THE UNIVERSITY OF CHICAGO

ADVANCING FRAMEWORKS FOR ANALYZING COMPLEX REACTION DYNAMICS FROM TRAJECTORY DATA

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMISTRY

BY
CHATIPAT LORPAIBOON

CHICAGO, ILLINOIS
DECEMBER 2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisors, Aaron Dinner and Jonathan Weare, for their guidance and patience throughout my doctoral studies. I would also like to thank my colleagues in the Dinner and Weare groups. In particular, I thank John Strahan, whose many productive discussions contributed substantially to this work. I would also like to acknowledge my other co-authors: Adam Antoszewski, Spencer Guo, Zihan Pengmei, Steven Redford, Erik Thiede, Bodhi Vani, and Rob Webber. I greatly enjoyed collaborating with them on these projects. Finally, I thank my family for their unwavering support throughout my studies.

ABSTRACT

One approach to understanding complex reaction dynamics is to estimate dynamical observables—such as committors, mean first passage times, and transition rates—from simulation data. This dissertation extends the applicability of these data-driven methods to more challenging systems—such as those with memory effects or multiple pathways—with a view towards enabling the estimation of richer dynamical observables. Key contributions include a robust method for estimating slow modes, the incorporation of memory effects via the Mori–Zwanzig formalism for accurate estimation of committors and related quantities, and a generalization of transition path theory that allows for direct analysis of specific reaction steps and pathways. Together, these advances enable detailed analysis of reaction kinetics and mechanism directly from simulation data.

CHAPTER 1

INTRODUCTION

Molecular simulation provides atomistic insight into reaction kinetics and mechanisms. However, many important processes occur on timescales far beyond those accessible by direct simulation. A promising strategy is to estimate dynamical observables—such as committors, mean first passage times, and transition rates—from ensembles of short trajectories that collectively sample the state space. Even with the same computational cost, short simulations are often more effective than a few long ones because they can be concentrated in regions of interest (e.g., transition states), improving sampling where it matters most, whereas long simulations tend to remain trapped in metastable states. This dissertation advances data-driven analysis of complex reaction dynamics by developing frameworks that extend applicability to more challenging systems—such as those with memory effects or multiple pathways—with a view towards enabling the estimation of richer dynamical observables, including those for specific reaction pathways or steps.

Throughout this work, we adopt a data-driven approach that makes minimal assumptions about the dynamics. We assume that the dynamics can be modeled by a Markov process X_t with stationary distribution π and transition operator

$$\mathcal{T}^\tau f(x) = \mathbb{E}[f(X_\tau) \mid X_0 = x], \tag{1.1}$$

where τ is a time lag. Rather than fitting explicit dynamical models, we compute observables by solving equations that involve inner products of $\mathcal{T}^\tau$ and related operators, which we estimate directly from data. This avoids unnecessary intermediate approximations and provides greater flexibility in representing dynamical observables.

One application of this approach is to identify slow modes—collective variables (CVs) that capture long-lived patterns in the dynamics. These modes are typically defined as

eigenvectors ψ of the transition operator with the longest implied timescales (autocorrelation times) σ in the eigenvalue problem

$$\mathcal{T}^\tau \psi_k(x) = e^{-\tau/\sigma_k} \psi_k(x). \tag{1.2}$$

Time-lagged independent component analysis (TICA), also known as extended dynamical mode decomposition (EDMD), approximates these modes via basis expansion $\psi_k(x) = \sum_j \phi_j(x) v_{jk}$ and solves the generalized eigenvalue problem

$$\sum_j \langle \phi_i, \mathcal{T}^\tau \phi_j \rangle_\pi v_{jk} = \sum_j \langle \phi_i, \phi_j \rangle_\pi v_{jk} e^{-\tau/\sigma_k}. \tag{1.3}$$

When the dynamics satisfy detailed balance, the variational approach to conformational dynamics (VAC) finds these modes by maximizing the time-lagged autocorrelation

$$\sum_i \frac{\langle \psi_i, \mathcal{T}^\tau \psi_i \rangle_\pi}{\langle \psi_i, \psi_i \rangle_\pi}, \tag{1.4}$$

where $\langle \psi_i, \psi_j \rangle_\pi = \delta_{ij}$, $\langle f, g \rangle_\pi = \int f(x) g(x) \pi(x) \, \mathrm{d}x$, and δ_{ij} is the Kronecker delta. VAC reduces to TICA when ψ are approximated with a linear basis expansion. As shown in Webber et al. [1] and Schultze and Grubmüller [2], VAC can be sensitive to the choice of time lag. In Chapter 2, we introduce integrated VAC (IVAC), which improves robustness by integrating over a range of time-lagged correlations.

When the process of interest is a transition between from state A to state B, the key observables are statistics of the transition path ensemble (TSE), i.e., paths that start in A, end in B, and are within the transition region $D \subseteq (A \cup B)^\mathsf{c}$ between these times. Transition path theory (TPT) [3] provides a formalism to compute collective variables and kinetic statistics that characterize these transitions. TPT statistics are expressed as time integrals of

observables $F(t)$ along transitions paths,

$$\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} F(t) \mathbb{1}_A(X_{S^-_{D^\mathsf{c}}(t)}) \mathbb{1}_B(X_{S^+_{D^\mathsf{c}}(t)}) \, \mathrm{d}t, \tag{1.5}$$

where the first time after and last time before t that the system is in D^c are, respectively,

$$S^+_{D^\mathsf{c}}(t) = \min\{s \mid s \geq t, X_s \in D^\mathsf{c}\}, \tag{1.6}$$

$$S^-_{D^\mathsf{c}}(t) = \max\{s \mid s \leq t, X_s \in D^\mathsf{c}\}, \tag{1.7}$$

and $\mathbb{1}_C(x) = 1$ is the indicator function of set C. To estimate these statistics from finite trajectories, TPT reformulates them in terms of the forward committor, the probability that the system at x will enter B before $D^\mathsf{c} \setminus B$,

$$q_+(x) = \mathbb{E}[\mathbb{1}_A(X_{S^+_{D^\mathsf{c}}(t)}) \mid X_t = x], \tag{1.8}$$

and the backward committor, the probability that the system at x exited A after $D^\mathsf{c} \setminus A$,

$$q_-(x) = \mathbb{E}[\mathbb{1}_A(X_{S^+_{D^\mathsf{c}}(t)}) \mid X_t = x]. \tag{1.9}$$

Analogous to TICA for slow modes, dynamical Galerkin approximation [4, 5] (DGA) is a linear method for estimating committors and other prediction functions. Its original formulation by Thiede et al. [4] assumes Markovian dynamics and infinitesimal time lags. However, molecular simulation data are typically sampled at finite time intervals, and longer time lags are often needed to recover approximately Markovian behavior when degrees of freedom are omitted (e.g., velocities or solvent coordinates) or the linear model is insufficiently expressive. Similar issues affect estimators for TPT statistics based on infinitesimal time lags. In Chapter 3, we reformulate both DGA and estimators for TPT statistics to use finite time lags, improving estimates by better approximating Markovian behavior.

While long time lags can improve estimates by better approximating Markovian behavior, memory enables accurate estimates even when such assumptions are violated. Even when the true dynamics are Markovian, projected dynamics (e.g., when velocities or solvent degrees of freedom are omitted) can exhibit memory effects. Models with limited expressivity can also produce apparent non-Markovian behavior because it cannot fully represent the dynamics. Earlier work [6, 7] incorporated memory effects to improve slow mode estimation. In Chapter 4, we lift the Markov assumption for estimating committors and related statistics by incorporating memory into DGA using the Mori–Zwanzig formalism.

Linear models often require carefully chosen basis functions, which is especially challenging for systems with high dimension. Although the dynamics often lie on a low-dimensional manifold, a linear basis may need to be much larger to capture relevant observables. To address this, we develop nonlinear methods for learning dynamical observables using highly expressive models such as neural networks. In Chapter 5, we use a variational approach for learning committors and transition rates, which requires detailed balance and equilibrium sampling (or appropriate reweighting). In Chapter 6, we introduce an iterative approach that does not require detailed balance or equilibrium sampling, at the cost of instability.

Several approaches have been proposed to analyze individual pathways from short simulations. One strategy is to construct a simplified model of the dynamics—such as a Markov state model—and generate long trajectories from it to obtain a transition path ensemble. Because this provides complete transition paths, estimating statistics is straightforward, regardless of the complexity of the specified path ensemble or observable. The drawback, however, is that the simplified dynamics often poorly approximate the true dynamics. Another strategy is to analyze the gross or net reactive flux between partitions of the transition region, which can be estimated using TPT. For example, Noé et al. [8] used the net reactive flux to dissect folding pathways of a PinWWW domain. A related idea is to project the reactive current onto chosen CVs, as we did in Strahan et al. [5] to study Trp-cage folding and unfolding.

Flux- and current-based approaches examine effective dynamics, which can miss important details (e.g., rates of interchange between pathways) and do not yield an ensemble of physical trajectories (so general path functionals cannot be computed). In Chapter 7, we overcome these limitations by developing augmented transition path theory (ATPT), which extends TPT to compute statistics for specific reaction steps and pathways directly from the true dynamics. Unlike flux- and current-based approaches, ATPT defines an explicit ensemble of physical transition paths and enables computation of general path functionals. This enables detailed analysis of complex reaction mechanisms, allowing us to compare competing pathways, identify key intermediates, and quantify stepwise contributions—all without relying on simplified models or effective dynamics.

We conclude by outlining connections between these methods, and discussing remaining challenges and future directions in the field.

CHAPTER 2

INTEGRATED VARIATIONAL APPROACH TO CONFORMATIONAL DYNAMICS

This chapter is adapted with permission from C. Lorpaiboon et al., "Integrated variational approach to conformational dynamics: a robust strategy for identifying eigenfunctions of dynamical operators", J. Phys. Chem. B **124**, 9354–9364 (2020).

One approach to analyzing the dynamics of a physical system is to search for long-lived patterns in its motions. This approach has been particularly successful for molecular dynamics data, where slowly decorrelating patterns can indicate large-scale conformational changes. Detecting such patterns is the central objective of the variational approach to conformational dynamics (VAC), as well as the related methods of time-lagged independent component analysis and Markov state modeling. In VAC, the search for slowly decorrelating patterns is formalized as a variational problem solved by the eigenfunctions of the system's transition operator. VAC computes solutions to this variational problem by optimizing a linear or nonlinear model of the eigenfunctions using time series data. Here, we build on VAC's success by addressing two practical limitations. First, VAC can give poor eigenfunction estimates when the lag time parameter is chosen poorly. Second, VAC can overfit when using flexible parameterizations such as artificial neural networks with insufficient regularization. To address these issues, we propose an extension that we call integrated VAC (IVAC). IVAC integrates over multiple lag times before solving the variational problem, making its results more robust and reproducible than VAC's.

2.1 Introduction

Many physical systems exhibit motion across fast and slow timescales. Whereas individual subcomponents may relax rapidly to a quasi-equilibrium, large collective motions occur over timescales that are orders of magnitude longer. These slow motions are often the most scientifically significant. For instance, observing the large-scale conformational changes that govern protein function requires microseconds to days, even though individual atomic vibrations have periods of femtoseconds. However, when exploring new systems, such slow collective processes may not be fully understood from the outset. Rather, they must be detected from time series data.

One approach for automating this process is the "variational approach to conformational dynamics" (VAC) [10–12]. In the VAC framework, slow dynamical processes are identified using functions that decorrelate slowly. These functions are the eigenfunctions of a self-adjoint operator associated with the system's dynamics known as the *transition operator*. The transition operator evolves expectations of functions over the system's state forward in time and completely defines the dynamics on a distributional level. VAC estimates the transition operator's eigenfunctions by constructing a linear or nonlinear model and using data to optimize parameters in the model. VAC encompasses commonly used approaches such as time-lagged independent component analysis [12–15] and eigenfunction estimates constructed using Markov state models. [15–18] In addition, recent VAC approaches use artificial neural networks to learn approximations to the eigenfunctions. [19, 20]

While VAC has been successful in some applications, the approach has limitations. The accuracy of the estimated eigenfunctions depends strongly on the function space in which the eigenfunctions are approximated, the amount of data available, and a hyperparameter known as the lag time. In our previous work [1] we gave a comprehensive error analysis for the linear VAC algorithm. This error analysis showed that the choice of lag time can be critical to achieving an accurate VAC scheme. Choosing a lag time that is too short can

cause substantial systematic bias in estimated eigenfunctions, while choosing a lag time that is too long can make VAC exponentially sensitive to sampling error.

In this chapter, we present an extension of the VAC procedure in which we integrate the correlation functions in VAC over a time window. We term this approach integrated VAC (IVAC). Because IVAC is less sensitive to the choice of lag time, it reduces error compared to VAC. Additionally, when IVAC is applied using an approximation space parameterized by a neural network, the approach leads to stable training and mitigates the overfitting problems associated with VAC.

We organize the rest of the chapter as follows. In the theory section, we review the role of the transition operator and its eigenfunctions, and we introduce the VAC approach for estimating eigenfunctions. We then present the procedure for IVAC. In the results section, we evaluate the performance of IVAC on two model systems. We conclude with a summary and a discussion of further ways IVAC can be extended. Software implementing IVAC is available at https://github.com/chatipat/ivac.

2.2 Methods

2.2.1 Background

In this section, we review the VAC theoretical framework [10, 21] that shows how the slowly decorrelating functions in a physical system can be identified using a linear operator known as the transition operator.

We assume that the system of interest is a continuous-time Markov process $X_t \in \mathbb{R}^n$ with a stationary, ergodic distribution μ (specifically a Feller process [22]). We use $\mathbb{E}$ to denote expectations of the process X_t started from μ. For example, if μ is the Boltzmann distribution associated with the Hamiltonian H and temperature T, then expectations of the

process satisfy

$$\mathbb{E}[f(X_t)] = \frac{\int f(x)e^{-H(x)/k_BT}dx}{\int e^{-H(x)/k_BT}dx} \tag{2.1}$$

for all $t \geq 0$. However, our results are valid for systems with other, more general, stationary distributions.

The transition operator

To begin, we consider the space of real-valued functions with finite second moment ($\mathbb{E}[f(X_0)^2] < \infty$). Equipped with the inner product

$$\langle f, g \rangle = \mathbb{E}[f(X_0)g(X_0)] \tag{2.2}$$

this forms a Hilbert space, which we denote L^2_μ. We define the transition operator [22] at a lag time τ to be the operator

$$\mathcal{T}^\tau f(x) = \mathbb{E}[f(X_\tau) \mid X_0 = x] \tag{2.3}$$

applied to a function $f \in L^2_\mu$. Here, we are interpreting the conditional expectation as a function of the initial point x.

The transition operator is also called the Markov or (stochastic) Koopman operator [15, 23]. We use the term transition operator as it is well-established in the literature on stochastic processes, and the terminology emphasizes the connection with finite-state Markov chains. For a finite-state Markov chain, f is a column vector and $\mathcal{T}^\tau$ is a row-stochastic transition matrix.

The transition operator lets us rewrite correlation functions in terms of inner products in

L^2_μ:

$$\mathbb{E}[f(X_0)g(X_\tau)] = \langle f, \mathcal{T}^\tau g\rangle. \tag{2.4}$$

Moreover, we can express the slow motions of a system's dynamics in terms of the transition operator. The slow motions are identified by functions f for which the normalized correlation function

$$\frac{\mathbb{E}[f(X_0)f(X_\tau)]}{\mathbb{E}[f(X_0)f(X_0)]} = \frac{\langle f, \mathcal{T}^\tau f\rangle}{\langle f, f\rangle} \tag{2.5}$$

is large. We will show in the next subsection that these slowly decorrelating functions lie in the linear span of the top eigenfunctions of the transition operator.

Eigenfunctions of the transition operator

We can immediately see that $\mathcal{T}^\tau$ has the constant function as an eigenfunction, because

$$\mathcal{T}^\tau 1 = \mathbb{E}[1 \mid X_0 = x] = 1. \tag{2.6}$$

However, there is no guarantee that any other eigenfunctions exist. We must therefore impose additional assumptions.

We first assume that X_t obeys detailed balance. For any functions $f, g \in L^2_\mu$, we have

$$\mathbb{E}[f(X_0)g(X_\tau)] = \mathbb{E}[f(X_\tau)g(X_0)], \tag{2.7}$$

or equivalently

$$\langle f, \mathcal{T}^\tau g\rangle = \langle \mathcal{T}^\tau f, g\rangle. \tag{2.8}$$

This detailed balance condition ensures that $\mathcal{T}^\tau$ is a self-adjoint operator on L^2_μ.

Next we assume that $\mathcal{T}^\tau$ is a compact operator. In our context, assuming compactness is

the same as assuming that the action of $\mathcal{T}^\tau$ can be decomposed as an infinite sum involving eigenfunctions and eigenvalues:

$$\mathcal{T}^\tau f(x) = \sum_{i=1}^{\infty} e^{-\sigma_i \tau} \langle \eta_i, f \rangle \eta_i(x). \tag{2.9}$$

Our assumption of compactness is made for the sake of simplicity; in fact a weaker assumption of quasi-compactness is sufficient. We refer the reader to Webber et al. [1] for a more general treatment.

At all lag times $\tau > 0$, the function η_i is an eigenfunction of the transition operator T^τ with eigenvalue

$$\lambda_i^\tau = e^{-\sigma_i \tau}. \tag{2.10}$$

The eigenvalues are indexed so that

$$0 = \sigma_1 < \sigma_2 \leq \sigma_3 \leq \cdots \tag{2.11}$$

and $\lim_{i \to \infty} \sigma_i = \infty$. Because the process is ergodic, it is known that the largest eigenvalue $\lambda_1^\tau = 1$ is a simple eigenvalue and all other eigenvalues are bounded away from 1. The particular dependence of the eigenvalues on τ occurs because the transition operator can be written as

$$\mathcal{T}^\tau = e^{\mathcal{L}\tau}, \, \forall \tau \geq 0 \tag{2.12}$$

where $\mathcal{L}$ is an operator known as the infinitesimal generator [22]. We note that it is also common to consider the *implied timescale* (ITS) associated with eigenfunction i, defined as

$$\mathrm{ITS}_i = \sigma_i^{-1}. \tag{2.13}$$

We can use the eigenvalues and eigenvectors of the transition operator to rewrite the

normalized correlation function (2.5). Observing that $\mathcal{T}^0 f(x) = f(x)$ and substituting (2.9) into the numerator and denominator of (2.5) gives

$$\frac{\mathbb{E}[f(X_0)f(X_\tau)]}{\mathbb{E}[f(X_0)f(X_0)]} = \frac{\sum_{i=1}^{\infty} e^{-\sigma_i \tau}\langle \eta_i, f\rangle^2}{\sum_{i=1}^{\infty}\langle \eta_i, f\rangle^2}. \tag{2.14}$$

We now consider which functions maximize the normalized correlation function. Applying (2.11), we find that the normalized correlation function is maximized when we set f to be the constant function $f(x) = \eta_1(x) = 1$, because

$$\frac{\sum_{i=1}^{\infty} e^{-\sigma_i \tau}\langle \eta_i, f\rangle^2}{\sum_{i=1}^{\infty}\langle \eta_i, f\rangle^2} \leq \frac{\sum_{i=1}^{\infty} e^{-\sigma_1 \tau}\langle \eta_i, f\rangle^2}{\sum_{i=1}^{\infty}\langle \eta_i, f\rangle^2} \tag{2.15}$$

$$= e^{-\sigma_1 \tau} \tag{2.16}$$

for all functions $f \in L^2_\mu$. If we constrain the search to functions that are orthogonal to η_1, i.e., functions where

$$\langle \eta_1, f\rangle = \mathbb{E}[f(x)] = 0 \tag{2.17}$$

and assume $\sigma_2 > \sigma_3$, the normalized correlation function is maximized when $f = \eta_2$. If we constrain f to be orthogonal to both η_1 and η_2, then the next slowest decorrelating function would be η_3, and so forth. Maximizing the normalized correlation function at any lag time τ is therefore equivalent to identifying the eigenfunctions of the transition operator.

Because of the connection to slowly decorrelating functions, the eigenfunctions provide a natural coordinate system for dimensionality reduction. The first few eigenfunctions provide a compact representation of all the slowest motions of the system. Additionally, clustering data based on the eigenfunction coordinates makes it possible to identify metastable states.

The variational approach to conformational dynamics

The "variational approach to conformational dynamics" (VAC) is a procedure for identifying eigenfunctions by maximizing the normalized correlation function. The first eigenfunction, $\eta_1(x) = 1$, is known exactly and is set to the constant function. To identify subsequent eigenfunctions, we parameterize a candidate solution f using a vector of parameters θ. We then construct an estimate γ_i for the ith eigenfunction by tuning the parameters to maximize (2.5). We set $\gamma_i = f_{\theta'}$, where

$$\theta' = \underset{\theta}{\operatorname{argmax}} \frac{\mathbb{E}[f_\theta(X_0) f_\theta(X_\tau)]}{\mathbb{E}[f_\theta(X_0) f_\theta(X_0)]} \tag{2.18}$$

subject to $\langle f_\theta, \gamma_j \rangle = 0$ for all $j < i$. In practice, we use empirical estimates of the correlations constructed from sampled data. For instance, if our data set consists of a single equilibrium trajectory $x_0, x_\Delta, \ldots x_{T-\Delta}$, we would then construct the estimate

$$\hat{\mathbb{E}}[f(X_0)g(X_\tau)] = \frac{\Delta}{T-\tau} \sum_{s=0}^{\frac{T-\Delta-\tau}{\Delta}} \frac{f(x_{s\Delta})g(x_{s\Delta+\tau}) + f(x_{s\Delta+\tau})g(x_{s\Delta})}{2}. \tag{2.19}$$

Here and in the rest of the chapter, we use the $\hat{}$ symbol to indicate quantities constructed using sampled data.

Once we have obtained an estimated eigenfunction $\hat{\gamma}_i$ using data, we can estimate the associated eigenvalue and implied timescale using

$$\hat{\lambda}_i^\tau = \frac{\hat{\mathbb{E}}[\hat{\gamma}_i(X_0)\hat{\gamma}_i(X_\tau)]}{\hat{\mathbb{E}}[\hat{\gamma}_i(X_0)\hat{\gamma}_i(X_0)]} \tag{2.20}$$

$$\hat{\sigma}_i = -\frac{1}{\tau} \log \hat{\lambda}_i^\tau. \tag{2.21}$$

If the sampling is perfect, the variational principle ensures that VAC eigenvalues and VAC implied timescales are bounded from above by the true eigenvalues $e^{-\sigma_i \tau}$ and implied

timescales σ_i^{-1}, and the upper bound is achieved when the VAC eigenfunction is the true eigenfunction η_i. However, since the empirical estimate (2.20) is used in practice, it is possible to obtain estimates that exceed the variational upper bound.

The earliest VAC approaches estimated the eigenfunctions of the transition operator by using linear combinations of basis functions $\{\phi_i\}$, a procedure now known as linear VAC. In linear VAC, the optimization parameters are the unknown linear coefficients v, which solve the generalized eigenvalue problem

$$\hat{C}(\tau)v_i = \hat{\lambda}_i^{\tau}\hat{C}(0)v_i, \tag{2.22}$$

where

$$\hat{C}_{jk}(t) = \hat{\mathbb{E}}[\phi_j(X_0)\phi_k(X_t)]. \tag{2.23}$$

In approaches known as time-lagged independent component analysis [13] and relaxation mode analysis [21, 24], the basis functions $\{\phi_i\}$ were chosen to be the system's coordinate axes. This choice of approximation space is still commonly used to construct collective variable spaces either for analyzing dynamics or for streamlining further sampling. Markov state models (MSMs) provide an alternative approach for estimating eigenfunctions using linear combinations of basis functions [16–18, 25]. MSMs can serve as general dynamical models for the estimation of metastable structures and chemical rates [25–28]. When MSMs are applied to estimate eigenfunctions and eigenvalues, the approach is equivalent to performing linear VAC using a basis of indicator functions on disjoint sets [15].

Noé and Nuske [10] unified the linear VAC approaches and exploited a general variational principle for identifying eigenvalues and eigenfunctions of the transition operator. Subsequent work further developed the methodology and introduced more general linear basis functions [11, 29–31]. Moreover, it was observed that the general variational principle allows one to model the eigenfunctions using nonlinear approximation spaces such as the output of a neural

network [19, 20]. This can lead to very flexible and powerful approximation spaces. However, in our experience, the greater flexibility can also lead to overfitting problems that need to be addressed through regularization.

In a common nonlinear VAC approach, a neural network outputs a set of functions $\phi_1, \phi_2, \ldots, \phi_S$ that serves as a basis set for linear VAC calculations. The network parameters are then optimized to maximize the VAMP score [32], which under our assumption of detailed balance can be calculated using

$$\text{VAMP-}k = \sum_{i=1}^{S} |\hat{\lambda}_i^\tau|^k. \tag{2.24}$$

The hyperparameter k is typically set to 1 or 2. In this chapter, we use the VAMP-1 score, since we find that it leads to more robust training. We note that the score function we use is also called the generalized matrix Rayleigh quotient [33].

Challenges in VAC calculations

A major challenge in VAC calculations is selecting the lag time τ. Since the early days of VAC, it was noted that lag times that are too short or too long can lead to inaccurate eigenfunction estimates [34, 35]. Our recent work [1] revealed that the sensitivity to lag time is caused by a combination of approximation error at short lag times and estimation error at long lag times. In this section, we describe the impact of approximation error and estimation error and provide a schematic (Figure 2.1) that illustrates the tradeoff between approximation error and estimation error at different lag times.

Approximation error is the systematic error of VAC that exists even when VAC is performed with an infinite data set. We expect approximation error to dominate the calculation when the basis set is of poor quality and our approximation space cannot faithfully represent the eigenfunctions of the transition operator. The approximation error

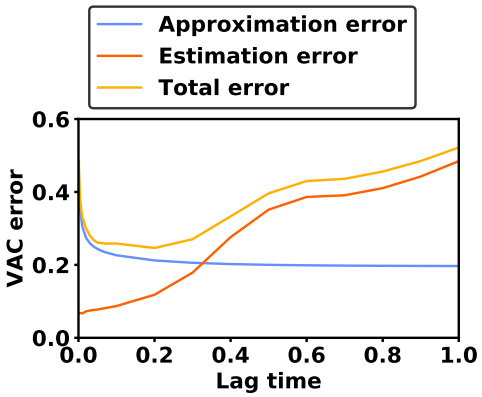

Figure 2.1: Schematic illustrating the sources of VAC error at different lag times. Even without sampling, VAC solutions have approximation error. Random variation due to sampling contributes additional estimation error.

is greatest at short lag times, and it decreases and eventually stabilizes as the lag time is increased. Therefore, VAC users can typically reduce approximation error by avoiding the very shortest lag times.

Estimation error is the random error of VAC that comes from statistical sampling. As shown in our previous work [1], with increasing lag time the results of VAC become exponentially sensitive to small variations in the data set, leading to high estimation error. At large enough lag times, all the eigenfunction estimates $\hat{\gamma}_2^\tau, \hat{\gamma}_3^\tau, \ldots$ are essentially random noise.

In Webber et al. [1], we proposed measuring VAC's sensitivity to estimation error using the condition number κ^τ. The condition number measures the largest possible changes that can occur in the subspace of VAC eigenfunctions $\{\gamma_j^\tau, \gamma_{j+1}^\tau, \ldots, \gamma_k^\tau\}$ when there are small errors in the entries of $C(0)$ and $C(\tau)$. The condition number is calculated using the expression

$$\kappa^\tau = \frac{1}{\min\{\hat{\lambda}_{j-1}^\tau - \hat{\lambda}_j^\tau, \hat{\lambda}_k^\tau - \hat{\lambda}_{k+1}^\tau\}}. \tag{2.25}$$

For a given problem and a given lag time, we can use the condition number to determine which subspaces of VAC eigenfunctions are highly sensitive to estimation error and which

subspaces are comparatively less sensitive to estimation error.

Although we rigorously derived the condition number only in the case of linear VAC, we find that the condition number is also helpful for measuring estimation error in nonlinear VAC. If $\kappa^\tau \gtrsim 5$ at all lag times τ, then identifying eigenfunctions is very difficult and requires a large data set. We recommend that authors report the condition number along with their VAC results, helping readers to assess whether the results are potentially sensitive to estimation error.

2.2.2 Integrated VAC

To address the difficulty inherent in choosing a good lag time, we propose an extension of VAC called "integrated VAC" (IVAC) where we integrate over a range of different lag times before solving a variational problem. We find that the new approach is more robust to lag time selection and it often gives better results overall.

Just as VAC maximizes the correlation function in (2.5), IVAC solves a variational problem by identifying a subspace of functions f that maximize the integrated correlation function

$$\int_{\tau_{\min}}^{\tau_{\max}} \frac{\mathbb{E}[f(X_0)f(X_s)]}{\mathbb{E}[f(X_0)f(X_0)]}\, ds\,. \tag{2.26}$$

As in VAC, the functions solving the variational problem are the eigenfunctions of the transition operator. When the eigenfunction η_i is substituted into the integrated correlation function (2.26), the resulting expression is related to the implied timescales by

$$\int_{\tau_{\min}}^{\tau_{\max}} \frac{\mathbb{E}[\eta_i(X_0)\eta_i(X_s)]}{\mathbb{E}[\eta_i(X_0)\eta_i(X_0)]}\, ds = \frac{e^{-\sigma_i\tau_{\min}} - e^{-\sigma_i\tau_{\max}}}{\sigma_i}. \tag{2.27}$$

Therefore, like VAC, IVAC is a variational approach for identifying both eigenfunctions and implied timescales.

IVAC is a natural extension of VAC; in the limit as $\tau_{\max}$ approaches $\tau_{\min}$, IVAC gives the

same eigenfunction and implied timescale estimates as regular VAC. However, when $\tau_{\max}$ and $\tau_{\min}$ are separated from each other, the results of IVAC and VAC start to diverge. We find that IVAC with minimal tuning performs comparably to VAC with optimal tuning. IVAC has the desirable feature that it is not very sensitive to the values of $\tau_{\min}$ and $\tau_{\max}$.

Previous approaches for estimating eigenfunctions using multiple time lags have attempted to reduce approximation error by accounting for unobserved degrees of freedom [4, 7, 36, 37]. In contrast, IVAC uses multiple time lags to reduce estimation error and improve robustness to parameter choice.

Linear IVAC

Linear IVAC uses linear combinations of basis functions to maximize the integrated autocorrelation function (2.26). However, as simulation data are sampled at discrete time points, we cannot directly calculate the integral. We therefore replace (2.26) with a discrete sum taken over uniformly spaced lag times. We seek to maximize

$$\sum_{\tau=\tau_{\min}}^{\tau_{\max}} \frac{\mathbb{E}[f(X_0)f(X_\tau)]}{\mathbb{E}[f(X_0)f(X_0)]}, \tag{2.28}$$

where $\tau = \tau_{\min},\ \tau_{\min} + \Delta,\ \tau_{\min} + 2\Delta, \ldots,\ \tau_{\max}$ and Δ is the sampling interval. The discrete sum (2.28) approximates (2.26) up to a constant multiple, and its value is maximized when f lies within the span of the top eigenfunctions of the transition operator. Setting f to be the eigenfunction η_i, we can sum the resulting finite geometric series:

$$\sum_{\tau=\tau_{\min}}^{\tau_{\max}} \frac{\mathbb{E}[\eta_i(X_0)\eta_i(X_\tau)]}{\mathbb{E}[\eta_i(X_0)\eta_i(X_0)]} = \frac{e^{-\sigma_i \tau_{\min}} - e^{-\sigma_i(\tau_{\max}+\Delta)}}{1 - e^{-\sigma_i \Delta}}. \tag{2.29}$$

In linear IVAC, we optimize linear combinations of basis functions $\{\phi_i\}$ to maximize the functional (2.28). The optimization parameters are the unknown linear coefficients v, which

solve the generalized eigenvalue problem

$$\hat{I}(\tau_{\min}, \tau_{\max}) v_i = \hat{\lambda}_i \hat{C}(0) v_i, \tag{2.30}$$

where we have defined

$$\hat{C}_{jk}(t) = \hat{\mathbb{E}}[\phi_j(X_0)\phi_k(X_t)] \tag{2.31}$$

$$\hat{I}(\tau_{\min}, \tau_{\max}) = \sum_{\tau=\tau_{\min}}^{\tau_{\max}} \hat{C}(\tau). \tag{2.32}$$

We solve the generalized eigenvalue problem to obtain estimates $\hat{\gamma}_i$ for the transition operator's eigenfunctions. Then, we form the sum

$$\sum_{\tau=\tau_{\min}}^{\tau_{\max}} \frac{\hat{\mathbb{E}}[\hat{\gamma}_i(X_0)\hat{\gamma}_i(X_\tau)]}{\hat{\mathbb{E}}[\hat{\gamma}_i(X_0)\hat{\gamma}_i(X_0)]}, \tag{2.33}$$

and we estimate implied timescales by solving (2.29) for $\hat{\sigma}_i$ using a root-finding algorithm.

Nonlinear IVAC

Nonlinear IVAC maximizes the integrated correlation function (2.26) by constructing approximations in a nonlinear space of functions, for example, those represented by a neural network. Specifically, the nonlinear model provides a set of functions $\phi_1, \phi_2, \ldots, \phi_S$ that serves as a basis set for linear IVAC. The parameters are trained to maximize the VAMP-k score

$$\sum_{i=1}^{S} |\hat{\lambda}_i|^k \tag{2.34}$$

where the eigenvalues $\hat{\lambda}_i$ are defined using equation (2.30). In a linear approximation space, all values of VAMP-k scores lead to identical eigenfunction estimates. In a nonlinear approximation space, it is theoretically possible that minimizing with different values of

k would lead to different estimates. However, in practice we find there is little difference between estimates at the minima. We present our results using $k = 1$ because it leads to the most stable convergence; we found that higher values of k are prone to large gradients and, in turn, unstable training. When $k = 1$, the score function can be computed using

$$\mathrm{tr}(\hat{C}(0)^{-1}\hat{I}(\tau_{\min}, \tau_{\max})). \tag{2.35}$$

The main practical challenge in an application of nonlinear IVAC is that the basis functions $\phi_1, \phi_2, \ldots, \phi_S$ change at every iteration, requiring costly re-evaluation of $\hat{C}(0)$, $\hat{I}(\tau_{\min}, \tau_{\max})$, and the gradient of (2.35) with respect to the parameters. To reduce this cost, we have developed the batch subsampling approach described in Algorithm 1, which we apply at the start of each optimization iteration.

Algorithm 1 subsampling routine

Input: data $x_0, \ldots, x_{T-\Delta}$, $\tau_{\min}$, $\tau_{\max}$, number of samples N
for $n \in 1, 2, \ldots, N$ **do**
  Sample τ_n from $\{\tau_{\min}, \ldots, \tau_{\max}\}$
  Sample s_n from $\{0, \ldots, T - \tau_n - \Delta\}$
end for
Output: sampled pairs $(x_{s_n}, x_{s_n+\tau_n})$

In the subsampling approach, we draw a randomly chosen set of data points, which allow us to estimate the matrix entries $\hat{C}_{ij}(0)$ using

$$\sum_{n=1}^{N} \frac{\phi_i(x_{s_n})\phi_j(x_{s_n}) + \phi_i(x_{s_n+\tau})\phi_j(x_{s_n+\tau})}{2N} \tag{2.36}$$

and the matrix entries $\hat{I}_{ij}(\tau_{\min}, \tau_{\max})$ using

$$\sum_{n=1}^{N} \frac{\phi_i(x_{s_n})\phi_j(x_{s_n+\tau_n}) + \phi_i(x_{s_n+\tau_n})\phi_j(x_{s_n})}{2N\Delta/(\tau_{\max} - \tau_{\min} + \Delta)}. \tag{2.37}$$

After constructing these random matrices, we calculate the score function 2.35. We then use automatic differentiation to obtain the gradient of the score function with respect to the parameters, and we perform an optimization step. By randomly drawing new data points at each optimization step, we ensure a thorough sampling of the data set and we are able to train the nonlinear representation at reduced cost. Typically, we find that 10^3–10^4 data points per batch is enough for the score function (2.35) to be estimated with low bias.

2.3 Results and discussion

In this section, we provide evidence that IVAC is more robust than VAC and can give more accurate eigenfunction estimates. First, we show results from applying IVAC and VAC to the alanine dipeptide. VAC can provide accurate eigenfunction estimates for this test problem owing to the large spectral gap and the approximation space that overlaps closely with the eigenfunctions of the transition operator. However, VAC requires a careful tuning of the lag time. In contrast, IVAC is much less sensitive to lag time choice. IVAC gives solutions that are comparable to VAC with the optimal lag time parameter and substantially better than VAC with a poorly chosen lag time.

Second, we show results for the villin headpiece protein. Because the data set has a small number of independent samples and the neural network approximation space is flexible and prone to overfitting, VAC and IVAC suffer from estimation error at long lag times. Despite these challenges, we present a robust protocol for choosing parameters in IVAC to limit the estimation error, and we show that IVAC is less sensitive to overfitting for this problem compared with VAC.

2.3.1 Application to the alanine dipeptide

In this section we compare linear IVAC and VAC applied to Langevin dynamics simulations of the alanine dipeptide (i.e., N-acetyl-alanyl-N'-methylamide) in aqueous solvent; further

simulation details are given in the supporting information.

The alanine dipeptide is a well-studied model for conformational changes in proteins. Like many protein systems, the alanine dipeptide has dynamics that are dominated by transitions between metastable states. The top eigenfunctions are useful for locating barriers between states, as these eigenfunctions change sharply when passing from one well to another. We focus on estimating η_2 and η_3, as large changes in these eigenfunctions correspond to transitions over the alanine dipeptide's two largest barriers. We refer to the span of η_1, η_2, and η_3 as the 3D subspace.

In our experiments, we consider trajectories of length 10 ns and 20 ns. The trajectories are long enough to observe approximately 15 or 30 transitions respectively along the dipeptide's slowest degree of freedom. Folding simulations of proteins, such as the villin headpiece considered below, often have a similar number of transitions between the folded and unfolded states.

There are several features that make it possible for VAC to perform well on this example. First, the linear approximation space, which consists of all the dihedral angles in the molecular backbone, is small (just 9 basis functions), and it is known to overlap heavily with the top eigenfunctions of the dynamics. Second, we are estimating a well-conditioned subspace with a minimum condition number of just

$$\min_{\tau} \kappa^{\tau} = \min_{\tau} (\hat{\lambda}_3^{\tau} - \hat{\lambda}_4^{\tau})^{-1} = 1.4, \tag{2.38}$$

and therefore we do not expect a heavy amplification of sampling error that degrades eigenfunction estimates.

To evaluate the error in our eigenfunction estimates, we compare to "ground truth" eigenfunctions computed using a Markov state model built with a very long time series (1.5 µs) and a fine discretization of the dihedral angles. We measure error using the projection distance [38], which evaluates the overlap between one subspace and the orthogonal complement of

another subspace. For subspaces $\mathcal{U}$ and $\mathcal{V}$ with orthonormal basis functions $\{u_i\}$ and $\{v_i\}$, the projection distance is given by

$$d(\mathcal{U}, \mathcal{V}) = \sqrt{\sum_{i,j} (\delta_{ij} - \langle u_i, v_j \rangle^2)}. \tag{2.39}$$

This measure, which combines the error in the different eigenfunctions into a single number, is useful because VAC is typically used to identify subspaces of eigenfunctions rather than individual eigenfunctions. The maximum possible error when estimating k eigenfunctions is $\sqrt{k}$.

Our main result from the alanine dipeptide application is that IVAC is more robust to the selection of lag time parameters than VAC. In Figure 2.2, we report the accuracy of IVAC and VAC for different lag times and trajectory lengths. In the left column, we show the root mean square errors (RMSE) for IVAC (orange) and VAC (purple), aggregated over thirty independent trajectories. From the aggregated results, IVAC performs nearly as well as VAC with the best possible τ and consistently gives results much better than VAC with a poorly chosen τ. The RMSE of IVAC is just 0.58 with 10 ns trajectories and 0.45 with 20 ns trajectories. These low error levels are not far from the minimum error of 0.37 that is possible using our linear approximation space.

In the right column of Figure 2.2, we show results for a 10 ns trajectory and a 20 ns trajectory. The trajectories were selected to help illustrate differences in the error profiles for VAC and IVAC; similar plots for all other trajectories can be found in the supporting information. We observe two key differences. First, VAC error can exhibit high-frequency stochastic variability as a function of lag time, a source of variability that does not affect integrated VAC results. Second, VAC can have high error levels at very short and long lag times. The projection distance against our reference often reaches 1.0, which might indicate that a true eigenfunction is completely orthogonal to our estimated subspace. The error of

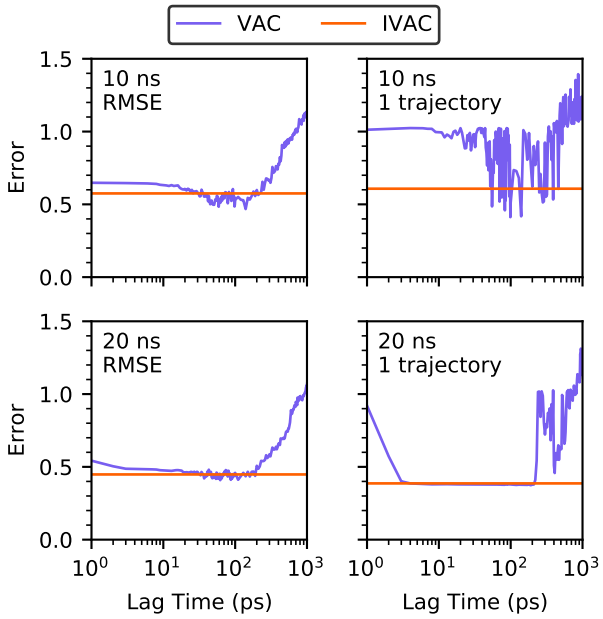

Figure 2.2: Linear IVAC and VAC errors for alanine dipeptide trajectories. IVAC was applied with $\tau_{\text{min}} = 1\,\text{ps}$ and $\tau_{\text{max}} = 1\,\text{ns}$. VAC was applied with variable lag time τ (horizontal axis). Errors are computed using the projection distance to the MSM reference for the span of η_2 and η_3. (left) Root mean square errors (RMSE) over 30 independent trajectories. (right) Errors for a single trajectory.

