

State predictive information bottleneck

Cite as: J. Chem. Phys. 154, 134111 (2021); doi: 10.1063/5.0038198

Submitted: 19 November 2020 • Accepted: 23 March 2021 •

Published Online: 5 April 2021



View Online



Export Citation



CrossMark

Dedi Wang¹ and Pratyush Tiwary^{2,a)}

AFFILIATIONS

¹ Biophysics Program and Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742, USA

² Department of Chemistry and Biochemistry and Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742, USA

Note: This paper is part of the 2021 JCP Emerging Investigators Special Collection.

^{a)}Author to whom correspondence should be addressed: ptiwar@umd.edu

ABSTRACT

The ability to make sense of the massive amounts of high-dimensional data generated from molecular dynamics simulations is heavily dependent on the knowledge of a low-dimensional manifold (parameterized by a reaction coordinate or RC) that typically distinguishes between relevant metastable states, and which captures the relevant slow dynamics of interest. Methods based on machine learning and artificial intelligence have been proposed over the years to deal with learning such low-dimensional manifolds, but they are often criticized for a disconnect from more traditional and physically interpretable approaches. To deal with such concerns, in this work we propose a deep learning based state predictive information bottleneck approach to learn the RC from high-dimensional molecular simulation trajectories. We demonstrate analytically and numerically how the RC learnt in this approach is connected to the committor in chemical physics and can be used to accurately identify transition states. A crucial hyperparameter in this approach is the time delay or how far into the future the algorithm should make predictions about. Through careful comparisons for benchmark systems, we demonstrate that this hyperparameter choice gives useful control over how coarse-grained we want the metastable state classification of the system to be. We thus believe that this work represents a step forward in systematic application of deep learning based ideas to molecular simulations.

Published under license by AIP Publishing. <https://doi.org/10.1063/5.0038198>

I. INTRODUCTION

Rapid advances in computational power have made molecular dynamics (MD) a powerful tool for studying systems in biophysics, chemical physics, and beyond. However, there are still at least two open questions in this area: first, how to make use of the deluge of data generated from MD simulation understandable for a human; second, how to further extend timescales that can be reached in MD. The unifying aspect to overcoming both these difficulties is to efficiently uncover a low-dimensional manifold (parameterized by a reaction coordinate or RC) on which the dynamics of the system can be projected.¹

Over the past decades, various approaches have been developed to learn the RC from trajectory data. It has been argued that for given two states, the committor, defined next, is a perfect candidate for the RC as it provides a quantitative description of the dynamics along a trajectory.^{2,3} Let A and B denote the reactant and product states;

then, the committor probability $p_B(x)$ is defined as the probability of the trajectories that reach state B prior to the state A from a conformation X. Through the analysis of the committor distribution, much insight has been obtained in a variety of phenomena ranging from ion solvation to biomolecular isomerization.^{2,4–6} A range of approaches have been proposed over the years to obtain the committor. Transition path sampling (TPS), which focuses on sampling the pathways connecting metastable states, is a powerful tool to analyze the committor.^{3,7,8} Based on it, some physically meaningful RC can then be identified through a genetic algorithm⁹ or a likelihood maximization approach.^{10,11}

Another approach to obtaining the RC is to learn the relevant slow modes of dynamics. Coifman, Kevrekidis *et al.* first used a diffusion map to determine collective reaction coordinates for macromolecular dynamics.^{12–14} Thereafter, Noé and co-workers proposed the variational approach to conformation dynamics (VAC) and combined it with the time-lagged independent component

analysis (TICA) to identify the optimal “slow subspace” from a large set of prior order parameters.^{15,16} More recently, a generalized version called VAMPnets was developed by the same group leveraging the power of neural networks.¹⁷ In a similar spirit, the SGOOP method by Tiwary and Berne used an iterative approach to find RC through a maximum path entropy framework.¹⁸ Though all these slow-mode based methods are highly interpretable, the optimization can usually be difficult unless some simplifications are made. For instance, in TICA and SGOOP, these simplifications could include learning the RC as linear combinations of pre-selected order parameters.^{16,18}

Besides these approaches, other statistical approaches have also been developed to learn RC through a more flexible framework, such as VDE¹⁹ and RAVE.^{20,21} In the RAVE approach, for instance, the RC is interpreted as a bottleneck or a low-dimensional space that predicts the most important features of the simulated trajectories. Such a RC can then be learned by making a trade-off between prediction and model complexity through an objective function. Typically, a variational Bayesian approach is employed to allow these methods to parameterize the objective function using a neural network and achieve highly efficient training.^{22,23} However, these approaches can be uninterpretable¹ and therefore unreliable since the relationship between the statistics-based RC and the traditional physics-based RC is still unclear.

In this work, we develop a State Predictive Information Bottleneck (SPIB) framework that allows us to efficiently and accurately learn a RC from MD trajectories. Most importantly, we demonstrate rigorously how the slow bottleneck variable learnt in RAVE and related deep learning based methods can qualify as a good RC with the same attributes as expected from the committor. Similar to RAVE,^{20,21} we also assume that RC should carry only the minimal information of the past to still be able to reliably predict the future state of the system. The key feature that makes SPIB stand out is that a discrete-state representation of this system is learned on-the-fly during the training process and guides our RC to focus only on the motion related to the state-to-state transitions. We show analytically and numerically that the RC learned by our algorithm is related to the committor and demonstrate that it can capture the important information from the trajectory to identify the correct transition state. Moreover, we demonstrate how our algorithm can automatically figure out the metastable states in a complex system and generate an accurate but still highly understandable description of their inter-conversion dynamics. Given these promising properties, we believe that our algorithm can be a powerful tool to analyze generic complex systems.

II. METHOD

A. Information bottleneck

The Information Bottleneck (IB) principle provides a general framework to learn a concise representation \mathbf{z} of an input source \mathbf{X} that is maximally informative about some target \mathbf{y} .^{23,24} Here, typically the representation \mathbf{z} has much smaller dimensionality than the source \mathbf{X} , while the target \mathbf{y} can be of low or high dimensionality depending on the task at hand. The IB principle postulates that the desired representation \mathbf{z} should use minimal information from the input \mathbf{X} to predict the target \mathbf{y} . Mathematically, such a learning

process can be formulated as maximizing the objective function,

$$\mathcal{L}_{IB} \equiv I(\mathbf{z}, \mathbf{y}) - \beta I(\mathbf{X}, \mathbf{z}). \quad (1)$$

Here, the function $I(\mathbf{x}, \mathbf{y}) \equiv \int d\mathbf{x} d\mathbf{y} p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}$ denotes the mutual information between any two random variables. The trade-off between the prediction capacity $I(\mathbf{z}, \mathbf{y})$ and model complexity $I(\mathbf{X}, \mathbf{z})$ is controlled by a hyper-parameter $\beta \in [0, \infty)$. Unfortunately, the direct optimization of the information bottleneck shown in Eq. (1) is impractical as the calculation of mutual information, in general, is computationally expensive.^{21,23} Thus, following Ref. 23, we can obtain a variational lower bound on the original objective function from Eq. (1),

$$\begin{aligned} \mathcal{L}_{IB} \geq & - \underbrace{\frac{1}{N} \sum_{n=1}^N \int d\mathbf{z} [-p(\mathbf{z}|\mathbf{X}^n) \log q(\mathbf{y}^n|\mathbf{z})]}_{\text{distortion}} \\ & - \beta \underbrace{\frac{1}{N} \sum_{n=1}^N \int d\mathbf{z} \left[p(\mathbf{z}|\mathbf{X}^n) \log \frac{p(\mathbf{z}|\mathbf{X}^n)}{r(\mathbf{z})} \right]}_{\text{rate}} + H(\mathbf{y}) = \mathcal{L}, \end{aligned} \quad (2)$$

