, “How fast-folding proteins fold,” *Science* **334**, 517–520 (2011).
- ⁴²T. Lelièvre, M. Rousset, and G. Stoltz, *Free energy computations: A mathematical perspective* (World Scientific, 2010).
- ⁴³L. Boninsegna, F. Nüske, and C. Clementi, “Sparse learning of stochastic dynamical equations,” *J. Chem. Phys.* **148**, 241723 (2018).

Appendix A: Derivation of Kernel Eigenvalue Problems

Our derivation of the kernel eigenvalue problems in Section II F follows^{23,24}. We compute the matrix elements of the operators $\hat{\mathcal{G}}$, $\hat{\mathcal{A}}^t$, $\hat{\mathcal{A}}^L$ on the finite-dimensional space \mathbb{H}_m , with respect to its canonical basis $\{\Phi(x_l)\}_{l=1}^m$:

$$\begin{aligned}
\left\langle \Phi(x_r), \hat{\mathcal{G}}\Phi(x_s) \right\rangle_{\mathbb{H}} &= \frac{1}{m} \sum_{l=1}^m \Phi(x_s)(x_l) \langle \Phi(x_r), \Phi(x_l) \rangle_{\mathbb{H}} \\
&= \frac{1}{m} \sum_{l=1}^m k(x_s, x_l) k(x_l, x_r) = \frac{1}{m} [\mathbf{K}_X \mathbf{K}_X](r, s), \\
\left\langle \Phi(x_r), \hat{\mathcal{A}}^t \Phi(x_s) \right\rangle_{\mathbb{H}} &= \frac{1}{m} \sum_{l=1}^m \Phi(x_s)(x_{l+1}) \langle \Phi(x_r), \Phi(x_l) \rangle_{\mathbb{H}} \\
&= \frac{1}{m} \sum_{l=1}^m k(x_s, x_{l+1}) k(x_l, x_r) = \frac{1}{m} [\mathbf{K}_X \mathbf{K}_Y](r, s), \\
\left\langle \Phi(x_r), \hat{\mathcal{A}}^L \Phi(x_s) \right\rangle_{\mathbb{H}} &= \frac{1}{m} \sum_{l=1}^m (\mathcal{L}\Phi(x_s))(x_l) \langle \Phi(x_r), \Phi(x_l) \rangle_{\mathbb{H}} \\
&= \frac{1}{m} \sum_{l=1}^m \left[\nabla F(x_l) \cdot \nabla \Phi(x_s)(x_l) + \frac{1}{2} a(x_l) : \nabla^2 \Phi(x_s)(x_l) \right] k(x_l, x_r) \\
&= \frac{1}{m} \sum_{l=1}^m \left[\nabla F(x_l) \cdot \langle \nabla_1 k(x_l, \cdot), \Phi(x_s) \rangle_{\mathbb{H}} + \frac{1}{2} a(x_l) : \langle \nabla_1^2 k(x_l, \cdot), \Phi(x_s) \rangle_{\mathbb{H}} \right] \\
&\quad k(x_l, x_r) \\
&= \frac{1}{m} \sum_{l=1}^m \left[\nabla F(x_l) \cdot \nabla_1 k(x_l, x_s) + \frac{1}{2} a(x_l) : \nabla_1^2 k(x_l, x_s) \right] k(x_l, x_r) \\
&= \frac{1}{m} [\mathbf{K}_X \mathbf{K}_X^L](r, s).
\end{aligned}$$

Here, we used the subscript ∇_1 for the nabla-operator to indicate differentiation with respect to the first variable. To get from the second to the third line, we used the derivative reproducing property

$$D^\alpha f(x) = \langle D_1^\alpha k(x, \cdot), f \rangle_{\mathbb{H}},$$

which holds for any multi-index α and all RKHS functions f as soon as the kernel is at least $2|\alpha|$ -times continuously differentiable in both arguments, see²⁰.

We see that in all cases, the full-rank matrix \mathbf{K}_X can be canceled out, which leaves us with (9). In the reversible case, we replace the operator $\hat{\mathcal{A}}^L$ by a symmetric bi-linear form

$$\hat{\mathcal{A}}^{\text{rev}}(f, g) = -\frac{1}{2m} \sum_{l=1}^m \nabla^\top f(x_l) a(x_l) \nabla g(x_l).$$

Computing its matrix elements on \mathbb{H}_m , we find

$$\begin{aligned} \hat{\mathcal{L}}^{\text{rev}}(\Phi(x_r), \Phi(x_s)) &= -\frac{1}{2m} \sum_{l=1}^m \nabla^\top \Phi(x_r)(x_l) a(x_l) \nabla \Phi(x_s)(x_l) \\ &= -\frac{1}{2m} \sum_{l=1}^m \nabla_1^\top k(x_l, x_r) a(x_l) \nabla_1 k(x_l, x_s) = \mathbf{K}_X^{\text{rev}}(r, s). \end{aligned}$$

In this case, we cannot cancel a factor \mathbf{K}_X , which leads to (10).