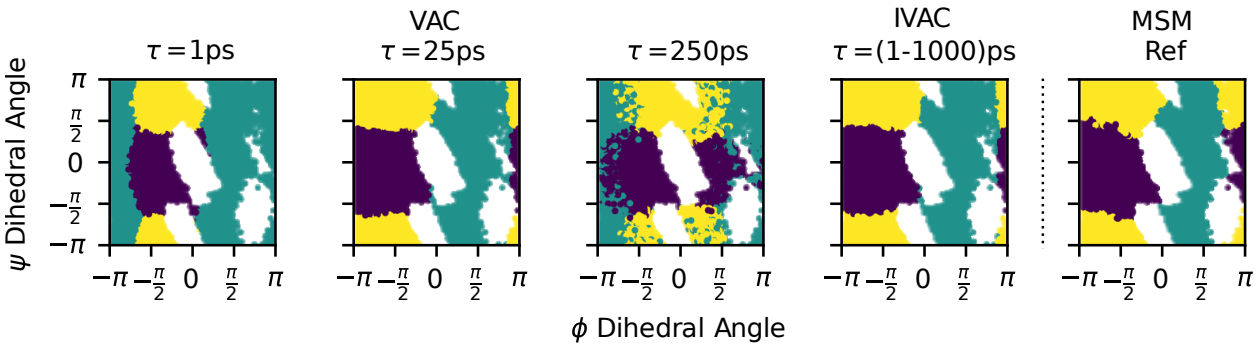

Figure 2.3: Clusters on the eigenfunctions estimated using VAC and IVAC compared with clusters on an accurate MSM. (left of the dashed line) VAC and IVAC results for the 20 ns trajectory from Figure 2.2. (right of the dashed line) Clustering on η_2 and η_3 evaluated using an accurate MSM reference.

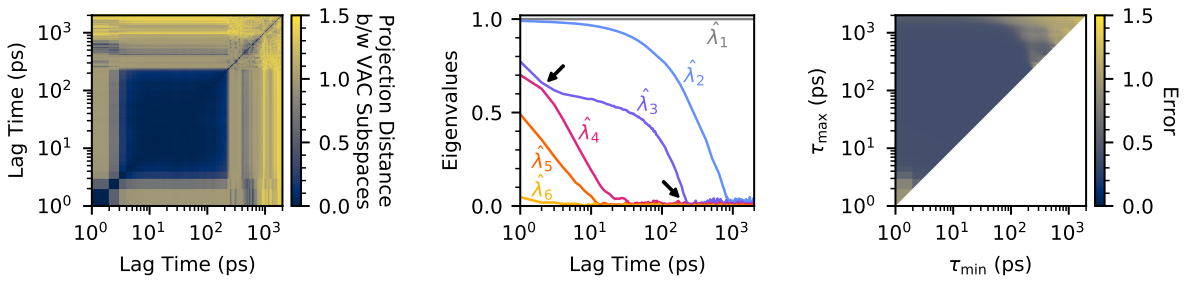

Figure 2.4: Lag time dependence of VAC and IVAC results. All results shown are for the single 20 ns alanine dipeptide trajectory in Figure 2.2. (left) Projection distance between VAC results at the horizontal axis lag time and VAC results at the vertical axis lag time. (center) First six estimated eigenvalues of the transition operator. (right) Error in IVAC results at different values of τ_{min} and τ_{max}, evaluated using the projection distance to the MSM reference for the span of η_2 and η_3.

IVAC is unlikely to reach such high extremes.

We note that the parameter values $\tau_{\text{min}} = 1\,\text{ps}$ and $\tau_{\text{max}} = 1\,\text{ns}$ used in IVAC are not hard to tune. The range $1\,\text{ps} - 1\,\text{ns}$ is a broad window of lag times over which VAC eigenvalues $\hat{\lambda}_2^\tau$ and $\hat{\lambda}_3^\tau$ decrease from values near one to values near zero. In contrast, it is much harder to tune the VAC lag time τ. VAC results are very sensitive to high or low lag times as seen in Figure 2.2.

When eigenfunction estimates are accurate, we expect that the eigenfunction coordinates will help identify the system's metastable states. In Figure 2.3, we compare the results of clustering configurations in the 20 ns alanine dipeptide trajectory in Figure 2 using the associated IVAC and VAC estimates. We plot the predicted metastable states against the dipeptide's ϕ and ψ dihedral angles. In the figure, we present VAC results taken at a short lag time, an intermediate lag time, and a long lag time. We also present results for the MSM reference. Comparing against the reference, we find that IVAC identifies clusters as accurately as VAC at a well-chosen lag time, and IVAC performs far better than VAC at a poorly-chosen lag time.

Next, we present additional analyses applied to a single 20 ns alanine dipeptide trajectory, that provide insight into why IVAC is more robust to lag time selection than VAC. To start,

we examine the discrepancy in VAC results at different lag times. In Figure 2.4, left, we performed VAC with a range of different lag times, and we measured the projection distance between the VAC results obtained at one lag time τ_1 (horizontal axis) and the VAC results obtained at a different lag time τ_2 (vertical axis). The square with low projection distance between 3 ps and 200 ps indicates that VAC results with lag times chosen within this range are similar to one another, but not to those with lag times taken from outside this range.

The discrepancy between VAC results at both low and high lag times can be explained by a plot of VAC eigenvalues (Figure 2.4, center). At 3 ps, there is an eigenvalue crossing between the eigenvalues $\hat{\lambda}_3^\tau$ and $\hat{\lambda}_4^\tau$ (shown in purple and magenta). The eigenvalue crossing causes VAC to misidentify the third VAC eigenfunction (which is inside the 3D subspace) and the fourth VAC eigenfunction (which is outside the 3D subspace). At 200 ps, there is a different problem related to insufficient sampling. The third eigenvalue descends into noise, causing VAC to fit the first two eigenfunctions at the expense of the 3D subspace.

With integrated VAC, the problem of finding a single good lag time is replaced with the problem of finding two endpoints for a range of lag times. This proves to be an easier task as IVAC is more tolerant of lag times outside the region where VAC gives good results. In Figure 2.4, right, we show the error of IVAC as a function of $\tau_{\min}$ and $\tau_{\max}$ (horizontal and vertical axes, respectively). This figure, which shows the error of IVAC estimates computed from comparison with the reference, is different from the figure on the left which shows only the discrepancy between VAC results at different lag times. Figure 2.4, right, also shows the error of VAC, which appears along the diagonal of the plot corresponding to the case $\tau_{\min} = \tau_{\max}$.

Figure 2.4, right, reveals that the range of lag time parameters for which IVAC exhibits low error levels is much broader than the range of lag times for which VAC exhibits low error levels. This supports our basic argument that choosing good parameters in IVAC is easier than choosing good parameters in VAC. To achieve low errors, we do not need to identify the

optimal VAC lag times but only integrate over a window that contains the optimal VAC lag times while ensuring that $\tau_{\max}$ is not excessively high.

2.3.2 Application to the villin headpiece

Next we apply IVAC to a difficult spectral estimation problem with limited data. We seek to estimate the slow dynamics for an engineered 35-residue subdomain of the villin headpiece protein. Our data consist of a 125 µs molecular dynamics simulation performed by Lindorff-Larsen et al. [39]. Villin is a common model system for protein folding for both experimental and computational studies [39–42], where the top eigenfunctions correlate with the folding and unfolding of the protein.

On the surface, the villin data set would seem to be much larger and more useful for spectral estimation compared to the 10 ns to 20 ns trajectories we examined for the alanine dipeptide. However, the villin headpiece relaxes to equilibrium orders of magnitude more slowly than the alanine dipeptide. The data set contains just 34 folding/unfolding events with a folding time of 2.8 µs. The limited number of observed events is characteristic of simulations of larger and more complex biomolecules, since simulations require massive computational resources and conformational changes take place slowly over many molecular dynamics time steps. The fact that the dynamics of villin are not understood nearly as well as the dynamics of the alanine dipeptide presents an additional challenge. Compared to the alanine dipeptide, villin has a more complex free energy surface and a larger number of degrees of freedom. Since the true eigenfunctions of the system are unknown, it is appropriate to apply spectral estimation using a large and diverse feature set. However, the large size and diversity of the feature set increases the risk of estimation error.

In contrast to the alanine dipeptide results, where we applied IVAC using linear combinations of basis functions, here we apply IVAC using a neural network. The increased flexibility of the neural network approximation reduces approximation error. However, the

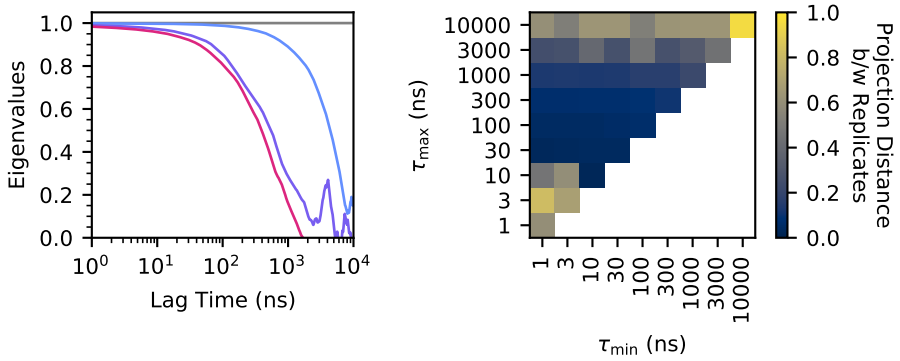

Figure 2.5: Nonlinear IVAC results for the 125 µs villin headpiece trajectory. (left) Estimated eigenvalues of the transition operator. (right) Root mean square projection distance between 10 replicates of nonlinear IVAC at the specified values of τ_{min} and τ_{max}.

procedure for optimizing the neural network is more complicated than the procedure for applying linear VAC. Moreover, the complexity of the neural network representation (around 5×10^4 parameters) makes overfitting a concern for this example.

We use a slight modification of the neural network architecture published in Sidky et al. [43], with 2 hidden layers of 50 neurons, tanh nonlinearities, and batch normalization between layers. The network is built on top of a rich set of features, consisting of all the C_α pairwise distances as well as sines and cosines of all dihedral angles. At each optimization step, we subsample 10^4 data points using Algorithm 1. We optimize the neural network parameters using AdamW [44] with a learning rate of 10^{-4} and a weight decay coefficient of 10^{-2}. Following standard practice, we use the first half of the data set for training and the second half for validation. We validate the neural network against the testing data set every 100 optimization steps, and perform early stopping with a patience of 10.

We present our results for villin in two parts. First we describe our procedure for selecting parameters in nonlinear IVAC. Next we highlight evidence that nonlinear IVAC shows greater robustness to overfitting compared to nonlinear VAC.

Selection of parameters

Here, we describe the protocols we use for selecting IVAC parameters. By establishing clear protocols, we help ensure that IVAC performs to the best of its ability, providing robust eigenfunction estimates even in a high-dimensional setting with limited data.

Our first protocol is to evaluate the condition number for the subspace of eigenfunctions that we are estimating. This protocol is motivated by the theoretical error analysis in Webber et al. [1], where we showed that spectral estimates are less sensitive to estimation error for a well-conditioned subspace. To ensure that we are estimating a well-conditioned subspace, we first use IVAC to estimate eigenvalues for the transition operator. We then identify a subspace of eigenfunctions $\eta_1, \eta_2, \ldots, \eta_k$ that is separated from all other eigenfunctions by a large spectral gap $\hat{\lambda}_k^\tau - \hat{\lambda}_{k+1}^\tau$.

For the villin data, we choose the subspace consisting only of the constant eigenfunction $\eta_1 = 1$ and the first nontrivial eigenfunction η_2. This is a well-conditioned subspace with a minimum condition number

$$\min_\tau \kappa^\tau = \min_\tau (\hat{\lambda}_2^\tau - \hat{\lambda}_3^\tau)^{-1} = 1.6. \tag{2.40}$$

Our second protocol for ensuring robustness is to check that eigenfunction estimates remain consistent when the random seeds used in initialization and subsampling are changed. We train ten nonlinear IVAC neural networks and quantify the inconsistency in the results using the root mean square projection distance between eigenspace estimates from different runs. The results of this calculation are plotted in Figure 2.5 across a range of $\tau_{\min}$ and $\tau_{\max}$ values. The results for VAC appear along the diagonal of the plot in Figure 2.5, corresponding to the case $\tau_{\min} = \tau_{\max}$.

Figure 2.5 reveals problems with consistency for both IVAC and VAC. IVAC is robust to the choice of $\tau_{\min}$. However, setting $\tau_{\max} < 30\,\text{ns}$ or $\tau_{\max} > 300\,\text{ns}$ leads to poor consistency.

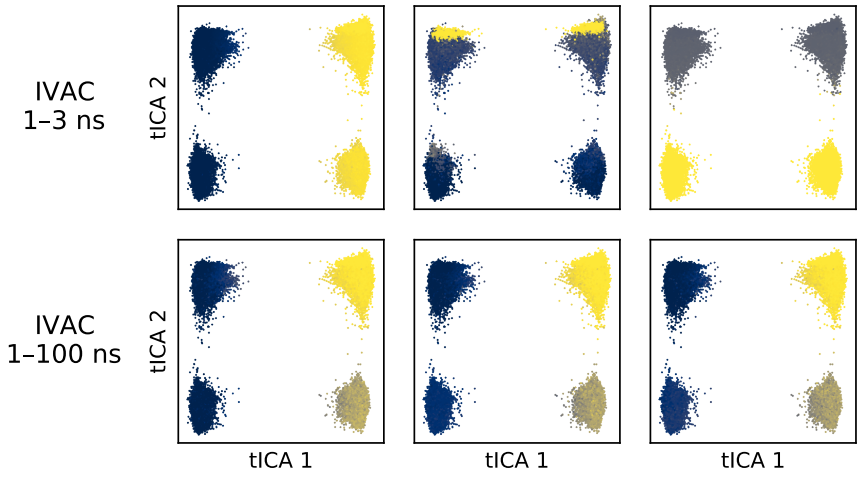

Figure 2.6: Nonlinear IVAC results plotted on the first two time-lagged independent component analysis (tICA) coordinates. (top) IVAC with a 1 ns to 3 ns integration window and three different random seeds. (bottom) IVAC with a 1 ns to 100 ns integration window and three different random seeds.

If we train the neural network with these problematic $\tau_{\max}$ values, then solutions can look very different depending on the random seeds that are used for optimizing. With VAC, setting $\tau < 10\,\text{ns}$ or $\tau > 300\,\text{ns}$ would lead to inconsistent results.

IVAC provides more flexibility to address the consistency issues compared to VAC, since we can integrate over a range of lag times. For the villin data, we choose to set $\tau_{\min} = 1\,\text{ns}$ and $\tau_{\max} = 100\,\text{ns}$. For these parameter values, the consistency score is very good. The typical projection distance between subspaces with different random seeds is just 0.05. Moreover, 1 ns to 100 ns is a wide range of lag times, helping to ensure that optimal or near-optimal VAC lag times are included in the integration window.

To help explain why the consistency is so poor for small $\tau_{\max}$ values, we present in Figure 2.6 a set of IVAC solutions obtained with an integration window of 1 ns to 3 ns and three different random seeds. We see that all three solutions identify clusters in the data, but the clusters are completely different in the three cases. We conjecture that IVAC is randomly fitting three different eigenspaces. This is supported by the eigenvalue plot in Figure 2.5, which shows that three nontrivial eigenvalues of the transition operator lie close together over

the 1 ns to 3 ns time window, making it possible that eigenspaces are randomly misidentified by IVAC.

In contrast to the inconsistent results obtained with an integration window of 1 ns to 3 ns, we obtain more reasonable results with an integration window of 1 ns to 100 ns. As shown in Figure 2.6, the IVAC solutions are nearly identical regardless of the random seed.

In summary, we propose a robust procedure for approximating eigenfunctions of the villin headpiece system. We choose to approximate a well-conditioned eigenspace that is separated from other eigenspaces by a wide spectral gap. Moreover, we ensure that IVAC results are consistent regardless of the random initialization and randomly drawn data subsets used to train the neural net. Because of these protocols, the neural network estimates shown in Figure 2.6 reliably identify clusters in the trajectory data indicative of folded/unfolded states.

Robustness to overfitting

In this section, we present results suggesting that nonlinear IVAC is more robust to overfitting than nonlinear VAC. This is crucial if the data set is too small for cross-validation.

To identify the overfitting issue with small data sets, we eliminate the early stopping and we train IVAC and VAC until the training loss stabilizes. We calculate implied timescales by performing linear VAC on the outputs of the networks trained using IVAC and VAC, which we present in Figure 2.7.

We first compare the estimated implied timescales between the training and validation data sets. For both algorithms, the implied timescales calculated on the training data are larger than those calculated on the validation data. This is clear evidence of overfitting. However, we see that IVAC gives larger implied timescales on the validation data compared to VAC. In combination with the variational principle associated with the implied timescales, this suggests that IVAC is giving an improved estimate for the slow eigenfunctions.

Examining the implied timescales estimated on training data show further signs of

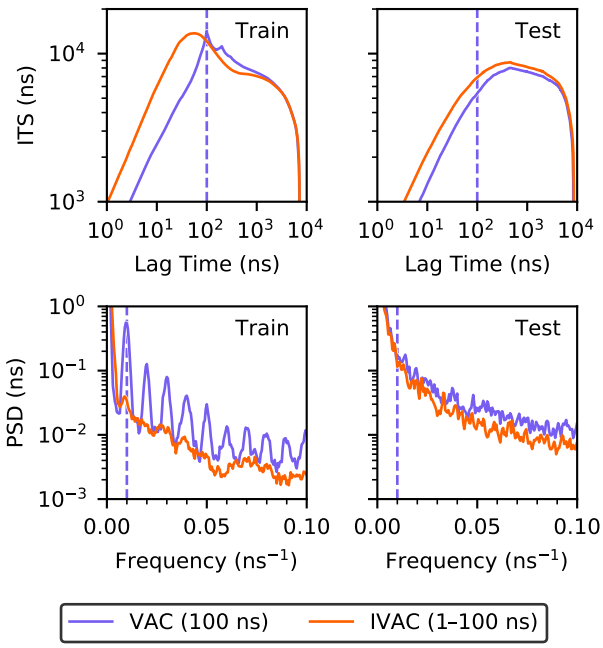

Figure 2.7: Implied time scales (ITS) and power spectral densities (PSD) obtained with nonlinear IVAC and VAC with neural network basis functions applied to the villin headpiece data set. The VAC training lag time is marked by the dotted line in each panel.

overfitting. The VAC implied timescale estimates for the training data exhibit sharp peaks at the training lag time that are absent in the implied timescale estimates of the validation data. This suggests a hypothesis for the mechanism of overfitting: with a sufficiently flexible approximation space, VAC is able to find spurious correlations between features that happen to be separated by τ. This explains the smaller peaks at integral multiples of the lag time, as features artificially correlated at τ will be correlated at 2τ as well.

To confirm our hypothesis, we plot the power spectral density (PSD) [45] of the time trace of eigenfunction estimates in Figure 2.7. The PSD confirms the existence of a periodic component in VAC results with a frequency at the inverse training lag time. In contrast, IVAC does not exhibit such a periodic component. In Figure 2.7, we see that the 1 ns to 100 ns integration window leads to implied timescale estimates that depend smoothly on the data both for the training and the test data set. The PSD shows no periodic components in the spectra for IVAC, providing further evidence that IVAC is comparatively robust while

VAC results can be very sensitive to the particular lag time that is used.

2.4 Conclusion

In this chapter, we have presented integrated VAC (IVAC), a new extension to the popular variational approach to conformational dynamics (VAC). By integrating correlation functions over a window of lag times, IVAC provides robust estimates of the eigenfunctions of a system's transition operator.

To test the efficacy of the new approach, we compared IVAC and VAC results on two molecular systems. First, we applied the spectral estimation methods to simulation data from the alanine dipeptide. This is a relatively simple system that permits generation of extensive reference data for validating our calculations. As we varied the lag time parameters and the amount of data available, we observed the improved robustness of IVAC compared to VAC. IVAC gives low-error eigenfunction estimates even when the lag times range over multiple orders of magnitude. In contrast, VAC requires more precise lag-time tuning to give reasonable results

Next we applied IVAC to analyze a folding/unfolding trajectory for the villin headpiece. These data contain relatively few folding/unfolding events despite pushing the limits of present computing technology. For this application, we used a flexible neural network representation built on top of a rich feature set. We presented a procedure for selecting parameters in IVAC that helps lead to robust performance in the face of uncertainty. For the application to villin data, we found that VAC exhibited pronounced artifacts from overfitting when precautions were not taken to specifically prevent it, while IVAC did not.

Our work highlights the sensitivity of VAC calculations to error from insufficient sampling. Examining our results on the villin headpiece, we see that regularization (here, by early stopping) and validation are crucial when running VAC with neural networks or other flexible approximation spaces. With insufficient regularization or poor validation these schemes

easily overfit. Even for the alanine dipeptide example, where we employ a simple basis on a statistically well-conditioned problem, we see that VAC has a high probability of giving spurious results with insufficient data.

Integrated VAC addresses this problem by considering information across multiple time lags. Future extensions of the work could further leverage this information. For instance, employing a well-chosen weighting function within the integral in (2.5) could further decrease hyperparameter sensitivity. Additionally, future numerical experiments could point to improved procedures for selecting τ_{min} and τ_{max} values. Finally, we could integrate over multiple lag times in other formalisms using the transition operator, such as schemes that estimate committors and mean-first-passage times [4]. These extensions would further strengthen the basic message of our work: combining information from multiple lag times leads to improved estimates of the transition operator and its properties.

2.5 Supplementary information

2.5.1 Simulation details for the alanine dipeptide experiments

All simulations were conducted using Gromacs 5.1.4 [46–51]. The molecule was represented by the CHARMM 27 force field [52] in a solvent modeled by 513 water molecules using a rigid TIP3P model [53]. Long-range electrostatics were performed using particle-mesh Ewald summation at fourth order with a Fourier spacing of 0.12 nm [54]. Each simulation used Langevin dynamics, integrated using the GROMACS leap-frog Langevin integrator with a 1 fs time step and a time constant of 0.1 ps at a temperature of 310 K. Hydrogen bonds were constrained to be rigid using LINCS [55], and water rigidity was enforced using SETTLE [56]. In each simulation, the system was initialized at a density of $1\,\text{kg}\,\text{L}^{-1}$. The system was then equilibrated for 50 ps at constant volume, followed by another 50 ps at constant pressure using the Parrinello-Rahman barostat [57]. Finally, the system was again equilibrated at

constant volume for 50 ns. The data set used was obtained from a production run of 50 ns, with structures saved every 500 fs. To construct our references for the true eigenfunctions, we ran 10 simulations each of length 150 ns, and constructed an MSM on all dihedral angles. This MSM had 500 Markov states; these were identified by k-means clustering, and we estimated the eigenfunctions and eigenvalues using pyEMMA [58].

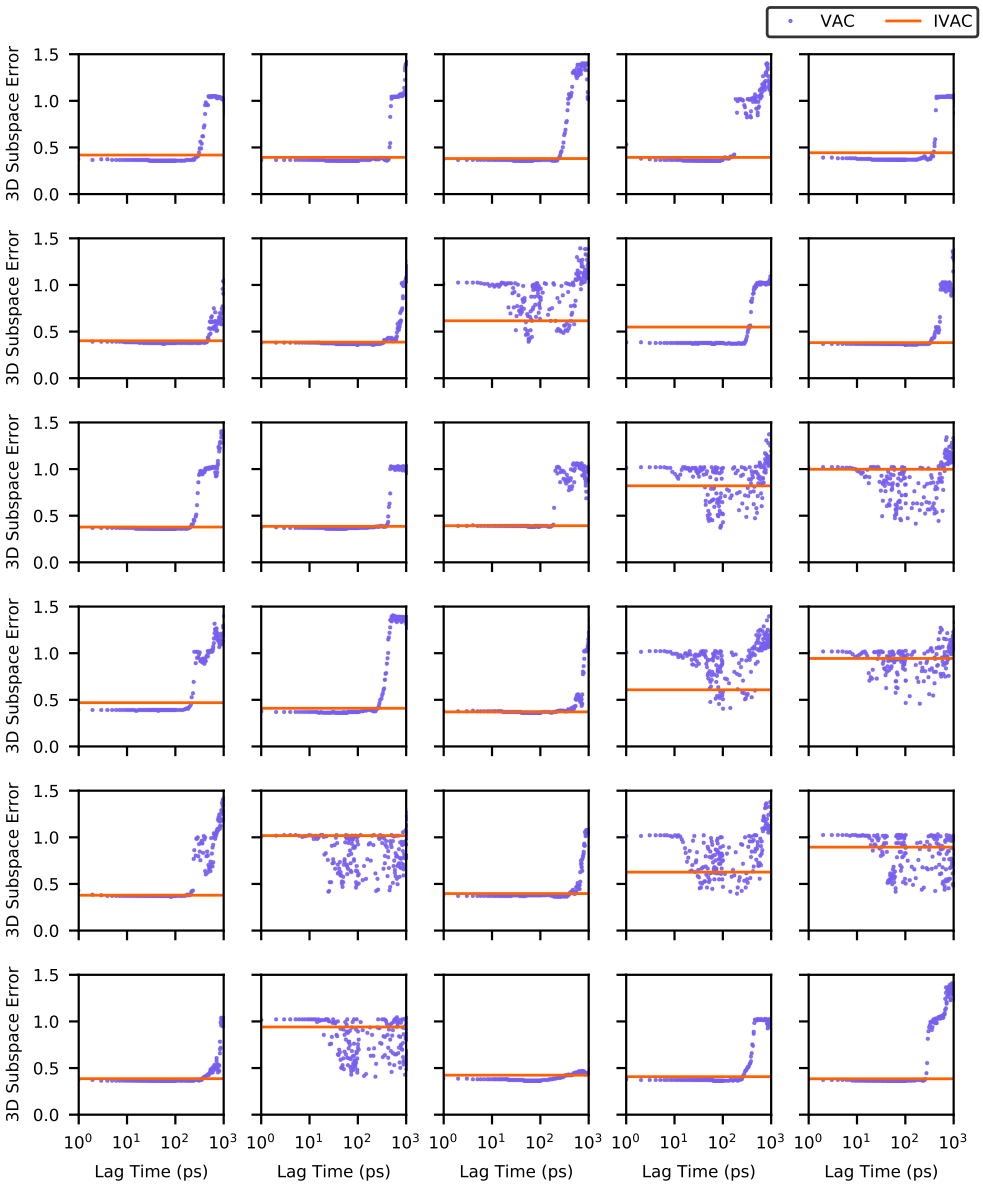

Figure 2.8: VAC (at the horizontal axis lag time) and IVAC (with $\tau_{\min} = 1\,\text{ps}$ and $\tau_{\max} = 1\,\text{ns}$) errors for all 30 of the 10 ns long alanine dipeptide trajectories.

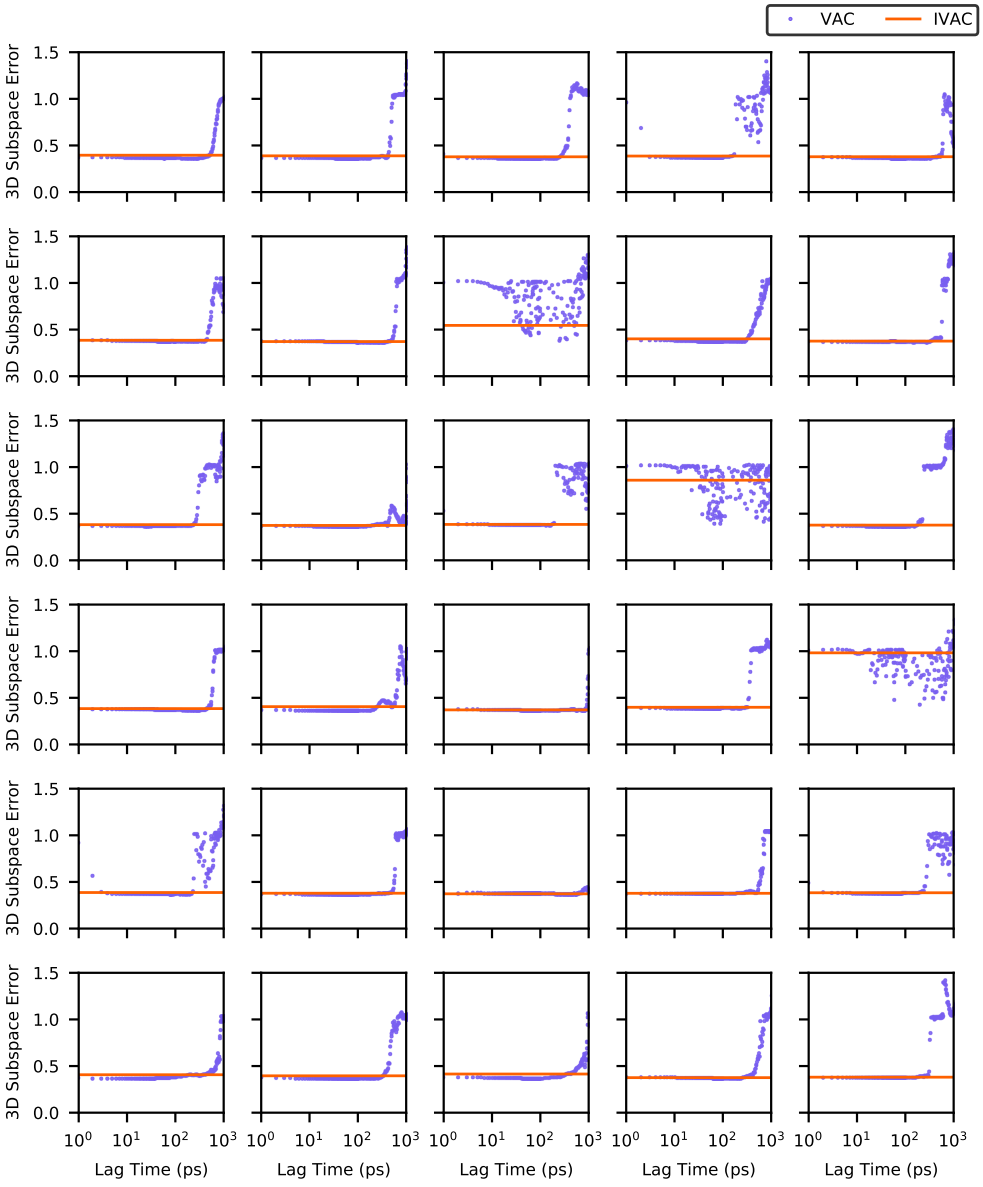

Figure 2.9: VAC (at the horizontal axis lag time) and IVAC (with $\tau_{\min} = 1\,\text{ps}$ and $\tau_{\max} = 1\,\text{ns}$) errors for all 30 of the 20 ns long alanine dipeptide trajectories.

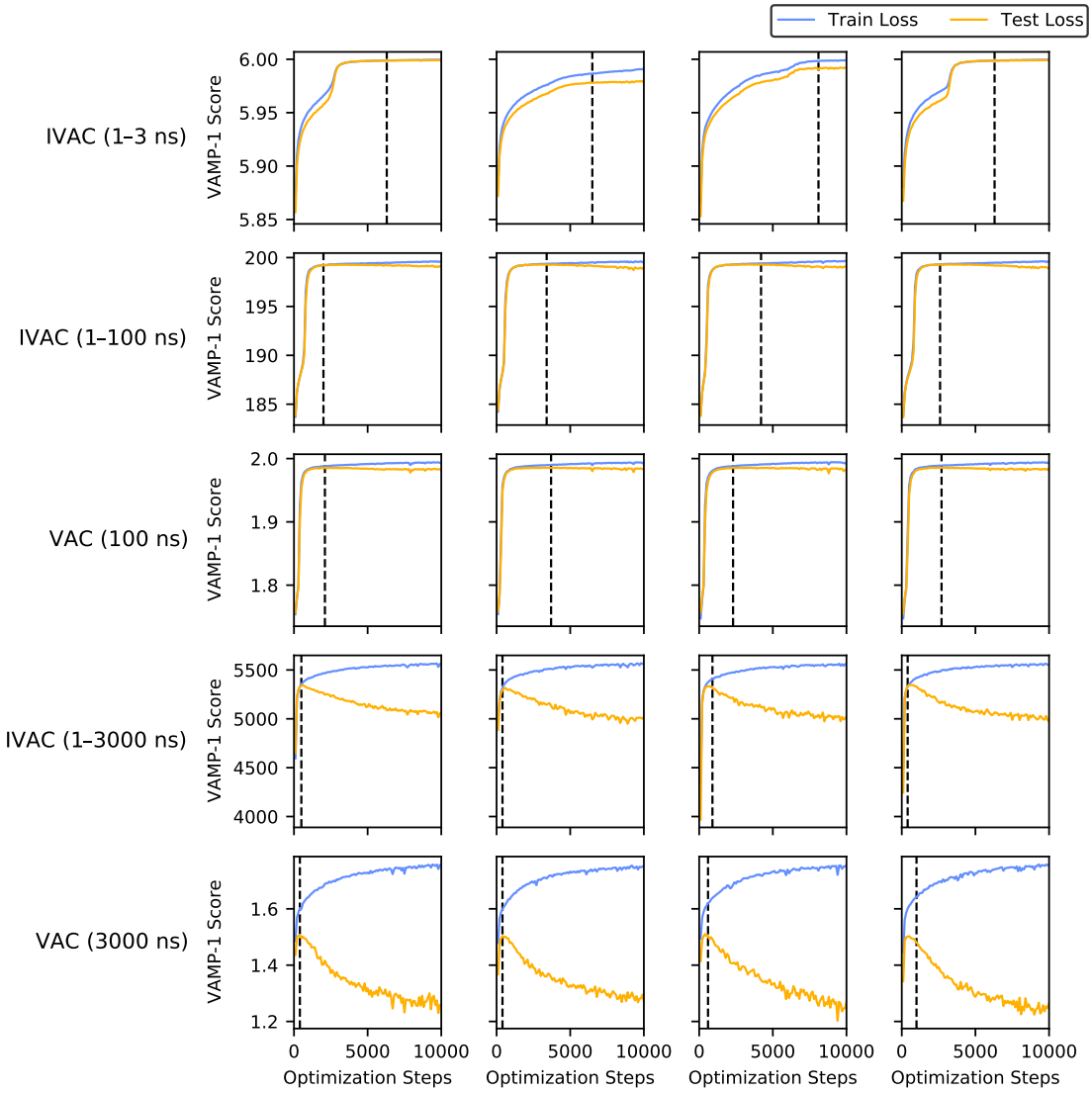

Figure 2.10: VAMP-1 scores as a function of optimization steps during the training of the neural networks on the villin headpiece data set. Early-stopping times are marked by the black dotted lines.

CHAPTER 3

DYNAMICAL GALERKIN APPROXIMATION AND TRANSITION PATH THEORY

This chapter is adapted with permission from J. Strahan et al., "Long-time-scale predictions from short-trajectory data: a benchmark analysis of the Trp-cage miniprotein", J. Chem. Theory Comput. **17**, 2948–2963 (2021).

Elucidating physical mechanisms with statistical confidence from molecular dynamics simulations can be challenging owing to the many degrees of freedom that contribute to collective motions. To address this issue, we recently introduced a dynamical Galerkin approximation (DGA) [E. H. Thiede et al., "Galerkin approximation of dynamical quantities using trajectory data", J. Chem. Phys. **150**, 244111 (2019)], in which chemical kinetic statistics that satisfy equations of dynamical operators are represented by a basis expansion. Here, we reformulate this approach, clarifying (and reducing) the dependence on the choice of lag time. We present a new projection of the reactive current onto collective variables and provide improved estimators for rates and committors. We also present simple procedures for constructing suitable smoothly varying basis functions from arbitrary molecular features. To evaluate estimators and basis sets numerically, we generate and carefully validate a data set of short trajectories for the unfolding and folding of the trp-cage miniprotein, a well-studied system. Our analysis demonstrates a comprehensive strategy for characterizing reaction pathways quantitatively. My primary contributions to this work are to the formula for the reactive current (3.35), and transition rate and reactive current estimators that converge at finite time lags (Sec. 3.7.1).

3.1 Introduction

Molecular dynamics simulations enable atomic-resolution investigation of complex processes. These investigations are often carried out by direct simulation: the equations of motion are numerically integrated forward in time to generate trajectories (times series of atomic positions and, as needed, momenta) for as long as possible given available computational resources. Since most events of interest occur on timescales longer than those accessible by direct simulation, many enhanced sampling schemes have been developed to allow more extensive interrogation of an event of interest without sacrificing model fidelity. Splitting methods, for example, branch and prune a collection of simultaneously evolving trajectories to promote progress in a small number of order parameters (or collective variables, CVs) [59–64]. Regardless of whether trajectory data are generated by direct simulation or enhanced sampling, an essential question remains: How can these data be analyzed to yield new understanding about the process under study?

We recently introduced dynamical Galerkin approximation (DGA) [4] to analyze trajectory data generated by direct simulation, as well as many enhanced sampling schemes. In this approach, conditional expectations such as committor functions are cast as solutions to equations involving the operator determining the statistics of the underlying process, its transition operator. The solution to the equation is then approximated as a linear combination of basis functions. This approach builds on an extensive literature from the last decade that shows that the eigenvalues and eigenvectors of the transition operator can be approximated from trajectory data, subject to a Markov assumption [11, 15, 19, 23, 25, 32, 65–72]. These spectral estimation methods aim to characterize the slowest dynamical features of the system (e.g., transitions between metastable states) as eigenvectors corresponding to the largest eigenvalues of the transition operator. When the goal is to study a particular event of interest, the indirect relationship between the eigenvectors of the transition operator and the specific event of interest is a weakness of the spectral estimation approach. Indeed, for many complex

systems the true slowest dynamical features of the system are too slow to be of any physical interest.

In contrast, the aim of DGA is *not* to extract spectral information. Instead DGA aims to compute statistics that directly characterize a particular event under study. For example, when transitions between particular metastable states are of interest, the statistics that DGA yields can be combined within the framework of transition path theory (TPT) [3, 73, 74] to obtain reactive fluxes and in turn reaction mechanisms. Because DGA analyzes short trajectory fragments, it can be used to process the data generated by many splitting schemes. Alternatively, the trajectory data can be generated by seeding initial conditions for short direct simulations throughout state space.

Though most often employed as a spectral estimation tool, Markov State Models (MSMs) [8, 75, 76] and related methods [77] have also been used to approximate TPT quantities. DGA can be viewed as an extension of these MSM variants to a more general class of target quantities and to more general representations (basis set expansions) of those quantities. However, even with parameters chosen as in MSMs, the DGA estimators introduced in this article improve upon their MSM counterparts in several ways including reduced dependence on the crucial lag time parameter and lower variance estimates of certain TPT quantities.

In our previous study [4], we compared diffusion map [78] and indicator basis sets for predicting mean first-passage times and committors for the Müller-Brown model [79] and the folding of the protein Fip35 using six long equilibrium trajectories from D. E. Shaw Research [80]. Because there were only a few folding events within those trajectories, it was difficult to assess the performance of the method. One goal of the present study is to generate a protein folding data set that enables robust application of the approach and to compare different basis sets and estimators systematically.

To this end, we study the trp-cage miniprotein, a 20-residue fast-folding artificial sequence (asn-leu-tyr-ile-glu-trp-leu-lys-asp-gly-gly-pro-ser-ser-gly-arg-pro-pro-pro-ser) that has been

studied extensively both experimentally and computationally [39, 43, 81–85]. In solution at 298 K, the protein folds on a 4 μs timescale and unfolds on a 12 μs timescale [81], which makes these processes difficult but not impossible to simulate directly. In particular, D. E. Shaw Research produced a 208 μs equilibrium simulation of the K8A mutant of trp-cage using the Anton supercomputer with the CHARMM 22* force field [39]. Although, like the Fip35 data, this trajectory contains relatively few folding events, it has been the subject of previous MSM [85] and variational approach for Markov processes (VAMP) studies [43]. These earlier studies serve as valuable points of comparison and enable us to identify CVs that provide good control over sampling. Though DGA does not depend directly on any choice of CV, its performance is strongly affected by the quality of the data set of sampled trajectories. We use our chosen set of CVs together with enhanced sampling methods to generate a new data set comprised of many short trajectories that are distributed evenly throughout the CV space.

In this article we reformulate DGA in terms of the transition operator of the underlying Markov process. This has two primary advantages relative to our previous formulation in terms of the generator of the process [4]. First, it clarifies the role of lag time in DGA estimates, showing that correctly constructed estimators should have no dependence on lag time in the infinite-basis, infinite-sampling limit. Second, the formulation in terms of the transition operator leads directly to estimators that correctly account for boundary conditions by stopping underlying trajectories appropriately. Using our improved DGA estimators we introduce new estimators for TPT reaction rates and reactive currents. To make computation of the reactive current tractable and the result readily interpretable, we introduce a projection formula for the reactive current onto a CV space which allows us to assign relative weights to transition paths in arbitrary CV spaces. We also introduce a new procedure for constructing a basis set from arbitrary molecular features (here, primarily pairwise distances between C_α atoms, though we also explore CVs with delay embedding) and compare it with two basis

sets that are used widely in the MSM literature: indicator functions on molecular features and indicator functions on time-lagged independent component analysis (TICA) coordinates [12–14]. We show that our DGA estimators with selected basis sets can robustly yield remarkably good agreement with published results for committors and pathways, even though the total simulation time of our trp-cage data set is only 30 μs, with a maximum trajectory length of 30 ns. The projection of the reactive currents on CVs facilitates both visualization and quantification of information about pathways, enabling immediate identification of the defining properties of transition states. This makes our approach an efficient one for exploring mechanisms.