where $q(\mathbf{y}|\mathbf{z})$ and $r(\mathbf{z})$ are variational approximations to the true probability distributions $p(\mathbf{y}|\mathbf{z})$ and $p(\mathbf{z})$, respectively. Note that the entropy of the targets $H(\mathbf{y}) \equiv -\int d\mathbf{y} p(\mathbf{y}) \log p(\mathbf{y})$ in Eq. (2) is independent of the optimization process and hence can be ignored. From a coding theory perspective,²⁵ as \mathbf{z} can be interpreted as a latent representation or a code, we usually refer to $p(\mathbf{z}|\mathbf{X})$ as a probabilistic encoder and $q(\mathbf{y}|\mathbf{z})$ as a probabilistic decoder. Interestingly, one can easily obtain from Eq. (2) the objective function used in variational autoencoders by assuming $\beta = 1$ and requiring the representation \mathbf{z} to reconstruct the input \mathbf{X} instead of predicting a target \mathbf{y} .^{22,23} Based on rate-distortion theory,^{26,27} the first term in Eq. (2) can be interpreted as the distortion, which measures the ability of our representation to predict the desired target, while the second term can be interpreted as the rate, which measures the number of bits per data sample to be transmitted. Thus, maximizing \mathcal{L} can also be viewed as the problem of determining the minimal number of bits, as measured by the rate, that should be communicated from a source through a channel so that the receiver can reconstruct the original signal without exceeding a desired value of the distortion.

There are many possible choices for the encoder $p(\mathbf{z}|\mathbf{X})$, the decoder $q(\mathbf{y}|\mathbf{z})$, and the approximate prior $r(\mathbf{z})$, depending on the particular application domain. We point out here that all these three probability distributions can depend collectively on some model parameters θ , which are learned during the training process. Therefore, in Secs. II B–II E, we will add a subscript θ to all these three distributions $\{p_\theta(\mathbf{z}|\mathbf{X}), q_\theta(\mathbf{y}|\mathbf{z}), r_\theta(\mathbf{z})\}$.

B. State predictive information bottleneck

The generic IB framework introduced in Sec. II A leaves ample scope for the specific flavor of implementation in many different ways, as, for instance, we demonstrated in our past publications through the RAVE family of methods,^{20,21} and it has been discussed more generally in Ref. 28. Based on the general IB framework, in this section, we advance our RAVE family of methods with a State Predictive Information Bottleneck (SPIB) framework. Similar

to existing RAVE formulations, here as well we aim to learn an accurate reaction coordinate (RC) for generic molecular systems, but we make RAVE significantly more robust in many aspects and draw rigorous and useful connections between the past–future information bottleneck and the committor-based definition of the RC in theoretical chemistry.² However, unlike RAVE, where the aim is to predict a time-delayed version of the entire input molecular configuration, here we set as target \mathbf{y} in Eq. (1) its future state, which is drawn from a dictionary of indices for possible metastable states. The target \mathbf{y} is relatively much lower in dimensionality than the exact molecular configuration. In this way, we require our RC to only predict which state the system will stay in after a time delay Δt , instead of its exact configuration. Typically, the number and location of such states are not available *a priori*, and our work makes it possible to estimate these robustly and on-the-fly, as we demonstrate in Sec. II D.

The main advantage of such a simplification of the prediction task is that only the motion related to the transitions between different states will be captured by the learnt RC, while the fluctuations inside any metastable state will be ignored. Thus, for a given unbiased trajectory $\{\mathbf{X}^1, \dots, \mathbf{X}^{M+s}\}$ and its corresponding state labels $\{\mathbf{y}^1, \dots, \mathbf{y}^{M+s}\}$ with large enough M , the objective function of SPIB can be formulated as

$$\mathcal{L} \approx \frac{1}{M \cdot L} \sum_{n=1}^M \sum_{l=1}^L \left[\log q_\theta(\mathbf{y}^{n+s} | \mathbf{z}^{(n,l)}) - \beta \log \frac{p_\theta(\mathbf{z}^{(n,l)} | \mathbf{X}^n)}{r_\theta(\mathbf{z}^{(n,l)})} \right], \quad (3)$$

where $\mathbf{z}^{(n,l)}$ is sampled from $p_\theta(\mathbf{z} | \mathbf{X}^n)$ and the time interval between \mathbf{X}^n and \mathbf{X}^{n+s} is the time delay Δt .

In SPIB, the trajectory $\{\mathbf{X}^n\}$ is usually expressed in terms of many order parameters or features, while the state labels $\{\mathbf{y}^n\}$ are mutually exclusive and expressed in terms of one-hot vectors, i.e., a binary vector with a single high (1) bit and all the others low (0). To implement this, we use a deep feed forward neural network with softmax outputs in our decoder $q_\theta(\mathbf{y} | \mathbf{z})$,

$$\log q_\theta(\mathbf{y}^{n+s} | \mathbf{z}^n) = \sum_{i=1}^D y_i^{n+s} \log \mathcal{D}_i(\mathbf{z}^n; \theta), \quad (4)$$

where the state label \mathbf{y} is a one-hot vector of D dimensions and the decoder function \mathcal{D} is the D -dimensional softmax output of a neural network.

Given that we expect the learnt RC should demarcate between different metastable states, it is natural to assume a multi-modal distribution for the prior $r_\theta(\mathbf{z})$. In our algorithm, we employ the variational mixture of posteriors prior (VampPrior) to obtain such a multi-modal prior distribution.²⁹ Here, the approximate prior $r_\theta(\mathbf{z})$ is a weighted mixture of different posteriors $p_\theta(\mathbf{z} | \mathbf{X})$ with pseudo-inputs $\{\mathbf{u}^k\}_{k=1}^K$ in lieu of \mathbf{X} ,

$$r_\theta(\mathbf{z}) = \sum_{k=1}^K \omega_k p_\theta(\mathbf{z} | \mathbf{u}^k), \quad (5)$$

where K is the number of pseudo-inputs, \mathbf{u}^k is a vector that has the same dimension as input \mathbf{X} , and ω_k represents the weight of $p_\theta(\mathbf{z} | \mathbf{u}^k)$ under the constraint $\sum_k \omega_k = 1$. The pseudo-inputs $\{\mathbf{u}^k\}$ and weights $\{\omega_k\}$ can be thought of the parameters of the prior, which are learned through backpropagation of the objective function [Eq. (3)]. In principle,

the number of pseudo-inputs should be equal to the number of metastable states in the system. In practical settings, however, for real-world applications to complex molecular systems, the number of metastable states is unknown *a priori*. To deal with such cases, the simple and powerful solution is to choose a large enough K , making the prior more flexible.

Finally, for simplicity, we take the encoder $p_\theta(\mathbf{z} | \mathbf{X})$ in Eq. (5) as a neural network with a multivariate Gaussian output,

$$\log p_\theta(\mathbf{z}^n | \mathbf{X}^n) = \log \mathcal{N}(\mathbf{z}^n; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 I), \quad (6)$$

where the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ are outputs of a neural network whose input is \mathbf{X}^n . I is the identity matrix. Then, we can use the reparameterization trick²² to write $p_\theta(\mathbf{z}^n | \mathbf{X}^n) d\mathbf{z}^n = p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}$ and $\mathbf{z}^n = \boldsymbol{\mu}(\mathbf{X}^n) + \boldsymbol{\sigma}(\mathbf{X}^n) \cdot \boldsymbol{\epsilon} = \mathcal{E}(\mathbf{X}^n, \boldsymbol{\epsilon}; \theta)$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and the encoder function \mathcal{E} is a deterministic nonlinear function parameterized by a neural network.

C. Dependence of SPIB on Δt

In RAVE²¹ as well as in this algorithm, the time delay Δt plays an important role in the simplification of the learning process. A time delay $\Delta t = 0$ is tantamount to ignoring the dynamics completely and simply clustering the input configuration into different states, while $\Delta t > 0$ can filter out all the fast modes, helping us ignore unnecessary details of the dynamical processes. Given its critical importance, in this section, we analyze Δt in detail.