3.2 Long time phenomena from short trajectory data

In this section, we introduce key dynamical statistics and explain how they can be defined in terms of an evolution operator (Section 3.2.1). An emphasis on forms that lead directly to practical and accurate numerical estimators causes several departures from the standard presentation of this material. We present our approach for solving the operator equations numerically by Galerkin (basis expansion) approximation [4] (Section 3.2.2), and distinguish forward-in-time statistics (Section 3.2.2) from backward-in-time statistics involving the adjoint of the evolution operator (Section 3.2.2); this is followed by a discussion of basis sets (Section 3.2.2) and an approach for constructing an approximately Markovian process when the molecular representation does not adequately capture the dynamics (delay embedding, Section 3.2.2). Finally, guided by TPT, we combine the dynamical statistics estimated by DGA to yield approximations of reaction rates and currents. (Section 3.2.3).

3.2.1 The transition operator and Feynman-Kac representation

The dynamics of a Markov process X_t can be encoded in its associated transition operator, $\mathcal{T}^t$, which specifies the evolution of the expectation of a function f over some interval of time

$t \geq 0$:

$$\mathcal{T}^t f(x) = \mathbb{E}\left[f(X_t) \,|\, X_0 = x\right]. \tag{3.1}$$

The time index t can be continuous or discrete. The transition operator (also known as the Koopman operator), and in particular its eigenvectors and eigenvalues, are the key quantities in well-established methods for discovering slowly decorrelating features of a Markov process [1]. The transition operator is also central to the DGA approach [4]. However, in DGA, instead of estimating the spectrum of the transition operator, the goal is to solve linear equations representing certain conditional expectations.

In Thiede et al. [4], we presented DGA in terms of the generator which, for a continuous time process is defined by the limit:

$$\mathcal{L}f(x) = \lim_{t \to 0} \frac{\mathcal{T}^t f(x) - f(x)}{t}. \tag{3.2}$$

For a discrete time process the limit is removed and t in (3.2) is replaced by the unit of a single time step. A presentation in terms of the generator has the advantage that it results in very concise equations for quantities of interest. For example, consider the (forward) committor, $q_+(x)$, which is the probability of entering a product state B before a reactant state A starting from $x \notin A \cup B$:

$$q_+(x) = \mathbb{P}[X_{S^+_{A \cup B}(0)} \in B | X_0 = x], \tag{3.3}$$

where $S^+_{A \cup B}(0) = \min\{t \geq 0 : X_t \in A \cup B\}$ is the time of first entrance into $A \cup B$. For $x \in A$, $q_+(x) = 0$, and, for $x \in B$, $q_+(x) = 1$.

The committor satisfies the Feynman-Kac relation

$$\mathcal{L}q_+(x) = 0 \text{ for } x \notin A \cup B, \qquad q_+(x) = \mathbb{1}_B(x) = \begin{cases} 1, & x \in B \\ 0, & x \notin B \end{cases} \text{ for } x \in A \cup B \tag{3.4}$$

(see Eqs. (18) and (19) of Thiede et al. [4]).

In this article we choose to work directly with the transition operator instead of the generator because it facilitates the implementation of numerical formulas. It also greatly simplifies our description of TPT and clarifies the relationship between DGA and the well-established VAC approach to approximating spectral properties of the transition operator (see Webber et al. [1]). In the case of the committor, we integrate (3.4) until a chosen time τ to obtain the equivalent form of the Feynman-Kac relation,

$$\mathcal{T}^{\tau}_{A\cup B}\, q_+(x) - q_+(x) = 0 \text{ for } x \notin A \cup B, \qquad q_+(x) = \mathbb{1}_B(x) \text{ for } x \in A \cup B. \tag{3.5}$$

In this expression we have introduced the notation $\mathcal{T}^t_{A\cup B}$ for the transition operator of the stopped process $X_{t\wedge S^+_{A\cup B}(0)}$, i.e.,

$$\mathcal{T}^t_{A\cup B}\, f(x) = \mathbb{E}\left[f(X_{t\wedge S^+_{A\cup B}(0)}) \,|\, X_0 = x \right]. \tag{3.6}$$

Here and below $t \wedge S^+_{A\cup B}(0) = \min\{t, S^+_{A\cup B}(0)\}$, indicating that the evolution process does not proceed beyond escape.

For a more general domain D and $S^+_{D^{\mathsf{c}}}(0) = \min\{t \geq 0 : X_t \notin D\}$, the conditional expectation

$$u(x) = \mathbb{E}\left[b(X_{S^+_{D^{\mathsf{c}}}(0)}) - \int_0^{S^+_{D^{\mathsf{c}}}(0)} a(X_t)dt \,\middle|\, X_0 = x \right] \text{ for } x \in D \tag{3.7}$$

solves the equation

$$\mathcal{T}_{D^{\mathsf{c}}}^{\tau}\, u(x) - u(x) = \int_0^{\tau} \mathcal{T}_{D^{\mathsf{c}}}^{t}\, a(x)dt \quad \text{for} \quad x \in D, \qquad u(x) = b(x) \text{ for } x \notin D. \tag{3.8}$$

To obtain (3.5) for the committor, choose $D = (A \cup B)^{\mathsf{c}}$, $b = \mathbb{1}_B$, and $a = 0$. In (3.7) and (3.8) we assume for simplicity that $a(x) = 0$ for $x \notin D$. For a discrete-time process the time integral in these expressions should be interpreted as a sum.

Crucially, (3.8) holds for any choice of $\tau \geq 0$ including relatively small values. For very large values of τ, (3.8) converges to (3.7). However, in most cases of interest, the escape time $S_{D^{\mathsf{c}}}^{+}(0)$ is very large, making estimation of u in (3.7) by direct simulation of sample trajectories of X_t prohibitively expensive. In the context of DGA, the significance of (3.8) is that it expresses u in terms of an expectation over short trajectories. The catch is that (3.8) must be "inverted" to solve for u.

3.2.2 Dynamical Galerkin Approximation (DGA)

We now describe a Galerkin approach to approximating conditional expectations from short trajectory data. We first introduce a "guess" function ψ that satisfies the boundary conditions (i.e., $\psi(x) = b(x)$ for $x \notin D$). Our approximation has the form

$$u(x) \approx \psi(x) + \sum_{j=1}^{n} \phi_j(x) v_j, \tag{3.9}$$

where $\{\phi_j(x)\}$ is a set of n basis functions satisfying $\phi_j(x) = 0$ for $x \notin D$, and v is a vector of n coefficients.

Forward-in-time predictions

We begin by approximating predictions of quantities forward-in-time as in (3.7) by expanding the solution u of (3.8) at a particular user chosen value of τ called the lag time. While the solution u itself is independent of τ in (3.8), the quality of our approximation of u with a finite basis may depend on the choice of lag time (even in the absence of sampling error). A similar phenomenon has recently been explained in detail in the context of the VAC algorithm [1]. Substituting (3.9) into (3.8), multiplying by ϕ_i and integrating over the distribution of sampled points μ to form the inner product $\langle f, g\rangle = \int f(x)g(x)\mu(dx)$, we obtain the linear system of equations:

$$(C^\tau - C^0)v = r^\tau, \tag{3.10}$$

with matrices $C^s \in \mathbb{R}^{n\times n}$ for $s = 0, \tau$,

$$C^s_{ij} = \langle \phi_i, \mathcal{T}^s_{D^\mathsf{c}}\, \phi_j\rangle_\mu \tag{3.11}$$

and vector $r^\tau \in \mathbb{R}^n$,

$$r^\tau_i = \left\langle \phi_i, \psi(x) - \mathcal{T}^\tau_{D^\mathsf{c}}\, \psi(x) + \int_0^\tau \mathcal{T}^t_{D^\mathsf{c}}\, a(x)dt \right\rangle_\mu . \tag{3.12}$$

Given (3.11) and (3.12), (3.10) can be readily solved for v by standard methods of linear algebra.

In models that represent molecules with high fidelity, (3.11) and (3.12) cannot be evaluated directly because a closed form of $\mathcal{T}^\tau_{D^\mathsf{c}}$ is not known.

DGA overcomes this issue by approximating the action of the transition operator using short molecular dynamics trajectories: if X_0 is a sample drawn from μ and $\{X_t\}^\tau_{t=0}$ is a trajectory segment of length τ starting from X_0, then we can estimate C^s_{ij} (for $s = 0, \tau$) and

r_i^τ as

$$C_{ij}^s \approx \frac{1}{M}\sum_{m=1}^{M} \phi_i(X_0^{(m)})\phi_j(X^{(m)}_{s\wedge S^+_{D^\mathsf{c}}(0)}) \tag{3.13}$$

$$r_i^\tau \approx \frac{1}{M}\sum_{m=1}^{M} \phi_i(X_0^{(m)}) \left(\psi(X_0^{(m)}) - \psi(X^{(m)}_{\tau\wedge S^+_{D^\mathsf{c}}(0)}) + \Delta \sum_{p=0}^{N-1} h(X^{(m)}_{p\Delta})\right) \tag{3.14}$$

where m indexes trajectory segments, Δ is the sampling interval, and N satisfies $N\Delta = \tau \wedge S^+_{D^\mathsf{c}}(0)$. To avoid overhead, it is advantageous to generate trajectories much longer than τ (but still much shorter than typical values of $S^+_{D^\mathsf{c}}(0)$) and use a rolling window to generate short trajectories of length τ. We further note that in practice configurations are not saved at every molecular dynamics step. This limits the resolution of both the lag time and the stopping time, which we take to be the time of the first saved configuration outside the domain D.

Adjoints, the steady state, and backward-in-time predictions

To compute many important quantities we need not only to solve equations involving the transition operator but also equations involving its adjoint $(\mathcal{T}^t)^\dagger_\mu$ in the μ-weighted inner product, which by definition satisfies

$$\int f(x)\mathcal{T}^t g(x)\,\mu(dx) = \int g(x)(\mathcal{T}^t)^\dagger_\mu f(x)\,\mu(dx). \tag{3.15}$$

One such equation is for the change of measure $w = d\pi/d\mu$, which can be used to reweight from the sampling distribution μ to the stationary distribution π:

$$\int f(x)\pi(dx) = \int f(x)w(x)\mu(dx), \tag{3.16}$$

assuming μ and w are normalized such that $\int w(x)\mu(dx) = 1$. Owing to the time translational

invariance of averages over the stationary distribution π, (3.15), and (3.16), the change of measure satisfies the equation

$$(\mathcal{T}^\tau)^\dagger_\mu w(x) - w(x) = 0. \tag{3.17}$$

(3.17) can be solved analogously to (3.8), but, in this case, there are no boundary conditions. The introduction of a basis leads to a linear system of equations of the form

$$(\bar{C}^\tau - \bar{C}^0)^\top v = 0, \tag{3.18}$$

with $\bar{C}^s$ (for $s = 0, \tau$) differing from C^s only in the choice of basis (which is no longer restricted to D) and the use of $\mathcal{T}^s$ in place of $\mathcal{T}^s_{D^\mathsf{c}}$; $\top$ denotes the transpose. We note that by including $\phi_1(x) = 1$ in the basis we can guarantee that the equation for v has a solution. Given an approximate w, (3.16) can be computed as

$$\int f(x)\pi(dx) \approx \sum_{j=1}^{M} f(X_0^{(j)}) w(X_0^{(j)}), \tag{3.19}$$

with the weights normalized such that

$$\sum_{j=1}^{M} w(X_0^{(j)}) = 1. \tag{3.20}$$

That the change of measure can be estimated from short nonequilibrium trajectory data was previously observed in Wu et al. [32].

Another important quantity expressible in terms of an equation involving an adjoint of the transition operator is the backwards committor

$$q_-(x) = \mathbb{P}\left[X_{-T^-_{A\cup B}} \in A \,|\, X_0 = x\right], \quad T^-_{A\cup B} = \min\{t \geq 0 :\, X_{-t} \in A \cup B\}, \tag{3.21}$$

for $x \notin A \cup B$, where X_{-t}, $t \geq 0$ is the steady-state backward-in-time process governed by the transition operator

$$\mathcal{T}^{-t} f(x) = (\mathcal{T}^t)^\dagger_\pi f(x) = \frac{1}{w(x)} (\mathcal{T}^t)^\dagger_\mu [fw](x) \tag{3.22}$$

(the last equality can be verified using (3.15)). The backward committor is the probability that a trajectory currently at position x last came from the reactant state A rather than the product state B. It satisfies the Feynman-Kac relation

$$\mathcal{T}^{-\tau}_{A\cup B}\, q_-(x) - q_-(x) = 0 \text{ for } x \notin A \cup B, \qquad q_-(x) = \mathbb{1}_A(x) \text{ for } x \in A \cup B. \tag{3.23}$$

Consistent with our definition of $\mathcal{T}^t_{A\cup B}$ above, $\mathcal{T}^{-t}_{A\cup B}$ is the transition operator for the steady-state backward-in-time process stopped upon first entrance in $A \cup B$.

To expand and approximate q_- according to the DGA recipe described above, we need to estimate μ-weighted inner products involving $\mathcal{T}^{-t}_{A\cup B}$. To that end we note that, as long as $g = 0$ on $A \cup B$,

$$\left\langle g, \mathcal{T}^{-t}_{A\cup B}\, f \right\rangle_\mu = \int \mathbb{E}\left[f(X_{S_{A\cup B}(t)}) \frac{g(X_t)}{w(X_t)} \,\middle|\, X_0 = x \right] w(x)\mu(dx) \tag{3.24}$$

where

$$S_{A\cup B}(t) = \max\{s \leq t :\ X_s \in A \cup B\} \tag{3.25}$$

(with $S_{A\cup B}(t) = 0$ if $X_s \notin A \cup B$ for all $0 \leq s \leq t$). We provide a derivation of (3.24) in Appendix 3.6.1. Just as for the forward committor, we expect that use of a sampling measure μ with high resolution in transition regions will lead to higher approximation accuracy (i.e., better ability of a finite basis to capture the dynamics). However, in our experience the factor of $w^{-1}(X_t)$ in (3.24) leads to significant sampling errors for larger values of t. For our

backward committor calculation we therefore weight inner products by π, using the formula

$$\left\langle g, \mathcal{T}_{A\cup B}^{-t} f \right\rangle_\pi = \int \mathbb{E}\left[f(X_{S_{A\cup B}(t)}) g(X_t) \,\middle|\, X_0 = x \right] w(x)\mu(dx). \tag{3.26}$$

(3.26) allows inner products involving $\mathcal{T}_{A\cup B}^{-t}$ to be computed using forward trajectories of X initiated according to μ, i.e., exactly the same ingredients required to make forward-in-time predictions by DGA.

Following our procedure for forward quantities outlined in Section 3.2.2, given a guess function ψ satisfying $\psi = 1$ on A and $\psi = 0$ on B and basis functions ϕ_j that are zero on $A \cup B$, we can build an approximation

$$q_-(x) \approx \psi(x) + \sum_{j=1}^{n} \phi_j(x) v_j \tag{3.27}$$

by solving

$$(C^{-\tau} - C^0)v = r^{-\tau} \tag{3.28}$$

with

$$C_{ij}^{-\tau} = \left\langle \phi_i, \mathcal{T}_{A\cup B}^{-\tau} \phi_j \right\rangle_\pi = \int \mathbb{E}\left[\phi_j(X_{S_{A\cup B}(\tau)}) \phi_i(X_\tau) \,\middle|\, X_0 = x \right] w(x)\mu(dx) \tag{3.29}$$

and

$$\begin{aligned} r_i^{-\tau} &= \langle \phi_i,\, \psi \rangle_\pi - \left\langle \phi_i,\, \mathcal{T}_{A\cup B}^{-\tau} \psi \right\rangle_\pi \\ &= \int \mathbb{E}\left[\Big(\psi(X_\tau) - \psi(X_{S_{A\cup B}(\tau)}) \Big) \phi_i(X_\tau) \,\middle|\, X_0 = x \right] w(x)\mu(dx) \end{aligned} \tag{3.30}$$

where the second equality in each display follows from (3.26).

Along with the forward committor q_+ and the stationary change of measure w, the backward committor is a key ingredient of TPT. In Section 3.2.3 we describe how DGA

estimates of these quantities can be combined with TPT to reveal key properties of steady-state transition paths from the reactant state A to the product state B. However, before that, we complete our presentation of DGA with a discussion of molecular representations and basis sets, with emphasis on those that we employ in the present study to analyze trp-cage miniprotein unfolding and folding.

Basis functions

A key determinant of the performance of DGA is the choice of basis set. Constructing a basis set that respects the boundary conditions of the problem and captures the dynamics with relatively few functions requires care. Here we discuss how we generated the basis sets that we compare later in our numerical experiments, and explain why we chose them over alternatives.

In addition to the choice of functions, there is also a choice of molecular representation (i.e., the features that serve as inputs to the functions). Although molecular dynamics trajectories are generally recorded as sequences of Cartesian coordinates, the inputs to the basis functions are generally internal coordinates. This removes the effects of trivial translations and rotations, and it can improve the statistics. The internal coordinates that we use are pairwise distances between all C_α atoms, except those pairs which are less than three sequence positions apart; for trp-cage, there are 153 such distances. In other words, the process X_t to which we apply DGA (and TPT) is the length 153 vector of pairwise distance values. In our tests we found that including additional features, such as backbone dihedral angles, did not improve performance.

We assume that the reactant state A and product state B of interest can be characterized in terms of these variables. We construct basis functions of these variables that satisfy the homogeneous boundary condition on the domain $D = (A \cup B)^{\mathsf{c}}$.

In this work, we compare three choices of basis set: indicator functions on the pairwise

distances, indicator functions constructed on the top 10 TICA coordinates [12–14] computed from the pairwise distances at a lag time of 0.5 ns, and smooth functions of pairwise distances that satisfy the boundary conditions. We refer to these henceforth as the distance indicator, TICA indicator, and modified distance basis sets.

We constructed the distance indicator and TICA indicator basis sets and their guess functions as follows:

1. For the distance indicator basis set, we constructed 200 indicator functions by mini-batch k-means clustering as implemented in PYEMMA on the values of the 153 pairwise distances. For the TICA indicator basis set, the clustering was performed on the top 10 TICA coordinates constructed on the pairwise distances.

2. We retained all resulting indicator functions with non-zero regions fully contained in $(A \cup B)^c$ as the basis set. We split any indicator functions with non-zero regions overlapping with A or B, and we redefined them to be non-zero only in the portions in $(A \cup B)^c$. For the change of measure calculations, boundary conditions are not present, so we used all indicator functions unmodified.

3. For the forward committor calculation we took the the guess function to be $\psi(x) = \mathbb{1}_B(x)$. For the backward committor calculation we took the guess function to be $\psi(x) = \mathbb{1}_A(x)$.

With an indicator basis, the DGA and MSM estimator (with appropriate state definitions) of the forward committor q_+ and change of measure w become similar [4]. We note however that the DGA (as formulated here) and MSM approaches diverge both in DGA's use of stopped trajectories and in the way q_+ and w (and q_-) are used to estimate TPT quantities as described in Section 3.2.3.

We constructed the distance basis set and its guess function as follows:

1. We computed d_A and d_B as the distance in feature space (i.e. in 153-dimensional Euclidean space) to the sampled points in states A and B, respectively.

2. We set $h(x) = d_A d_B/(d_A + d_B)^2$, which obeys the homogeneous boundary conditions by construction.

3. We computed basis functions obeying the boundary conditions by multiplying each coordinate of the pairwise distance vector x by $h(x)$: $\phi_i(x) = x_i\, h(x)$. For the change of measure calculation, we use $\phi_i = x_i$ and add the constant function into the set of chosen features.

4. To remove any linear dependencies introduced by enforcing the boundary conditions, and to ensure numerical stability, we orthogonalized the basis set ϕ_i with respect to the sampling measure (up to sampling error) using a singular value decomposition.

5. For the forward committor calculation we took the guess function to be $\psi(x) = d_B^2/(d_A + d_B)^2$. For the backward committor calculation we took the guess function to be $\psi(x) = d_A^2/(d_A + d_B)^2$.

Although here we use the backbone pairwise distances, we note that this construction procedure could be used to generate basis sets obeying the homogeneous boundary conditions for a choice of variables other than the pairwise distances such as dihedral angles, radial basis functions, or soft indicator functions.

The indicator and TICA basis sets are the most widely used in the MSM literature. Various alternatives have been proposed specifically in the context of spectral estimation [29, 30, 86, 87]. In our previous work [4], we considered a basis set based on diffusion maps [78]. Due to the size of our trp-cage data set ($\sim 10^6$ datapoints), the $O(N^3)$ scaling of the matrix diagonalization associated with the diffusion map proved prohibitively computationally costly without subsampling and out of sample extension.

Delay Embedding

Application of DGA as described so far assumes that the underlying process X_t is Markovian; the conditional expectations that DGA seeks to approximate are not fully defined if X_t is not Markovian. Yet, in the previous section we described an approach to building a basis set for DGA consisting of functions of only a subset of the full collection of variables (selected pairwise distances). Though the dynamics of this subset are not strictly Markovian, in Section 3.4 we show that, at least in the specific context of the trp-cage system, the remaining degrees of freedom relax sufficiently fast that DGA yields accurate results.

However, in some circumstances, one may only have access to a small number of variables that are insufficient to specify the dynamics. This situation is typical when the data are from an experiment. In this case, we can construct a more expressive representation of the system from time-lagged images, i.e., if X_t is not itself Markovian we can instead apply DGA to the augmented process $\bar{X}_t = (X_{t-M\delta}, X_{t-(M-1)\delta}, \ldots, X_t)$ [4]. For large enough M one can expect $\bar{X}_t$ to be nearly Markovian. State space augmentation was also used in the history-augmented MSM (haMSM) approach of Suárez et al. [37] to obtain accurate MFPT estimates at all lag times. Our approach differs in that we construction our basis on the delay embedded space, whereas in the haMSM approach the transition probabilities are conditioned on visiting multiple clusters in sequence. In principle, one can explicitly include memory as defined by the Mori-Zwanzig formalism [88], though our delay-embedding approach is computationally more straightforward because it does not require choosing a form for the memory kernel and then estimating it from data, both of which are quite challenging [6, 89].

In Section 3.4.5, we show that delay embedding can significantly improve DGA estimates when a small number of CVs is used to characterize molecular configurations. Writing the values of the CVs at time t as the vector X_t, we construct the delay embedded process $\bar{X}_t$. We then construct a basis set following the recipe in Section 3.2.2 for the modified distance basis, but replacing X with $\bar{X}$. We then extend other functions f of the CV space to the

delay-embedded space by $f(\bar{X}_t) = f(X_{t-\lfloor M/2 \rfloor \delta})$. This allows us to extend the states A and B (which can both be defined in terms of the CVs) as well as the functions a and b in (3.7). We then apply DGA as outlined above directly on the delay-embedded space.

3.2.3 Reaction rates and currents

Estimates of rates from simulations are frequently of interest because they can be compared directly with experimental measurements, and they can provide indirect information about mechanisms. TPT in principle provides not just rate estimates but reactive currents or fluxes, which provide direct information about mechanisms. However, previous calculations of reactive current have been limited to toy models and depictions of the reactive flux between metastable states can been difficult to interpret. Working within the TPT framework and building upon DGA approximations of w, q_+, and q_-, in this section we introduce robust estimates of the reaction rate and of an easily interpretable projection of the reactive current onto CVs (as opposed to over the network of metastable states).

There are various expressions for the rate in TPT. One approach is based on the rate at which trajectories transition from A to B, R_{AB}.

If U is any set for which $A \subset U$ and $B \subset U^{\mathsf{c}}$ then, for a continuous time process,

$$\begin{aligned} R_{AB} &= \lim_{t \to 0} \frac{1}{t} \int \left(\mathbb{1}_U(x) \mathcal{T}^t [\mathbb{1}_{U^{\mathsf{c}}} q_+](x) - \mathbb{1}_{U^{\mathsf{c}}}(x) \mathcal{T}^t [\mathbb{1}_U q_+](x) \right) q_-(x) \pi(dx) \\ &= \lim_{t \to 0} \frac{1}{t} \int \left(\mathbb{1}_U(x) \mathcal{T}^t q_+(x) - \mathcal{T}^t [\mathbb{1}_U q_+](x) \right) q_-(x) \pi(dx), \end{aligned} \tag{3.31}$$

where the second line is obtained by noting that $\mathbb{1}_{U^{\mathsf{c}}}(x) = 1 - \mathbb{1}_U(x)$. Here and below, for a discrete time process the limit is removed and t is replaced by the unit of a single time step.

Expression (3.31) simply counts trajectories with forward crossings of the surface dividing U and U^{c}, weighted by their probabilities that they start in A and end in B. Consequently, when using this formula to estimate rates from data, only those trajectories that cross

the surface dividing U and U^{c} contribute. Because these trajectories are generally a small fraction of the data, this results in relatively large variances in estimates. We can obtain considerably better estimates by considering the isocommittor surfaces: $U(z) = \{x : q_+(x) \leq z,\ x \in D\}$ for $z \in (0,1)$, and noting that R_{AB} is independent of z. Integrating (3.31) with respect to z [3], then exchanging integrals over z with applications of $\mathcal{T}^t$ and noting that $\int_0^1 \mathbb{1}_{[0,z]}\big(q_+(x)\big)\, dz = 1 - q_+(x)$, we find that

$$\begin{aligned} R_{AB} &= \lim_{t\to 0} \frac{1}{t} \int \int_0^1 \left\{ \mathbb{1}_{[0,z]}\big(q_+(x)\big)\mathcal{T}^t q_+(x) - \mathcal{T}^t \left[q_+ \mathbb{1}_{[0,z]}(q_+) \right](x) \right\} dz\, q_-(x)\pi(dx) \\ &= \lim_{t\to 0} \frac{1}{t} \int \left\{ \big[1 - q_+(x)\big]\mathcal{T}^t q_+(x) - \mathcal{T}^t \big[q_+(1-q_+)\big](x) \right\} q_-(x)\pi(dx) \\ &= \lim_{t\to 0} \frac{1}{t} \int \Big(\mathcal{T}^t q_+^2(x) - q_+(x)\mathcal{T}^t q_+(x) \Big)\, q_-(x)\pi(dx), \end{aligned} \tag{3.32}$$

where we have made use of the fact that the integral of the Heaviside function (which enters through the indicator functions) is the ramp function.

This expression for R_{AB} immediately suggests the estimator:

$$R_{AB} \approx \frac{1}{t} \sum_i q_+(X^{(i)}_{t\wedge S^+_{A\cup B}(0)}) \left[q_+(X^{(i)}_{t\wedge S^+_{A\cup B}(0)}) - q_+(X_0^{(i)}) \right] q_-(X_0^{(i)}) w(X_0^{(i)}) \tag{3.33}$$

for some small choice of t. Note the use of stopped trajectories in (3.33). For very small values of t the inclusion of the stopping time $S^+_{A\cup B}(0)$ has no impact. However, in our numerical experiments we find that use of stopped trajectories improves the accuracy of (3.33) and, in particular, (3.37) below, for most choices of t. Given an estimate of R_{AB}, the rate constant is

$$k_{AB} = \frac{R_{AB}}{\sum_i q_-(X_0^{(i)}) w(X_0^{(i)})}. \tag{3.34}$$

The denominator in 3.34 is the mean of the backward committor, which is the fraction of time the system spends having last visited state A.

As noted above, we can also use simulations to understand how reactive trajectories flow through a CV space. One way to do this is to partition the space into discrete states and then estimate the reactive fluxes between pairs of states [74]. However, the resulting directed graph can be complicated and difficult to interpret. When the sample paths are continuous the reactive flux between neighboring values in CV space is can be summarized as a single vector field in CV space. If θ is a vector-valued CV and ds is a bin in CV space of volume $|ds|$, the reactive current at point s is

$$J^{\theta}_{AB}(s) = \lim_{t,\,|ds|\to 0} \frac{1}{2t\,|ds|} \int \Big(\mathcal{T}^t[\theta q_+](x) - \theta(x)\mathcal{T}^t q_+(x)\Big)\, \mathbb{1}_{\{\theta\in ds\}}(x) q_-(x)\pi(dx) \\ + \Big(\mathcal{T}^t[\mathbb{1}_{\{\theta\in ds\}}\theta q_+](x) - \theta(x)\mathcal{T}^t[\mathbb{1}_{\{\theta\in ds\}} q_+](x)\Big)\, q_-(x)\pi(dx) \tag{3.35}$$

In Appendices 3.6.2 and 3.7, we show that $J^{\theta}_{AB}(s) = \int J_{AB}\cdot\nabla\theta(x)\delta(\theta(x)-s)\pi(dx)$, and we establish that the projected reactive current satisfies

$$\int_{\partial C^\theta} J^{\theta}_{AB}(s)\cdot n_{C^\theta} d\sigma_{C^\theta} = \int_{\partial C} J_{AB}\cdot n_C d\sigma_C, \tag{3.36}$$

where C^θ is any region of CV space such that its inverse image (under the CV mapping) in the full configuration space, C, contains A and does not intersect B. To estimate J^{θ}_{AB} from trajectory data we have the following estimator:

$$J^{\theta}_{AB}(s) \approx \frac{1}{2t|ds|}\sum_{i=1}^{M} q_+(X^{(i)}_{t\wedge S^+_{A\cup B}(0)})\left(\theta(X^{(i)}_{t\wedge S^+_{A\cup B}(0)}) - \theta(X^{(i)}_0)\right) \times \mathbb{1}_{\theta\in ds}(X^{(i)}_0) q_-(X^{(i)}_0) w(X^{(i)}_0) + \frac{1}{2t|ds|}\sum_{i=1}^{M} q_+(X^{(i)}_t)\left(\theta(X^{(i)}_t) - \theta(X^{(i)}_{S_{A\cup B}(t)})\right) \times \mathbb{1}_{\theta\in ds}(X^{(i)}_t) q_-(X^{(i)}_{S_{A\cup B}(t)}) w(X^{(i)}_0) \quad (3.37)$$

Note that the lag time t in (3.33) and (3.37) need not be the same as the lag time τ used to estimate the committors q_+ and q_-. Even with perfect sampling and a perfect basis, estimates of TPT quantities will depend on t, in contrast to τ. Several considerations are involved in the choice of t. For larger values of t (3.33) and (3.37) incur significant bias due to poor approximation of the $t \to 0$ limit in (3.32) and (3.35). On the other hand, for small values of t, we found that (3.33) and (3.37) suffer large statistical errors.

Alternative estimators for the rate and reactive current based on expressions that are exact at any lag time up to error from the discrete sampling interval Δ are given in Appendix 3.7.1.

A full analysis of error sources is beyond the scope of this work, and in practice we choose a lag time that gives reasonable results for the change of measure and reasonable smoothness in the vector field.

3.3 Simulation methods and choices

In this section, we specify the computational procedure to generate and analyze the data set for the unfolding and folding of trp-cage. We describe preparing the system and its underlying dynamics (Section 3.3.1), choosing collective variables based on their ability to distinguish metastable states (Section 3.3.2), generating and validating the data set of short trajectories

(3.3.3), and defining the unfolded and folded states (Section 3.3.4).

3.3.1 System setup

Unless otherwise noted, all molecular dynamics simulations were performed with GROMACS 5.1.4 [46] and PLUMED 2.3 [90–92] using the CHARMM36m force field [93–95] in the NVT ensemble at 300 K using the Langevin thermostat with a temperature coupling constant of $10\,\mathrm{ps}^{-1}$ applied to all atoms, and a time step of 2 fs. Bonds to hydrogen atoms were constrained using the LINCS algorithm [55]. Electrostatic interactions were computed using particle-mesh Ewald summation with a cutoff of 1.2 nm. Lennard-Jones interactions were switched off from 1.0 to 1.2 nm using the default GROMACS switching function.

The system was prepared from an NMR structure of trp-cage (PDB code 1L2Y [96]). The protein was solvated in a 50 Å cubic box with the TIP3P water model [97] using CHARMM-GUI 3.0 [98, 99]. 10 K^+ and 11 Cl^- ions were added, bringing the system to charge neutrality and 150 mM KCl. The energy of the system was minimized until the maximum force was below $1000\,\mathrm{kJ\,mol^{-1}\,nm^{-1}}$. The system was then equilibrated for 1 ns in the NVT ensemble with position restraints (using a 1 fs timestep), 10 ns in the NPT ensemble with harmonic restraints on non-hydrogen atom positions (force constant $400\,\mathrm{kJ/mol/nm^2}$ for backbone atoms and $40\,\mathrm{kJ/mol/nm^2}$ for side chain atoms.) and a Parrinello-Rahman barostat with a pressure coupling constant of $5\,\mathrm{ps}^{-1}$, 5 ns in the NPT ensemble without position restraints, and then 10 ns in the NVT ensemble without position restraints. The cubic box length was determined from the restraint-free NPT equilibration run to be 4.48 nm and fixed at that value after that run.

3.3.2 Choice of CVs

The performance of DGA rests on having a data set with good sampling of all states that contribute to the reaction mechanism. As mentioned in the Introduction, the available

physically weighted molecular dynamics data for trp-cage [39] contain few unfolding and folding transitions. We thus sought to use enhanced sampling methods to generate a data set with improved representation outside the stable states. To this end, we evaluated CVs for their ability to control sampling and resolve the unfolded and folded states.

Based on previous studies [43, 83], we considered five CVs:

1. The radius of gyration of the C_α atoms (R_g);
2. The root mean squared deviation (RMSD) of all C_α atoms from their positions in an equilibrated structure ($\text{RMSD}_{\text{full}}$);
3. The RMSD of the C_α atoms of residues 2 to 9, which make up the α helix in the native state (RMSD_{hx});
4. The RMSD of the C_α atoms of residues 11 to 15, which make up the 3-10 helix in the native state ($\text{RMSD}_{\text{3-10}}$);
5. The end-to-end distance (d).

R_g, $\text{RMSD}_{\text{full}}$, and RMSD_{hx} were used in Juraszek and Bolhuis [83], and $\text{RMSD}_{\text{3-10}}$ was used in Sidky et al. [43] (there defined only to residue 14), where they found that it was able to resolve several metastable states identified by spectral clustering.

To explore how these collective variables change as trp-cage unfolds, we ran a series of Adiabatic Bias Molecular Dynamics (ABMD) [100] simulations to drive unfolding from the equilibrated native structure. ABMD uses a ratchet-and-pawl-like bias to trap spontaneous fluctuations that move the system forward in selected CVs. By applying ABMD with different combinations of the CVs above, we found that $\text{RMSD}_{\text{full}}$ and $\text{RMSD}_{\text{3-10}}$ yielded reasonable control of the system and enabled exploration of all metastable states characterized in previous studies.

3.3.3 Generation of the DGA data set

To initialize a data set of short trajectories for DGA, we defined a grid of 64 points in the space of $\mathrm{RMSD}_{\mathrm{full}}$ and $\mathrm{RMSD}_{\text{3-10}}$ (Figure 3.1). We then used 64 independent ABMD simulations to steer the system to each of these points from the final structure from the equilibration simulations described in Section 3.3.1. We ran each ABMD simulation for 1 ns, saving the structure every 5 ps; the force constants were $1.25\,\mathrm{kJ/\AA^2}$ and $1.0\,\mathrm{kJ/\AA^2}$ for $\mathrm{RMSD}_{\mathrm{full}}$ and $\mathrm{RMSD}_{\text{3-10}}$, respectively. From the set of all recorded structures, we chose the 64 structures closest to the targets and equilibrated each for 1 ns with a harmonic restraint with the same force constants as in the AMBD simulations. From each of the resulting structures, we then launched 14 free simulations (with different random number generator seeds) of length 30 ns each, saving structures every 5 ps.

From this data set, we computed all possible two-dimensional potentials of mean force (PMFs) involving the CVs listed in Section 3.3.2. We compared these PMFs with corresponding ones from replica exchange umbrella sampling (REUS). Based on the DGA PMFs, we used the RMSD of the α helix ($\mathrm{RMSD}_{\mathrm{hx}}$), and the RMSD of the 3-10 helix ($\mathrm{RMSD}_{\text{3-10}}$), and the end-to-end distance (d) to control the sampling. REUS window centers were placed on a uniform $8 \times 8 \times 8$ grid of these three CVs, with $\mathrm{RMSD}_{\mathrm{hx}}$ ranging from 0.3 to 2.8 Å, $\mathrm{RMSD}_{\text{3-10}}$ ranging 0.3 to 3.3 Å, and d ranging from 6 to 38 Å. This grid fully covered the relevant areas of CV space identified by previous simulations. The force constants for the harmonic potentials for each window were $29.2\,\mathrm{kJ/\AA^2}$ for $\mathrm{RMSD}_{\mathrm{hx}}$, $20.3\,\mathrm{kJ/\AA^2}$ for $\mathrm{RMSD}_{\text{3-10}}$, and $0.178\,\mathrm{kJ/\AA^2}$ for d, following Park et al. [101]. To initialize each window, structures were taken from the DGA database that were closest to each window center. The built-in replica exchange functionality of GROMACS was used to create a three-dimensional replica exchange procedure, where structures from nearby windows were periodically exchanged [102]. Every window was first simulated for 100 ps, with swaps attempted between adjacent windows in d space (i.e., window centers with the same $\mathrm{RMSD}_{\mathrm{hx}}$ and $\mathrm{RMSD}_{\text{3-10}}$ values, but neighboring d values) every 10 ps.

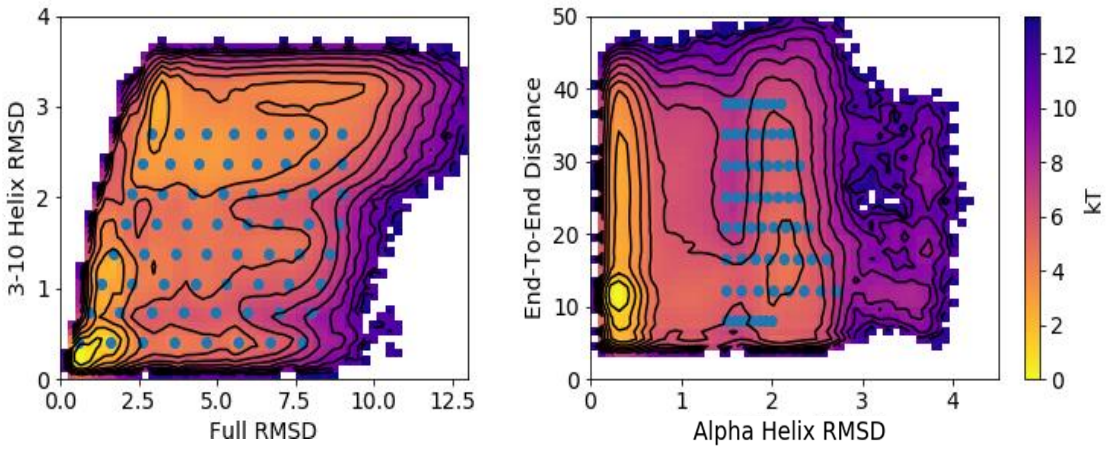

Figure 3.1: Initialization points for the data set of short trajectories. ABMD targets (symbols) are overlaid on DGA PMFs (color scale and contours, spaced every 1 k_BT) for the CVs used for steering. (left) The initial 64 ABMD targets were based on $\mathrm{RMSD}_{\mathrm{full}}$ and $\mathrm{RMSD}_{\text{3-10}}$; 14 free simulations of length 30 ns were launched from each of the structures resulting from these ABMD simulations. (right) 64 ABMD targets in $\mathrm{RMSD}_{\mathrm{hx}}$ and end-to-end distance added to ensure adequate sampling of the unfolded state; 2 free simulations of length 30 ns were launched from each of the structures resulting from these ABMD simulations.

This was repeated for a total of three 100 ps iterations, with the second and third iterations proposing swaps between neighboring windows in $\mathrm{RMSD_{hx}}$ and $\mathrm{RMSD_{3\text{-}10}}$, respectively. This 300-ps procedure was repeated until a total simulation time of 10 ns was reached for each window, with structures saved every 10 ps. Following this protocol, structures were exchanged across all of the three-dimensional grid, with exchange probabilities in the range 10-60%. The PMF was constructed by using the Eigenvector Method for Umbrella Sampling (EMUS) [103] extended to REUS [104]. The REUS simulations were run until the asymptotic variance of the PMF dropped below 0.1 $(k_BT)^2$ (Figure 3.10).

The REUS PMFs suggested that the initial DGA data set did not adequately sample configurations with $\mathrm{RMSD_{hx}} > 1.5$ Å (Figure 3.12, note the lack of sampling toward the upper right areas of the plots compared with those in Figure 3.2). In this case several of the basins are missing, and the RMSD over all bins is $> 1.3\ k_BT$. Therefore, we selected 64 more points from a grid with $\mathrm{RMSD_{hx}} > 1.5$ Å and a range of end-to-end distances from our short trajectory data set. From each of these points, we released two new free molecular dynamics simulations of length 30 ns (Figure 3.1B). With these additional trajectories, we obtained good agreement between DGA and REUS PMFs. Adding the extra sampling improved the PMFs involving $\mathrm{RMSD_{hx}}$ the most, but other PMFs were also noticeably improved. The data set used for all further DGA calculations thus contains a total of 1024 trajectories, each of length 30 ns, with structures saved every 5 ps.