Given a Markov process \mathbf{X}_t , if the initial probability distribution is given by ρ_0 , the corresponding probability distribution after a lag time τ is

$$\rho_\tau(\mathbf{X}) = \int \rho_0(\mathbf{X}') P_\tau(\mathbf{X} | \mathbf{X}') d\mathbf{X}' \equiv \mathcal{P}(\tau) \rho_0, \quad (7)$$

where the time evolution of the probability density is governed by a linear operator $\mathcal{P}(\tau)$, called the propagator for the process \mathbf{X}_t . To explain the role of the time delay Δt in our algorithm, we can derive a spectral decomposition for this operator $\mathcal{P}(\tau)$ by assuming that the dynamics is reversible,³⁰

$$\rho_\tau = v_1 + \sum_{i=2}^{\infty} a_i(\rho_0) \lambda_i(\tau) v_i, \quad (8)$$

where $\{v_i\}$ are the propagator's eigenfunctions and $\lambda_i = \exp(-k_i \tau)$ are the eigenvalues that decay exponentially in time with rates k_i . $a_i(\rho_0)$ are coefficients depending on the initial density ρ_0 . In principle, SPIB will ignore the dynamical processes whose timescale $t_i = 1/k_i$ is comparable to or even smaller than the time delay Δt , as its corresponding component in Eq. (8) will decay exponentially. Thus, we interpret the time delay Δt as the minimal time resolution that we seek to maintain for the dynamical system.

As discussed in Subsection II B, SPIB predicts from the present configuration \mathbf{X}^n the future state \mathbf{y}^{n+k} instead of the exact configuration \mathbf{X}^{n+k} . A subtle assumption made to justify this simplification is that the fluctuation inside each metastable state should be much faster than the transitions between different states. Such a timescale separation allows us to rewrite Eq. (8) as a sum of the stationary state eigenvector v_1 and two other parts,

$$\begin{aligned} \rho_{\tau} &= v_1 + \sum_{i=2}^m a_i(\rho_0) \lambda_i(\tau) v_i + \sum_{i=m+1}^{\infty} a_i(\rho_0) \lambda_i(\tau) v_i \\ &= v_1 + \sum_{i=2}^m a_i(\rho_0) \lambda_i(\tau) v_i + \mathcal{P}_{\text{fast}}(\tau) \rho_0. \end{aligned} \quad (9)$$

The first m slow processes $\{v_i\}_{i=1}^m$ correspond to the state-to-state transitions that we are interested in, while the fast processes $\mathcal{P}_{\text{fast}}(\tau)$ represent the motions related to the molecular relaxation within these states. Therefore, an appropriate time delay Δt should satisfy $t^{m+1} < \Delta t \ll t^m$ in order to screen out all the fast processes. In practice, this can be checked by examining the robustness of the results against different values of Δt , as we show in Sec. III.

D. Discrete-state representation and iterative retraining algorithm

The SPIB framework introduced thus far requires a prior knowledge of states in the system, which is usually intractable especially for complex systems. To surmount this limitation, here we introduce an iterative technique to obtain a converged discrete-state representation based on the selected time delay Δt we described in Sec. II C. The central idea is that if one configuration was located at state i at a certain time, then after time delay Δt , it should still have the largest probability to be found at state i , since Δt is much shorter than the typical escape time from a metastable state.

Given a set of initial state labels $\{y^n\}$, we can write down the optimal predictor \hat{y}^* by assuming ergodic dynamics and setting it equal to a vector of probabilities $K^* = \{K_i^*(\mathbf{X}; \Delta t)\}$,

$$K_i^*(\mathbf{X}; \Delta t) = \frac{1}{\rho(\mathbf{X})} \lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T h_i(\mathbf{X}_{t+\Delta t}) \delta(\mathbf{X} - \mathbf{X}_t) dt$$

where

$$\rho(\mathbf{X}) = \lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T \delta(\mathbf{X} - \mathbf{X}_t) dt. \quad (10)$$

Here, $\mathbf{h}(\mathbf{X}) = \{h_i(\mathbf{X}) \text{ for } i \in [1, D]\}$ is the state label function that maps the trajectory $\{\mathbf{X}^n\}$ to the D -dimensional state labels $\{y^n\}$, and $\rho(\mathbf{X})$ represents the equilibrium density of \mathbf{X} . $K_i^*(\mathbf{X}; \Delta t)$ can be interpreted as the probability that the system starting from \mathbf{X} will be found in state i after a time delay Δt . As it is a function of the input configuration \mathbf{X} and represents a state-transition probability, we call the function $K^*(\mathbf{X}; \Delta t)$ as the state-transition density.

With this setup, we now introduce a simple iterative scheme that is at the heart of our SPIB approach, as it allows us to learn the number and location of states on-the-fly with minimal human intervention. We start with an arbitrary set of labels $\{1, \dots, D\}$ for the system, where both the number and location of labels are some initial guess. If the system initiated from a certain high-dimensional configuration \mathbf{X} has the largest probability to be found after time delay Δt in some state i from these initial labels, then the label of the configuration \mathbf{X} will be refined and updated to state i . We can denote the deterministic output of SPIB as $\hat{y} = K(\mathbf{X}; \Delta t, \theta) \equiv \mathcal{D}(\mu(\mathbf{X}); \Delta t, \theta)$, which tries to approximate the best predictor $\hat{y}^* = K^*(\mathbf{X}; \Delta t)$. Then, a set of new state labels can be generated by

$$h_i(\mathbf{X}) = \begin{cases} 1 & (i = \operatorname{argmax}_j K_j(\mathbf{X}; \Delta t, \theta)) \\ 0 & (\text{otherwise}) \end{cases} \quad \text{for } i = 1, \dots, D. \quad (11)$$

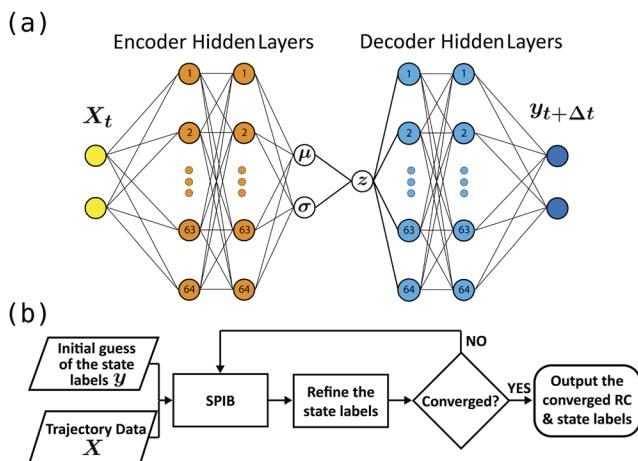


FIG. 1. (a) Network architecture used for SPIB. Generalizing Ref. 21, both the encoder and decoder are nonlinear deep neural networks. (b) A flowchart illustrating SPIB.

This label refinement step might very well lead to null assignments for some of the labels we started with, as shown in Sec. III for actual test cases.

Based on Eqs. (10) and (11), an iterative retraining can be performed and the whole algorithm is summarized through Fig. 1 and Algorithm 1. Thus, as illustrated above, we expect that such a converged discrete-state representation $\mathbf{h}(\mathbf{X})$, by this self-consistent design, should only depend on the dynamic properties of the system and the time delay Δt . Moreover, on account of the screening property of the time delay Δt , the final representation will automatically ignore transient intermediate states and only figure out those long-lived metastable states. This, in fact, offers us a powerful tool to obtain a dynamics-based coarse-grained description of the complex system.

E. State-transition density and committor

Recently, a few methods have been proposed to calculate the committor through the construction of Markov state models (MSMs).^{31–35} By constructing an efficient MSM, the committor can

ALGORITHM 1. SPIB.