3.3.4 State definitions

We found that PMFs projected onto only global measures of unfolding ($\mathrm{RMSD_{full}}$, R_g, and d) did not have clearly identifiable unfolded basins (Figures 3.1 and 3.2). By contrast, the PMF on the CVs tracking secondary structure ($\mathrm{RMSD_{hx}}$ and $\mathrm{RMSD_{3\text{-}10}}$) had clearly identifiable unfolded and folded basins, as well as several intermediates. Based on this analysis, we took

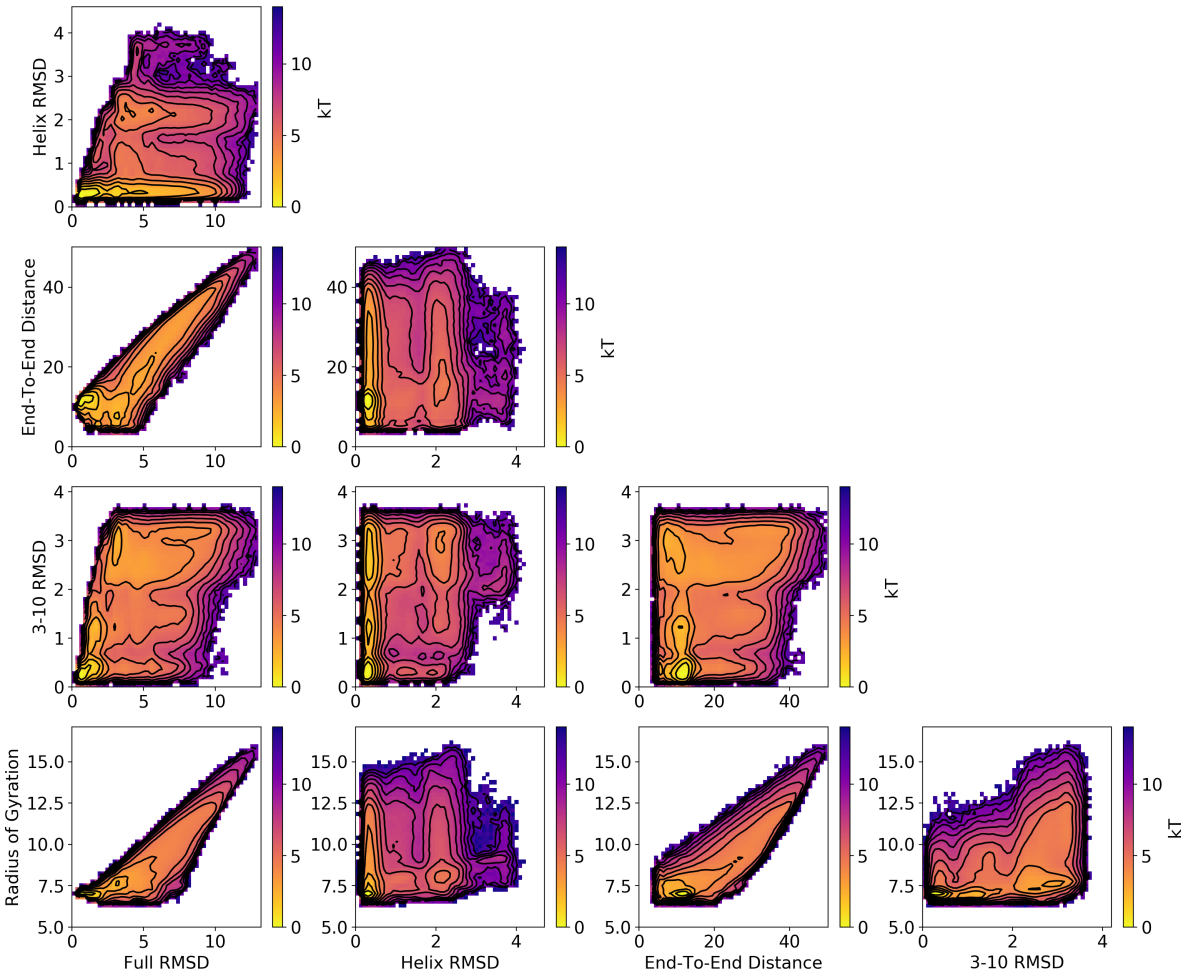

Figure 3.2: PMFs for the indicated CVs. Results shown are computed by DGA with the modified distance basis set and a lag time of 0.5 ns. We use a 50×50 grid to compute each PMF. Similar results are obtained with other basis sets and REUS; see Figures 3.13, 3.14, and 3.15.

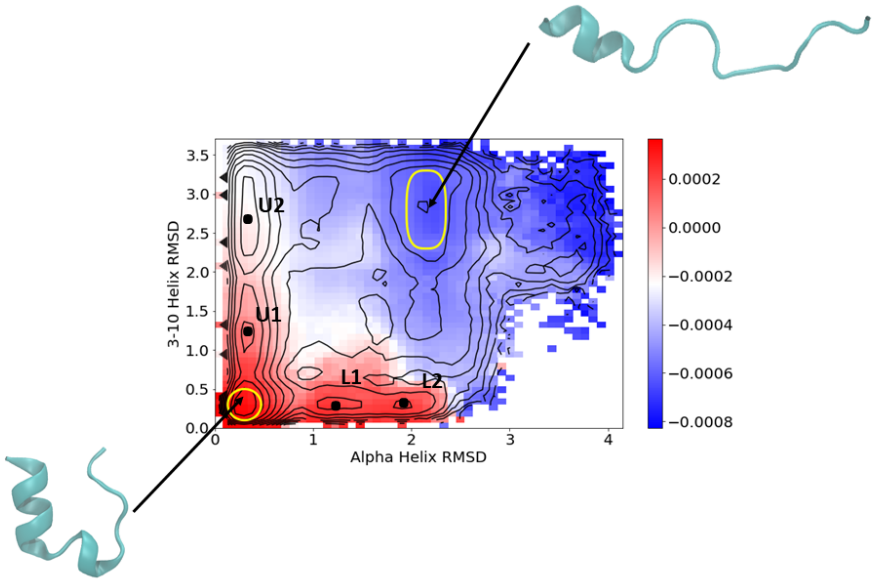

Figure 3.3: Top nontrivial TICA eigenvector averaged on the $\mathrm{RMSD_{hx}}$ and $\mathrm{RMSD_{3\text{-}10}}$ CVs with physical weighting. The unfolded and folded states are indicated in yellow with representative structures. Intermediate states in Table 3.1 are marked and labeled.

the unfolded state to be

$$\frac{|\mathrm{RMSD_{hx}} - 2.15\,\text{Å}|^3}{0.008\,\text{Å}^3} + \frac{|\mathrm{RMSD_{3\text{-}10}} - 2.8\,\text{Å}|^3}{0.125\,\text{Å}^3} < 1. \tag{3.38}$$

The folded state is

$$\frac{(\mathrm{RMSD_{hx}} - 0.3\,\text{Å})^2}{0.0289\,\text{Å}^2} + \frac{(\mathrm{RMSD_{3\text{-}10}} - 0.3\,\text{Å})^2}{0.04\,\text{Å}^2} < 1 \text{ and } d < 17\,\text{Å}. \tag{3.39}$$

We included the end-to-end distance constraint on the folded state to exclude structures which are extended but have the secondary structure intact.

Heterogeneous structures contribute to the unfolded state, making it challenging to define, and there is no guarantee that the choices above are optimal in any sense. Because we expect unfolding and folding to be among the slowest motions of the system, an alternative

would be to define the states in terms of the slowest mode of the system identified by a dimensionality-reduction algorithm. However, data-driven state definitions are often difficult to interpret physically, despite their theoretical justifications. Furthermore, data-driven state definitions can be difficult to incorporate into sampling algorithms. We thus use physical CVs for path sampling, stratification, and state definitions, and we then check for consistency with a data-driven state choice.

Figure 3.3 shows that the slowest mode of the system identified by TICA applied to the DGA data set correlates with the PMF and switches between low and high values in going between the unfolded and folded states. Here and going forward, all functions we project onto CVs are conditional averages of the form $\int f(x)\delta(\theta(x)-s)\pi(dx)/\int \delta(\theta(x)-s)\pi(dx)$. We estimate these by binning our CV space into bins, and for each bin ds, plotting:

$$\frac{\int f(x)\delta(\theta(x)-s)\pi(dx)}{\int \delta(\theta(x)-s)\pi(dx)} \approx \frac{\sum_i f(X_0^{(i)})w(X_0^{(i)})\mathbb{1}_{\theta\in ds}(X_0^{(i)})}{\sum_i w(X_0^{(i)})\mathbb{1}_{\theta\in ds}(X_0^{(i)})}. \tag{3.40}$$

We furthermore show in Section 3.4.2 that this mode correlates with the committor. We thus feel that $\mathrm{RMSD_{hx}}$ and $\mathrm{RMSD_{3\text{-}10}}$ enable the clearest two-dimensional projection of the reaction and present most of our results in terms of these CVs. In addition to the unfolded and folded states, we define four intermediate states U1, U2, L1, and L2 shown on Figure 3.3. In the next section, we apply our DGA and TPT formalism to show that trp-cage can fold along an upper path through intermediates U1 and U2, or a lower path through L1 and L2.

3.4 Trp-cage analysis

In this section, we evaluate how the three basis sets described in Section 3.2.2 (indicator functions of pairwise distances, indicator functions of TICA coordinates, and pairwise distances modified to satisfy the boundary conditions) impact the performance of DGA for estimating PMFs, rates, committors, and reactive currents for the unfolding and folding of the trp-cage

miniprotein. Where possible, we compare our results with references obtained by independent means.

3.4.1 Comparison of PMFs

Figure 3.2 shows PMFs computed on each pair of the physically motivated CVs with DGA with the modified distance basis set. The corresponding PMFs from REUS are shown in Figure 3.11; difference maps comparing the results obtained with the two methods and three basis sets are shown in Figures 3.13, 3.14, and 3.15. All of the main basins identified by REUS are present in the DGA PMFs, and there is good quantitative agreement between REUS and DGA, with RMSDs of $< 1\ k_BT$ for all three basis sets (that said, of these, the distance indicator basis set results in the largest deviations). Consistent with their agreement with the REUS PMF, the three DGA PMFs are in agreement with each other. We did observe that REUS tends to give slightly flatter PMFs than DGA with all three basis sets. In principle, there are two sources of error in the DGA PMFs: (i) approximation error from representing the true change of measure with a basis expansion and (ii) estimation (sampling) error. Analysis of error in DGA will be the subject of future work. Error in US is discussed in Thiede et al. [103], Antoszewski et al. [104], and Dinner et al. [105].

Table 3.1: CV values for metastable states.

State	$\mathrm{RMSD}_{\mathrm{full}}$/Å	$\mathrm{RMSD}_{\mathrm{hx}}$/Å	d/Å	$\mathrm{RMSD}_{\text{3-10}}$/Å	R_g/ Å
Folded	1.1	0.30	11.1	0.30	7.0
Unfolded	5.8	2.1	20.2	2.8	9.2
U1	2.4	0.34	13.1	1.2	7.3
U2	5.2	0.34	19.3	2.8	8.8
L1	2.2	1.2	9.5	0.30	7.2
L2	2.6	1.9	14.5	0.30	7.3

We found that the projection onto the $\mathrm{RMSD}_{\mathrm{hx}}$ and $\mathrm{RMSD}_{\text{3-10}}$ coordinates was best able to separate the pathways and states of interest, so we now focus on this projection. Figure 3.3 indicates the folded (lower left) and unfolded (upper right) basins, as well four intermediates.

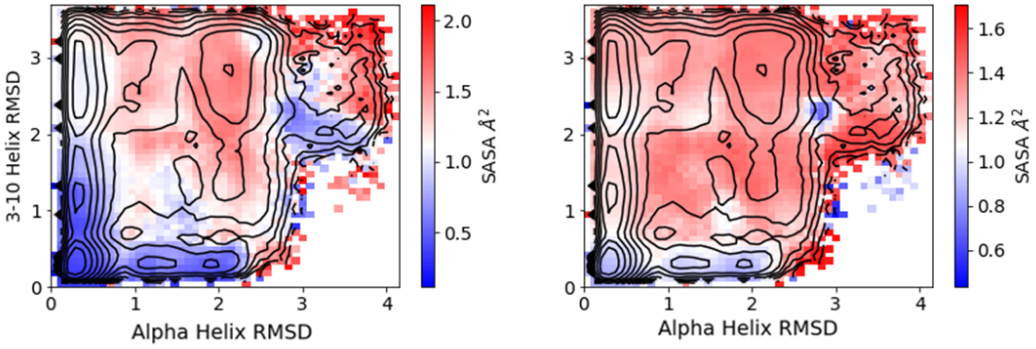

Figure 3.4: Equilibrium average solvent accessible surface area (SASA) projected onto the RMSD_{hx} and $\text{RMSD}_{\text{3-10}}$ CVs for (left) trp-6 and (right) proline-12.

The intermediates define two pathways, which we label upper (with intermediates U1 and U2) and lower (with intermediates L1 and L2). Table 3.1 gives the five CV values for each of the six states.

To understand the characteristics of the intermediate states, we turn to Figure 3.4, which shows the solvent-accessible surface area (SASA) of trp-6 on the left, and pro-12 on the right. We find that the U2 intermediate state is characterized by partial solvation of the hydrophobic core, measured by the SASA of trp-6, and nearly full detachment of pro-12. Furthermore, the U2 state is significantly more extended than the lower pathway intermediates as measured both by R_g and end-to-end distance. In addition to being more compact, with near-native R_g values, L1 and L2 have near-native trp-6 and pro-12 SASA values, suggesting the hydrophobic core is fully formed. These intermediate states can be mapped to those previously reported in the literature. Bolhuis and Jurazek [82] identified three folding intermediates. Our U1 and U2 intermediates roughly map onto their P_d and I intermediates, and our L1 and L2

intermediates roughly map onto their L intermediate. U1 and U2 also correspond to states S_7 and S_0 identified by Sidky *et al.* [43].

3.4.2 Comparison of committors

We next calculated both forward and backward committors using DGA with the three basis sets and lag times ranging from 0.5 ns to 12 ns (Figure 3.5 and Figure 3.16). As they should, the backward committors mirror the forward committors, so we focus our discussion on the latter. The timescale of trp-cage folding is on the order of 5 μs from both experiment [81] and simulation [83], thus both our trajectory lengths (30 ns) and lag times are several orders of magnitude shorter than the motions of interest, providing an appropriate setting in which we expect DGA to show benefits.

In contrast to the PMFs, we found the committors to be sensitive to the choice of basis set (and associated guess function). The modified distance basis set, in addition to being substantially faster to construct as it avoids slow and unstable high-dimensional clustering, is less prone to discontinuities at the boundary than the distance indicator function basis set. The TICA indicator function basis set performs similarly to the modified distance basis set and has the advantage over the distance indicator basis set that clustering on the lower-dimensional subspace is significantly faster and more stable.

For a given basis set, we found relatively little variation in the committors across lag times. This is in contrast to variational approach for conformational dynamics (VAC) algorithm, where the results can strongly depend on the lag time [1] (although this can be mitigated by using multiple lag times [9]). We postpone a full investigation of DGA's error properties, and in particular its dependence on the choice of lag time, to future work.

Because we expect unfolding and folding to be among the slowest motions of the system, we can validate the DGA committors by comparing them with the slowest mode of the system identified by TICA. Comparing Figures 3.3 and 3.5 shows that the largest TICA eigenvector

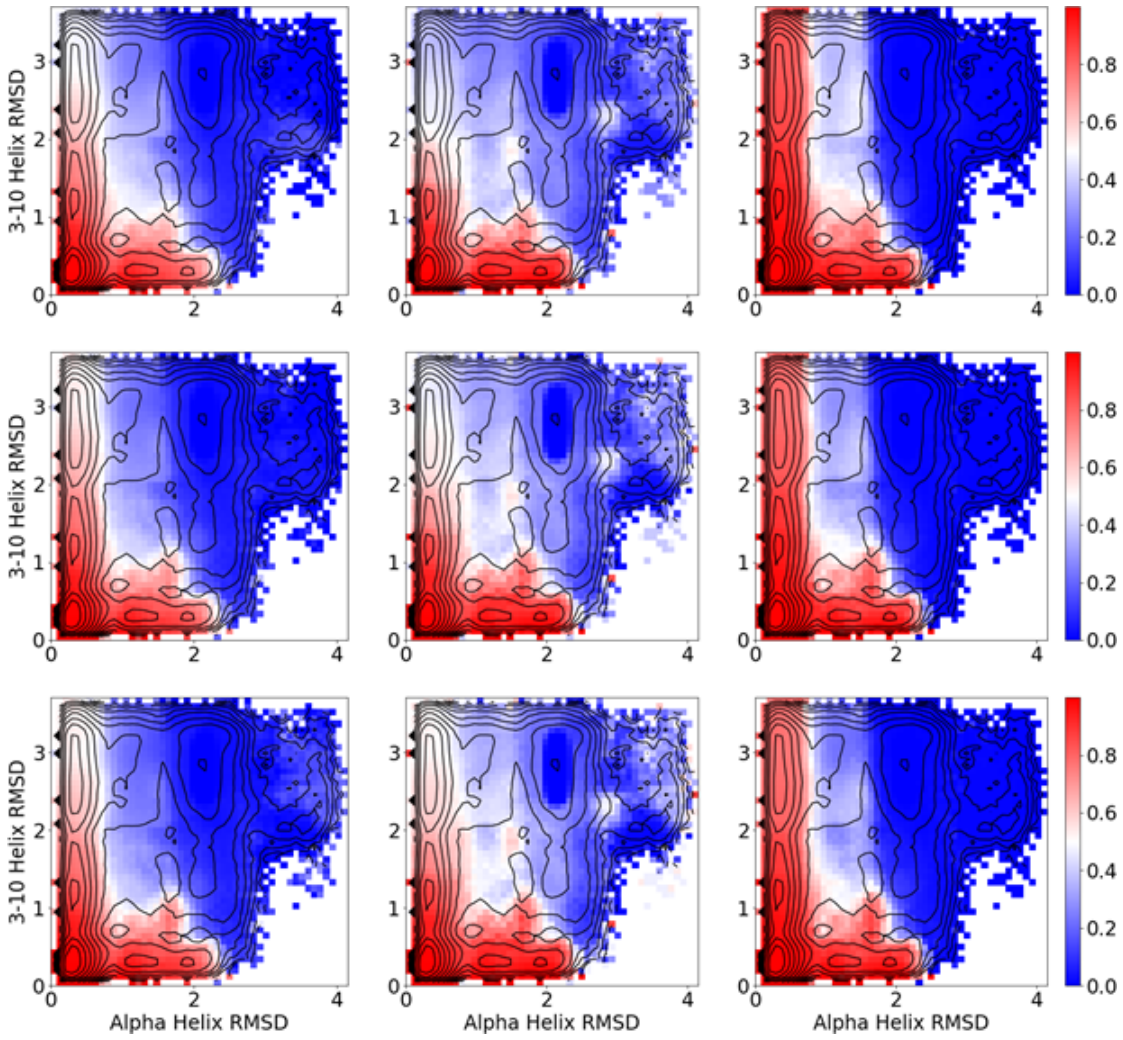

Figure 3.5: DGA forward committors. Left, middle, and right columns are computed with the modified distance, distance indicator, and TICA indicator basis sets, respectively. Top, middle, and bottom rows are computed with lag times of 0.5, 2.5, and 7.5 ns, respectively.

(estimated with a lag time of 0.5 ns) correlates almost perfectly with the estimated committors obtained with the modified distance basis set, when projected onto $\mathrm{RMSD}_{\mathrm{hx}}$ and $\mathrm{RMSD}_{\text{3-10}}$. The agreement between these two independent calculations furthermore suggests that the physically motivated CVs capture the behavior detected by the data-driven method. In this projection, we see that the transition states fall where the SASA of trp-6 (Figure 3.4) changes rapidly.

As an additional validation, we used DGA with the modified distance basis set and a lag time of 0.5 ns (Figure 3.6) to compute committors on the CVs used by Juraszek and Bolhuis [82]. When projected onto RMSD and $\mathrm{RMSD}_{\mathrm{hx}}$, the positions of the transition states in Figure 4 of Juraszek and Bolhuis [82] fall in areas estimated to have $q_+ = 0.5$ (white in Figure 3.6). The traditional shooting approach employed in Juraszek and Bolhuis [82] is quite computationally costly and provides information about only a limited number of structures. Our ability to capture the transition states thus makes clear the benefit of DGA. We discuss DGA's ability to provide mechanistic information further in the next section.

3.4.3 Reactive currents

We computed reactive currents for the three basis sets using the estimator in (3.37) and the committors from the the previous section (Figure 3.7). For this calculation, we use the shortest lag time of 0.5 ns for both the committor and reactive current, though in principle they could be chosen separately. As previously, we primarily present our results projected onto $\mathrm{RMSD}_{\mathrm{hx}}$ and $\mathrm{RMSD}_{\text{3-10}}$. Overall the results for the three basis sets are similar, though the distance indicator basis set exhibits greater noise around ($\mathrm{RMSD}_{\mathrm{hx}}$, $\mathrm{RMSD}_{\text{3-10}}$) = (1.3 Å, 1.3 Å), consistent with the plateau in the committor in this region.

The currents, which provide information directly about dynamics, confirm the presence of two paths for the folding process: an upper path with formation of the α helix prior to formation of the 3-10 helix, and a lower path with the order of these events transposed. The

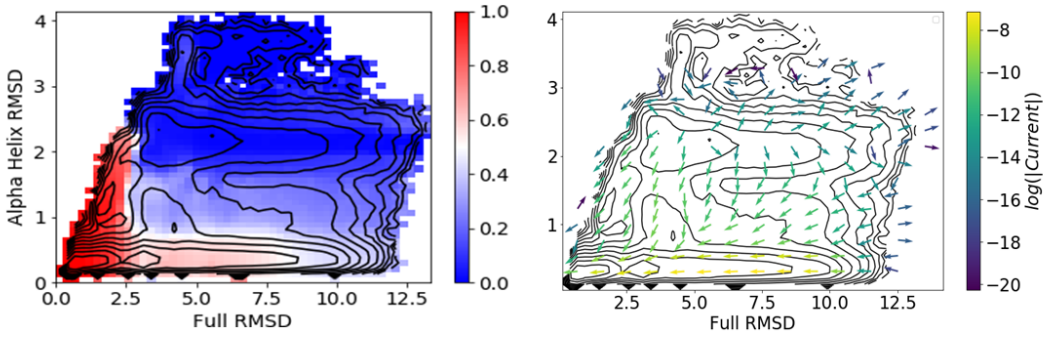

Figure 3.6: Forward committor (left) and reactive current (right) projected onto the $\mathrm{RMSD}_{\mathrm{hx}}$ and full RMSD CVs used in Juraszek and Bolhuis [82]. Results shown are computed with the modified distance basis set and a lag time of 0.5 ns.

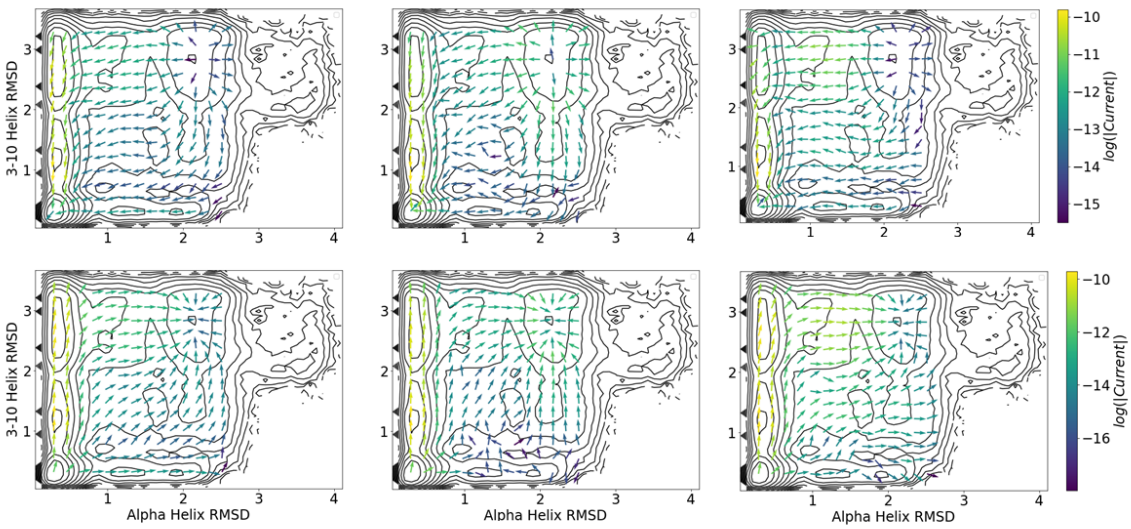

Figure 3.7: Folding (top) and unfolding (bottom) reactive current projected onto the $\mathrm{RMSD}_{\mathrm{hx}}$ and $\mathrm{RMSD}_{\text{3-10}}$ CVs using (3.37) with the three choices of basis set. Left, middle, and right columns are computed with the modified distance, distance indicator, and TICA indicator basis sets, respectively. All computations use a lag time of 0.5 ns.

upper path proceeds through intermediates U1 and U2, with folding beginning with formation of the α helix and partial desolvation of trp-6, followed by full formation of the 3-10 helix. The lower path proceeds through L1 and L2, with folding beginning with collapse into the L2 intermediate with no α helix, but the hydrophobic core fully formed, followed by formation of the α helix. Both of these paths correspond to troughs in the PMFs on these CVs.

Previous studies have found multiple pathways resembling the ones we find here. Kim et al. [106] used diffusion maps to identify two pathways: one with tertiary contacts forming first, followed by α helix formation, and another with the order transposed. Jurazek and Bolhius came to similar conclusions using transition path sampling [82].

An advantage of the reactive current is that we can use it to assign weights to the two paths. By computing the relative flux crossing $\mathrm{RMSD}_{\text{3-10}} = 1.8\,\text{Å}$ with either $\mathrm{RMSD}_{\mathrm{hx}} < 1.4\,\text{Å}$ (upper pathway) or $\mathrm{RMSD}_{\mathrm{hx}} > 1.4\,\text{Å}$ (lower pathway), we conclude that 88% of the reactive paths proceed by first forming the α helix, and then the 3-10 helix and hydrophobic core (i.e., the upper pathway). Although we are not aware of a previous estimate of the reactive current for this system, we can compare these numbers to the frequencies with which transition path sampling sampled the pathways in Juraszek and Bolhuis [82]. There, Juraszek and Bolhuis observed the pathway in which tertiary contacts form first (i.e., the lower pathway) 80% of the time. The difference may be due to different CV and state definitions (Jurazek and Bolhuis [82] used 5 CVs in their state definitions , whereas we consider only $\mathrm{RMSD}_{\text{3-10}}$ and $\mathrm{RMSD}_{\mathrm{hx}}$) or force field and setup differences.

3.4.4 Rates

Finally, we computed rates using the estimator in (3.31). We present our results as inverse rates (unfolding and folding times) to make comparisons to lag times and trajectory lengths clear. As mentioned previously, these times are expected to be on the order of microseconds. In particular, Juraszek and Bolhuis used transition interface sampling to estimate inverse

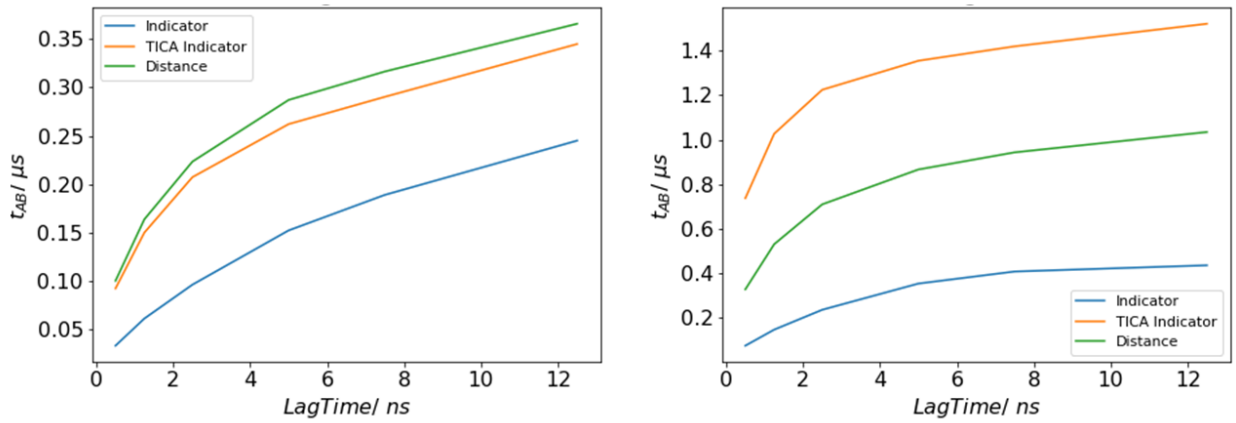

Figure 3.8: Inverse rates estimated for folding (left) and unfolding (right).

unfolding and folding rates of 1.2 μs and 0.4 μs [83], though as noted previously those results are for a different model.

All three basis sets gave rate estimates that were within an order of magnitude of those numbers. However, the results for the distance indicator basis were markedly faster. Furthermore, in all three cases, the inverse rate exhibited significant dependence on lag time. We do not show lag times >12 ns since they suffer from pronounced statistical error due to the limitations of our short-trajectory data set. Our analysis of the trajectory of the K8A mutant suggests the need for a lag time of at least 100 ns (consistent with Sidky et al. [43]), though as discussed in the Introduction, those data do not contain a sufficient number of unfolding and folding events to obtain accurate rate estimates. Juxtaposed with the lack of sensitivity to lag time for the committor and reactive current, these observations suggest that DGA's strength is in its ability to give statistical insight into mechanisms with relatively little data, but that rates may be more efficiently computed by methods that directly sample

relevant statistics such as stratification schemes [63].

3.4.5 Demonstration of delay embedding

As described in Section 3.2.2, delay embedding can be used to construct an approximately Markovian process when the feature space does not fully capture the dynamics. To illustrate this idea using our trp-cage data set, we restrict the feature space to the five physical CVs and apply DGA with the modified distance basis set on either the feature space itself or the delay-embedded feature space. Figure 3.9 shows the reactive currents and committors resulting from DGA on these two spaces. We find that the committor and current constructed from the delay embedded representation largely agree with the DGA result constructed on the 153 pairwise distances. Without delay embedding, we find several qualitative disagreements, in particular the U2 state has a committor value close to zero, and the reactive current does not resolve the two pathways since many of the arrows point directly towards the folded state.

3.5 Conclusions

In this chapter, we have cast the dynamical Galerkin approximation (DGA) [4] for computing chemical kinetic statistics from short trajectories in terms of the stopped transition operator. This formulation can be immediately translated into expressions that can be applied to simulation data. It also clarifies the role of the lag time, showing that estimates of conditional expectations computed by DGA are exact in the infinite basis and data limit, independent of the choice of lag time.

To evaluate DGA's performance, we generated and carefully validated a data set of short trajectories for the unfolding and folding of the trp-cage miniprotein, a well-characterized system. We used umbrella sampling to validate our short trajectory data set by comparing the resulting PMFs. Quantitative agreement between the PMFs was observed, suggesting

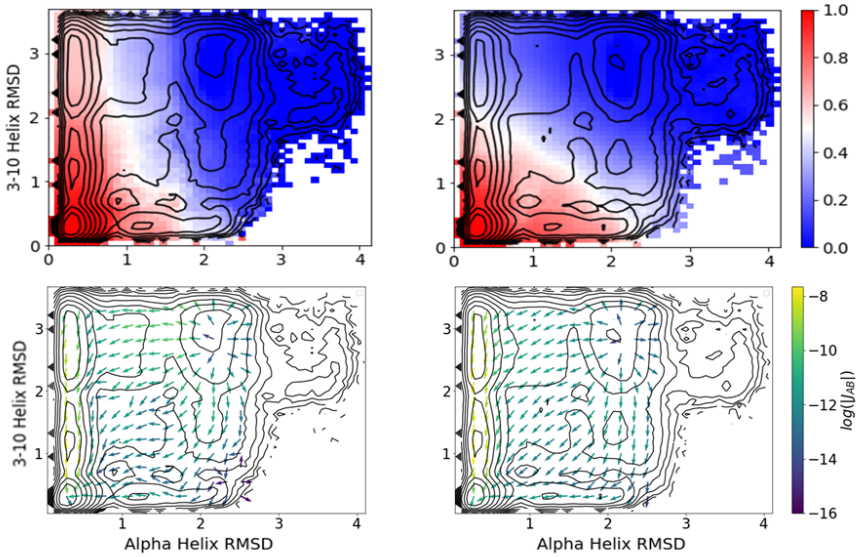

Figure 3.9: Comparison of DGA estimates for the forward committor (top) and reactive current for folding (bottom) with the modified distance basis set on a feature space restricted to the five physical CVs (right) and a delay-embedded feature space (left). The delay-embedded results are obtained with a delay of $\delta = 0.125$ ns, $N = 40$ images, and a DGA lag time of 0.5 ns.

that our short trajectory data set had sufficient sampling to compute dynamical statistics. The PMF calculations furthermore enabled us to rapidly assess different combinations of CVs for their abilities to separate metastable states. The α helix RMSD and 3-10 helix RMSD in particular allowed us to resolve intermediates to a greater degree than found in previous studies.

We next applied DGA to compute forward and backward committors between the unfolded and folded states.

We evaluated a number of competing estimators for the backward committor and found that one based on forward trajectories weighted by the stationary distribution gave the best results. The committors by themselves are not able to identify reaction pathways or transition states, but they can be combined according to transition path theory to extract this information. Specifically, we introduce a new estimator for the TPT rate, and a projection formula and corresponding estimator for the reactive current in a CV space. Our projected reactive current allows us to easily resolve and visualize the pathways that the system takes in arbitrary CV spaces, and even lets us assign relative weights to these pathways. Acquiring this kind of mechanistic information has previously been possible only through transition path sampling and related methods; such methods do not as readily allow exploration of CVs and state definitions because the sampling is linked directly to them.

We introduced a simple procedure that takes an arbitrary set of molecular features and adapts them to produce a basis set that satisfies the homogeneous boundary conditions. Using pairwise distances as the molecular features, we compared the performance of such a basis set with indicator functions on the molecular features and indicator functions on TICA coordinates. Other basis constructions such as diffusion maps and radial basis functions are possible, and we expect that the best choice will be system dependent. We applied our DGA and TPT formalism to our data set, and identified intermediate states and pathways which have been previously reported in the literature, providing further validation of our methods.

We found that the estimates of the TPT rate, while on the same order of magnitude as previous estimates, nevertheless show significant dependence on lag time. Finally, we showed that delay embedding can be an effective strategy for constructing a molecular representation with approximately Markovian dynamics from a low-dimensional feature space.

Our results suggest several interesting directions for future investigation. We have seen that in our trp-cage application the choice of lag time has only a modest effect on DGA estimates of conditional expectations, while TPT quantities, in particular the rate, depend sensitively on lag time. Recently, we showed that integrating over lag times for VAC improves the robustness of that method [9]. It will be interesting to see if an analogous strategy can improve rate estimates from DGA. An in depth mathematical study of DGA's error and its dependence on lag time along the lines of our previous analysis of VAC [1] is also in order. By showing how DGA's results depend on the sampling measure, such an analysis could lead to a practical scheme for targeting sampling to selected regions, just as our analysis of US [103, 107] did [104]. This will be particularly important for systems that are not amenable to the strategy that we took in the present study of using REUS for identifying regions of CV space that require more sampling.

Though DGA has performed well in our tests so far, looking ahead to larger and more complex systems, it may become necessary to move away from a Galerkin approach and toward more flexible representations of the kinetic functions we seek to approximate. This would be consistent with a trend toward using neural networks to represent eigenfunctions in spectral estimation [9, 19, 43]. Indeed, some of the first estimates of committors from data used neural networks [108, 109]. Introducing this higher level of representational flexibility while maintaining the reliability we observe in our trp-cage application of DGA will be a challenge.

3.6 Supporting information

3.6.1 Backward-in-time inner products

In this appendix we provide an elementary derivation of (3.24) which is key to our estimates of inner products involving the stopped backward-in-time transition operator $\mathcal{T}^t_{A\cup B}$. For the purposes of this derivation we assume that X_t is a discrete time process (so that t is a non-negative integer) with probability density $p(x(1)|x(0))$ for transition from $x(0)$ to $x(1)$ and stationary density π. The steady state backward-in-time process X_{-t} then has transition density

$$q(x(0)|x(1)) = \frac{p(x(1)|x(0))\pi(x(0))}{\pi(x(1))}. \tag{3.41}$$

From this expression we immediately find that

$$\begin{aligned}
&\pi(x(t))q(x(t-1)|x(t))q(x(t-2)|x(t-1))\cdots q(x(0)|x(1)) \\
&\qquad = \pi(x(0))p(x(t)|x(t-1))p(x(t-1)|x(t-2))\cdots p(x(1)|x(0))
\end{aligned} \tag{3.42}$$

relating the steady state backward-in-time path density to the steady state forward-in-time path density. Therefore, for any path function $F(x(0), x(1), \ldots, x(t))$ and any density μ (equivalent to π) we find (recalling that here $w = \pi/\mu$) that

$$\begin{aligned}
&\int \mathbb{E}\left[F(X_0, X_{-1}, \ldots, X_{-t}) \,|\, X_0 = x\right] \mu(x)dx \\
&\quad = \int \frac{F(x(0), \ldots, x(-t))}{w(x(0))} \pi(x(0))q(x(-1)|x(0))\cdots q(x(-t)|x(-t+1))dx(0)\cdots dx(-t) \\
&\quad = \int \frac{F(x(t), \ldots, x(0))}{w(x(t))} \pi(x(t))q(x(t-1)|x(t))\cdots q(x(0)|x(1))dx(t)\cdots dx(0) \\
&\quad = \int \mathbb{E}\left[\frac{F(X_t, X_{t-1}, \ldots, X_0)}{w(X_t)} \,\middle|\, X_0 = x\right] w(x)\mu(x)dx.
\end{aligned} \tag{3.43}$$

We will use (3.43) to find an expression for

$$\langle g, \mathcal{T}_{A\cup B}^{-t} f\rangle = \int \mathbb{E}\left[g(x) f(X_{-(T_{A\cup B}^{-}\wedge t)}) \,|\, X_0 = x\right] \mu(x)dx \tag{3.44}$$

in terms of the forward-in-time process. If we choose

$$F(X_0, X_{-1}, \ldots, X_{-t}) = g(X_0) f(X_{-(T_{A\cup B}^{-}\wedge t)}), \tag{3.45}$$

then

$$\langle g, \mathcal{T}_{A\cup B}^{-t} f\rangle = \int \mathbb{E}\left[F(X_0, X_{-1}, \ldots, X_{-t}) \,|\, X_0 = x\right] \mu(x)dx. \tag{3.46}$$

In terms of the forward process

$$F(X_t, X_{t-1}, \ldots, X_0) = f(X_{S_{A\cup B}(t)}) g(X_t), \tag{3.47}$$

where we remind the reader of (3.25):

$$S_{A\cup B}(t) = \max\{s \le t :\, X_s \in A \cup B\}$$

with $S_{A\cup B}(t) = 0$ if $X_s \notin A \cup B$ for all $0 \le s \le t$.

Applying (3.43) with this choice of F yields (3.24):

$$\left\langle g, \mathcal{T}_{A\cup B}^{-t} f \right\rangle = \int \mathbb{E}\left[f(X_{S_{A\cup B}(t)}) \frac{g(X_t)}{w(X_t)} \,\middle|\, X_0 = x \right] w(x)\mu(dx).$$

3.6.2 A formula for the reactive current

It has been shown [3] that for a diffusion with generator

$$\mathcal{L}f(x) = \sum_j b_j(x) \frac{\partial f}{\partial x_j}(x) + \frac{1}{2} \sum_{ij} a_{ij}(x) \frac{\partial^2 f}{\partial x_i \partial x_j}(x) \tag{3.48}$$

the reactive current is the vector field given by

$$\begin{aligned}(J_{AB})_i &= q_+(x)q_-(x)J_i + \\ &\quad \pi(x)q_-(x)\sum_j a_{ij}(x)\frac{\partial q_+}{\partial x_j}(x) - \pi(x)q_+(x)\sum_j a_{ij}(x)\frac{\partial q_-}{\partial x_j}(x),\end{aligned} \tag{3.49}$$

where J is the equilibrium current:

$$J_i = \pi(x)b_i(x) - \sum_j \frac{\partial[\pi a_{ij}]}{\partial x_j}(x). \tag{3.50}$$

To project the current onto a CV space of interest, we take the dot product with $\nabla\theta$ for any smooth CV θ and, using the identity

$$J_i \cdot \nabla f(x) = \frac{\pi(x)}{2}\left(\mathcal{L}f(x) - \mathcal{L}_\pi^\dagger f(x)\right), \tag{3.51}$$

which follows from direct manipulations, we can write

$$J_{AB} \cdot \nabla\theta(x) = \frac{\pi(x)}{2}\left(q_-(x)\mathcal{L}[q_+\theta](x) - q_+(x)\mathcal{L}_\pi^\dagger[q_-\theta](x)\right) \tag{3.52}$$

for $x \in (A \cup B)^{\mathrm{c}}$. This formula is not useful computationally since it still contains a backward-in-time generator. To compute statistics from data, we need to formulate their estimators as expectations against the stationary distribution since this (1) permits the use of the adjoint relation to clear away backward transition operators and (2) is consistent with our reweighting scheme. To this end, we define the projected reactive current as

$$J_{AB}^\theta(s) = \int J_{AB}(x) \cdot \nabla\theta(x)\delta(\theta(x) - s)dx = \lim_{|ds|\to 0}\frac{1}{|ds|}\int_{\{\theta(x)\in ds\}} J_{AB}(x) \cdot \nabla\theta(x)dx, \tag{3.53}$$

where $ds \in (A \cup B)^{\mathsf{c}}$ is an infinitesimal region of CV space with $s \in ds$, and $\{x : \theta(x) \in ds\}$ does not intersect $A \cup B$. Using (3.52) and the fact that $\mathcal{L}q_+ = 0$ and $\mathcal{L}_\pi^\dagger q_- = 0$ on $(A \cup B)^{\mathsf{c}}$, we have

$$\begin{aligned} J_{AB}^\theta(s) &= \lim_{|ds|\to 0} \frac{1}{|ds|} \int \mathbb{1}_{\{\theta(x)\in ds\}} \frac{\pi(x)}{2} \Big(q_-(x)\mathcal{L}[q_+\theta](x) - q_+(x)\mathcal{L}_\pi^\dagger[q_-\theta](x) \Big)\, dx \\ &= \lim_{|ds|\to 0} \frac{1}{|ds|} \int \mathbb{1}_{\{\theta(x)\in ds\}} \frac{\pi(x)}{2} \Big(q_-(x)\mathcal{L}[q_+\theta](x) - q_-(x)\mathcal{L}q_+(x)\theta(x) \\ &\qquad\qquad - q_+(x)\mathcal{L}_\pi^\dagger[q_-\theta](x) + q_+(x)\mathcal{L}_\pi^\dagger q_-(x)\theta(x) \Big) dx. \end{aligned} \tag{3.54}$$

Writing this expression in terms of the transition operator and canceling terms, we find that

$$\begin{aligned} J_{AB}^\theta(s) &= \lim_{t,|ds|\to 0} \frac{1}{2t\,|ds|} \int \mathbb{1}_{\{\theta(x)\in ds\}} \pi(x) \Big(q_-(x)\mathcal{T}^t[q_+\theta](x) - q_-(x)\mathcal{T}^t q_+(x)\theta(x) \\ &\qquad\qquad - q_+(x)(\mathcal{T}^t)_\pi^\dagger[q_-\theta](x) + q_+(x)(\mathcal{T}^t)_\pi^\dagger q_-(x)\theta(x) \Big) dx \\ &= \lim_{t,|ds|\to 0} \frac{1}{2t\,|ds|} \int \pi(x) q_-(x) \Big(\mathbb{1}_{\{\theta(x)\in ds\}} \Big(\mathcal{T}^t[q_+\theta](x) - \mathcal{T}^t q_+(x)\theta(x) \Big) \\ &\qquad\qquad + \Big(\mathcal{T}^t[q_+\theta\, \mathbb{1}_{\{\theta\in ds\}}](x) - \mathcal{T}^t[q_+ \mathbb{1}_{\{\theta\in ds\}}](x)\theta(x) \Big) \Big) dx, \end{aligned} \tag{3.55}$$

where the second equality follows from the definition of the adjoint $(\mathcal{T}^t)_\pi^\dagger$.