Input: a long unbiased trajectory $\{\mathbf{X}^n\}$, a set of initial state labels $\{y^n\}$, RC dimensionality d , the number of pseudo-inputs K , time delay Δt

- 1: **repeat**
- 2: **for** i from 0 to $m-1$ **do**
- 3: Sample a minibatch $\{\mathbf{X}^n\}$ and $\{y^n\}$
- 4: Calculate the objective function \mathcal{L}
- 5: Update the neural network parameters θ , pseudo-inputs $\{\mathbf{u}^k\}_{k=1}^K$, pseudo-weights $\{\omega_k\}_{k=1}^K$
- 6: **end for**
- 7: Update the state labels $\{y^n\}$ by Eq. (11)
- 8: **until** convergence of RC, state-transition density, and state labels

be calculated directly from the transition matrix by solving a system of linear equations.³² However, this usually requires a large number of discrete states in order to estimate the MSM transition matrix and thus the committor accurately,^{33–35} thereby severely diminishing the interpretability of the model. This also means requiring a very well sampled trajectory moving accurately and capturing transitions between the large number of different states, which might be hard to achieve.³⁶ As we will show in this section, our SPIB approach can efficiently estimate the transition density by relegating the need to know the exact transition probabilities within the metastable states of the system. In other words, we will demonstrate that SPIB can learn the approximate committor and identify correct transition state regions even with a small number of discrete states relative to MSM type approaches.

As discussed in the Sec. II C, the timescale separation between the state-to-state transitions and the fluctuations inside each metastable state allows the factorization of the transition density $P_{\Delta t}(\mathbf{X}|\mathbf{X}_0)$ into two parts,

$$P_{\Delta t}(\mathbf{X}|\mathbf{X}_0) \approx \sum_{i=1}^D P_i(\mathbf{X}) K_i^*(\mathbf{X}_0; \Delta t)$$

where

$$P_i(\mathbf{X}) = \frac{\rho(\mathbf{X}) h_i(\mathbf{X})}{\int h_i(\mathbf{X}) \rho(\mathbf{X}) d\mathbf{X}}. \quad (12)$$

In Eq. (12), the first part $P_i(\mathbf{X})$ represents the equilibrium probability density for any state $i \in [1, D]$, while the second part $K_i^*(\mathbf{X}_0; \Delta t)$ is exactly the state-transition density defined in Eq. (10). Given the definition of two states A and B whose committor attracts our interest, the committor $p_B(\mathbf{X})$ then can be obtained by solving following linear integral equations:

$$\begin{aligned} p_B(\mathbf{X}) &= \int p_B(\mathbf{X}') P_{\Delta t}(\mathbf{X}'|\mathbf{X}) d\mathbf{X}' && \text{if } \mathbf{X} \notin A \cup B, \\ p_B(\mathbf{X}) &= 1 && \text{if } \mathbf{X} \in B, \\ p_B(\mathbf{X}) &= 0 && \text{if } \mathbf{X} \in A. \end{aligned} \quad (13)$$

However, instead of solving Eq. (13) explicitly and tabulating configurations where $p_B(\mathbf{X}) \approx 0.5$, the transition state ensemble (TSE) can also be identified using this state-transition density directly. If a set of trajectories starting from some \mathbf{X} have the largest, approximately equal probabilities of transitioning to two different states after time delay Δt , then \mathbf{X} can be considered to belong to the TSE. Through the numerical examples in Sec. III, we will illustrate that this new definition of transition states is, in fact, as valid as the original committor-based definition. Besides, we would also like to highlight that based on this definition and Algorithm 1, the ensemble of transition states will eventually form the boundaries of the finally converged discrete states. Therefore, the state-transition density $K^*(\mathbf{X}; \Delta t)$ provides us with an accurate but intuitive way to characterize the transitions between different metastable states.

Overall, we believe that such a state-transition density finally generated by our algorithm is a reasonable substitute for the committor as it can quantitatively describe the dynamical behaviors of every states along a trajectory and further identify the correct transition states. Thus, the most informative representation \mathbf{z} given by the encoder $p_\theta(\mathbf{z}''|\mathbf{X}'')$ about this state-transition density learned by SPIB should naturally serve as a reasonable RC approximating the

committor. However, for simplicity, in the following discussion, our RC will refer specifically to the deterministic part or the mean value μ of the representation \mathbf{z} to better compare with traditional deterministic prescriptions.

III. RESULTS

A. Model systems

To demonstrate our SPIB approach, in practice, here we benchmark it for different model potentials, including two analytical potentials and the small biomolecule alanine dipeptide in vacuum. The first analytical potential $U_{DW}(x, y)$ comprises a double well in two dimensions, shown in Fig. 2. The second potential is made of four wells also in two dimensions, shown in Fig. 5. The governing potentials are given by

$$U_{DW}(x, y) = (x^2 - 1)^2 + y^2 \quad (14)$$

and

$$\begin{aligned} U_{FW}(x, y) = 2 & \left[x^8 + 0.6e^{-80x^2} + 0.2e^{-80(x-0.5)^2} \right. \\ & \left. + 0.5e^{-40(x+0.5)^2} \right] + (x^2 - 1)^2 + y^2. \end{aligned} \quad (15)$$

The trajectories for these two potentials were generated using Langevin dynamics simulation³⁷ with the integration time step of 0.001 units, inverse temperature $(k_B T)^{-1} = 3.0$, and friction coefficient $\gamma = 4.0$, where k_B is Boltzmann constant. For either potential, we used a long equilibrium trajectory equaling 60 000 time units with a temporal resolution of 0.01 units.

For the study of conformation transitions in alanine dipeptide in vacuum, the simulation was performed with the software GROMACS 5.0,^{38,39} patched with PLUMED 2.4.⁴⁰ The temperature was kept constant at 450 K using the velocity rescaling thermostat,⁴¹ and the integration time step was 2 fs. An 800 ns trajectory with a temporal resolution of 0.01 ps was employed to train and test our algorithm. Through this 800 ns trajectory, a total of about 500 transitions are observed between C_{eq} and C_{ax} defined in Fig. 9(b). However, only about 30 transitions are through the TS region [also defined in Fig. 9(b)].

B. Neural network architecture and training

In this paper, both the encoder and decoder are nonlinear and parameterized by fully connected neural networks with two hidden

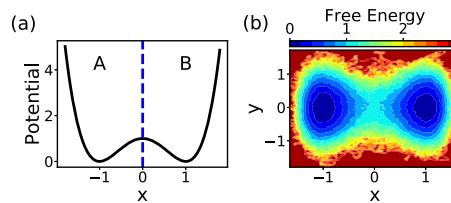


FIG. 2. Double-well analytical potentials projected along the x -axis (a) and its corresponding probability $P(x, y)$ distribution of the generated trajectory (b), plotted as the free energy $-k_B T \log P(x, y)$.

layers, as shown in Fig. 1(a). Each hidden layer in both the encoder and decoder has 16 nodes for the two analytical potentials and 64 nodes for alanine dipeptide. All these hidden layers use a rectified linear unit (ReLU) as the activation function.

The networks were trained using the Adam optimizer⁴² with a learning rate of 0.001 and a batch size of 2048 for all the numerical examples. The state labels are refined every 1000 training steps for analytical potentials and every 2000 training steps for alanine dipeptide.

C. Double-well analytical potential

We first demonstrate SPIB for the double-well analytical potential. For this first example, we assume that we already know the system relatively well by setting the RC dimension $d = 1$, the number of pseudo-inputs $K = 2$, and the dimension of state labels $D = 2$. In Secs. III D and III E, we remove the need for making any such assumptions and show how SPIB still works very well. For this double-well system, in order to generate an initial guess of state labels, the samples are labeled as state A if $x < b$ and B otherwise. Here, the initial boundary point b can be changed to test the robustness of SPIB.

The final converged results are shown in Fig. 3. Figure 3(a) illustrates that SPIB can learn the correct state labels, where the boundary is located at around $x = 0$. In Fig. 3(b), as the y -direction is pure noise, the learned RC is almost independent of the y -direction, suggesting that SPIB is able to distinguish important features from noise. In addition, Fig. 3(b) also shows that as desired, the fluctuations inside each state are not captured by SPIB, as they are almost mapped to a single point in RC. Figures 3(c) and 3(d) present the state-transition density learned by our algorithm, which is highly correlated with our RC.

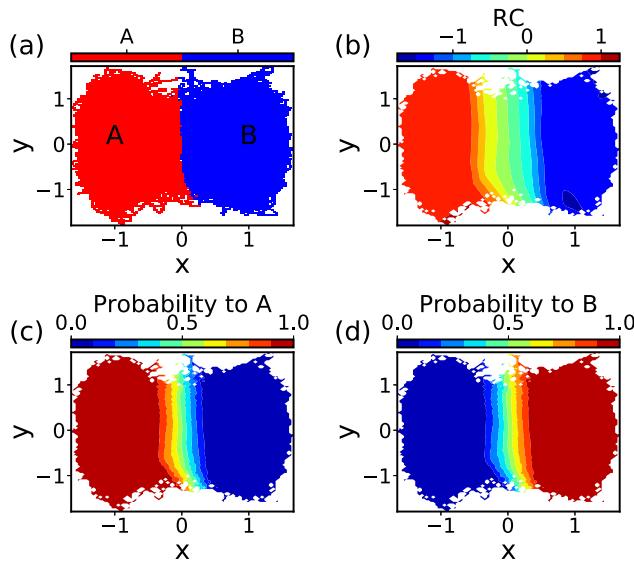


FIG. 3. The results of SPIB for the double-well potential. (a) The converged state labels A and B. (b) Different values of the RC illustrated in the x - y plane. (c) and (d) are the state-transition density learned by SPIB, where (c) represents the transition density to state A and (d) represents the transition density to state B.

We now further demonstrate that our results obtained above are robust to the initial boundary demarcating parameter b and the time delay Δt . As shown in Fig. 4, a large range of b and Δt values can result in the same state definition. The fractional population of state A is defined by the ratio of the number of samples finally labeled as state A to the total number of samples ($f_A = \sum_{j=1}^N y_A^j / N$). For the initial boundary point b , the only constraint is that it should not be large than 1 or smaller than -1; otherwise, state A and state B will be regarded as one state by our algorithm. Δt can be anywhere between the molecular relaxation timescale ($\Delta t \gtrsim 0.5$) and the interconversion timescale between state A and B, which is around the implied timescale of $t_1 = 54$ shown in the [supplementary material](#).

Here, all the results are obtained by setting the hyper-parameter $\beta = 0.03$ in Eq. (3) [not to be confused with the inverse temperature $(k_B T)^{-1}$], which can be determined by choosing the turning point on the rate-distortion plot (see the [supplementary material](#)). However, as long as β is not too large, we found that our results are still very robust to the selection of β .

D. Four-well analytical potential

We now apply SPIB to a four-well analytical potential where we do not assume any prior knowledge about the system such as the number of metastable states. In this case, we arbitrarily discretized the input data space into sufficiently fine grids as our initial state labels, shown in Fig. 6(a). We set the RC dimension $d = 1$, the number of pseudo-inputs $K = 10$, and the dimension of state labels $D = 10$. In other words, here we have deliberately taken K and D to be arbitrarily large relative to the true number of metastable states. In order to let the RC only contain the important information, we chose $\beta = 0.01$ (see the [supplementary material](#)).

Figure 6 shows the state labels and RC learned by SPIB using different time delays Δt . There are several interesting observations that can be made here. First, in Fig. 6(b), we find that SPIB can still obtain the correct state labels by choosing an appropriate time delay ($\Delta t = 0.5$) without any prior information. This is very promising for practical problems as a precise state definition or even the number of states is usually unavailable in complex systems. Second, we also find that a dynamically truthful discrete-state representation can be obtained by SPIB using different time delays Δt . When the time delay increases ($\Delta t = 2.0$), the original state C and state D shown in Fig. 5(a) cannot be distinguished by SPIB any

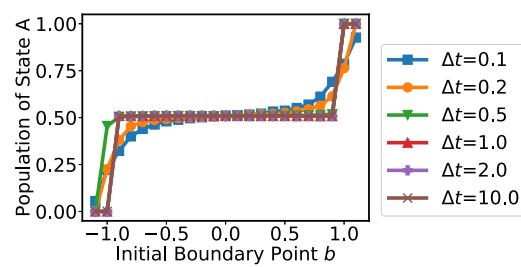


FIG. 4. The robustness of SPIB in the double-well analytical potential. The x axis represents the initial boundary point b , while y axis represents the converged fractional population of state A ($\sum_{j=1}^N y_A^j / N$). The lines start to overlap especially when $\Delta t \gtrsim 0.5$.

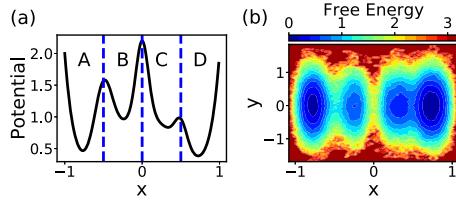


FIG. 5. Four-well analytical potentials projected along the x axis (a) and its corresponding probability $P(x, y)$ distribution of the generated trajectory (b), plotted as the free energy $-k_B T \log P(x, y)$.

more [Fig. 6(c)]. If we further increase the time delay ($\Delta t = 10$), even the original state A and state B will become indistinguishable [Fig. 6(d)]. The time dependence of state labels can be explained by the different timescales of transitions between states (Fig. 7). These results unequivocally shed light into the role of the time delay Δt in our algorithm—it filters out the fast modes of dynamics and provides a dynamics-based coarse-grained understanding of the complex

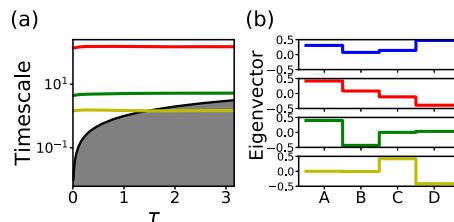


FIG. 7. The implied timescales (a) and corresponding eigenvectors (b) for the four-well analytical potential. (a) The converged values of the implied timescales are $t_2 = 151.7$ (red), $t_3 = 5.3$ (green), and $t_4 = 1.5$ (yellow). The gray area under the black line represents the timescale that is smaller than the lag time τ . (b) The first eigenvector (blue line) represents the stationary probability distribution; the second eigenvector (red line) mainly represents the transition between state A and state D; the third eigenvector (green line) represents the transition between state A and state B; the last eigenvector (yellow line) represents the transition between state C and state D.

system. We also point out that although our results depend on the selection of time delay, they are, in fact, still very robust to changes of Δt . Figure 8 shows that a broad range of Δt can result in the same discrete-state representation.