Expression (3.55) for $J_{AB}^\theta(s)$ can be directly translated into an estimator for computing from short-trajectory data:

$$\begin{aligned} J_{AB}^\theta(s) \approx \frac{1}{2t|ds|} &\sum_{i=1}^M q_+(X_t^{(i)}) \Big(\theta(X_t^{(i)}) - \theta(X_0^{(i)}) \Big) \\ &\times q_-(X_0^{(i)}) \mathbb{1}_{\theta\in ds}(X_0^{(i)}) w(X_0^{(i)}) \\ + \frac{1}{2t|ds|} &\sum_{i=1}^M q_+(X_t^{(i)}) \Big(\theta(X_t^{(i)}) - \theta(X_0^{(i)}) \Big) \\ &\times q_-(X_0^{(i)}) \mathbb{1}_{\theta\in ds}(X_t^{(i)}) w(X_0^{(i)}). \end{aligned} \tag{3.56}$$

Finally, without affecting the $t \to 0$ limit, we can stop our trajectories when they exit or enter $A \cup B$, yielding the estimator

$$\begin{aligned} J^{\theta}_{AB}(s) \approx \frac{1}{2t|ds|} \sum_{i=1}^{M} & q_+(X^{(i)}_{t \wedge S^+_{A\cup B}(0)}) \left(\theta(X^{(i)}_{t \wedge S^+_{A\cup B}(0)}) - \theta(X^{(i)}_0) \right) \\ & \times q_-(X^{(i)}_0) \mathbb{1}_{\theta \in ds}(X^{(i)}_0) w(X^{(i)}_0) \\ + \frac{1}{2t|ds|} \sum_{i=1}^{M} & q_+(X^{(i)}_t) \left(\theta(X^{(i)}_t) - \theta(X^{(i)}_{S_{A\cup B}(t)}) \right) \\ & \times q_-(X^{(i)}_{S_{A\cup B}(t)}) \mathbb{1}_{\theta \in ds}(X^{(i)}_t) w(X^{(i)}_0) \end{aligned} \tag{3.57}$$

which, in our experience, outperformed (3.56) for larger values of t.

Note that we could have canceled additional terms in (3.55) to yield a more concise estimator. However, we found that the estimator (3.57) gave less noisy results.

3.7 Reactive current on a CV space

We now establish that our projected reactive current gives the flux over surfaces in CV space. We assume that our CVs are smooth and that, for some subset C^θ of CV space with smooth boundary, the set $C = \{x : \theta(x) \in C^\theta\}$ contains A and does not intersect B. We will establish that for such a subset,

$$\int_{\partial C^\theta} J^{\theta}_{AB}(s) \cdot n_{C^\theta} d\sigma_{C^\theta} = \int_{\partial C} J_{AB} \cdot n_C d\sigma_C. \tag{3.36}$$

Here n_{C^θ} is the outward pointing normal vector to the boundary ∂C^θ of C^θ, n_C is the normal vector to the boundary ∂C of C, σ_{C^θ} is the surface measure on ∂C^θ and, σ_C is the surface measure on ∂C. The significance of (3.36) is that it shows that our definition of J^{θ}_{AB} preserves reactive flux across surfaces in the CV space so that statistics of reactive paths could, in principle, be computed directly from J^{θ}_{AB}.

Let f_δ be a smooth function on CV space that is equal to 1 on C^θ and equal to 0 for x a distance of more than δ from C^θ. Applying the divergence theorem and integrating by parts we find that

$$\begin{aligned}\int_{\partial C^\theta} J^\theta_{AB}(s) \cdot n_{C^\theta} d\sigma_{C^\theta} &= \int_{C^\theta} \operatorname{div} J^\theta_{AB}(s) ds \\ &= \lim_{\delta \to 0} \int f_\delta(s) \operatorname{div} J^\theta_{AB}(s) ds \\ &= -\lim_{\delta \to 0} \int J^\theta_{AB}(s) \cdot \nabla f_\delta(s) ds. \end{aligned} \tag{3.58}$$

Inserting our definition of J^θ_{AB} we find that

$$\begin{aligned}\int_{\partial C^\theta} J^\theta_{AB}(s) \cdot n_{C^\theta} d\sigma_{C^\theta} &= -\lim_{\delta \to 0} \sum_j \int \int J_{AB}(x) \cdot \nabla \theta_j(x) \delta(\theta(x) - s) \frac{\partial f_\delta(s)}{\partial s_j} dx ds \\ &= -\lim_{\delta \to 0} \sum_j \int J_{AB}(x) \cdot \nabla \theta_j(x) \frac{\partial f_\delta}{\partial s_j}(\theta(x)) dx. \end{aligned} \tag{3.59}$$

Using the chain rule the last expression can be rewritten as

$$\int_{\partial C^\theta} J^\theta_{AB}(s) \cdot n_{C^\theta} d\sigma_{C^\theta} = -\lim_{\delta \to 0} \int J_{AB}(x) \cdot \nabla f_\delta(\theta(x)) dx. \tag{3.60}$$

Integrating by parts, taking the $\delta \to 0$ limit, and applying the divergence theorem again yields (3.36).

3.7.1 Rate and Current Estimators for Long Lag Times

The estimators for the reaction rate and the reactive current presented in the main text incur significant bias at longer lag times. Here, we derive estimators which at all lag times converge to the TPT quantities with perfect sampling, committors, and change of measure.

We introduce the notation

$$S^+_{A\cup B}(t) = \min\{s \geq t : X_s \in A \cup B\} \tag{3.61}$$

$$S^-_{A\cup B}(t) = \max\{s \leq t : X_s \in A \cup B\} \tag{3.62}$$

for forward and backward stopping times starting at time t.

For the reaction rate estimator, we start with equation (3.32). Because X_0 is integrated over the stationary distribution, we can shift the term inside the expectation by an arbitrary time s. Averaging over $0 \leq s < \tau$ then yields

$$\begin{aligned}
R_{AB} &= \lim_{t\to 0} \frac{1}{t} \int (\mathcal{T}^t q_+^2(x) - q_+(x)\mathcal{T}^t q_+(x)) q_-(x)\pi(dx) \\
&= \lim_{t\to 0} \frac{1}{t} \int \mathbb{E}[q_+(X_t)(q_+(X_t) - q_+(X_0))q_-(X_0) \mid X_0 = x]\pi(dx) \\
&= \lim_{t\to 0} \frac{1}{t} \int \mathbb{E}[q_+(X_{s+t})(q_+(X_{s+t}) - q_+(X_s))q_-(X_s) \mid X_0 = x]\pi(dx) \\
&= \lim_{t\to 0} \frac{1}{t} \int \frac{1}{\tau} \int_0^\tau \mathbb{E}[q_+(X_{s+t})(q_+(X_{s+t}) - q_+(X_s)) \\
&\qquad\qquad q_-(X_s) \mid X_0 = x] ds\, \pi(dx) \\
&= \lim_{t\to 0} \frac{1}{t} \int \frac{1}{\tau} \int_0^\tau \mathbb{E}[q_+(X_{\tau\wedge S^+_{A\cup B}(s+t)})(q_+(X_{s+t}) - q_+(X_s)) \\
&\qquad\qquad q_-(X_{0\vee S^-_{A\cup B}(s)}) \mid X_0 = x] ds\, \pi(dx)
\end{aligned} \tag{3.63}$$

where (3.63) follows from (3.43) and the equations solved by the forward and backward committors.

This suggests the estimator

$$\begin{aligned}
R_{AB} \approx \frac{1}{\tau} \sum_i \sum_{p=0}^{\frac{\tau}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((p+1)\Delta)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta})) \\
& q_-(X^{(i)}_{0\vee S^-_{A\cup B}(p\Delta)}) w(X^{(i)}_0).
\end{aligned} \tag{3.64}$$

In contrast to the estimator in equation (3.33), we have taken the limit over the sampling interval Δ rather than the lag time, which is now τ in this expression. This is the same as the approximation made in equation (3.14).

To further relate this estimator to equation (3.33), we manipulate it to obtain a similar form. To do that, we require two identities.

First, we express forward stopping times in terms of backward stopping times. We can derive this by case analysis (Figure 3.17):

1. If $X_{0\vee S^-_{A\cup B}(s)} \notin A\cup B$, then $X_r \notin A\cup B$ for $0 \le r \le s$. This condition is equivalent to that of $X_{s\wedge S^+_{A\cup B}(0)} \notin A\cup B$.
2. If $X_{0\vee S^-_{A\cup B}(s)} \in A\cup B$ and $X_s \in A\cup B$, the process stopped immediately.
3. If $X_{0\vee S^-_{A\cup B}(s)} \in A\cup B$ and $X_s \notin A\cup B$, the process hit $A\cup B$ after propagating backward in time. The time $0\vee S^-_{A\cup B}(s)$ can be found by finding a time $r\in[0,s)$ such that $X_t \notin A\cup B$ for all times $t\in(r,s]$.

Together, for all $s\ge 0$, this yields

$$\begin{aligned} f(X_{0\vee S^-_{A\cup B}(s)}) &= \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\wedge S^+_{A\cup B}(0)})f(X_0) + \mathbb{1}_{A\cup B}(X_s)f(X_s) \\ &\quad + \lim_{t\to 0}\frac{1}{t}\int_0^s \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\wedge S^+_{A\cup B}(r+t)})\mathbb{1}_{A\cup B}(X_r)f(X_r)\,dr \\ &\approx \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\wedge S^+_{A\cup B}(0)})f(X_0) + \mathbb{1}_{A\cup B}(X_s)f(X_s) \\ &\quad + \sum_{p=0}^{\frac{s}{\Delta}-1} \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\wedge S^+_{A\cup B}((p+1)\Delta)})\mathbb{1}_{A\cup B}(X_{p\Delta})f(X_{p\Delta}). \end{aligned} \tag{3.65}$$

Second, we have the identity

$$S^+_{A\cup B}(r) = \lim_{t\to 0} S^+_{A\cup B}(s+t) \approx S^+_{A\cup B}(s+\Delta) \text{ if } X_{s\wedge S^+_{A\cup B}(r)} \in (A\cup B)^c \text{ and } s\ge r. \tag{3.66}$$

This follows from the definition of the stopping time: $S^+_{A\cup B}(r) = \min\{t'\ge r : X_{t'}\in A\cup B\}$,

and so if $X_{t'} \notin A \cup B$ for $r \leq t' \leq s$, then $\min\{t' \geq r : X_{t'} \in A \cup B\} = \min\{t' > s : X_{t'} \in A \cup B\}$.

We now apply equations (3.65) and (3.66) to equation (3.64), yielding

$$
\begin{aligned}
R_{AB} \approx \frac{1}{\tau}\sum_i \sum_{p=0}^{\frac{\tau}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}(0)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta})) \\
& \qquad \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X^{(i)}_{p\Delta\wedge S^+_{A\cup B}(0)})q_-(X^{(i)}_0)w(X^{(i)}_0) \\
+ \frac{1}{\tau}\sum_i \sum_{p=0}^{\frac{\tau}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((p+1)\Delta)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta})) \\
& \qquad \mathbb{1}_{A\cup B}(X^{(i)}_{p\Delta})q_-(X^{(i)}_{p\Delta})w(X^{(i)}_0) \\
+ \frac{1}{\tau}\sum_i \sum_{p=0}^{\frac{\tau}{\Delta}-1}\sum_{r=0}^{p-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta})) \\
& \qquad \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X^{(i)}_{p\Delta\wedge S^+_{A\cup B}((r+1)\Delta)})\mathbb{1}_{A\cup B}(X^{(i)}_{r\Delta})q_-(X^{(i)}_{r\Delta})w(X^{(i)}_0) \\
= \frac{1}{\tau}\sum_i \sum_{p=0}^{\frac{\tau\wedge S^+_{A\cup B}(0)}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}(0)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta}))q_-(X^{(i)}_0)w(X^{(i)}_0) \\
+ \frac{1}{\tau}\sum_i \sum_{p=0}^{\frac{\tau}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((p+1)\Delta)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta})) \\
& \qquad \mathbb{1}_{A\cup B}(X^{(i)}_{p\Delta})q_-(X^{(i)}_{p\Delta})w(X^{(i)}_0) \\
+ \frac{1}{\tau}\sum_i \sum_{r=0}^{\frac{\tau}{\Delta}-2} \sum_{p=r+1}^{\frac{\tau\wedge S^+_{A\cup B}((r+1)\Delta)}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)})(q_+(X^{(i)}_{(p+1)\Delta}) - q_+(X^{(i)}_{p\Delta})) \\
& \qquad \mathbb{1}_{A\cup B}(X^{(i)}_{r\Delta})q_-(X^{(i)}_{r\Delta})w(X^{(i)}_0) \\
= \frac{1}{\tau}\sum_i & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}(0)})(q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}(0)}) - q_+(X^{(i)}_0))q_-(X^{(i)}_0)w(X^{(i)}_0) \\
+ \frac{1}{\tau}\sum_i \sum_{r=0}^{\frac{\tau}{\Delta}-1} & q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)})(q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)}) - q_+(X^{(i)}_{r\Delta})) \\
& \qquad \mathbb{1}_{A\cup B}(X^{(i)}_{r\Delta})q_-(X_{r\Delta})w(X^{(i)}_0) \qquad (3.67)
\end{aligned}
$$

This is equation (3.33) with an additional term

$$\frac{1}{\tau}\sum_i \sum_{r=0}^{\frac{\tau}{\Delta}-1} q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)})(q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)}) - q_+(X^{(i)}_{r\Delta})) \\ \mathbb{1}_{A\cup B}(X^{(i)}_{r\Delta})q_-(X^{(i)}_{r\Delta})w(X^{(i)}_0) \tag{3.68}$$

which adds the contributions from trajectories leaving $A \cup B$ within the lag time τ, which becomes significant at longer lag times since a larger portion of the trajectories hit (or start in) and then exit $A \cup B$.

We proceed analogously for the reactive current estimator. We start with expression (3.35) and integrate to obtain

$$\begin{aligned}
J^\theta_{AB}(s) &= \lim_{t,|ds|\to 0} \frac{1}{2t|ds|}\int (\mathcal{T}^t[\theta q_+](x) - \theta(x)\mathcal{T}^t q_+(x))\mathbb{1}_{\{\theta\in ds\}}(x)q_-(x)\pi(dx) \\
&\qquad + (\mathcal{T}^t[\mathbb{1}_{\{\theta\in ds\}}\theta q_+](x) - \theta(x)\mathcal{T}^t[\mathbb{1}_{\{\theta\in ds\}}q_+](x))q_-(x)\pi(dx) \\
&= \lim_{t,|ds|\to 0} \frac{1}{2t|ds|}\int \mathbb{E}[q_+(X_{r+t})(\theta(X_{r+t}) - \theta(X_r))q_-(X_r) \\
&\qquad (\mathbb{1}_{\{\theta\in ds\}}(X_{r+t}) + \mathbb{1}_{\{\theta\in ds\}}(X_r)) \mid X_0 = x]\pi(dx) \\
&= \lim_{t,|ds|\to 0} \frac{1}{2t|ds|}\int \frac{1}{\tau}\int_0^\tau \mathbb{E}[q_+(X_{r+t})(\theta(X_{r+t}) - \theta(X_r))q_-(X_r) \\
&\qquad (\mathbb{1}_{\{\theta\in ds\}}(X_{r+t}) + \mathbb{1}_{\{\theta\in ds\}}(X_r)) \mid X_0 = x]dr\,\pi(dx) \\
&= \lim_{t,|ds|\to 0} \frac{1}{2t|ds|}\int \frac{1}{\tau}\int_0^\tau \mathbb{E}[q_+(X_{\tau\wedge S^+_{A\cup B}(r+t)})(\theta(X_{r+t}) - \theta(X_r)) \\
&\qquad q_-(X_{0\vee S^-_{A\cup B}(r)})(\mathbb{1}_{\{\theta\in ds\}}(X_{r+t}) - \mathbb{1}_{\{\theta\in ds\}}(X_r)) \mid X_0 = x]dr\,\pi(dx),
\end{aligned} \tag{3.69}$$

which suggests the estimator

$$J^\theta_{AB}(s) \approx \frac{1}{2\tau|ds|}\sum_i \sum_{p=0}^{\frac{\tau}{\Delta}-1} q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((p+1)\Delta)})(\theta(X^{(i)}_{(p+1)\Delta}) - \theta(X^{(i)}_{p\Delta})) \\ q_-(X^{(i)}_{0\vee S^-_{A\cup B}(p\Delta)})(\mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{(p+1)\Delta}) + \mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{p\Delta}))w(X^{(i)}_0). \tag{3.70}$$

We note that if we integrate over s and substitute $\theta(x) = q_+(x)$, we obtain the reaction rate estimator (3.64). As with equation (3.64), this estimator converges to the reactive current with perfect sampling, committors, and change of measure. In comparison to equation (3.37), we find that this estimator has larger variance with limited data. This results from the contribution of $\theta(X^{(i)}_{(p+1)\Delta}) - \theta(X^{(i)}_{p\Delta})$ being assigned locally to $X_{(p+1)\Delta}$ and $X_{p\Delta}$, whereas equation (3.37) assigns $\theta(X^{(i)}_{\tau\wedge S^+_{A\cup B}(0)}) - \theta(X_0)$ to X_0 and $\theta(X_\tau) - \theta(X_{0\vee S^-_{A\cup B}(\tau)})$ to X_τ which leads to mixing over longer length scales. Thus, when using (3.70), we recommend smoothing the reactive current vector field using a kernel density estimate. Specifically, to estimate the reactive current at a point s, we evaluate

$$\frac{\int e^{-|s-s'|^2/2\sigma^2} J^\theta_{AB}(s')ds'}{\int e^{-|s-s'|^2/2\sigma^2} ds'}, \tag{3.71}$$

where σ is chosen to be the smallest value where the vector field is smooth.

Using equations (3.65) and (3.66), we can express this estimator using a similar procedure as above for the rate as

$$\begin{aligned} J^\theta_{AB}(s) \approx \frac{1}{2\tau|ds|} \sum_i q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}(0)}) q_-(X^{(i)}_0) w(X^{(i)}_0) \\ \sum_{p=0}^{\frac{\tau\wedge S^+_{A\cup B}(0)}{\Delta}-1} (\theta(X^{(i)}_{(p+1)\Delta}) - \theta(X^{(i)}_{p\Delta})) \\ (\mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{(p+1)\Delta}) + \mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{p\Delta})) \\ + \frac{1}{2\tau|ds|} \sum_i \sum_{r=0}^{\frac{\tau}{\Delta}-1} q_+(X^{(i)}_{\tau\wedge S^+_{A\cup B}((r+1)\Delta)}) q_-(X^{(i)}_r) \mathbb{1}_{A\cup B}(X^{(i)}_{r\Delta}) w(X^{(i)}_0) \\ \sum_{p=r}^{\frac{\tau\wedge S^+_{A\cup B}((r+1)\Delta)}{\Delta}-1} (\theta(X^{(i)}_{(p+1)\Delta}) - \theta(X^{(i)}_{p\Delta})) \\ (\mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{(p+1)\Delta}) + \mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{p\Delta})). \end{aligned} \tag{3.72}$$

Likewise, by expressing the forward stopping time in terms of the backward stopping time as

$$\begin{aligned} f(X_{\tau\wedge S^+_{A\cup B}(s)}) &= \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\vee S^-_{A\cup B}(\tau)})f(X_\tau) + \mathbb{1}_{A\cup B}(X_s)f(X_s) \\ &\quad + \lim_{t\to 0}\frac{1}{t}\int_s^\tau \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\vee S^-_{A\cup B}(r-t)})\mathbb{1}_{A\cup B}(X_r)f(X_r)\,dr \\ &\approx \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\vee S^-_{A\cup B}(\tau)})f(X_\tau) + \mathbb{1}_{A\cup B}(X_s)f(X_s) \\ &\quad + \sum_{p=\frac{s}{\Delta}+1}^{\frac{\tau}{\Delta}} \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(X_{s\vee S^-_{A\cup B}((p-1)\Delta)})\mathbb{1}_{A\cup B}(X_{p\Delta})f(X_{p\Delta}) \end{aligned} \tag{3.73}$$

for $s \le \tau$ and using the identity

$$S^-_{A\cup B}(r) = \lim_{t\to 0} S^-_{A\cup B}(s-t) \approx S^-_{A\cup B}(s-\Delta) \text{ if } X_{s\vee S^-_{A\cup B}(r)} \notin A\cup B \text{ and } s \le r \tag{3.74}$$

we obtain

$$\begin{aligned} J^\theta_{AB}(s) &\approx \frac{1}{2\tau|ds|}\sum_i q_+(X^{(i)}_\tau)q_-(X^{(i)}_{0\vee S^-_{A\cup B}(\tau)})w(X^{(i)}_0) \\ &\qquad \sum_{p=\frac{0\vee S^-_{A\cup B}(\tau)}{\Delta}+1}^{\tau} (\theta(X^{(i)}_{p\Delta}) - \theta(X^{(i)}_{(p-1)\Delta})) \\ &\qquad\qquad (\mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{p\Delta}) + \mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{(p-1)\Delta})) \\ &\quad + \frac{1}{2\tau|ds|}\sum_i\sum_{r=1}^{\frac{\tau}{\Delta}} q_+(X^{(i)}_{r\Delta})\mathbb{1}_{A\cup B}(X^{(i)}_{r\Delta})q_-(X^{(i)}_{0\vee S^-_{A\cup B}((r-1)\Delta)})w(X^{(i)}_0) \\ &\qquad \sum_{p=\frac{0\vee S^-_{A\cup B}((r-1)\Delta)}{\Delta}+1}^{r} (\theta(X^{(i)}_{p\Delta}) - \theta(X^{(i)}_{(p-1)\Delta})) \\ &\qquad\qquad (\mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{p\Delta}) + \mathbb{1}_{\{\theta\in ds\}}(X^{(i)}_{(p-1)\Delta})). \end{aligned} \tag{3.75}$$

The reactive current estimator (3.37) is obtained by averaging the first term of equation (3.72) together with the first term of equation (3.75), after performing the approximations

$\mathbb{1}_{\{\theta \in ds\}}(X_s^{(i)}) \approx \mathbb{1}_{\{\theta \in ds\}}(X_0^{(i)})$ and $\mathbb{1}_{\{\theta \in ds\}}(X_s^{(i)}) \approx \mathbb{1}_{\{\theta \in ds\}}(X_\tau^{(i)})$ for the terms from equations (3.72) and (3.75), respectively. The terms that we gain here relative to (3.37) in the main text are similar in origin to those introduced into the reaction rate estimator in this section. For equation (3.72), they account for the portions of the trajectories which leave $A \cup B$ within τ; the same holds for equation (3.75), but for the time-reversed trajectory. In effect, equation (3.37) ignores half the contribution of the trajectories which hit $A \cup B$ and half the contribution of the time-reversed trajectories which hit $A \cup B$.

3.7.2 Supplemental Figures

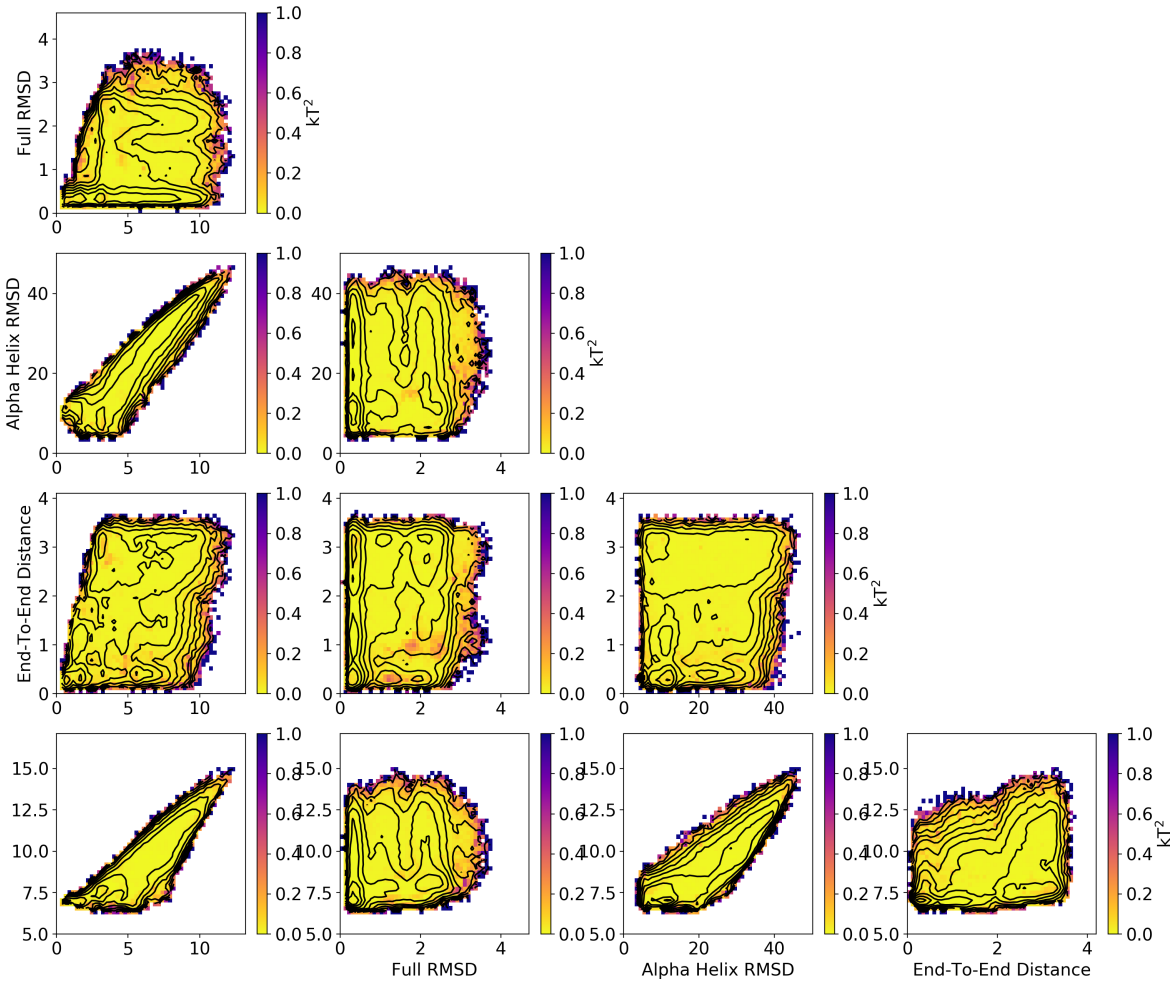

Figure 3.10: EMUS asymptotic variance for REUS PMFs.

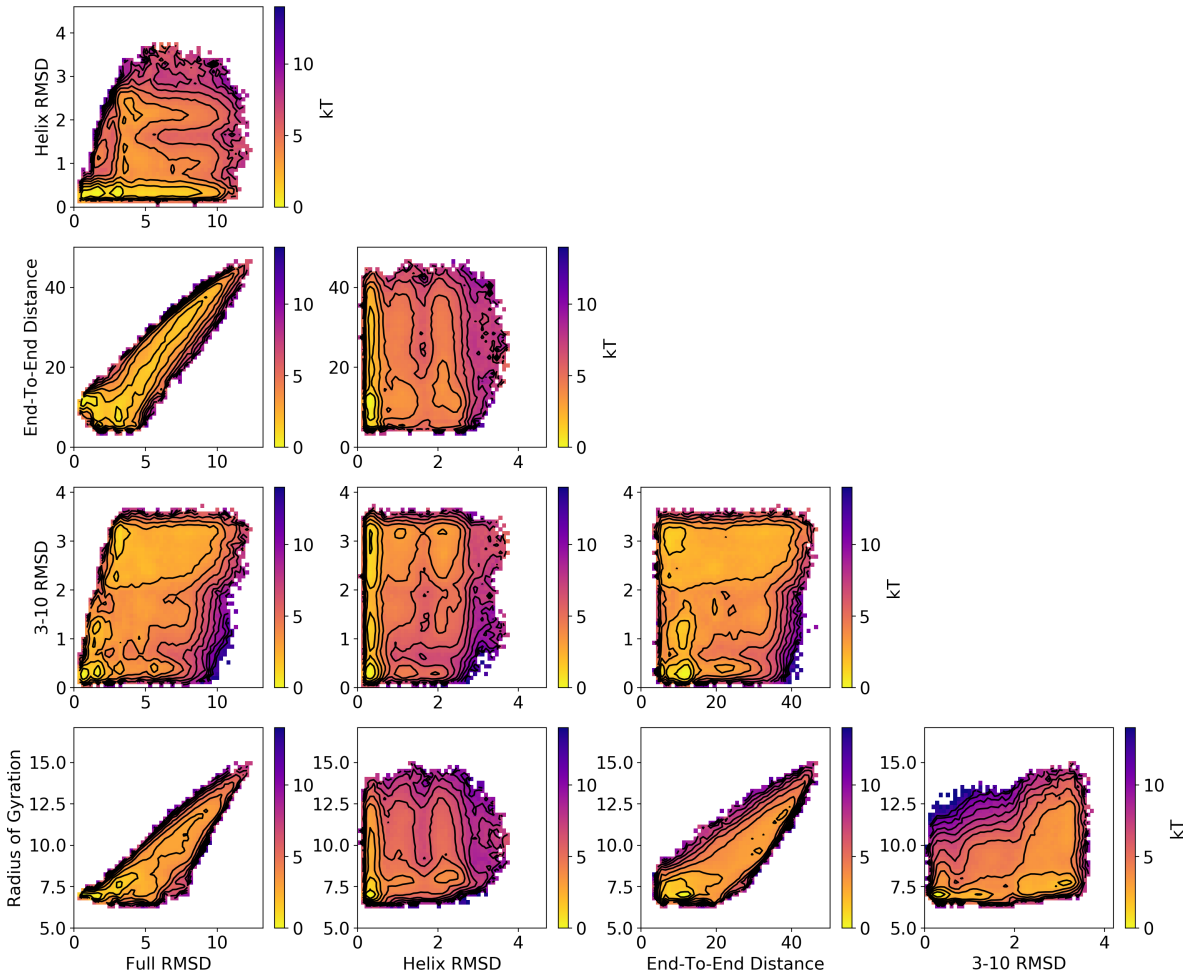

Figure 3.11: REUS PMFs.

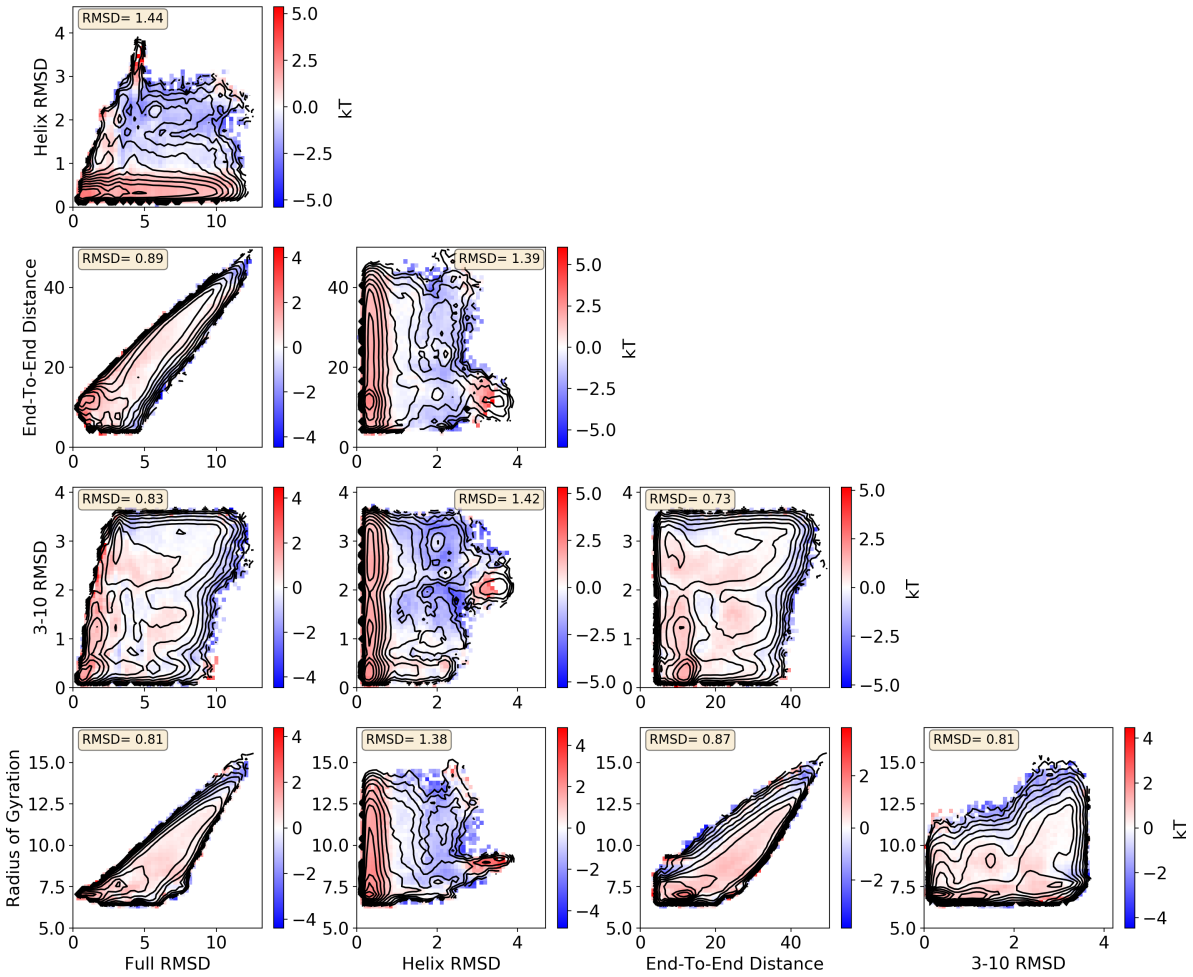

Figure 3.12: Difference between DGA with the modified distance basis set without the α helix resampling and REUS.

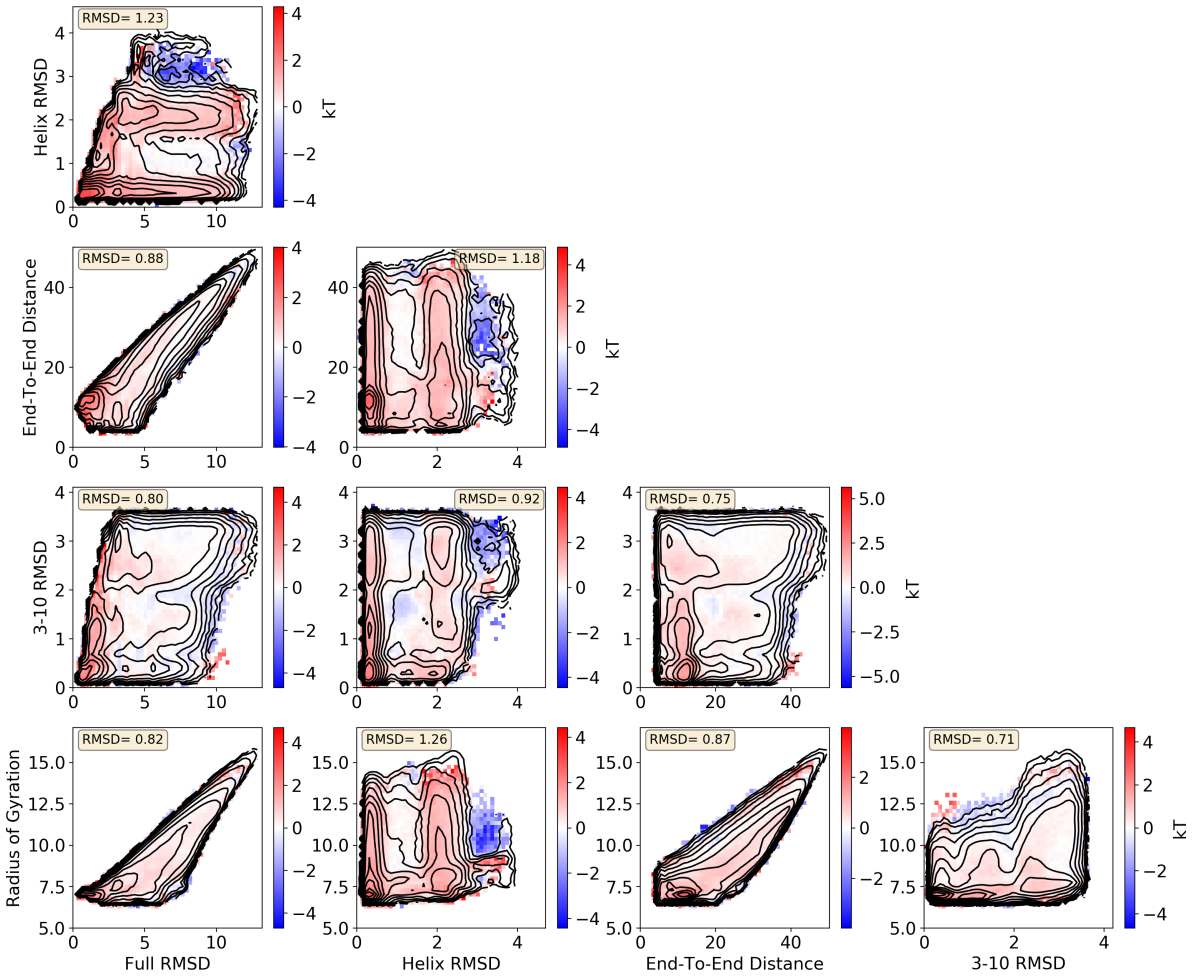

Figure 3.13: Difference between the PMF from DGA with the distance indicator basis set and the PMF from REUS.

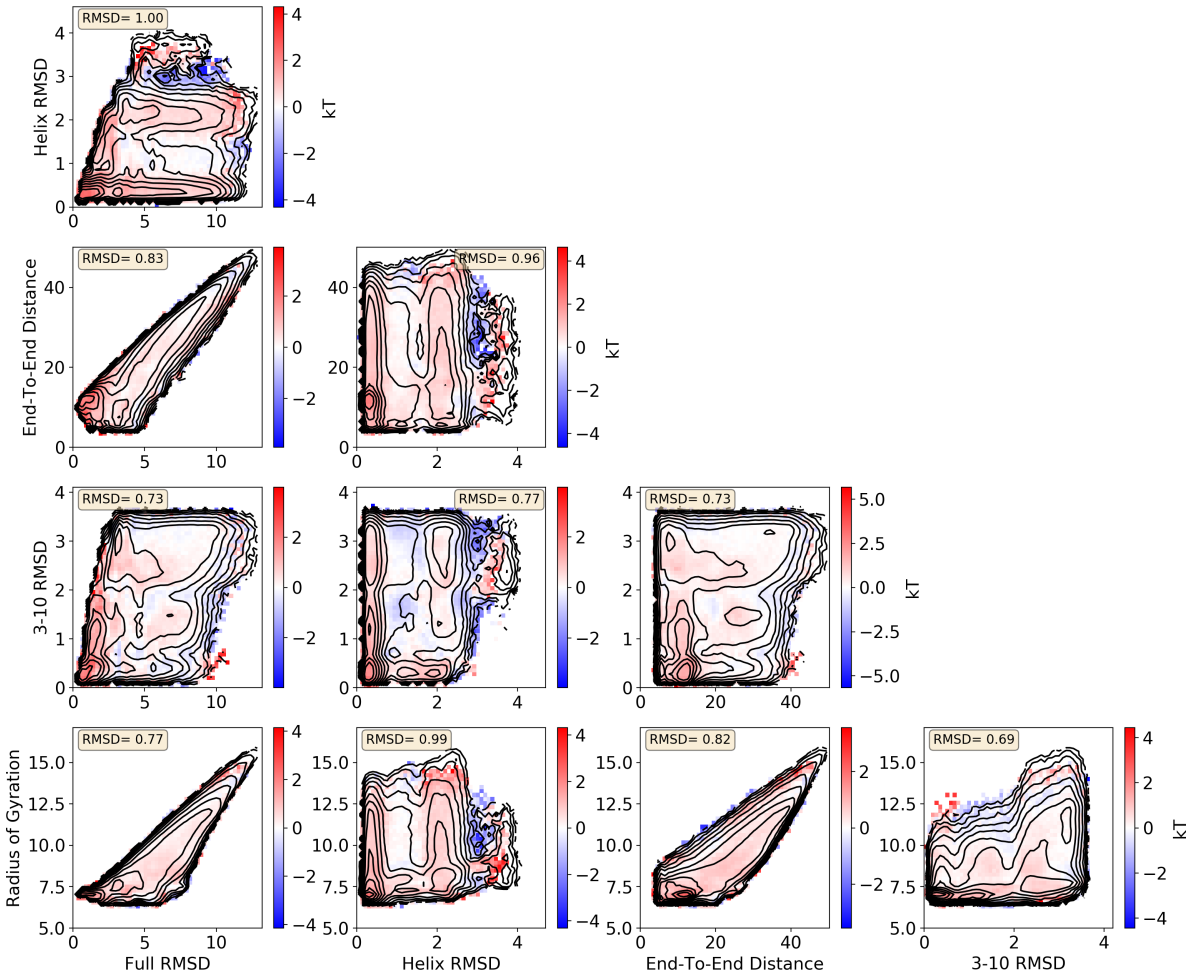

Figure 3.14: Difference between the PMF from DGA with the TICA indicator basis set and the PMF from REUS.

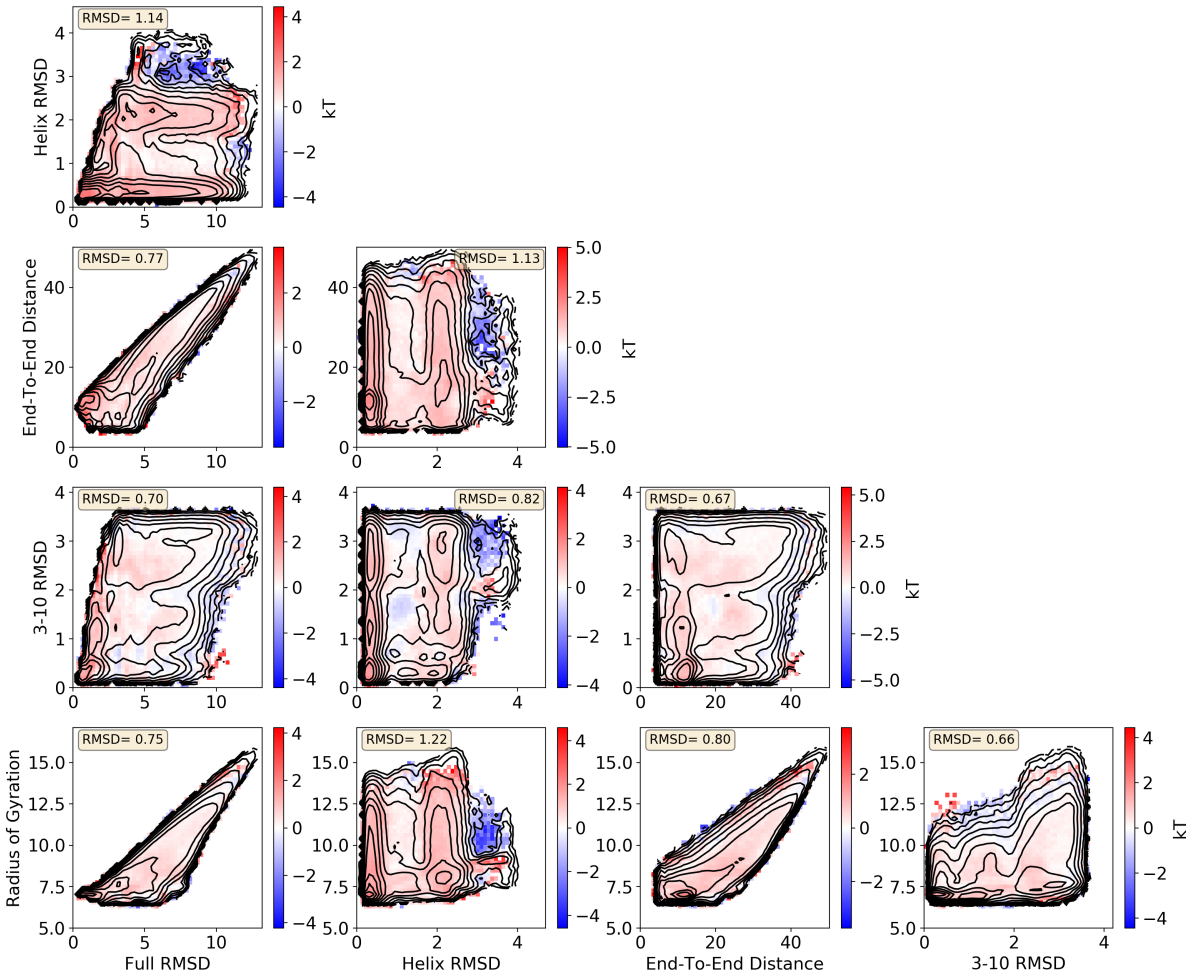

Figure 3.15: Difference between the PMF from DGA with the modified distance basis set and the PMF from REUS.

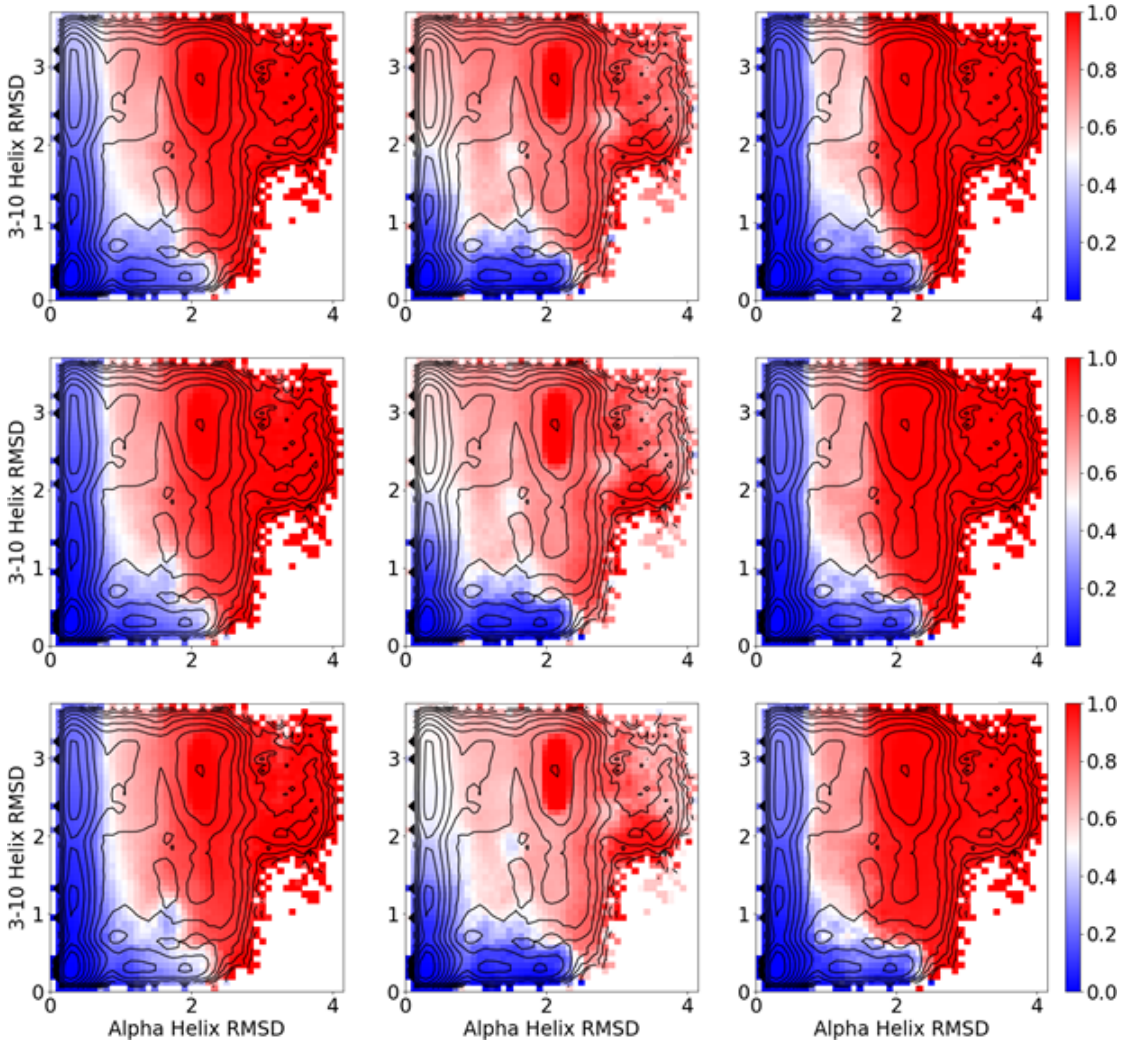

Figure 3.16: DGA backward committors. Left, middle, and right columns are computed with the modified distance, distance indicator, and TICA indicator basis sets, respectively. Top, middle, and bottom rows are computed with lag times of 0.5, 2.5, and 7.5 ns, respectively.

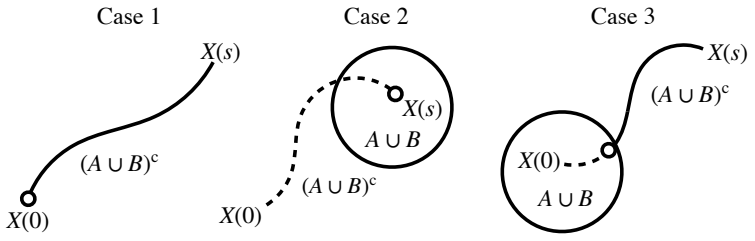

Figure 3.17: Case analysis for expressing the backward stopping time in terms of forward stopping times.

CHAPTER 4

DYNAMICAL GALERKIN APPROXIMATION WITH MEMORY

This chapter is adapted from C. Lorpaiboon et al., "Accurate estimates of dynamical statistics using memory", J. Chem. Phys. **160**, 084108 (2024), with the permission of AIP Publishing.

Many chemical reactions and molecular processes occur on timescales that are significantly longer than those accessible by direct simulation. One successful approach to estimating dynamical statistics for such processes is to use many short time series of observations of the system to construct a Markov state model (MSM), which approximates the dynamics of the system as memoryless transitions between a set of discrete states. The dynamical Galerkin approximation (DGA) is a closely related framework for estimating dynamical statistics, such as committors and mean first passage times, by approximating solutions to their equations with a projection onto a basis. Because the projected dynamics are generally not memoryless, the Markov approximation can result in significant systematic error. Inspired by quasi-Markov state models, which employ the generalized master equation to encode memory resulting from the projection, we reformulate DGA to account for memory and analyze its performance on two systems: a two-dimensional triple well and helix-to-helix transitions of the AIB_9 peptide. We demonstrate that our method is robust to the choice of basis and can decrease the time series length required to obtain accurate kinetics by an order of magnitude.

4.1 Introduction

Models that seek to treat complex systems with high fidelity typically have many variables (e.g., the positions and velocities of all the atoms in a molecular system). However, often the dynamics of interest of such models can be largely captured by a relatively small number of variables that are functions of the original ones (collective variables or CVs). Projecting the dynamics to these CVs can facilitate analysis, by both improving convergence of statistics

and simplifying interpretation. However, the coupling of the CVs to orthogonal ones generally introduces a form of memory in which the projected dynamics depend not only on the states of the CVs but also on their histories.

When the dynamics have a separation of time scales, it may be possible to find CVs that minimize the memory, so that it can be neglected. This is the idea behind Markov State Models (MSMs), in which the dynamics are treated as hops between discrete states, and large numbers of states are often used in the hope of minimizing the time for equilibration within states. However, not all dynamics have a clear separation of time scales, and, even if they do, finding CVs that minimize the memory can be challenging [89].

Thus various strategies have been introduced to account for memory. One is to consider explicit histories of the CVs, as in delay embedding [4, 5, 111–114]. However, this increases the effective number of variables, mitigating the advantages of projecting described above. An alternative is to account for the memory through the Mori–Zwanzig formalism [115, 116]. In this approach, the projected dynamics are represented through a generalized Langevin/master equation (GLE/GME) that involves a time integral over the CV histories and a kernel. This approach is physically well-motivated and can in principle yield highly accurate dynamics. In practice, it can be hard to determine the memory kernel.

Traditionally, researchers chose the memory kernel based on physical intuition [117, 118], and many studies still assume a specific functional form for the kernel (most often exponential) when estimating it from data. A few studies allow for more flexibility by introducing an expansion in basis functions [6, 119–121] or alternative fitting forms [122, 123] for the kernel or its Laplace/Fourier transforms [124–126], and these show that it does not have a simple universal functional form. However, solving for the coefficients of the expansion can be numerically ill-posed [127], and data-driven parameterization of GLEs/GMEs remains an active area of research [128–132].

Recently, approaching the problem from the perspective of improving the accuracy of

MSMs with relatively small numbers of states [7], Huang, Markland, Montoya-Castillo, and co-workers obtained the memory kernel at successive discrete time points by directly inverting a GME for the state-to-state transition matrix [7, 133, 134]. This enabled them to significantly shorten the lag times needed to construct the transition matrix, which reduced the amount of simulation needed to converge statistics and increased the time resolution of the model; the small number of states facilitates interpretation.

Like many MSM studies, the studies described immediately above focus on computing implied time scales (via time correlation functions rather than directly from the eigenvalues of the transition matrix). However, MSMs can be used to solve for statistics that provide more direct insight into the mechanisms of specified events, such as the committor [8], and we recently introduced a closely related framework that solves operator equations for such statistics, that we term dynamical Galerkin approximation (DGA) [4, 5]. The goal of this chapter is to show how memory can improve the accuracy of such statistics.

To this end, we incorporate memory into DGA by interpreting it as an iterative algorithm and applying a discrete-time Mori–Zwanzig formalism. We present formulas for computing the aforementioned statistics and numerically demonstrate that memory indeed improves results for a two-dimensional triple well and on the AIB_9 peptide.

4.2 Theory

In this section, we briefly review DGA as previously presented [4, 5]. We then recast it as an iteration to show that its solutions can be viewed as projections; the iteration and resulting projection operators lead naturally to a discrete-time GME and, in turn, an improved estimator. This estimator can be used without iteration.

4.2.1 Form of the problem

Let X_t denote a time-homogeneous ergodic Markov process at time t. Our goal is to compute a statistic u of this process that satisfies an equation of the form

$$\dot{\mathcal{A}}^0 u = -\dot{b}^0, \tag{4.1}$$

where $\dot{\mathcal{A}}^0$ is a linear operator that describes the time evolution of expectations of functions of X_t (akin to a time derivative), and $\dot{b}^0$ is the rate of change in u at X_t. We can interpret (4.1) as finding a time-invariant solution to the equation specifying the dynamics of u,

$$\frac{\mathrm{d}u_t}{\mathrm{d}t} = \dot{\mathcal{A}}^0 u_t + \dot{b}^0. \tag{4.2}$$

Integrating (4.2) over a time interval $\tau > 0$ that we call the lag time yields the Richardson iteration

$$u_t = u_{t-\tau} + (\mathcal{A}^\tau u_{t-\tau} + b^\tau), \tag{4.3}$$

where $\mathcal{A}^\tau = \int_0^\tau \mathcal{C}^{\tau'} \dot{\mathcal{A}}^0 \,\mathrm{d}\tau' = \mathcal{C}^\tau - \mathcal{I}$, $b^\tau = \int_0^\tau \mathcal{C}^{\tau'} \dot{b}^0 \,\mathrm{d}\tau'$, $\mathcal{C}^\tau = e^{\dot{\mathcal{A}}^0 \tau}$, and $\mathcal{I}$ is the identity operator. Eq. (4.3), as we discuss further in Section 4.2.3, corresponds to an iteration in u that is consistent with the dynamics in (4.2). The fixed point of the iteration is the finite time analogue of (4.1):

$$\mathcal{A}^\tau u = -b^\tau. \tag{4.4}$$

For example, the mean time to first enter the set B starting from state x, $u(x) := m(x) = \mathbb{E}[S_B^+(0) \mid X_0 = x]$, where $S_B^+(0) = \min\{t \geq 0 \mid X_t \in B\}$, satisfies [5]

$$(\mathcal{T}_B^\tau - \mathcal{I})m = -\int_0^\tau \mathcal{T}_B^t \mathbb{1}_{B^\mathsf{c}} \,\mathrm{d}t, \tag{4.5}$$

Algorithm 2 DGA

$G^\tau = \langle \phi, \mathcal{A}^\tau \phi^\mathsf{T} \rangle$
$h^\tau = \langle \phi, \mathcal{A}^\tau \hat{u}_0 + b^\tau \rangle$
$v = -(G^\tau)^{-1} h^\tau$
$\hat{u}^\tau = \hat{u}_0 + \phi^\mathsf{T} v$
$u^\tau = \hat{u}^\tau + (\mathcal{A}^\tau \hat{u}^\tau + b^\tau)$

subject to the boundary condition $m(x) = 0$ for $x \in B$. Here,

$$\mathcal{T}_B^\tau f(x) = \mathbb{E}[f(X_{\tau \wedge S_B^+(0)}) \mid X_0 = x] \tag{4.6}$$

is the stopped transition operator,

$$\int_0^\tau \mathcal{T}_B^t \mathbb{1}_{B^\mathsf{c}}(x)\, \mathrm{d}t = \mathbb{E}[\tau \wedge S_B^+(0) \mid X_0 = x], \tag{4.7}$$

$\tau \wedge S_B^+(0) = \min\{\tau, S_B^+(0)\}$, and $\mathbb{1}_{B^\mathsf{c}}(x)$ is an indicator function that equals one for $x \in B^\mathsf{c}$ and zero for $x \notin B^\mathsf{c}$. We define other statistics of interest in Section 4.3.

4.2.2 Dynamical Galerkin approximation (DGA)

DGA solves (4.4) by expanding u in a basis and estimating the resulting matrix elements by averages over random samples. We represent the basis by a vector ϕ of functions that satisfy homogeneous boundary conditions. Then we approximate u by

$$\hat{u} = \hat{u}_0 + \phi^\mathsf{T} v, \tag{4.8}$$

where $\hat{u}_0$ is a "guess function" for u that satisfies the boundary conditions, v is a vector of coefficients, and we use the symbol $\hat{}$ to denote functions that can be represented by (4.8). To solve for the coefficients, we substitute (4.8) into (4.4), multiply from the left by ϕ, and

integrate over an arbitrary distribution of samples of initial conditions μ. This yields

$$G^\tau v = -h^\tau, \tag{4.9}$$

where

$$G^\tau = \langle \phi, \mathcal{A}^\tau \phi^\mathsf{T} \rangle, \tag{4.10}$$

$$h^\tau = \langle \phi, \mathcal{A}^\tau \hat{u}_0 + b^\tau \rangle, \tag{4.11}$$

and $\langle f, g \rangle = \int f(x) g(x) \mu(x) \, \mathrm{d}x$. The matrix G^τ and vector h^τ are estimated from averages over trajectories of length τ; explicit expressions are given in Sec. 4.3. Given these estimates, (4.9) can be solved directly for v. DGA is summarized in Algorithm 2. We select τ by increasing its value until the estimate of u appears to converge.

4.2.3 Iterative solution

The form in (4.3) and our recent use of it in the context of inexact numerical linear algebra [135] suggests that one can view DGA as the fixed point of the *inexact* iteration

$$u_t^\tau = \hat{u}_{t-\tau}^\tau + (\mathcal{A}^\tau \hat{u}_{t-\tau}^\tau + b^\tau), \tag{4.12a}$$

$$\hat{u}_t^\tau = \hat{u}_{t-\tau}^\tau + \mathcal{P}(u_t^\tau - \hat{u}_{t-\tau}^\tau), \tag{4.12b}$$

where $\mathcal{P}$ is an operator that projects functions onto linear combinations of the basis functions:

$$\mathcal{P} f = \phi^\mathsf{T} (K^0)^{-1} \langle \phi, f \rangle, \tag{4.13}$$

and

$$K^t = \langle \phi, \mathcal{C}^t \phi^\mathsf{T} \rangle. \tag{4.14}$$

We use the superscript τ on u_t^τ to distinguish that it comes from the inexact iteration (4.12) involving the projected function $\hat{u}_t^\tau$ (and thus a layer of approximation) rather than the exact iteration (4.3). Fixed points are indicated without subscripts: $u^\tau = \lim_{t\to\infty} u_t^\tau$ and $\hat{u}^\tau = \lim_{t\to\infty} \hat{u}_t^\tau$.

We now identify the approximation associated with DGA by comparing the inexact iteration (4.12) with the *exact* iteration

$$u_t = u_{t-\tau} + (\mathcal{A}^\tau u_{t-\tau} + b^\tau), \tag{4.15a}$$

$$\hat{u}_t = \hat{u}_{t-\tau} + \mathcal{P}(u_t - \hat{u}_{t-\tau}). \tag{4.15b}$$

Let us define the complementary projection

$$\mathcal{Q} = \mathcal{I} - \mathcal{P}. \tag{4.16}$$

We note that $\mathcal{Q}(\hat{u}_t - \hat{u}_{t'}) = 0$ for any t and t' since $\mathcal{Q}\phi = 0$ and $\hat{u}_t - \hat{u}_{t'} = \phi^\mathsf{T} v$ for some v (we work with differences because we do not assume $\mathcal{Q}\hat{u}_0 = 0$). By substituting $\mathcal{P} = \mathcal{I} - \mathcal{Q}$ into (4.15b) and replacing t with $t - \tau$, we find that $u_{t-\tau} = \hat{u}_{t-\tau} + \mathcal{Q}(u_{t-\tau} - \hat{u}_{t-2\tau})$; in turn, by substituting this expression into (4.15a), we can express (4.15) as

$$u_t = \hat{u}_{t-\tau} + (\mathcal{A}^\tau \hat{u}_{t-\tau} + b^\tau) + \mathcal{C}^\tau \mathcal{Q}(u_{t-\tau} - \hat{u}_{t-2\tau}), \tag{4.17a}$$

$$\hat{u}_t = \hat{u}_{t-\tau} + \mathcal{P}(u_t - \hat{u}_{t-\tau}). \tag{4.17b}$$

The fixed point $\hat{u} = \lim_{t\to\infty} \hat{u}_t$ satisfies $\mathcal{P}(\mathcal{A}^\tau \hat{u} + b^\tau) + \mathcal{P}\mathcal{C}^\tau \mathcal{Q}(u - \hat{u}) = 0$. Comparing (4.12a) and (4.15a) shows that they both use the previous estimate ($\hat{u}_{t-\tau}^\tau$ or $u_{t-\tau}^\tau$, respectively) to construct an improved estimate u_t^τ. Because $\hat{u}_{t-\tau}^\tau$ can be represented using the basis, we only need length τ trajectories to approximate u_t^τ. Hence, DGA makes the approximation $\mathcal{P}\mathcal{C}^\tau \mathcal{Q}(u - \hat{u}) \approx 0$, which avoids the need for trajectories longer than τ. In contrast, the exact

iteration requires infinite-length trajectories since the last term of (4.17a) prevents $\hat{u}_t$ from depending on $\hat{u}_{t-\tau}$ alone.

To understand the need for memory, we observe that (4.15a) specifies Markovian dynamics for u_t: the value of u_t at current iteration t is fully determined by the value of $u_{t-\tau}$ at the last iteration $t-\tau$. DGA amounts to making the approximation that the dynamics of the *projected* function $\hat{u}_t$ observed at discrete time intervals τ is Markovian. This perspective suggests that we can derive an improved estimator by mitigating the Markov approximation. We do this by using projected statistics at multiple times (i.e., using memory), rather than a single time, as we now describe.

4.2.4 Iterative solution with memory

Our starting point is (4.17a) with τ replaced by σ, where τ/σ is a positive integer:

$$u_t = \hat{u}_{t-\sigma} + (\mathcal{A}^\sigma \hat{u}_{t-\sigma} + b^\sigma) + \mathcal{C}^\sigma \mathcal{Q}(\underline{u_{t-\sigma}} - \hat{u}_{t-2\sigma}). \tag{4.18}$$

The underlined term is not representable by the basis and prevents solving for $\hat{u}$ by fixed point iteration with length τ trajectories. By repeatedly substituting (4.18), we obtain

$$\begin{aligned} u_t &= \hat{u}_{t-\sigma} + \sum_{n=1}^{2} (\mathcal{C}^\sigma \mathcal{Q})^{n-1} (\mathcal{A}^\sigma \hat{u}_{t-n\sigma} + b^\sigma) + (\mathcal{C}^\sigma \mathcal{Q})^2 (\underline{u_{t-2\sigma}} - \hat{u}_{t-3\sigma}) \\ &\vdots \\ &= \hat{u}_{t-\sigma} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^\sigma \mathcal{Q})^{n-1} (\mathcal{A}^\sigma \hat{u}_{t-n\sigma} + b^\sigma) + (\mathcal{C}^\sigma \mathcal{Q})^{\tau/\sigma} (\underline{u_{t-\tau}} - \hat{u}_{t-\tau-\sigma}), \end{aligned} \tag{4.19}$$

where the underline indicates the location of the next substitution. We now have an alternative iteration to (4.17) that depends on the last τ/σ projections:

$$u_t = \hat{u}_{t-\sigma} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^\sigma \mathcal{Q})^{n-1} (\mathcal{A}^\sigma \hat{u}_{t-n\sigma} + b^\sigma) + (\mathcal{C}^\sigma \mathcal{Q})^{\tau/\sigma} (u_{t-\tau} - \hat{u}_{t-\tau-\sigma}), \tag{4.20a}$$

$$\hat{u}_t = \hat{u}_{t-\sigma} + \mathcal{P}(u_t - \hat{u}_{t-\sigma}). \tag{4.20b}$$

Eq. (4.20a) is similar to a discrete-time GME (e.g., Cao et al. [7]) or a discrete-time Mori–Zwanzig decomposition (e.g., Darve et al. [136] for eigendecomposition of the transition operator). It describes u_t at iteration t by its projection $\hat{u}_{t-\sigma}$ at iteration $t-\sigma$ (*Markov approximation*, first term and $n=1$ of the second term), the projections $\hat{u}_{t-n\sigma}$ at previous iterations $t-n\sigma$ (*memory*, $n>1$ of the second term), and the complementary projection $\mathcal{Q}u_{t-\tau}$ at time $t-\tau$ (*orthogonal dynamics*, last term). The approximation used in DGA is the first two terms of (4.20a) with $\sigma=\tau$ (i.e., the previous solution and the first term of the sum). The remaining terms correct for the Markov approximation by introducing memory.

Omitting the last term of (4.20a) yields a computable iteration

$$u_t^{\sigma,\tau} = \hat{u}_{t-\sigma}^{\sigma,\tau} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^\sigma \mathcal{Q})^{n-1} (\mathcal{A}^\sigma \hat{u}_{t-n\sigma}^{\sigma,\tau} + b^\sigma), \tag{4.21a}$$

$$\hat{u}_t^{\sigma,\tau} = \hat{u}_{t-\sigma}^{\sigma,\tau} + \mathcal{P}(u_t^{\sigma,\tau} - \hat{u}_{t-\sigma}^{\sigma,\tau}), \tag{4.21b}$$

with a fixed point that solves

$$\sum_{n=1}^{\tau/\sigma} \mathcal{P}(\mathcal{C}^\sigma \mathcal{Q})^{n-1} (\mathcal{A}^\sigma \hat{u}^{\sigma,\tau} + b^\sigma) = 0. \tag{4.22}$$

Comparing (4.20) and (4.22), we see that the latter corresponds to making the approximation $\mathcal{P}(\mathcal{C}^\sigma \mathcal{Q})^{\tau/\sigma}(u - \hat{u}) \approx 0$. When the basis captures much of the slower dynamics of the system, we expect the DGA with memory approximation $\mathcal{P}(\mathcal{C}^\sigma \mathcal{Q})^{\tau/\sigma}(u - \hat{u}) \approx 0$ to be

Algorithm 3 DGA with memory

for $n \in \{1, \ldots, \tau/\sigma\}$ **do**
$K^{n\sigma} = \langle \phi, \mathcal{C}^{n\sigma} \phi^{\mathsf{T}} \rangle$
$G^{n\sigma} = \langle \phi, \mathcal{A}^{n\sigma} \phi^{\mathsf{T}} \rangle$
$h^{n\sigma} = \langle \phi, \mathcal{A}^{n\sigma} \hat{u}_0 + b^{n\sigma} \rangle$
$G^{\sigma,n\sigma} = G^{n\sigma} - \sum_{n'=1}^{n-1} K^{(n-n')\sigma} (K^0)^{-1} G^{\sigma,n'\sigma}$
$h^{\sigma,n\sigma} = h^{n\sigma} - \sum_{n'=1}^{n-1} K^{(n-n')\sigma} (K^0)^{-1} h^{\sigma,n'\sigma}$
end for
$v = -(G^{\sigma,\tau})^{-1} h^{\sigma,\tau}$
$\hat{u}^{\sigma,\tau} = \hat{u}_0 + \phi^{\mathsf{T}} v$
$u^{\sigma,\tau} = \hat{u}^{\sigma,\tau} + (\mathcal{A}^{\tau} \hat{u}^{\sigma,\tau} + b^{\tau}) - \sum_{n=1}^{\tau/\sigma} \mathcal{C}^{\tau - n\sigma} \phi^{\mathsf{T}} (K^0)^{-1} (G^{\sigma,n\sigma} v + h^{\sigma,n\sigma})$

better than the DGA approximation $\mathcal{P}\mathcal{C}^{\tau}\mathcal{Q}(u - \hat{u}) \approx 0$ because the additional applications of $\mathcal{Q}$ remove these slower dynamics. We show this numerically in Sec. 4.4.

4.2.5 DGA with memory

In this section, we derive and summarize the algorithm (Algorithm 3). As in (4.8), we expand $\hat{u}^{\sigma,\tau}$ in a basis. Substituting $\hat{u}^{\sigma,\tau} = \hat{u}_0 + \phi^{\mathsf{T}} v$ into (4.22) and applying $\langle \phi_i, \mathcal{P} f \rangle = \langle \phi_i, f \rangle$ leads to the linear system

$$G^{\sigma,\tau} v = -h^{\sigma,\tau}, \tag{4.23}$$

where

$$G^{\sigma,\tau} = \sum_{n=1}^{\tau/\sigma} \langle \phi, (\mathcal{C}^{\sigma}\mathcal{Q})^{n-1} \mathcal{A}^{\sigma} \phi^{\mathsf{T}} \rangle, \tag{4.24}$$

$$h^{\sigma,\tau} = \sum_{n=1}^{\tau/\sigma} \langle \phi, (\mathcal{C}^{\sigma}\mathcal{Q})^{n-1} (\mathcal{A}^{\sigma} \hat{u}_0 + b^{\sigma}) \rangle. \tag{4.25}$$

We can calculate matrices $G^{\sigma,\tau}$ and $h^{\sigma,\tau}$ by expanding each $\mathcal{Q}$ in the expectations as $\langle f, \mathcal{Q} g \rangle = \langle f, g \rangle - \langle f, \mathcal{P} g \rangle = \langle f, g \rangle - \langle f, \phi^{\mathsf{T}} \rangle (K^0)^{-1} \langle \phi, g \rangle$ until none remain. For instance, to

calculate the term $h^{\sigma,3\sigma}$, we expand:

$$\begin{aligned}
h^{\sigma,3\sigma} &= \langle \phi, b^\sigma + \mathcal{C}^\sigma \mathcal{Q} b^\sigma + (\mathcal{C}^\sigma \mathcal{Q})^2 b^\sigma \rangle \\
&= \langle \phi, b^\sigma + \mathcal{C}^\sigma b^\sigma + \mathcal{C}^{2\sigma} b^\sigma \rangle - \langle \phi, \mathcal{C}^\sigma \mathcal{P}(b^\sigma + \mathcal{C}^\sigma \mathcal{Q} b^\sigma) \rangle - \langle \phi, \mathcal{C}^{2\sigma} \mathcal{P} b^\sigma \rangle \\
&= h^{3\sigma} - K^\sigma (K^0)^{-1} \underline{h^{\sigma,2\sigma}} - K^{2\sigma} (K^0)^{-1} h^\sigma, \\
h^{\sigma,2\sigma} &= \langle \phi, b^\sigma + \mathcal{C}^\sigma \mathcal{Q} b^\sigma \rangle \\
&= \langle \phi, b^\sigma + \mathcal{C}^\sigma b^\sigma \rangle - \langle \phi, \mathcal{C}^\sigma \mathcal{P} b^\sigma \rangle \\
&= h^{2\sigma} - K^\sigma (K^0)^{-1} h^\sigma.
\end{aligned}$$

The underlined expectations contain $\mathcal{Q}$ and must be expanded in terms of matrices that do not contain $\mathcal{Q}$, which can be directly evaluated as averages over random samples. We have rearranged and grouped terms for numerical stability because $G^{\sigma,\tau}$ and $h^{\sigma,\tau}$ can be very sensitive to how they are computed. Using this approach, we can derive the iterative formulas

$$G^{\sigma,\tau} = G^\tau - \sum_{n=1}^{(\tau/\sigma)-1} K^{\tau-n\sigma} (K^0)^{-1} G^{\sigma,n\sigma}, \tag{4.26a}$$

$$h^{\sigma,\tau} = h^\tau - \sum_{n=1}^{(\tau/\sigma)-1} K^{\tau-n\sigma} (K^0)^{-1} h^{\sigma,n\sigma}. \tag{4.26b}$$

A similar iteration was used to calculate the discretized memory kernel in Cao et al. [7] and the transfer tensors in Cerrillo and Cao [137].

After we solve (4.23), we can compute a stochastic approximation of u:

$$u^{\sigma,\tau} = \hat{u}^{\sigma,\tau} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^\sigma \mathcal{Q})^{n-1} (\mathcal{A}^\sigma \hat{u}^{\sigma,\tau} + b^\sigma), \tag{4.27}$$

which is the fixed point of (4.21a). In this case, we expand each $\mathcal{Q} f = f - \phi^{\mathsf{T}} (K^0)^{-1} \langle \phi, f \rangle$

from left to right until only $\mathcal{Q}$s inside expectations remain (which can be grouped into matrices $G^{\sigma,n\sigma}$ and $h^{\sigma,n\sigma}$):

$$u^{\sigma,\tau} = \hat{u}^{\sigma,\tau} + (\mathcal{A}^\tau \hat{u}^{\sigma,\tau} + b^\tau) - \sum_{n=1}^{\tau/\sigma} \mathcal{C}^{\tau-n\sigma} \delta u_n^{\sigma,\tau}, \tag{4.28}$$

where $\delta u_n^{\sigma,\tau} = \phi^{\mathsf{T}}(K^0)^{-1}(G^{\sigma,n\sigma} v + h^{\sigma,n\sigma})$. As in (4.26), we have combined terms in (4.28) for numerical stability.

We select σ and τ by increasing τ with τ/σ fixed until a statistic appears to converge; we try this for a few small integer values of τ/σ. As we show numerically in Sec. 4.4, only a few correction terms (i.e., small τ/σ) are generally needed to obtain most of the benefit of including memory.

4.3 Dynamical statistics

In this section, we introduce the statistics that we want to compute and the equations that they solve. We describe a variety of statistics beyond the MFPT that can be solved using DGA or DGA with memory. Provided that a statistic can be written in the form (4.4), we can solve for it with DGA or DGA with memory by evaluating (4.9) or (4.23), respectively, with appropriate K^t, G^t, and h^t matrices.

4.3.1 Stationary distribution

The stationary distribution π is invariant with respect to time translation and therefore

$$\int \pi(x) f(x)\, \mathrm{d}x = \int \pi(x) \mathcal{T}^t f(x)\, \mathrm{d}x \tag{4.29}$$

for all functions f, where the transition operator

$$\mathcal{T}^t f(x) = \mathbb{E}[f(X_t) \mid X_0 = x] \tag{4.30}$$

propagates expectations of functions f forward-in-time. The sampling distribution μ can be reweighted to obtain the stationary distribution π via $w = \pi/\mu$ [5, 32], which satisfies an operator equation in the form of (4.4):

$$((\mathcal{T}^t)^\dagger - \mathcal{I})w = 0, \tag{4.31}$$

where $(\mathcal{T}^t)^\dagger$ denotes the adjoint of $\mathcal{T}^t$ with respect to μ: $\langle f, (\mathcal{T}^t)^\dagger g\rangle = \langle g, \mathcal{T}^t f\rangle$ for all functions f and g. We expand $\hat{w} = \hat{w}_0 + \phi^\mathsf{T} v$, where $\int \hat{w}_0(x)\mu(x)\,\mathrm{d}x = 1$ and $\int \phi(x)\mu(x)\,\mathrm{d}x = 0$; these conditions ensure that the resulting estimates of $\hat{w}$ and w are normalized. The corresponding matrices are

$$\begin{aligned}
K^t &= \langle \phi, (\mathcal{T}^t)^\dagger \phi^\mathsf{T}\rangle \\
&= \mathbb{E}[\phi(X_t)\phi^\mathsf{T}(X_0) \mid X_0 \sim \mu], && \text{(4.32a)} \\
G^t &= \langle \phi, ((\mathcal{T}^t)^\dagger - \mathcal{I})\phi^\mathsf{T}\rangle \\
&= \mathbb{E}[(\phi(X_t) - \phi(X_0))\phi^\mathsf{T}(X_0) \mid X_0 \sim \mu], && \text{(4.32b)} \\
h^t &= \langle \phi, ((\mathcal{T}^t)^\dagger - \mathcal{I})\hat{w}_0\rangle \\
&= \mathbb{E}[(\phi(X_t) - \phi(X_0))\hat{w}_0(X_0) \mid X_0 \sim \mu], && \text{(4.32c)}
\end{aligned}$$

and we approximate expectations of functions f with respect to π as

$$\langle f, w^{\sigma,\tau}\rangle = \mathbb{E}\left[f(X_\tau)\hat{w}^{\sigma,\tau}(X_0) - \sum_{n=1}^{\tau/\sigma} f(X_{\tau-n\sigma})\delta w_n^{\sigma,\tau}(X_0) \;\middle|\; X_0 \sim \mu\right]. \tag{4.33}$$

4.3.2 Mean first passage time

The mean first passage time (MFPT)

$$m(x) = \mathbb{E}[S_B^+(0) \mid X_0 = x] \tag{4.34}$$

is the expected time until a trajectory starting at x first enters B. As noted above, it satisfies an equation of the form (4.4):

$$(\mathcal{T}_B^t - \mathcal{I})m = -\int_0^t \mathcal{T}_B^{t'} \mathbb{1}_{B^\mathsf{c}} \, \mathrm{d}t', \tag{4.35}$$

subject to the boundary condition $m(x) = 0$ for $x \in B$ (cf. (4.5)–(4.7)). Physically, this equation states that applying $\mathcal{T}_B^t$ to $m(x)$ reduces the mean remaining time to enter by $\mathbb{E}[t \wedge S_B^+(0) \mid X_0 = x]$. The corresponding matrices are

$$\begin{aligned}
K^t &= \langle \phi, \mathcal{T}_B^t \phi^\mathsf{T} \rangle \\
&= \mathbb{E}[\phi(X_0)\phi^\mathsf{T}(X_{t \wedge S_B^+(0)}) \mid X_0 \sim \mu], && \text{(4.36a)} \\
G^t &= \langle \phi, (\mathcal{T}_B^t - \mathcal{I})\phi^\mathsf{T} \rangle \\
&= \mathbb{E}[\phi(X_0)(\phi^\mathsf{T}(X_{t \wedge S_B^+(0)}) - \phi^\mathsf{T}(X_0)) \mid X_0 \sim \mu], && \text{(4.36b)} \\
h^t &= \left\langle \phi, (\mathcal{T}_B^t - \mathcal{I})\hat{m}_0 + \int_0^t \mathcal{T}_B^{t'} \mathbb{1}_{B^\mathsf{c}} \, \mathrm{d}t' \right\rangle \\
&= \mathbb{E}[\phi(X_0)(\hat{m}_0(X_{t \wedge S_B^+(0)}) - \hat{m}_0(X_0) + (t \wedge S_B^+(0))) \mid X_0 \sim \mu], && \text{(4.36c)}
\end{aligned}$$

and m can be approximated from its projection using

$$w^{\sigma,\tau}(x) = \mathbb{E}\left[\hat{m}^{\sigma,\tau}(X_{\tau \wedge S_B^+(0)}) + (\tau \wedge S_B^+(0)) - \sum_{n=1}^{\tau/\sigma} \delta m_n^{\sigma,\tau}(X_{(\tau - n\sigma) \wedge S_B^+(0)}) \;\middle|\; X_0 = x\right]. \tag{4.37}$$

4.3.3 Forward committor

The forward committor

$$q_+(x) = \mathbb{E}[\mathbb{1}_B(X_{S^+_{A\cup B}(0)}) \mid X_0 = x] \tag{4.38}$$

is the probability that a trajectory starting at x will enter B before A. The forward committor satisfies

$$(\mathcal{T}^t_{A\cup B} - \mathcal{I})q_+ = 0 \tag{4.39}$$

with boundary condition $q_+(x) = \mathbb{1}_B(x)$ for $x \in A \cup B$. The corresponding matrices are

$$\begin{aligned}
K^t &= \langle \phi, \mathcal{T}^t_{A\cup B}\phi^{\mathsf{T}}\rangle \\
&= \mathbb{E}[\phi(X_0)\phi(X_{t\wedge S^+_{A\cup B}(0)}) \mid X_0 \sim \mu], && \text{(4.40a)} \\
G^t &= \langle \phi, (\mathcal{T}^t_{A\cup B} - \mathcal{I})\phi^{\mathsf{T}}\rangle \\
&= \mathbb{E}[\phi(X_0)(\phi(X_{t\wedge S^+_{A\cup B}(0)}) - \phi(X_0)) \mid X_0 \sim \mu], && \text{(4.40b)} \\
h^t &= \langle \phi, (\mathcal{T}^t_{A\cup B} - \mathcal{I})\hat{q}_{+,0}\rangle \\
&= \mathbb{E}[\phi(X_0)(\hat{q}_{+,0}(X_{t\wedge S^+_{A\cup B}(0)}) - \hat{q}_{+,0}(X_0)) \mid X_0 \sim \mu], && \text{(4.40c)}
\end{aligned}$$

and q_+ can be approximated as

$$q^{\sigma,\tau}_+(x) = \mathbb{E}\left[\hat{q}^{\sigma,\tau}_+(X_{\tau\wedge S^+_{A\cup B}(0)}) - \sum_{n=1}^{\tau/\sigma} \delta q^{\sigma,\tau}_{+,n}(X_{(\tau-n\sigma)\wedge S^+_{A\cup B}(0)}) \;\middle|\; X_0 = x\right]. \tag{4.41}$$

4.3.4 Backward committor

The backward committor is the probability that a trajectory ending at x exited A after B,

$$q_-(x) = \mathbb{E}[\mathbb{1}_A(X_{S^-_{A\cup B}(0)}) \mid X_0 = x], \tag{4.42}$$

where $S^-_{A\cup B}(0) = \min\{t \le 0 \mid X_t \in A \cup B\}$. The backward committor satisfies

$$(\mathcal{T}^{-t}_{A\cup B} - \mathcal{I})q_- = 0 \tag{4.43}$$

with boundary condition $q_-(x) = \mathbb{1}_A(x)$ for $x \in A \cup B$.

$$\mathcal{T}^{-t}_{A\cup B} f(x) = \mathbb{E}[f(X_{(-t)\vee S^-_{A\cup B}(0)}) \mid X_0 = x] \tag{4.44}$$

is the stopped transition operator for the time-reversed process. We take conditional expectations backward-in-time with respect to the *time-reversed process*: given an infinite, statistically stationary trajectory X, we look backward in time from each time t where $X_t = x$. Mathematically,

$$\begin{aligned}&\mathbb{E}[f(X_{(-t)\vee S^-_{A\cup B}(0)}) \mid X_0 = x] \\ &\quad = \int \mathbb{E}[f(X_{(-t)\vee S^-_{A\cup B}(0)}) \mid X_0 = x, X_{-\tau} = x']\pi(x')\,\mathrm{d}x',\end{aligned} \tag{4.45}$$

which follows from the definition of conditional probability and $\pi(x) = \mathbb{P}[X_{-\tau} = x]$. Because $\pi(x) = w(x)\mu(x) = \mathbb{P}[X_0 = x] = \mathbb{P}[X_{-\tau} = x]$,

$$\begin{aligned}\langle g, \mathcal{T}^{-t}_{A\cup B} f\rangle &= \int \mathbb{E}[g(X_0)f(X_{(-t)\vee S^-_{A\cup B}(0)}) \mid X_0 = x]\mu(x)\,\mathrm{d}x \\ &= \mathbb{E}\left[g(X_0)f(X_{(-t)\vee S^-_{A\cup B}(0)})\frac{w(X_{-\tau})}{w(X_0)} \;\middle|\; X_{-\tau} \sim \mu\right],\end{aligned} \tag{4.46}$$

as we show explicitly in Strahan et al. [5] (with the zero of time shifted). The corresponding matrices are

$$\begin{aligned} K^t &= \langle \phi, \mathcal{T}_{A\cup B}^{-t} \phi^{\mathsf{T}} \rangle \\ &= \mathbb{E}\left[\phi(X_0) \phi(X_{(-t)\vee S^-_{A\cup B}(0)}) \frac{w(X_{-\tau})}{w(X_0)} \;\middle|\; X_{-\tau} \sim \mu \right], \end{aligned} \tag{4.47a}$$

$$\begin{aligned} G^t &= \langle \phi, (\mathcal{T}_{A\cup B}^{-t} - \mathcal{I}) \phi^{\mathsf{T}} \rangle \\ &= \mathbb{E}\left[\phi(X_0) (\phi(X_{(-t)\vee S^-_{A\cup B}(0)}) - \phi(X_0)) \frac{w(X_{-\tau})}{w(X_0)} \;\middle|\; X_{-\tau} \sim \mu \right], \end{aligned} \tag{4.47b}$$

$$\begin{aligned} h^t &= \langle \phi, (\mathcal{T}_{A\cup B}^{-t} - \mathcal{I}) \hat{q}_{-,0} \rangle \\ &= \mathbb{E}\left[\phi(X_0) (\hat{q}_{-,0}(X_{(-t)\vee S^-_{A\cup B}(0)}) - \hat{q}_{-,0}(X_0)) \frac{w(X_{-\tau})}{w(X_0)} \;\middle|\; X_{-\tau} \sim \mu \right]. \end{aligned} \tag{4.47c}$$

We estimate q_- as

$$q_-^{\sigma,\tau}(x) = \mathbb{E}\left[\hat{q}_-^{\sigma,\tau}(X_{(-\tau)\vee S^-_{A\cup B}(0)}) - \sum_{n=1}^{\tau/\sigma} \delta q_{-,n}^{\sigma,\tau}(X_{(-\tau+n\sigma)\vee S^-_{A\cup B}(0)}) \;\middle|\; X_0 = x \right]. \tag{4.48}$$

4.4 Numerical examples

In this section, we demonstrate our method on two systems. We first illustrate our method with a two-dimensional triple-well potential, for which an exact reference can be calculated using finite-differences. We next test our method on a more complex system, AIB_9, for which we have long, unbiased trajectories as a reference.