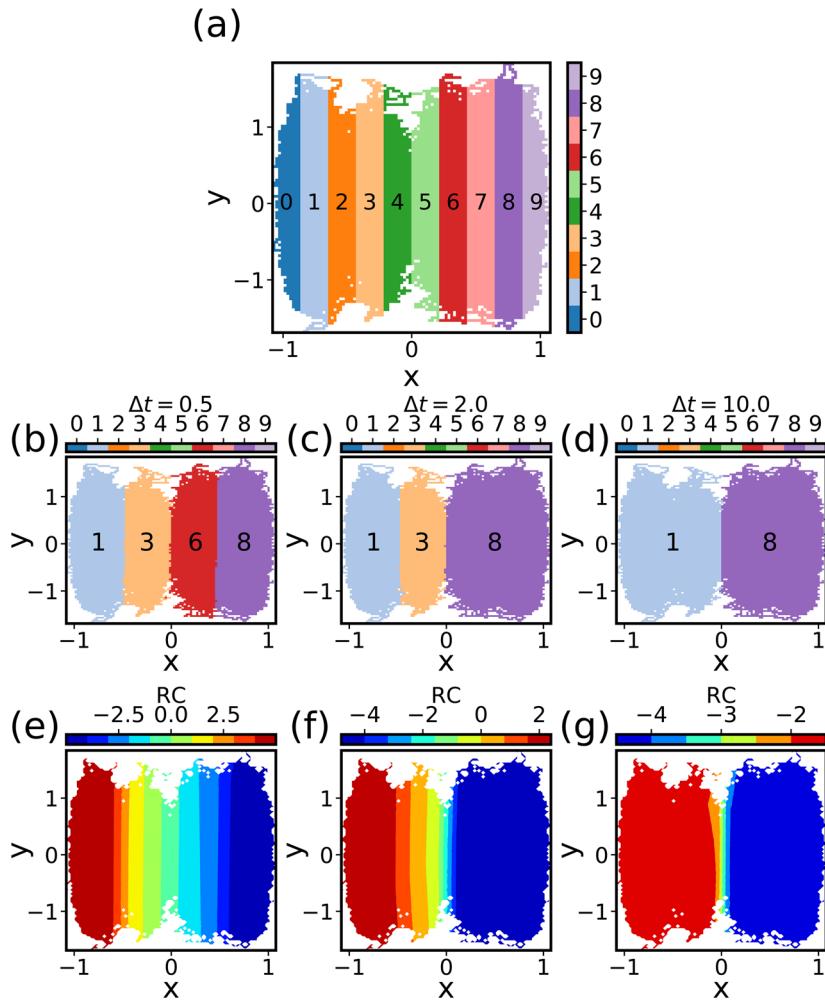


FIG. 6. The time delay dependent discrete-state representation of the four-well potential model. The initial state labels are shown in (a), while the converged results for different time delays are presented in (b)–(g). The middle row shows the state labels for different time delays, while the bottom row shows the corresponding RC. The state labels and RC were learned using the time delay 0.5 (b) and (e), 2 (c) and (f), and 10 (d) and (g), respectively.

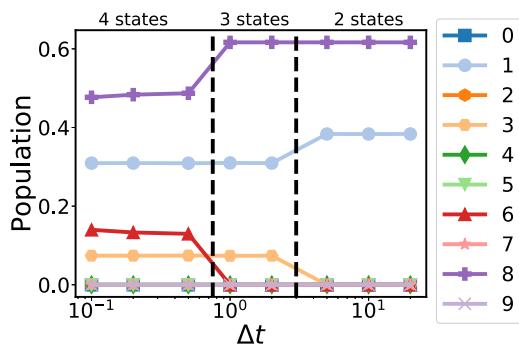


FIG. 8. The robustness of SPIB on the four-well analytical potential can be seen by plotting the fractional population of different states ($f_i = \sum_{j=1}^N y_i^j / N$ for $i = 0, \dots, 9$). With different time resolutions (or time delays Δt), the system is coarse grained into four states, three states, and two states.

E. Alanine dipeptide in vacuum

Finally, we employ SPIB to study conformation transitions in the small biomolecule alanine dipeptide. The trajectory here was expressed in terms of four dihedral angles ϕ , ψ , θ , and ω , illustrated in Fig. 9. Here, we discretized the input data space along ϕ into ten grids as our initial state labels, as shown in Fig. 10(a), and set the RC dimension $d = 2$, the number of pseudo-inputs $K = 10$, and the dimensionality of state labels $D = 10$. $\beta = 0.01$ was chosen to generate the most informative RC.

Similar to our previous results for the four-well analytical potential, in Fig. 10(b), we show how SPIB can still learn successfully the state labels corresponding to the three well-known free energy minima in the ϕ - ψ space shown in Fig. 9(b). When the time delay $\Delta t = 2$ ps, the two free energy minima located in the top-left corner of the ϕ - ψ space [Fig. 9(b)] become indistinguishable from a dynamical perspective, given that the interconversion times between these two metastable states is now close to Δt (Fig. 12). Thus, only two states are obtained in Fig. 10(c). We then further demonstrate in Fig. 11 that such a coarse-grained understanding obtained by SPIB is still very robust in alanine dipeptide, as the same state labels are obtained with a broad range of Δt .

The 2D RC so learnt through SPIB is presented in Figs. 10(d)–10(g). The free energy surface in the RC space shown in Figs. 10(f)

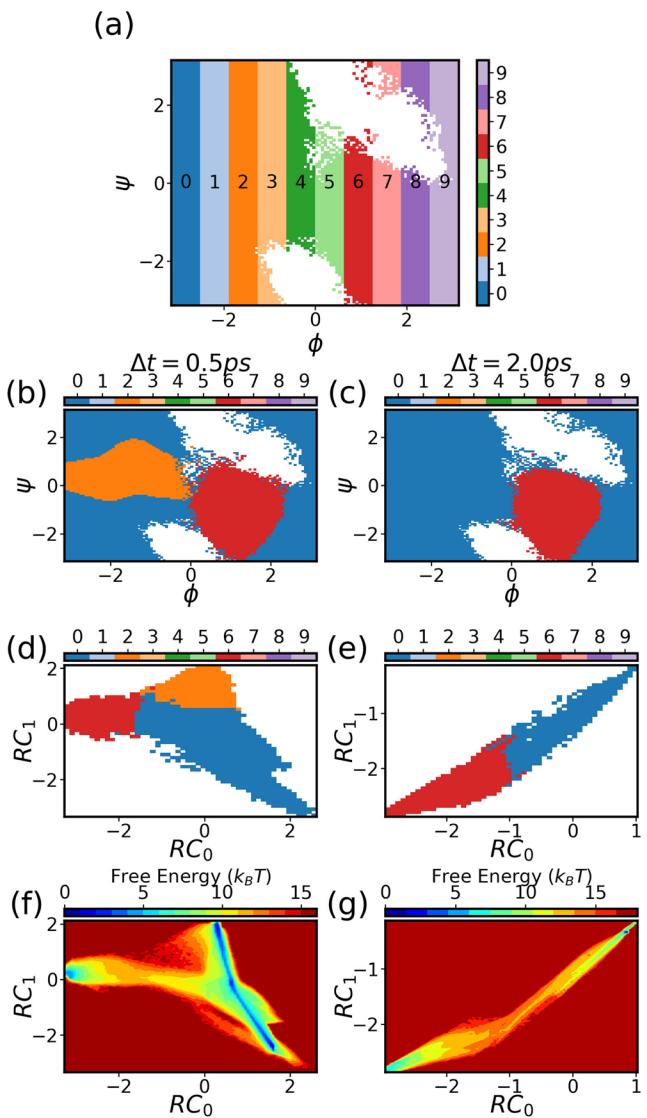


FIG. 10. The time-dependent discrete-state representation of alanine dipeptide in vacuum. The initial state labels are shown in (a). A three-state representation was learned by using the time delay $\Delta t = 0.5$ ps (b), (d), and (f), and a two-state representation was obtained by using the time delay $\Delta t = 2$ ps (c), (e), and (f). (b) and (c) in the second row are the state labels projected to ϕ - ψ space. The color (or state label) in each grid corresponds only to the state label with the highest fraction of samples for the respective grid point. (d) and (e) in the third row are the state labels learned in the 2D RC space. (f) and (g) in the fourth row shows the free energy surface $[-k_B T \log P(RC_0, RC_1)]$ in the 2D RC space.