4.4.1 Two-dimensional triple-well

In this section, we illustrate DGA with memory on a simple model system. The system is a drift-diffusion process obeying the Fokker–Planck equation

$$\partial_t p_t(x) = \nabla \cdot (p_t(x) \nabla V(x)) + \beta^{-1} \nabla^2 p_t(x) \tag{4.49}$$

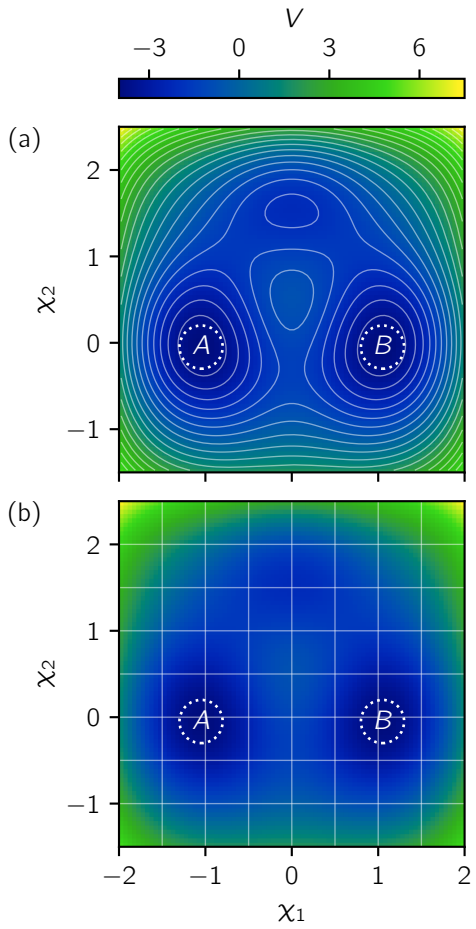

Figure 4.1: Potential energy surface of the two-dimensional triple-well. The boundaries of sets A and B are indicated using dotted lines. (a) Contours are drawn every $0.5\beta^{-1}$. (b) 8×8 grid basis used for DGA and DGA with memory.

where β is the inverse temperature, which we set to $\beta = 2$, $x = (\chi_1, \chi_2)$, and V is the potential [138]

$$\begin{aligned} V(\chi_1, \chi_2) = 3e^{-\chi_1^2-(\chi_2-1/3)^2} - 3e^{-\chi_1^2-(\chi_2-5/3)^2} \\ - 5e^{-(\chi_1-1)^2-\chi_2^2} - 5e^{-(\chi_1+1)^2-\chi_2^2} \\ + 0.2\chi_1^4 + 0.2(\chi_2 - 1/3)^4. \end{aligned} \tag{4.50}$$

We define the reactant state A and product state B as circles with radii 0.25 centered at $(\pm 1.05, -0.05)$, respectively (Fig. 4.1a).

We discretize the system on a 80×80 grid of equally-spaced points (with spacing $h = 0.05$)

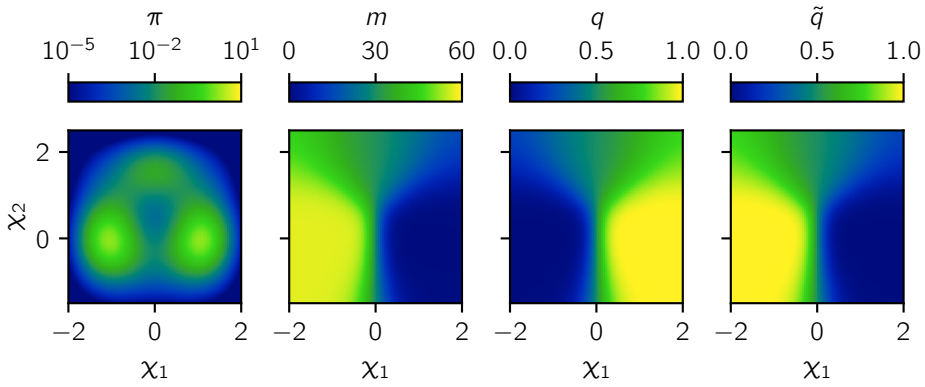

Figure 4.2: Reference statistics for the two-dimensional triple-well.

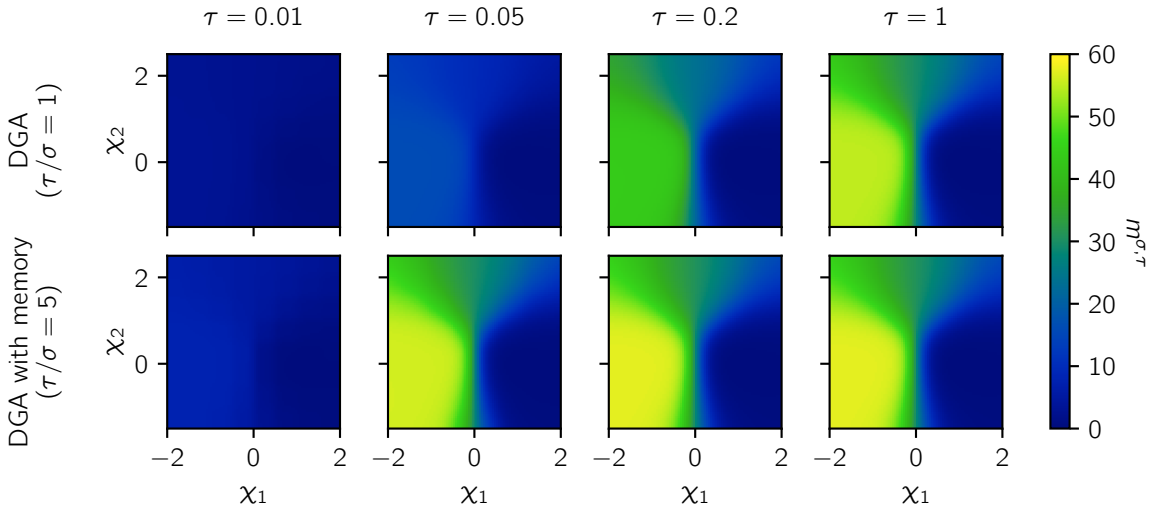

Figure 4.3: Mean first passage time for the two-dimensional triple-well, estimated using DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$). DGA with memory requires shorter lag times to achieve comparable results to DGA.

in the range $[-2, 2] \times [-1.5, 2.5]$. The dynamics of this discretized system follow the master equation [4, 135]

$$\partial_t p_t(x) = \sum_{x'} p_t(x') \mathcal{L}(x', x), \tag{4.51}$$

where the elements of the matrix $\mathcal{L}$ are

$$\mathcal{L}(x, x') = \frac{2\beta^{-1}}{h^2} \frac{1}{1 + e^{-\beta(V(x) - V(x'))}} \tag{4.52}$$

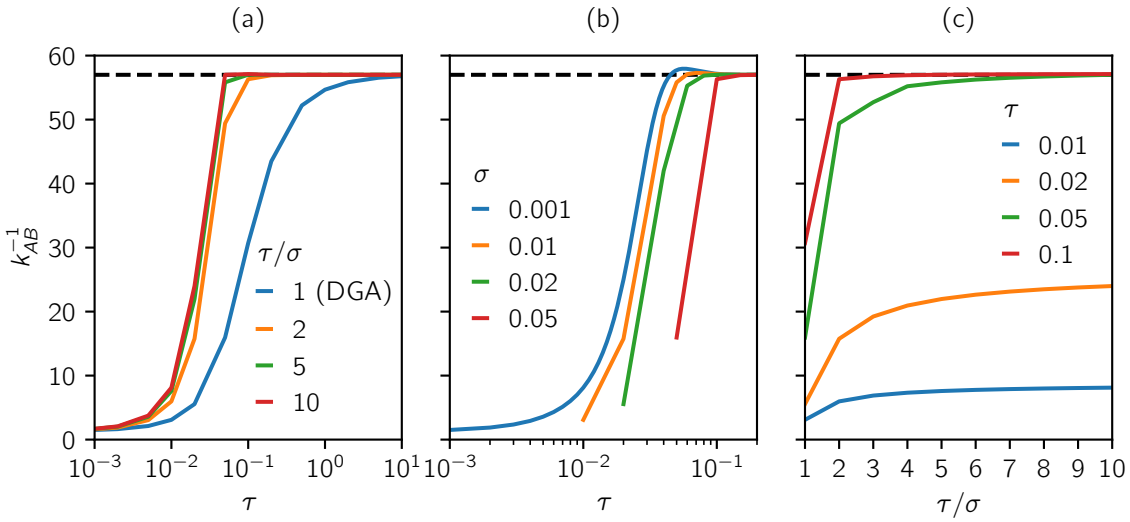

Figure 4.4: Dependence of the estimated inverse rate constant k_{AB}^{-1} for the two-dimensional triple-well on (a) τ with a fixed number of memory terms, (b) τ with fixed σ, and (c) the number of memory terms with fixed τ. The true value of the inverse rate constant is indicated by the dashed line.

for $x - x' \in \{(0, \pm h), (\pm h, 0)\}$, $\mathcal{L}(x, x) = -\sum_{x' \neq x} \mathcal{L}(x, x')$, and $\mathcal{L}(x, x') = 0$ otherwise. We use $\mathcal{L}$ to calculate finite-time operators such as $(\mathcal{T}^t)^\dagger$ and $\mathcal{T}_B^t$. These expressions are given in Appendix 4.7. We then directly compute the DGA matrices by taking inner products. For example, we compute (4.32a) as

$$\langle \phi, (\mathcal{T}^t)^\dagger \phi^{\mathsf{T}} \rangle = \sum_{x,x'} \mu(x)\phi(x)(\mathcal{T}^t)^\dagger(x, x')\phi^{\mathsf{T}}(x'),$$

where $(\mathcal{T}^t)^\dagger f(x) = \sum_{x'} (\mathcal{T}^t)^\dagger(x, x') f(x')$.

We calculate reference statistics by directly solving (4.31), (4.35), (4.39), and (4.43). For the DGA calculations, we use a basis of 64 indicator functions on an equally-spaced 8×8 grid in the range $[-2, 2] \times [-1.5, 2.5]$, as shown in Fig. 4.1b. Operationally, we first calculate the stationary distribution π using (4.33) with an arbitrary distribution of initial conditions and $\hat{w}_0 = 1$; we subtract the mean from each basis function to normalize π. For other statistics, we use the resulting stationary distribution (estimated with the same σ and τ) for μ, and we

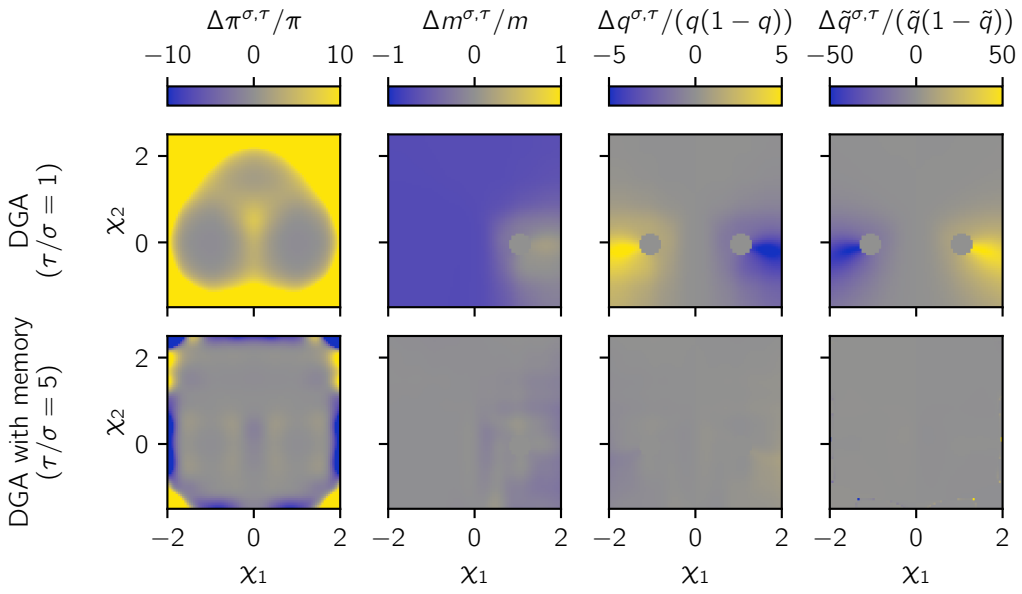

Figure 4.5: Relative error in the DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$) estimates for each dynamical statistic at $\tau = 0.05$, for the two-dimensional triple-well.

set the value of each basis function to zero for points in B (for m) or $A \cup B$ (for q_+ and q_-). We use guess functions $\hat{m}_0 = 0$, $\hat{q}_{+,0} = \mathbb{1}_B$, and $\hat{q}_{-,0} = \mathbb{1}_A$.

In Fig. 4.2, we show the stationary distribution (π), the MFPT to B (m), and the forward and backward committors for the reaction A to B (q_+ and q_-). The improvement due to memory is most noticeable in the MFPT, which we plot in Fig. 4.3. For DGA ($\tau/\sigma = 1$), a lag time of $\tau = 1$ is required for $w^{\sigma,\tau}$ to qualitatively match the reference in Fig. 4.2. On the

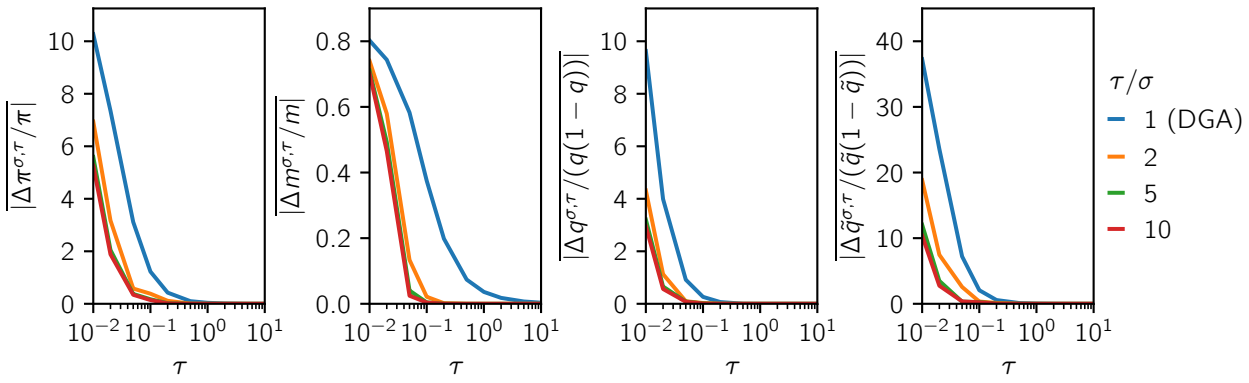

Figure 4.6: Effect of lag time and number of memory terms on the mean of the absolute value of the relative error for each dynamical statistic for the two-dimensional triple-well.

other hand, for DGA with memory ($\tau/\sigma = 5$), $\tau = 0.05$ is sufficient to obtain a reasonable approximation of the reference. When trajectory length is the limiting factor, this shorter lag time directly translates to an order of magnitude savings in simulation time.

We look at the effect of varying parameters σ and τ in Fig. 4.4. We focus on the rate constant k_{AB} for the transition from A to B because it is the most sensitive to errors; we calculate it as the inverse of the MFPT from configurations in the A state [139]:

$$k_{AB}^{-1} = \int_A m(x)\pi(x)\,\mathrm{d}x. \tag{4.53}$$

For this system, the true inverse rate constant is $k_{AB}^{-1} \approx 57$. In Fig. 4.4a, keeping the number of memory terms constant (by fixing τ/σ) and increasing τ makes the rate constant converge to the correct value. The best improvement that DGA with memory can give over DGA for a particular τ (which is limited by the data set) can be seen by comparing the $\tau/\sigma = 1$ (DGA) curve with the $\tau/\sigma = 10$ curve (which is approximately the same as the $\tau/\sigma = \infty$ limit). The inverse rate converges at $\tau \approx 10$ without memory and at $\tau \approx 0.05$ with memory. In Fig. 4.4b, keeping σ constant and increasing the number of memory terms also leads to convergence to the correct value. We note that DGA with memory overestimates the inverse rate around $\tau = 0.05$: memory can overcorrect for the error, so one must be careful when assessing convergence since the curve may appear flat at local extrema. In Fig. 4.4c, we consider the effect of increasing the number of memory terms with fixed τ. In this case, the result does not converge to the true value but instead retains a fixed error when τ is too short for the orthogonal variables to relax (see Appendix 4.6 for a graphical explanation).

These results inform our procedure for selecting τ and σ. Namely, we increase τ with fixed τ/σ until statistics of interest converge, as in Fig. 4.4a. Our examples suggest that we could instead hold σ fixed, as in Fig. 4.4b, but we expect this procedure to be more sensitive to statistical error in more demanding applications because it increases the number of terms in the memory sum.

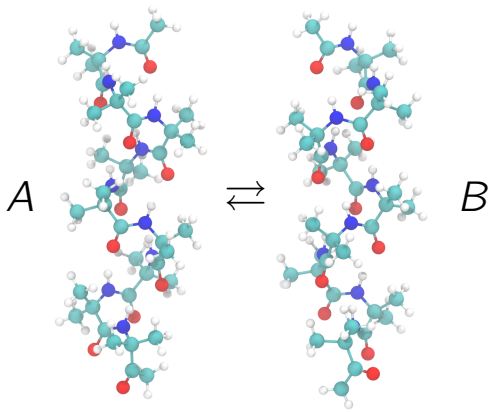

Figure 4.7: A and B configurations of AIB_9. Carbon, nitrogen, oxygen, and hydrogen atoms are colored cyan, blue, red, and white, respectively.

To demonstrate that these improvements are not specific to the MFPT and the rate constant, we plot the errors in other dynamical statistics in Figs. 4.5 and 4.6. We calculate relative errors for π and m as $\Delta u^{\sigma,\tau}/u$, and for q_+ and q_- as $\Delta u^{\sigma,\tau}/(u(1-u))$, where $\Delta u^{\sigma,\tau} = u^{\sigma,\tau} - u$. In Fig. 4.5, we look at the relative error in each dynamical statistic at $\tau = 0.05$. In all cases, we observe that DGA with memory ($\tau/\sigma = 5$) has a significantly lower error than DGA ($\tau/\sigma = 1$). We note that DGA has large regions of errors with the same sign, while DGA with memory has smaller regions of oscillating errors from memory-corrections with opposing signs. In Fig. 4.6, we look at the mean of the absolute value of the relative error in regions with $V \leq 0$ (where the system spends the most time; see Fig. 4.1) for each of the dynamical statistics. In all cases, DGA with memory outperforms DGA at the same lag time, especially at shorter lag times. We observe that, for $\tau/\sigma > 2$, DGA with memory estimates for all statistics converge at $\tau \approx 0.1$, even though each statistic has a different memory kernel. In contrast, for DGA ($\tau/\sigma = 1$), m requires a much larger τ to converge than other statistics.

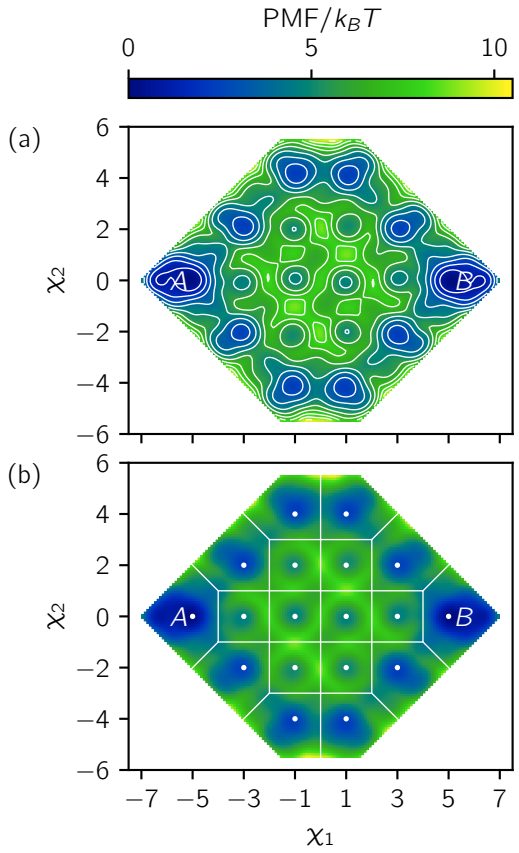

Figure 4.8: (a) Potential of mean force of AIB_9 in the (χ_1, χ_2) CV space with the 2 metastable states A and B labeled. Contours are drawn every k_BT. (b) Location and coordinates of the centers (dots) of the 18 states in the CV space. Note that multiple different intermediates (30 different intermediates appear as 16 states) may overlap in the same location in the CV space. Edges of the basis functions are indicated by lines.

4.4.2 Left-to-right helix transition in AIB_9

To demonstrate DGA with memory for a molecular system, we analyze the dynamics of AIB_9, a peptide consisting of nine α-aminoisobutyric acid (AIB) residues. AIB is an unnatural achiral amino acid which forms both left-handed and right-handed 3_{10} helices with equal probability. The left-to-right helix transition was previously studied using MSMs and long, unbiased molecular dynamics simulations [140, 141]. We recently performed both short and long simulations and used the resulting data to estimate statistics with MSMs and neural networks [135].

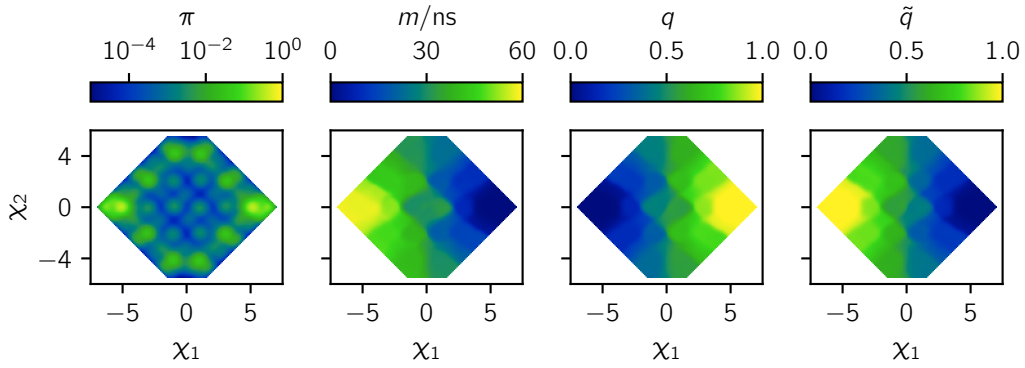

Figure 4.9: Reference statistics for AIB_9 calculated from the long trajectories. From left to right, the stationary distribution (π), the MFPT to the all-right state B (m), the forward committor for the transition from the all-left state A to all-right state B (q_+), and the backward committor for the same transition (q_-).

Each amino acid can isomerize between left-handed (l) and right-handed (r) states, which are defined by the ϕ and ψ dihedral angles. Similarly to Strahan et al. [135], we take an amino acid to be in the l state if its dihedral angle values are within a circle of radius 25° centered at $(41°, 47°)$, that is $(\phi - 41°)^2 + (\psi - 47°)^2 \leq (25°)^2$. Amino acids are classified as being in the r state using the same radius, but centered instead at $(-41°, -47°)$. States A and B are defined by the amino acids at sequence positions 3–7 being all l or all r, respectively. We show such configurations in Fig. 4.7. We do not use the two residues on each end of AIB_9 in defining the states as these are typically more flexible [141].

The data sets that we use are from Strahan et al. [135]. For that work, we ran simulations in OpenMM 7.7.0 [142] using the Langevin integrator at 300 K with a friction coefficient of $1\,\mathrm{ps}^{-1}$ and a time step of 4 fs. We used AIB parameters from Forcefield NCAA [143] with a hydrogen mass repartitioning scheme [144] and the GBNeck2 implicit-solvent model [145]. The short-trajectory data set that we analyze consists of 10 trajectories of duration 20 ns from each of the 691 starting configurations in Perez et al. [141]. The data set thus contains 6,910 trajectories, corresponding to a total sampling time of 138.2 µs. As a reference for comparison, we use 20 simulations of 15 µs with the same simulation parameters, corresponding to a total sampling time of 300 µs. In addition, we augment the reference data set with its reflection

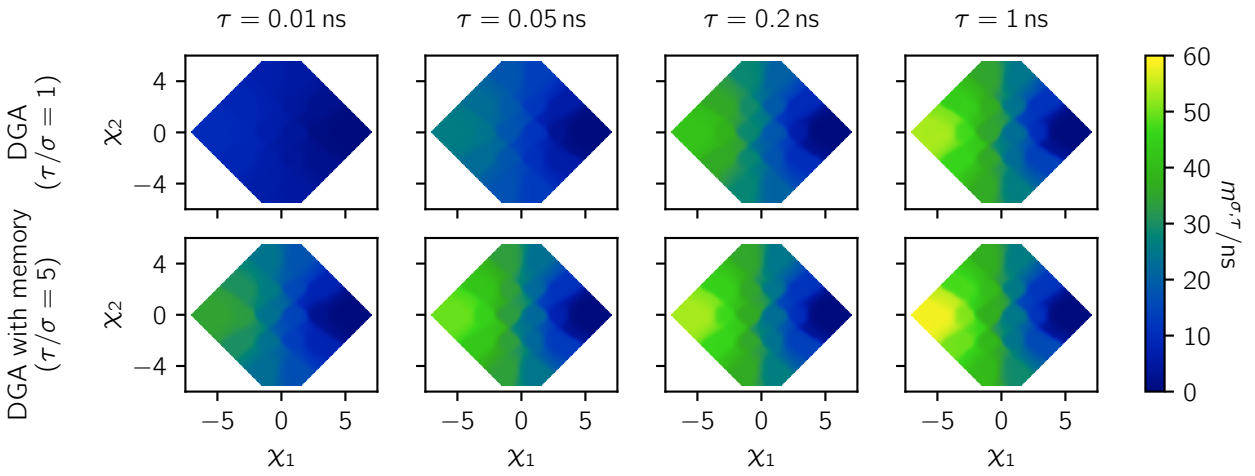

Figure 4.10: Mean first passage time to the all-right B state for AIB_9 estimated using DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$), at different lag times τ. DGA with memory requires shorter lag times to achieve comparable results to DGA.

(since AIB_9 is achiral).

We visualize results on the CV space

$$\chi_1 = \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6 + \gamma_7, \tag{4.54a}$$

$$\chi_2 = \gamma_3 + \gamma_4 - \gamma_6 - \gamma_7, \tag{4.54b}$$

where $\gamma_i = -0.8(\sin\phi_i + \sin\psi_i)$. The l state has $\gamma \approx -1$ and the r state has $\gamma \approx 1$, although these values also contain states other than l and r. We choose these CVs because they can distinguish the 10 most populated states and have physical meaning: χ_1 measures the chirality of a configuration, and χ_2 measures the difference in chirality between the two halves of the molecule. These CVs are similar to the first two principal components of the dihedral angle dynamics, which were used in Strahan et al. [135], Perez et al. [141], and Sittel et al. [146]. We project statistics onto the CV space using a kernel density estimate with a Gaussian kernel with a scale parameter of 0.2.

We compute the potential of mean force (PMF) on the (χ_1, χ_2) space from a histogram of CV pairs sampled in the reference data set (Fig. 4.8a). This shows the metastable A and B

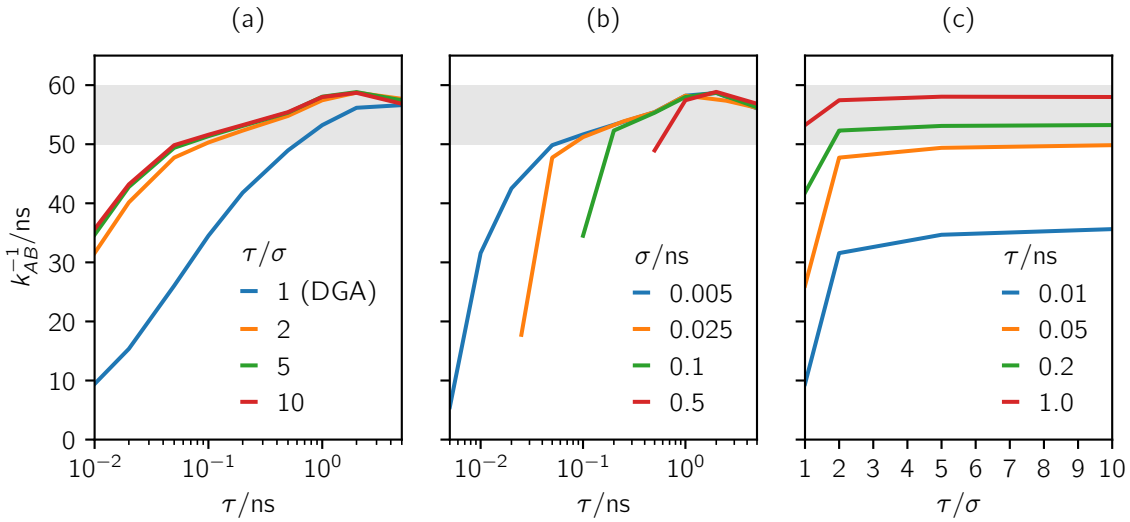

Figure 4.11: Dependence of the estimated AIB_9 inverse rate constant on (a) τ with fixed τ/σ, (b) τ with fixed σ, and (c) τ/σ with fixed τ. The true inverse rate constant is between 50 ns and 60 ns, which is indicated by the gray region.

states on the left and right sides, respectively, as well as the intermediate states connecting them. In Fig. 4.9, we show the reference statistics for the stationary distribution, the MFPT to the B state, and the forward and backward committors from the A state to the B state.

For the DGA calculations, we use a basis of 18 indicator functions on the (χ_1, χ_2) space as shown in Fig. 4.8b. These indicator functions are Voronoi cells with centers at points (χ_1, χ_2) with each $\gamma_i = \pm 1$. We compute statistics using the same procedure as in Sec. 4.4.1, where we first estimate the stationary distribution and then use it for other statistics. However, here we use short-trajectory data. We use a rolling window, so all choices of σ and τ use the same amount of data.

In Fig. 4.10, we look at the MFPT, which, as for the two-dimensional triple-well, most clearly shows the improvement due to memory. At $\tau = 0.05$ ns, DGA with memory ($\tau/\sigma = 5$) is quite similar to the reference (Fig. 4.9); DGA ($\tau/\sigma = 1$), on the other hand, drastically underestimates the MFPT in all of the CV space. DGA with memory results at $\tau = 0.2$ ns are comparable to DGA results at $\tau = 1$ ns. Looking at the inverse rate constant from A to B (Fig. 4.11), which we estimate to be between 50 ns and 60 ns from the reference trajectories,

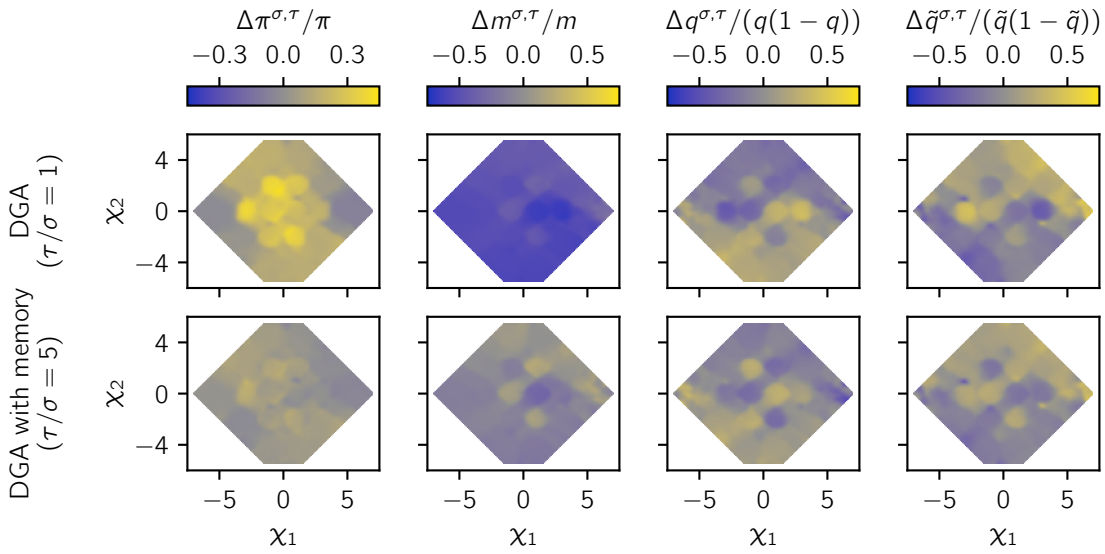

Figure 4.12: Relative errors in DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$) estimates of the AIB_9 dynamical statistics at lag time $\tau = 0.05\,\text{ns}$.

DGA with memory enters the correct range at $\tau \approx 0.05\,\text{ns}$, whereas DGA requires an order of magnitude more, $\tau \approx 0.5\,\text{ns}$.

We calculate errors by first projecting onto the (χ_1, χ_2) space, then using the relative error formulas as described in Sec. 4.4.1. We restrict the points used to calculate the mean absolute value of the relative absolute error to those with $|\chi_1| + |\chi_2| \leq 7$ and $|\chi_2| \leq 5.5$ due to a lack of samples outside of this region. The committors show limited improvement with

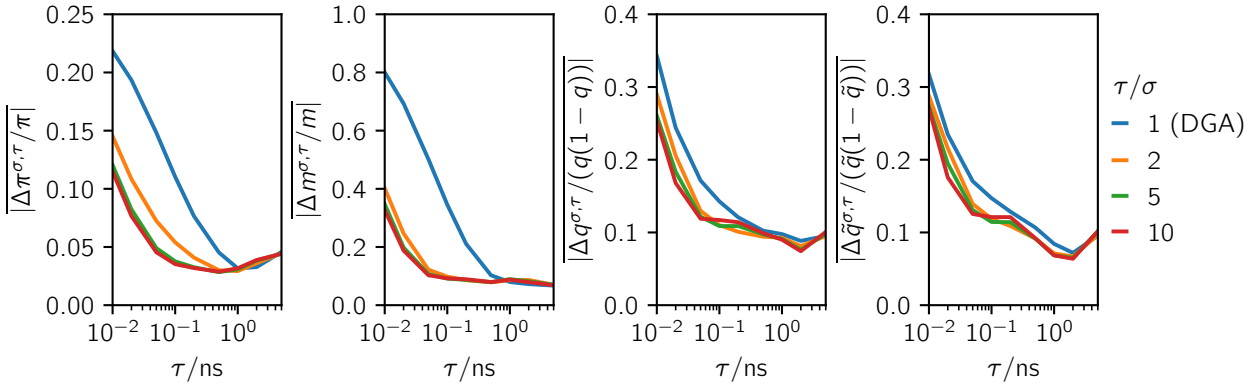

Figure 4.13: Effect of lag time and number of memory terms on the mean of the absolute value of the relative error for each dynamical statistic for AIB_9.

memory (as shown in Figs. 4.12 and 4.13), so we focus on the stationary distribution and the MFPT. In Fig. 4.12, we show the results of DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$) at $\tau = 0.05\,\mathrm{ns}$. Both algorithms accurately estimate the stationary probabilities of states A and B even at this short lag time, but DGA overestimates the weights of intermediates, especially the minor intermediates in the center. DGA drastically underestimates the MFPT to the B state in all regions. Fig. 4.13 shows that, for the stationary distribution and MFPT, DGA with memory requires an order magnitude shorter lag time to achieve comparable error to DGA. Note that the error fails to reach zero and even increases at longer lag times for π, q_+, and q_-; this is due to an increase in statistical error at longer lag times in both the DGA estimates and the reference statistics. Overall, the relative performances of DGA and DGA with memory for AIB_9 are qualitatively similar to those for the two-dimensional triple-well, despite the complications that we compute the statistics for AIB_9 from limited simulation data and quantify the error following projection to the (χ_1, χ_2) space.