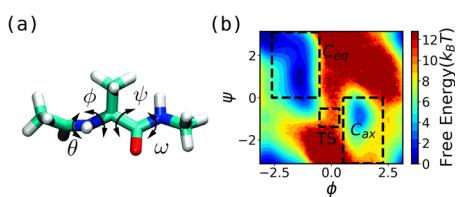


FIG. 9. (a) Alanine dipeptide molecule illustrated along with four dihedral angles: ϕ ($C-N-C_a-C$), ψ ($N-C_a-C-N$), θ ($O-C-N-C_a$), and ω ($C_a-C-N-C$). (b) The generated free energy surface of alanine dipeptide in vacuum at 450 K along the dihedral angles ϕ and ψ . The regions described in boxes are usually defined as state C_{eq} : $(-150^\circ \leq \phi \leq -30^\circ, 0^\circ \leq \psi \leq 180^\circ)$, C_{ax} : $(30^\circ \leq \phi \leq 130^\circ, -180^\circ \leq \psi \leq 0^\circ)$, and approximate TS: $(-30^\circ \leq \phi \leq 20^\circ, -80^\circ \leq \psi \leq -30^\circ)$.²

and 10(g) indicates that the barrier between C_{eq} and C_{ax} defined in Fig. 9(b) is much higher than the barrier between the two local minima [state 0 and state 2 in Figs. 10(b) and 10(d)] within C_{eq} . This also explains why different interconversion timescales are obtained in Fig. 12. From these results, we can see that a 1D RC is enough if we just want to identify the transitions between C_{eq} and C_{ax} . Thus, we reran the analysis of $\Delta t = 2$ ps with the RC dimension $d = 1$ and

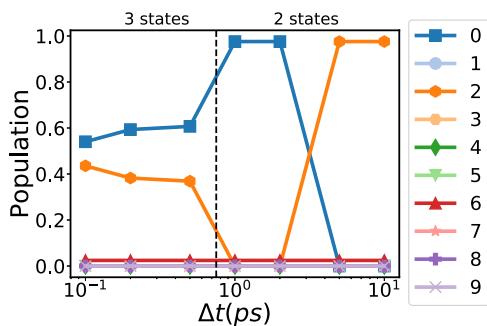


FIG. 11. The robustness of SPIB on alanine dipeptide through the fractional population of different states ($f_i = \sum_{j=1}^N y_j^i / N$ for $i = 0, \dots, 9$). With different time resolutions (or time delays Δt), the system is coarse grained into three states (e.g., $f_0 = 0.61$, $f_2 = 0.37$, and $f_6 = 0.02$ at $\Delta t = 0.5$ ps) and two states (e.g., $f_0 = 0.98$ and $f_6 = 0.02$ at $\Delta t = 2$ ps). Although it might appear that there is a flip between the state label 0 and 2 when $\Delta t > 2$ ps, the same metastable states are obtained.

show the new RC so-obtained in Fig. 13(a). Such a 1D RC can be easily used to identify the transition state, which has the same state-transition probability to two different metastable states. In Fig. 13(a), the transition state corresponds to $RC = -1.92$. To test whether our RC can identify the correct transition states, we chose to focus on the states located in the TS region shown in Fig. 9(b). By doing a traditional, detailed committor analysis, we obtained the reference committor. For this, we launched 50 1-ps trajectories with random initial Maxwell–Boltzmann velocities for each configuration in the vicinity of the TS under the constraint $-2.22 < RC < -1.62$, and then calculated their committor function $P_{C_{ax}}$ based on the fraction of trajectories reaching C_{ax} prior to C_{eq} . This committor probability distribution is shown in Fig. 13(b), where it can be seen clearly that the probability of $p_{C_{ax}}$ is characterized by a single peak centered at $p_{C_{ax}} \sim 0.5$. This shows that the RC from SPIB indeed meets the traditional expectations from a RC.²

We now show in Figs. 13(c) and 13(d) our learned state-transition density to the C_{ax} state projected in the ϕ – ψ plane and ϕ – θ plane. Figure 13(c) shows that the transition states in the TS region are aligned almost parallel to the ψ axis, suggesting that they are, in fact, almost irrelevant to ψ . Figure 13(d), however, indicates

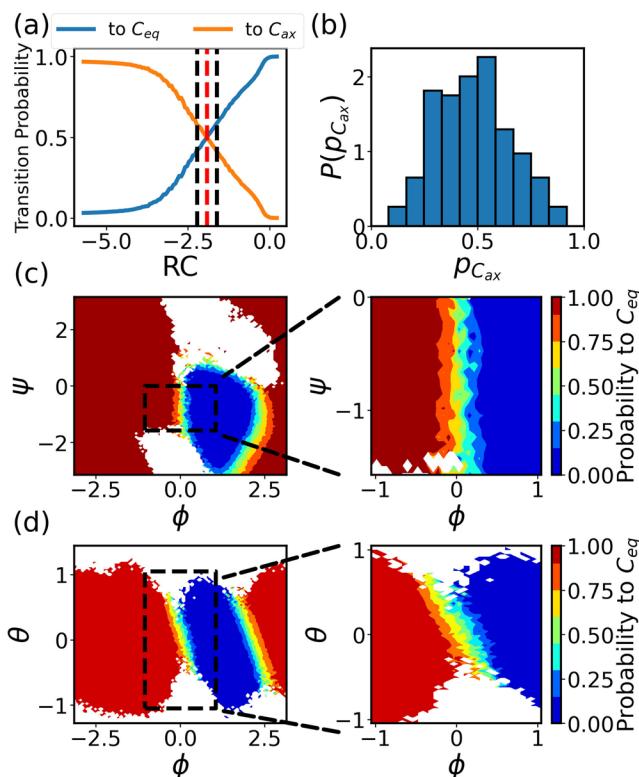


FIG. 13. (a) The 1D RC of alanine dipeptide learned by SPIB using $\Delta t = 2$ ps. The red dotted line represents the transition state ($RC^* = -1.92$), while the black dotted lines shows the neighborhood in the vicinity of the transition state range $[-2.22 < RC < -1.62]$. In (b), we show the committor probability distribution of $p_{C_{ax}}$ under the constraint $-2.22 < RC < -1.62$. (c) and (d) represent the state-transition density to C_{ax} projected to ϕ – ψ space and ϕ – θ space, respectively. In the bottom two rows, the figures in the right column are the zoomed plots around the TS region defined by Fig. 9(b).

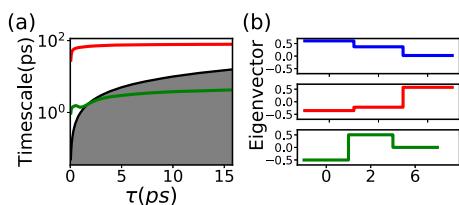


FIG. 12. The implied timescales (a) and corresponding eigenvectors (b) of alanine dipeptide. (a) The converged implied timescales $t_2 = 79$ ps (red) and $t_3 = 4.2$ ps (green). The gray area under the black line represents the timescale that is smaller than the lag time τ . (b) The first eigenvector (blue line) represents the stationary probability distribution; the second eigenvector (red line) mainly represents the transition between state C_{ax} (state 6) and state C_{eq} (state 0/2); the third eigenvector (green line) represents the transition inside state C_{eq} (or between state 0 and state 2).

that both ϕ and θ are required in order to identify the transition state. Both of these findings are in good agreement with previous reports for this system.^{2,9,43} Thus, our results confirm that the state-transition density generated by SPIB can be a reasonable substitute for the committor, and the RC learned can capture the most important dynamical information from the input trajectory to identify the correct transition states.

IV. DISCUSSION

In this work, we have proposed a deep learning based algorithm called the State Predictive Information Bottleneck (SPIB) to learn the RC from trajectory data. SPIB builds up on the insights we have introduced previously in the RAVE family of methods,^{20,21} and by changing the nature of the information bottleneck based objective function, it allows generating a more direct connection between artificial intelligence and notions from traditional chemical physics. We have first showed that the time delay Δt in a past–future information bottleneck can be interpreted as the time resolution that we

care about in a dynamical system, and through this, we can control the degree of coarse-graining obtained by our algorithm. Once a time delay Δt is selected, SPIB can automatically index the high-dimensional state space into metastable states through an iterative retraining algorithm and then characterize their dynamic behaviors in terms of state-transition density. This provides us with a promising way to analyze generic complex systems and interpret the massive data generated by MD simulations.

We have also demonstrated that the bottleneck variable learned in SPIB tries to carry the maximum information of the state-transition density, which, in principle, can be equivalent to the traditional committor function if there is a timescale separation between the state-to-state transitions and the fluctuations within metastable states. Then, through numerical tests on benchmark systems, we confirmed that the state-transition density generated by SPIB is a reasonable substitute for committor and demonstrated that our RC can focus only on the motion related to state transitions and capture the most important features from trajectories to identify the correct transition states.

While this manuscript is a proof-of-principle and theoretical demonstration of the ideas underlying SPIB, we finish this section by describing some of the new avenues pertaining to SPIB that we will explore in future work. The first one pertains to the use of SPIB for enhanced sampling. By choosing an appropriate time delay, the RC learned by SPIB can correctly identify different metastable states and even the transition states among them, which can be crucial to obtain better sampling.⁴⁴ This would involve reweighting input biased trajectories, obtained by biasing along some trial RC, for instance. The reweighting can then be performed in the manner described in Refs. 20 and 21. Second, it is natural to desire that the RC learned by SPIB can be interpreted in terms of a few human-understandable physical variables. However, unlike RAVE,²¹ here we use a nonlinear deep neural network as our encoder, and thus, the interpretation of our RC is not a trivial task. It could be possible to use approaches in representation learning in order to deal with this question of interpretability.⁴⁵ Overall, we believe our algorithm is a step toward a more complete understanding of complex systems by making use of the variability offered by the information bottleneck framework of AI^{23,24,28} and should be useful to a broad range of scientific communities.

SUPPLEMENTARY MATERIAL

See the [supplementary material](#) for other numerical details.

ACKNOWLEDGMENTS

This research was entirely supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, CPIMS Program, under Award No. DE-SC0021009. The authors thank Sun-Ting Tsai for sharing the code implementing Langevin dynamics, Luke Evans for sharing the GROMACS script, and Yihang Wang and Zachary Smith for in-depth discussions. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Bridges through Allocation Grant No. CHE180027P, which is supported by the National Science Foundation (Grant No. ACI-1548562). We also thank MARCC's Bluecrab HPC cluster for computing resources.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. The python code of SPIB using Pytorch is available for public use at <https://github.com/tiawarylab/State-Predictive-Information-Bottleneck>.

REFERENCES

- 1 Y. Wang, J. M. Lamm Ribeiro, and P. Tiwary, *Curr. Opin. Struct. Biol.* **61**, 139 (2020).
- 2 P. G. Bolhuis, C. Dellago, and D. Chandler, *Proc. Natl. Acad. Sci. U. S. A.* **97**, 5877 (2000).
- 3 P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, *Adv. Chem. Phys.* **53**, 291 (2002).
- 4 P. L. Geissler, C. Dellago, and D. Chandler, *J. Phys. Chem. B* **103**, 3706 (1999).
- 5 E. Pluharova, M. D. Baer, G. K. Schenter, P. Jungwirth, and C. J. Mundy, *J. Phys. Chem. B* **120**, 1749 (2016).
- 6 S. Roy, M. D. Baer, C. J. Mundy, and G. K. Schenter, *J. Phys. Chem. C* **120**, 7597 (2016).
- 7 C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, *J. Chem. Phys.* **108**, 1964 (1998).
- 8 R. B. Best and G. Hummer, *Proc. Natl. Acad. Sci. U. S. A.* **102**, 6732 (2005).
- 9 A. Ma and A. R. Dinner, *J. Phys. Chem. B* **109**, 6769 (2005).
- 10 B. Peters and B. L. Trout, *J. Chem. Phys.* **125**, 054108 (2006).
- 11 B. Peters, *Annu. Rev. Phys. Chem.* **67**, 669 (2016).
- 12 B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, *Appl. Comput. Harmonic Anal.* **21**, 113 (2006).
- 13 R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler, *Multiscale Model. Simul.* **7**, 842 (2008).
- 14 M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi, *J. Chem. Phys.* **134**, 124116 (2011).
- 15 F. Noé and F. Nüske, *Multiscale Model. Simul.* **11**, 635 (2013).
- 16 J. Espinosa-Garcia and J. C. Corchado, *J. Chem. Phys.* **112**, 5731 (2000).
- 17 A. Mardt, L. Pasquali, H. Wu, and F. Noé, *Nat. Commun.* **9**, 4443 (2018).
- 18 P. Tiwary and B. J. Berne, *Proc. Natl. Acad. Sci. U. S. A.* **113**, 2839 (2016).
- 19 C. X. Hernandez, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, *Phys. Rev. E* **97**, 062412 (2018).
- 20 J. M. L. Ribeiro, P. Bravo, Y. Wang, and P. Tiwary, *J. Chem. Phys.* **149**, 072301 (2018).
- 21 Y. Wang, J. M. L. Ribeiro, and P. Tiwary, *Nat. Commun.* **10**, 3573 (2019).
- 22 D. P. Kingma and M. Welling, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013).
- 23 A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, [arXiv:1612.00410](https://arxiv.org/abs/1612.00410) (2016).
- 24 N. Tishby, F. C. Pereira, and W. Bialek, [arXiv:physics/0004057](https://arxiv.org/abs/physics/0004057) (2000).
- 25 E. R. Berlekamp, *Algebraic Coding Theory*, Revised ed. (World Scientific, 2015).
- 26 C. E. Shannon, *IRE Natl. Conv. Rec.* **4**, 1 (1959).
- 27 A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurus, and K. Murphy, *An information-theoretic analysis of deep latent-variable models* (ICLR, 2018).
- 28 A. A. Alemi and I. Fischer, [arXiv:1807.04162](https://arxiv.org/abs/1807.04162) (2018).
- 29 J. M. Tomczak and M. Welling, [arXiv:1705.07120](https://arxiv.org/abs/1705.07120) (2017).
- 30 Y. M. Berezansky, Z. G. Sheftel, and G. F. Us, "Spectral decomposition of compact self-adjoint operators. Analytic functions of operators," in *Functional Analysis* (Springer, 1996), pp. 355–384.
- 31 W. C. Swope, J. W. Pitera, and F. Suits, *J. Phys. Chem. B* **108**, 6571 (2004).
- 32 F. Noé and S. Fischer, *Curr. Opin. Struct. Biol.* **18**, 154 (2008).
- 33 D. J. Wales, *J. Chem. Phys.* **130**, 204111 (2009).

³⁴T. J. Lane, G. R. Bowman, K. Beauchamp, V. A. Voelz, and V. S. Pande, *J. Am. Chem. Soc.* **133**, 18413 (2011).

³⁵E. H. Thiede, D. Giannakis, A. R. Dinner, and J. Weare, *J. Chem. Phys.* **150**, 244111 (2019).

³⁶M. Biswas, B. Lickert, and G. Stock, *J. Phys. Chem. B* **122**, 5508 (2018).

³⁷G. Bussi and M. Parrinello, *Phys. Rev. E* **75**, 056707 (2007).

³⁸H. J. C. Berendsen, D. van der Spoel, and R. van Drunen, *Comput. Phys. Commun.* **91**, 43 (1995).

³⁹M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, *SoftwareX* **1-2**, 19 (2015).

⁴⁰G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, and G. Bussi, *Comput. Phys. Commun.* **185**, 604 (2014).

⁴¹G. Bussi, D. Donadio, and M. Parrinello, *J. Chem. Phys.* **126**, 014101 (2007).

⁴²D. P. Kingma and J. Ba, *arXiv:1412.6980* (2014).

⁴³Y. Mori, K.-i. Okazaki, T. Mori, K. Kim, and N. Matubayasi, *J. Chem. Phys.* **153**, 054115 (2020).

⁴⁴G. Bussi and A. Laio, *Nat. Rev. Phys.* **2**, 200 (2020).

⁴⁵Y. Bengio, A. Courville, and P. Vincent, *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798 (2013).