4.5 Conclusions

In this chapter, we incorporate memory to improve the accuracy of statistics estimated by dynamical Galerkin approximation. Our work builds on numerical approaches for solving the orthogonal dynamics equation of the Mori–Zwanzig formalism [136] and quasi-MSMs [7, 133, 134] and shows that the basic concept behind these approaches can be generalized and used to compute much more than relaxation time scales. A key feature of quasi-MSMs and the present approach is that the memory corrections are obtained by directly inverting a GME, so that it is not necessary to assume a particular functional form for the memory. Quasi-MSMs and DGA with memory also contrast with traditional applications of the Mori–Zwanzig formalism in that the models are high-dimensional. As a result, the memory decays rapidly, and we find that even a few correction terms are sufficient to achieve high accuracy (with most of the correction coming from the first term beyond the Markov model). In our numerical

experiments, the memory correction decreases the data requirement by an order of magnitude.

Our approach has numerical advantages over quasi-MSMs and other methods that predict long-time statistics by integrating the GME. Integrating the GME requires estimating the full coarse-grained propagator and then using it to extrapolate in time. Solving statistic-specific equations, as we do here, allows better control of errors. In particular, because boundary conditions and other constraints are satisfied regardless of the choice of τ and σ, we can increase τ to decrease the error while keeping τ/σ small. A separate strategy that can be applied to improve the numerical performance further is to integrate over lag times [9, 133].

While DGA with memory can decrease the sensitivity of estimates to the choice of basis, a choice must still be made. To address this issue, we recently introduced an inexact subspace iteration to learn a basis represented by neural networks [135]. In Strahan et al. [135], we apply the memory correction after we learn the basis, but one could apply it after each subspace iteration to accelerate convergence. Even when the representation of a dynamical statistic does not have the form in (4.8) (e.g., Strahan et al. [147], or the Richardson iterate in Strahan et al. [135]), one could define a nonlinear projection operator and in turn compute a memory correction to compensate for deficiencies of the input features.

It would be interesting to compare the Mori–Zwanzig approach to memory taken here to alternative means of treating memory, such as delay embedding [4, 5, 111, 113, 114], with regard to data requirements and robustness to hyperparameter choices for high-dimensional models like those considered here.

4.6 Graphical depiction of the effect of memory

We compare DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 2$) graphically in Fig. 4.14. The projected dynamics move the system horizontally, while the orthogonal dynamics move it vertically. We show early iterations in Fig. 4.14a and the fixed point in Fig. 4.14b. The iterations alternate between using length-τ trajectories to propagate the projected function

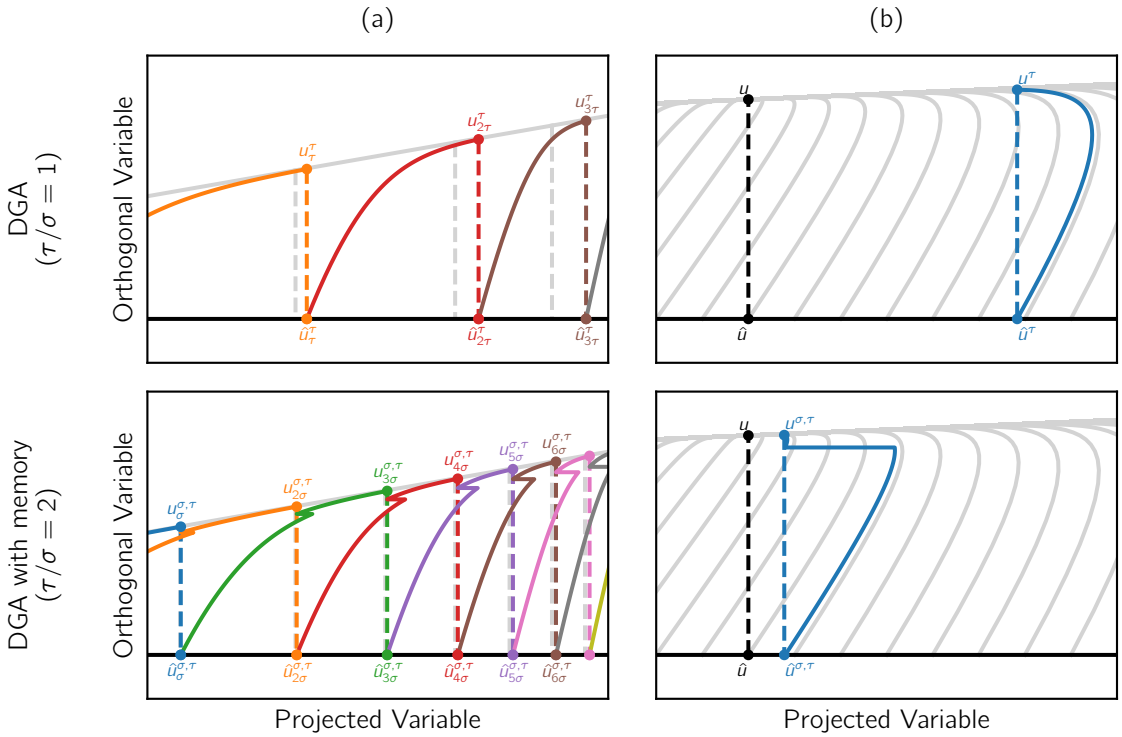

Figure 4.14: Schematic of the iterations and their solutions. (a) Inexact Richardson iteration corresponding to DGA ($\tau/\sigma = 1$, (4.12)) and DGA with memory ($\tau/\sigma = 2$, (4.21)). The exact Richardson iteration in (4.15) is plotted is gray. (b) Solutions to DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 2$). To guide the eye, exact Richardson iterations starting at different projections are plotted in gray.

$\hat{u}_t^\tau$ to $u_{t+\tau}^\tau$ (colored curves) and projecting back onto the subspace spanned by the basis (dashed lines).

In the case of DGA (Fig. 4.14a, $\tau/\sigma = 1$), the iteration (4.12) rapidly diverges from the exact iteration in (4.15), which is plotted in gray. The fixed point (4.12) occurs when propagating $\hat{u}^\tau$ for lag time τ to u^τ such that projecting returns the same $\hat{u}^\tau$ (Fig. 4.14b). As drawn, fast variables propagate $\hat{u}^\tau$ up and to the right, and slow variables ultimately return the system so as to be directly above $\hat{u}^\tau$, yielding the blue curve. In the absence of error due to sampling trajectories, $\hat{u}^\tau$ deviates from $\hat{u}$ because τ is insufficiently long for u^τ to reach u.

Comparing the iteration without memory ($\tau/\sigma = 1$) to the iteration with memory ($\tau/\sigma = 2$), we see that the latter approximates the exact iteration with significantly greater

accuracy. Each iteration with memory shown in the figure consists of four steps: (1) propagating $\hat{u}_t^{\sigma,\tau}$ by time σ (which initially moves the system up and to the right as drawn); (2) adjusting the projected variable with the orthogonal projection so that the projection is $\hat{u}_\sigma^{\sigma,\tau}$ (which moves the system to the left horizontally); (3) propagating by another time σ to $u_{t+2\sigma}^{\sigma,\tau}$ (which again moves the system up and to the right); (4) projecting to $\hat{u}_{t+2\sigma}^{\sigma,\tau}$ (which moves the system vertically). More generally, τ/σ steps, each of which propagates the system for time σ, and $(\tau/\sigma)-1$ projected value adjustments are performed in each iteration, and $(\tau/\sigma)-1$ iterations are inserted between those corresponding to DGA.

As for the case of DGA, the fixed point is reached when the final projection in each step of the iteration returns the system to $\hat{u}^{\sigma,\tau}$ (Fig. 4.14b, bottom). In DGA, the fast modes propagate $\hat{u}^\tau$ toward the upper right; τ must be long enough for the slow mode to correct this additional rightward shift. The impact of the memory correction is to eliminate this shift, so that if τ is sufficiently large to relax the fast modes that affect the projected variable, the deviation from u depends only on the interaction between the slow modes and the orthogonal variables.

4.7 Computing finite-time operators using the infinitesimal generator

In this section, we express the finite-time operators (e.g., $\mathcal{T}^t$ and $\mathcal{T}_B^t$) in terms of the infinitesimal generator

$$\mathcal{L} = \lim_{\epsilon \to 0^+} \frac{\mathcal{T}^\epsilon - \mathcal{I}}{\epsilon}. \tag{4.55}$$

For a continuous-time Markov jump process on a finite discrete state space, $\mathcal{L}$ is a matrix (e.g., (4.52)) and functions on the grid are vectors, so

$$\mathcal{L}f(x) = \sum_{x'} \mathcal{L}(x,x')f(x'), \tag{4.56}$$

and we can evaluate expressions in terms of $\mathcal{L}$ directly.

4.7.1 Transition operator

The transition operator can be expressed as

$$\mathcal{T}^t = (\mathcal{T}^\epsilon)^{t/\epsilon} = \lim_{\epsilon \to 0^+} (\mathcal{I} + \epsilon \mathcal{L})^{t/\epsilon} = e^{t\mathcal{L}}. \tag{4.57}$$

Similarly,

$$(\mathcal{T}^t)^\dagger(x, x') = e^{t\mathcal{L}^\dagger}(x, x'), \tag{4.58}$$

and

$$\mathcal{L}^\dagger(x, x') = \mu(x')\mathcal{L}(x', x)/\mu(x) \tag{4.59}$$

from the definition of the adjoint.

4.7.2 Stopped transition operator

Here, we express the stopped transition operator in terms of the infinitesimal generator. To this end, we observe that

$$f(X_{t \wedge S_B^+(0)}) = \lim_{\epsilon \to 0^+} \left(f(X_0) + \sum_{n=0}^{(t/\epsilon)-1} \left[\prod_{n'=0}^{n} \mathbb{1}_{B^\complement}(X_{n'\epsilon}) \right] (f(X_{(n+1)\epsilon}) - f(X_{n\epsilon})) \right). \tag{4.60}$$

That is, at time $n\epsilon$, we propagate the process forward-in-time only if $n\epsilon < S_B^+(0)$ (the product above is 1 only if $X_0, \ldots, X_{n\epsilon} \in B^{\mathsf{c}}$ and 0 otherwise). Therefore, we can express (4.6) as

$$\mathcal{T}_B^t f(x) = \mathbb{E}[f(X_{t \wedge S_B^+(0)}) \mid X_0 = x] \tag{4.61}$$

$$= \lim_{\epsilon \to 0^+} \left(f(x) + \sum_{n=0}^{(t/\epsilon)-1} (\mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}}) \mathcal{T}^\epsilon)^n \mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}})(\mathcal{T}^\epsilon - \mathcal{I}) f(x) \right) \tag{4.62}$$

$$= f(x) + \int_0^t e^{t' \mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}}) \mathcal{L}} \mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}}) \mathcal{L} f(x) \, \mathrm{d}t' \tag{4.63}$$

$$= e^{t \mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}}) \mathcal{L}} f(x). \tag{4.64}$$

We substitute (4.60) into (4.61) and express the resulting conditional expectation in terms of $\mathcal{T}^\epsilon(x, x') = \mathbb{P}[X_{t+\epsilon} = x' \mid X_t = x]$ to obtain (4.62). Taking the limit $\epsilon \to 0^+$ yields (4.63), and evaluating the integral yields (4.64). We define $\mathcal{D}(g)$ to be a diagonal matrix with entries $\mathcal{D}(g)(x, x) = g(x)$; the matrix $\mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}})\mathcal{L}$ has entries

$$(\mathcal{D}(\mathbb{1}_{B^{\mathsf{c}}})\mathcal{L})(x, x') = \mathbb{1}_{B^{\mathsf{c}}}(x) \mathcal{L}(x, x'). \tag{4.65}$$

We evaluate the integral $\int_0^t \mathcal{T}_B^{t'} \mathbb{1}_{B^\mathsf{c}} \,\mathrm{d}t' = \int_0^t e^{t'\mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L}} \mathbb{1}_{B^\mathsf{c}} \,\mathrm{d}t'$ in (4.36c) using the identity

$$\begin{aligned}
&\exp\left(t \begin{bmatrix} \mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L} & \mathbb{1}_{B^\mathsf{c}} \\ 0 & 0 \end{bmatrix}\right) \\
&\quad = \lim_{\epsilon \to 0^+} \left(\begin{bmatrix} \mathcal{I} & 0 \\ 0 & \mathcal{I} \end{bmatrix} + \epsilon \begin{bmatrix} \mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L} & \mathbb{1}_{B^\mathsf{c}} \\ 0 & 0 \end{bmatrix} \right)^{t/\epsilon} \\
&\quad = \lim_{\epsilon \to 0^+} \begin{bmatrix} (\mathcal{I} + \epsilon \mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L})^{t/\epsilon} & \epsilon \sum_{n=0}^{(t/\epsilon)-1} (\mathcal{I} + \epsilon \mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L})^n \mathbb{1}_{B^\mathsf{c}} \\ 0 & \mathcal{I} \end{bmatrix} \\
&\quad = \begin{bmatrix} e^{t\mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L}} & \int_0^t e^{t'\mathcal{D}(\mathbb{1}_{B^\mathsf{c}})\mathcal{L}} \mathbb{1}_{B^\mathsf{c}} \,\mathrm{d}t' \\ 0 & 1 \end{bmatrix},
\end{aligned} \tag{4.66}$$

That is, we compute the exponential and then take only the upper right element of the result.

4.7.3 Stopped transition operator for the time-reversed process

Similarly to (4.60) to (4.64), we use the identity

$$\begin{aligned}
&f(X_{(-t)\vee S^-_{A\cup B}(0)}) \\
&\quad = \lim_{\epsilon \to 0^+} \left(f(X_0) + \sum_{n=0}^{(t/\epsilon)-1} \left[\prod_{n'=0}^{n} \mathbb{1}_{(A\cup B)^\mathsf{c}}(X_{-n'\epsilon}) \right] (f(X_{-(n+1)\epsilon}) - f(X_{-n\epsilon})) \right)
\end{aligned} \tag{4.67}$$

to express (4.46) as

$$\begin{aligned}
&\langle g, \mathcal{T}_{A\cup B}^{-t} f\rangle \\
&\quad = \mathbb{E}\bigg[g(X_0) f(X_{(-t)\vee S^-_{A\cup B}(0)}) \frac{w(X_{-\tau})}{w(X_0)} \;\bigg|\; X_{-\tau}\sim\mu\bigg] && (4.68)\\
&\quad = \lim_{\epsilon\to 0^+} \sum_x \frac{g(x)\mu(x)}{w(x)} \bigg(f(x)(\mathcal{T}^\tau)^\dagger w(x) \\
&\qquad\qquad + \sum_{n=0}^{(t/\epsilon)-1} \Big[(\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})(\mathcal{T}^\epsilon)^\dagger)^n \mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}}) \\
&\qquad\qquad\qquad \times ((\mathcal{T}^\epsilon)^\dagger \mathcal{D}(f) - \mathcal{D}(f)(\mathcal{T}^\epsilon)^\dagger)(\mathcal{T}^{\tau-(n+1)\epsilon})^\dagger w(x) \Big] \bigg) && (4.69)\\
&\quad = \sum_x \frac{g(x)\mu(x)}{w(x)} \bigg(f(x) e^{\tau\mathcal{L}^\dagger} w(x) + \int_0^t \Big[e^{t'\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger} \mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}}) \\
&\qquad\qquad\qquad \times (\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger) e^{(\tau-t')\mathcal{L}^\dagger} w(x) \Big] \,\mathrm{d}t' \bigg). && (4.70)
\end{aligned}$$

We substitute (4.67) into (4.68) and expand the resulting expectation in terms of $(\mathcal{T}^\epsilon)^\dagger(x,x') = \mathbb{P}[X_{t+\epsilon} = x \mid X_t = x']\mu(x')/\mu(x)$ to obtain (4.69). Taking the limit $\epsilon \to 0^+$ yields (4.70), where the entries of the matrices $\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger$ and $\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger)$ are

$$(\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger)(x,x') = \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(x)\mathcal{L}^\dagger(x,x'), \tag{4.71}$$

$$(\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger))(x,x') = \mathbb{1}_{(A\cup B)^{\mathsf{c}}}(x)\mathcal{L}^\dagger(x,x')(f(x') - f(x)). \tag{4.72}$$

Likewise,

$$\begin{aligned}
&\langle g, (\mathcal{T}_{A\cup B}^{-t} - \mathcal{I})f\rangle \\
&\quad = \mathbb{E}\left[g(X_0)(f(X_{(-t)\vee S^-_{A\cup B}(0)}) - f(X_0))\frac{w(X_{-\tau})}{w(X_0)} \;\middle|\; X_{-\tau}\sim\mu \right] \qquad (4.73)\\
&\quad = \sum_x \frac{g(x)\mu(x)}{w(x)} \int_0^t e^{t'\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger} \mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger) e^{(\tau-t')\mathcal{L}^\dagger} w(x)\,\mathrm{d}t'. \qquad (4.74)
\end{aligned}$$

The integrals in (4.70) and (4.74) can be evaluated similarly to (4.66):

$$\begin{aligned}
&\exp\left(t \begin{bmatrix} \mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger & \mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger) \\ 0 & \mathcal{L}^\dagger \end{bmatrix} \right) \qquad (4.75)\\
&= \begin{bmatrix} e^{t\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger} & \int_0^t e^{t'\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger} \mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger) e^{(t-t')\mathcal{L}^\dagger}\,\mathrm{d}t' \\ 0 & e^{\mathcal{L}^\dagger t} \end{bmatrix}. \qquad (4.76)
\end{aligned}$$

The expressions for $\mathcal{T}_{A\cup B}^{-t}$ are more complicated than those for $\mathcal{T}_{A\cup B}^{t}$ because we want (4.70) and (4.74) to be equivalent to (4.68) and (4.73), respectively, when w is approximated. If we evaluate them without approximating w, $\mathcal{L}^\dagger w = 0$ and $e^{\mathcal{L}^\dagger t}w = w$; therefore, $\langle g, \mathcal{T}_{A\cup B}^{-t} f\rangle = \sum_{x,x'}(g(x)\mu(x)/w(x))e^{t\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger}(x,x')f(x')w(x')$ and $\langle g, (\mathcal{T}_{A\cup B}^{-t} - \mathcal{I})f\rangle = \sum_{x,x'}(g(x)\mu(x)/w(x))(e^{t\mathcal{D}(\mathbb{1}_{(A\cup B)^{\mathsf{c}}})\mathcal{L}^\dagger} - \mathcal{I})(x,x')f(x')w(x')$.

CHAPTER 5

EXACT VARIATIONAL COMMITTOR NETWORKS

For a transition between two stable states, the committor is the probability that the dynamics leads to one stable state before the other. It can be estimated from trajectory data by minimizing an expression for the transition rate that depends on a lag time. We show that an existing such expression is minimized by the exact committor only when the lag time is a single time step, resulting in a biased estimate in practical applications. We introduce an alternative expression that is minimized by the exact committor at any lag time. Numerical tests on benchmark systems demonstrate that our committor and resulting transition rate estimates are much less sensitive to the choice of lag time. We derive an additional expression for the transition rate, relate the transition rate expression to a variational approach for kinetic statistics based on the mean-squared residual, and discuss further numerical considerations with the aid of a decomposition of the error into dynamic modes.

5.1 Introduction

For a reaction in which a system transitions from a reactant state A to a product state B, the probability that the dynamics leads to B before A from a microscopic state x—known as the commitment probability or splitting probability, or simply the committor $q(x)$—is a natural (arguably optimal [73, 148–153]) reaction coordinate. Directly computing the committor for a microscopic state x involves simulating many trajectories from x to the A and B states. Though machine learning can be used to make efficient use of this information [108, 109, 154–156], directly computing the committor is prohibitively computationally expensive for many systems. To avoid this expense, many algorithms have been proposed which exploit the Markov structure of the underlying process to learn an estimate of the committor using a dataset of short trajectories (most neither beginning in A nor ending in B) initialized from

states distributed throughout the state space [4, 5, 135, 147, 157–168].

Specifically when the dynamics satisfy detailed balance and the trajectory data are from an equilibrium simulation, the committor can be approximated by minimizing the loss function [158]:

$$\Phi = \frac{1}{2\tau}\,\mathbb{E}[(q(X_\tau) - q(X_0))^2], \tag{5.1}$$

where X_t is the state of the Markov process at time t, with X_0 drawn from the equilibrium distribution. The parameter τ is known as the lag time. While (5.1) can be viewed as a finite-time approximation of the loss function $\mathbb{E}[\|\nabla q\|^2]$, which has also been minimized to approximate the committor function [157, 167, 169–171], (5.1) can also be derived by minimizing the steady-state flux between A and B [172–174]. Below, we obtain this expression from transition path theory [73, 150, 151].

Minimization of (5.1) yields the exact committor function when the lag time is a single time step. However, often analyses are performed for a projection onto a subset of variables (e.g., the positions without the velocities, the solute but not solvent degrees of freedom, distances between selected atoms, etc.), and the dynamics of these variables are generally non-Markovian at such short lag times [88]. At longer lag times, the Markov assumption better holds for projected dynamics [175], but then, as we discuss below, the exact committor no longer minimizes (5.1). In this chapter, we address this issue by introducing a variational expression that yields the exact committor and transition rates for arbitrary lag times.

The chapter is organized as follows. We review committors in Section 5.2.1 and outline the derivation of a multiple-time-step estimator for the transition rate in Section 5.2.2; details of the derivation are provided in Appendix 5.5. Based on the estimator for the transition rate, we introduce a multiple-time-step variational expression in Section 5.2.3 and discuss its relation to (5.1) and the loss function in Li et al. [157]. Then we discuss how the error in the committor can be quantified and introduce an improved estimator using two lag times in Section 5.2.4, with further discussion in Appendices 5.6 to 5.8. We describe the neural

network architectures that we use in Section 5.3.1 and the systems and data that we use for our numerical tests in Section 5.3.2. We show how our formulation improves committor and transition rate estimates in Sections 5.3.3 to 5.3.5. We discuss possible directions for future research in Section 5.4 and additional numerical considerations in Appendix 5.9.

5.2 Theory

We consider a discrete-time stationary ergodic Markov process X_t with time step ϵ. Continuous-time dynamics can be obtained by taking the limit $\epsilon \to 0$. We assume that states A and B are disjoint, and define the transition region $D = (A \cup B)^{\mathsf{c}}$.

5.2.1 Committor

The forward committor $q(x)$ is the probability that a system at x will enter B before A. The backward committor $\bar{q}(x)$ is the probability that a system at x exited A after B. Mathematically, they are defined as

$$q(x) = \mathbb{E}[\mathbb{1}_B(X_{S_t}) \mid X_t = x], \tag{5.2}$$

$$\bar{q}(x) = \mathbb{E}[\mathbb{1}_A(X_{\bar{S}_t}) \mid X_t = x], \tag{5.3}$$

where

$$S_t = \min\{s \mid s \geq t, X_s \notin D\}, \tag{5.4}$$

$$\bar{S}_t = \max\{s \mid s \leq t, X_s \notin D\}, \tag{5.5}$$

and

$$\mathbb{1}_C(x) = \begin{cases} 1 & \text{if } x \in C, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

When the dynamics obey detailed balance, the forward and backward committors are related by $\bar{q} = 1 - q$.

Committors can be estimated from single-step trajectories by solving the boundary value problems:

$$q(x) = \mathbb{E}[q(X_{t+\epsilon}) \mid X_t = x], \tag{5.7}$$

$$\bar{q}(x) = \mathbb{E}[\bar{q}(X_{t-\epsilon}) \mid X_t = x], \tag{5.8}$$

for $x \in D$, with boundary conditions $q(x) = \mathbb{1}_B(x)$ and $\bar{q}(x) = \mathbb{1}_A(x)$ for $x \notin D$. These boundary value problems follow from applying the law of total expectations (at times $t - \epsilon$ and $t + \epsilon$, respectively) to (5.2) and (5.3) when $x \in D$. We emphasize that (5.7) and (5.8) are exact only when ϵ is a single time step.

Definining $(t + \tau) \wedge S_t = \min\{t + \tau, S_t\}$ and $(t - \tau) \vee \bar{S}_t = \max\{t - \tau, \bar{S}_t\}$, for multiple time steps, the boundary value problems are instead

$$q(x) = \mathbb{E}[q(X_{(t+\tau)\wedge S_t}) \mid X_t = x], \tag{5.9}$$

$$\bar{q}(x) = \mathbb{E}[\bar{q}(X_{(t-\tau)\vee \bar{S}_t}) \mid X_t = x], \tag{5.10}$$

with the same boundary conditions as the single time step case. These boundary value problems follow from applying the law of total expectations (at times $(t + \tau) \wedge S_t$ and $(t - \tau) \vee \bar{S}_t$, respectively) to (5.2) and (5.3). They are exact for arbitrary $\tau \geq 0$ because S_t and $\bar{S}_t$ stop the process when it hits A or B. Unlike the single time step case, (5.9) and (5.10) also hold for $x \notin D$.

5.2.2 Transition rate

Denoting the number of transition paths from A to B within the trajectory segment $X_{[t,t']}$ by $N_{AB}(X_{[t,t']})$, the transition rate Φ is the mean of N_{AB} per unit time:

$$\Phi = \lim_{T\to\infty} \frac{1}{2T} N_{AB}(X_{[-T,T]}). \tag{5.11}$$

This rate is symmetric with respect to A and B and differs from the standard transition rate from A to B defined in transition path theory, which accounts for the time the process spends in state A [73, 150, 151]. In Appendix 5.5, we use transition path theory to derive an expression for the transition rate in terms of committors and finite-length trajectories:

$$\Phi = \frac{1}{\tau}\,\mathbb{E}[f(X_{[0,\tau]})], \tag{5.12}$$

where

$$\begin{aligned} f(X_{[t,t']}) = {} & \bar{q}(X_t)\mathbb{1}_D(X_{t'\wedge S_t})q(X_{t'})(\xi(X_{t'}) - \xi(X_t)) \\ & + \mathbb{1}_A(X_{t\vee \bar{S}_{t'}})q(X_{t'})(\xi(X_{t'}) - 0) \\ & + \bar{q}(X_t)\mathbb{1}_B(X_{t'\wedge S_t})(1 - \xi(X_t)) \\ & + N_{AB}(X_{[t,t']}), \end{aligned} \tag{5.13}$$

and ξ is a reaction coordinate that satisfies $\xi(x) = 0$ for $x \in A$ and $\xi(x) = 1$ for $x \in B$. The terms on the right hand side of (5.13) correspond respectively to contributions from transition paths that start before t and end after t'; start within $[t, t']$ and end after t'; start before t and end within $[t, t']$; and both start and end within $[t, t']$. In the case that $\tau = \epsilon$, (5.13) reduces to

$$f(X_{[t,t+\epsilon]}) = \bar{q}(X_t)q(X_{t+\epsilon})(\xi(X_{t+\epsilon}) - \xi(X_t)). \tag{5.14}$$

Expression (5.14) can also be derived from the transition path theory formula for the reactive flux through a separating surface between A and B, by integrating over level sets of ξ [5]. We emphasize that (5.13) and (5.14) do not rely on detailed balance.

When the dynamics satisfy detailed balance, such that $\bar{q} = 1 - q$, we can manipulate (5.12) to obtain an expression for the transition rate that is quadratic in the committor. To this end, we set $\xi = q$ and define the time-reversed trajectory $\bar{X}_{[t,t']} = (X_{t'}, X_{t'-\epsilon}, \ldots, X_t)$. Then, we can write (5.12) as

$$\Phi = \frac{1}{2\tau}\,\mathbb{E}[f(X_{[0,\tau]}) + f(\bar{X}_{[0,\tau]})] \tag{5.15}$$

$$= \frac{1}{\tau}\,\mathbb{E}[g(X_{[0,\tau]}, q)], \tag{5.16}$$

where

$$\begin{aligned} g(X_{[t,t']}, u) = \frac{1}{2}(&\mathbb{1}_D(X_{t'\wedge S_t})(u(X_{t'}) - u(X_t))^2 \\ &+ \mathbb{1}_{D^c}(X_{t'\wedge S_t})(\mathbb{1}_B(X_{t'\wedge S_t}) - u(X_t))^2 \\ &+ \mathbb{1}_{D^c}(X_{t\vee \bar{S}_{t'}})(u(X_{t'}) - \mathbb{1}_B(X_{t\vee \bar{S}_{t'}}))^2 \\ &+ N_{AB}(X_{[t,t']}) + N_{BA}(X_{[t,t']})). \end{aligned} \tag{5.17}$$

When $\tau = \epsilon$, (5.16) reduces to

$$\Phi = \frac{1}{2\epsilon}\,\mathbb{E}[(q(X_\epsilon) - q(X_0))^2]. \tag{5.18}$$

Expression (5.18) was previously obtained by Roux from the formula for the reactive flux through a separating surface. [172]

5.2.3 Variational loss functions

Expressions (5.16) and (5.18) can be used to optimize the committor and transition rate. Given a model $u(x)$ for the committor, we can define the loss function

$$\tilde{L}_\tau(u) = \frac{1}{\tau}\,\mathbb{E}[g(X_{[0,\tau]}, u)]. \tag{5.19}$$

We note that this loss function includes a soft penalty for the boundary conditions:

$$g(X_{[t,t']}, u) = g(X_{[t,t']}, \hat{u}) + \frac{1}{2}\mathbb{1}_{D^\mathsf{c}}(X_t)(u(X_t) - \mathbb{1}_B(X_t))^2 + \frac{1}{2}\mathbb{1}_{D^\mathsf{c}}(X_{t'})(u(X_{t'}) - \mathbb{1}_B(X_{t'}))^2, \tag{5.20}$$

where we have introduced the function $\hat{u}(x)$ that is identical to $u(x)$ for $x \in D$, but satisfies the boundary condition $\hat{u}(x) = \mathbb{1}_B(x)$ for $x \in D^\mathsf{c}$ exactly. The gradient of (5.19) with respect to parameters θ is

$$\nabla_\theta \tilde{L}_\tau(u) = \frac{1}{\tau}\,\mathbb{E}[\nabla_\theta u(X_0)(u(X_0) - \hat{u}(X_{\tau\wedge S_0})) + \nabla_\theta u(X_\tau)(u(X_\tau) - \hat{u}(X_{0\vee \bar{S}_\tau}))], \tag{5.21}$$

which is zero when $u = q$ by (5.9) and (5.10).

When $\tau = \epsilon$, (5.19) reduces to

$$L_\tau(u) = \frac{1}{2\tau}\,\mathbb{E}[(\hat{u}(X_\tau) - \hat{u}(X_0))^2 + \mathbb{1}_{D^\mathsf{c}}(X_0)(u(X_0) - \mathbb{1}_B(X_0))^2 + \mathbb{1}_{D^\mathsf{c}}(X_\tau)(u(X_\tau) - \mathbb{1}_B(X_\tau))^2]. \tag{5.22}$$

The last two terms in the expectation impose a soft penalty for the boundary conditions. The gradient of (5.22) is

$$\begin{aligned}\nabla_\theta L_\tau(u) = \frac{1}{\tau}\,\mathbb{E}[&\nabla_\theta u(X_0)(u(X_0) - (\mathbb{1}_B(X_0) + \mathbb{1}_D(X_0)\hat{u}(X_\tau))) \\ &+ \nabla_\theta u(X_\tau)(u(X_\tau) - (\mathbb{1}_B(X_\tau) + \mathbb{1}_D(X_\tau)\hat{u}(X_0)))],\end{aligned} \tag{5.23}$$

which is zero when $\tau = \epsilon$ and $u = q$ because of (5.7) and (5.8). We write these expression with τ rather than ϵ because we will later use them with $\tau > \epsilon$. However, minimizing (5.22) with $\tau > \epsilon$ does not yield q in the infinite data limit.

In principle, the boundary condition penalties are not needed—in Chen et al. [158], the authors minimized a variational expression that enforced boundary conditions directly on the model output and evaluated the loss only in D. However, we found that this approach often leads to non-physical (near) discontinuities close to A and B in practice. Because the variational expression in Chen et al. [158] was otherwise the same as (5.22), with u represented by a neural network, we follow them and term (5.22) the variational committor-based neural network (VCN) loss function; by extension, we term (5.19) the exact VCN (EVCN).

We now explicitly show that these loss functions are variational by analyzing how errors in the committor affect transition rate estimates. The EVCN loss function (5.19) is minimized when $u = q$ and $\tilde{L}_\tau(q) = \Phi$ for all τ. By adding a perturbation η to q, we have

$$\begin{aligned}\tilde{L}_\tau(q+\eta) = \Phi + \frac{1}{2\tau}\,\mathbb{E}[&2\hat{\eta}(X_0)(q(X_0) - q(X_{\tau\wedge S_0})) + 2\hat{\eta}(X_\tau)(q(X_\tau) - q(X_{0\vee\bar{S}_\tau})) \\ &+ \mathbb{1}_D(X_{\tau\wedge S_0})(\hat{\eta}(X_\tau) - \hat{\eta}(X_0))^2 \\ &+ \mathbb{1}_{D^\mathsf{c}}(X_{\tau\wedge S_0})\hat{\eta}(X_0)^2 + \mathbb{1}_{D^\mathsf{c}}(X_{0\vee\bar{S}_\tau})\hat{\eta}(X_\tau)^2 \\ &+ \mathbb{1}_{D^\mathsf{c}}(X_0)\eta(X_0)^2 + \mathbb{1}_{D^\mathsf{c}}(X_\tau)\eta(X_\tau)^2],\end{aligned} \tag{5.24}$$

where $\hat{\eta}(x) = \mathbb{1}_D(x)\eta(x)$. The first two terms inside the expectation of (5.24) are zero because of (5.9) and (5.10), and the remaining terms are quadratic, so $\tilde{L}_\tau(u) \geq \Phi$ and (5.19) is variational. Because (5.22) derives from (5.19), it follows immediately that (5.22) is variational as well when $\tau = \epsilon$. When $\tau > \epsilon$, (5.22) is not variational and can underestimate Φ.

The loss function in (5.19) is also closely related to that of Li et al. [157]. With infinite data, the two differ by a constant (they omit the N_{AB} and N_{BA} terms) and yield the same

gradient. Their method directly estimates the gradient from finite data using only the first term of (5.21), which is a consistent approximation of the true gradient but is not the gradient of any loss function on finite data (this issue does not arise in the limit of infinite data because both terms then have the same value due to detailed balance). By symmetrizing the loss with time-reversed trajectories, we directly compute the gradient from a loss function on finite data. Moreover, our loss is more interpretable, as its minimum is the transition rate and must be used when that quantity is of interest.

5.2.4 Mean squared error and alternative transition rate expression

To assess the accuracy of the committor in our numerical tests, we evaluate the mean squared error (MSE):

$$\mathrm{MSE}(u) = \langle u - q, u - q \rangle. \tag{5.25}$$

This is a natural metric for our method because we can rearrange (5.24) and take the limit $\tau \to \infty$ to obtain

$$\lim_{\tau\to\infty} \tau(\tilde{L}_\tau(q+\eta) - \Phi) = \mathrm{MSE}(q+\eta). \tag{5.26}$$

This result follows from $\lim_{\tau\to\infty} \mathbb{1}_D(X_{\tau\wedge S_0}) = \mathbb{1}_D(X_{S_0}) = 0$, $\lim_{\tau\to\infty} \mathbb{1}_{D^\mathsf{c}}(X_{\tau\wedge S_0}) = \mathbb{1}_{D^\mathsf{c}}(X_{S_0}) = 1$, $\lim_{\tau\to\infty} \mathbb{1}_{D^\mathsf{c}}(X_{0\vee\bar{S}_\tau}) = \lim_{\tau\to\infty} \mathbb{1}_{D^\mathsf{c}}(X_{\bar{S}_\tau}) = 1$, and $\hat{\eta}(x)^2 + \mathbb{1}_{D^\mathsf{c}}(x)\eta(x)^2 = \mathbb{1}_D(x)\eta(x)^2 + \mathbb{1}_{D^\mathsf{c}}(x)\eta(x)^2 = \eta(x)^2$. As noted above, the first two terms inside the expectation of (5.24) are zero because of (5.9) and (5.10).

Because we typically do not have the true committor at each configuration, we must approximate the MSE. If the dataset consists of long trajectories in equilibrium, we can estimate the MSE using

$$\begin{aligned}\mathrm{MSE}(u) &= \mathbb{E}[(u(X_0) - \mathbb{E}[\mathbb{1}_B(X_{S_0}) \mid X_0])^2] &(5.27)\\ &= \mathbb{E}[(u(X_0) - \mathbb{1}_B(X_{S_0}))(u(X_0) - \mathbb{1}_B(X_{\bar{S}_0}))]. &(5.28)\end{aligned}$$

As discussed in Strahan et al. [135, 147], (5.27) cannot be estimated using X_{S_0} alone because of the double sampling problem (that is, $\mathbb{E}[\mathbb{E}[\mathbb{1}_B(X_{S_0}) \mid X_0]^2] \neq \mathbb{E}[\mathbb{1}_B(X_{S_0})^2]$). We resolve this issue by using (5.28), which involves independent samples conditioned on X_0, X_{S_0} and $X_{\bar{S}_0}$. The MSE estimated from (5.28) can be negative due to sampling error, which introduces a constant offset. This is because the quantity in the expectation is not a squared difference but rather a product of two different differences. Nonetheless, lower MSE values (closer to negative infinity) generally indicate better performance, even when they are negative.

If the dataset consists of many short trajectories, the MSE cannot be estimated using (5.28) because S_0 and $\bar{S}_0$ are not generally available. Motivated by (5.26), we define approximations $\tilde{L}_{\tau_1,\tau_2}(u) \approx \Phi$ and $\mathrm{MSE}_{\tau_1,\tau_2}(u) \approx \mathrm{MSE}(u)$ by solving $\tau_1(\tilde{L}_{\tau_1}(u) - \tilde{L}_{\tau_1,\tau_2}(u)) = \mathrm{MSE}_{\tau_1,\tau_2}(u)$ and $\tau_2(\tilde{L}_{\tau_2}(u) - \tilde{L}_{\tau_1,\tau_2}(u)) = \mathrm{MSE}_{\tau_1,\tau_2}(u)$:

$$\tilde{L}_{\tau_1,\tau_2}(u) = \frac{\tau_1 \tilde{L}_{\tau_1}(u) - \tau_2 \tilde{L}_{\tau_2}(u)}{\tau_1 - \tau_2}, \tag{5.29}$$

$$\mathrm{MSE}_{\tau_1,\tau_2}(u) = \frac{\tilde{L}_{\tau_1}(u) - \tilde{L}_{\tau_2}(u)}{1/\tau_1 - 1/\tau_2}. \tag{5.30}$$

In Appendix 5.7, we show that $\tilde{L}_{\tau_1,\tau_2}(u)$ is an upper bound for Φ and $\mathrm{MSE}_{\tau_1,\tau_2}(u)$ is a lower bound for $\mathrm{MSE}(u)$. We relate $\mathrm{MSE}_{\tau_1,\tau_2}(u)$ to the mean squared residual loss function in Strahan et al. [147] in Appendix 5.8.

5.3 Numerical tests

In this section, we describe the neural network architecture and training procedure, introduce the test systems, and then discuss numerical results for the committor and transition rate. Because long trajectories are available for all of the systems, we can compare the results directly to the empirical rate, which we compute from the data as

$$\Phi_{\mathrm{emp}} = \frac{1}{2T}[N_{AB}(X_{[0,T]}) + N_{BA}(X_{[0,T]})], \tag{5.31}$$

where here T is the trajectory length. Similarly, we compute the empirical committor as

$$q_{\text{emp}}(X_t) = \frac{1}{2}[\mathbb{1}_B(X_{S_t}) + \mathbb{1}_B(X_{\bar{S}_t})]. \tag{5.32}$$

5.3.1 Neural networks

We approximate the committor using a multilayer perceptron with two hidden layers of 100 neurons each and SiLU activation functions, followed by a sigmoid function that constrains outputs to $[0, 1]$. The model is trained on either the VCN loss function (5.22) or the EVCN loss function (5.19).

Training the committor to convergence often requires many optimization steps (over 10^4), especially at short lag times. In this work, we trained models up to 20000 steps due to computational constraints. However, training often becomes unstable after many steps: the loss spikes and recovers repeatedly without converging. To mitigate this behavior, we found it essential to use a large batch size (at least 1000 trajectory segments), the Adam-atan2 optimizer [176], and a small learning rate (10^{-3}, which is just below the threshold at which training becomes unstable).

To make metrics on the training and validation datasets comparable, we partition the trajectories into two subsets. Five models are trained on each subset and validated on the other. Each full trajectory is segmented into length τ trajectory segments using a sliding window. At each optimization step, 1000 trajectory segments $X_{[t,t+\tau]}$ are uniformly sampled from the training dataset to compute the loss. Checkpoints are saved after $\{1, 2, 5, 10, 20, 50, \ldots, 20000\}$ optimization steps, and the checkpoint with the best loss on the validation dataset is selected. Unless otherwise indicated, metrics are averaged over the 10 models.

As we discuss further in Appendix 5.9, overfitting is a major issue for both VCN and EVCN loss functions. We thus tested a variety of regularization techniques. Dropout degraded performance and slowed training. Batch normalization [177] and layer normalization [178]

initially sped up training, but the models struggled to capture finer details in the committor. Standard weight decay [44] and weight decay toward initial parameters [179] both worsened performance. As such, we do not employ these methods.

5.3.2 Systems

We test our method on three molecular systems of increasing complexity: AIB_9, Trp-cage, and villin. In this section, we briefly describe these systems and define collective variables (CVs) and states that we use in our analysis.

AIB_9

AIB_9 is a 9-residue peptide of 2-aminoisobutyric acid (AIB), an achiral, unnatural amino acid that forms both left-handed and right-handed 3_{10} helices with equal probability. We examine the left-to-right helix transition. The dataset consists of 20 trajectories of 15 µs, which we generated for Strahan et al. [135] and analyzed in that study and Lorpaiboon et al. [110].

For each residue i, we define $\gamma_i = -0.8(\sin(\phi_i)+\sin(\psi_i))$. The left-handed and right-handed residue conformations have $\gamma_i \approx -1$ and $\gamma_i \approx 1$, respectively, though other configurations may also have these values. We define the left-handed helix state A and right-handed helix state B as having (ϕ, ψ) angles of residues 3–7 within 25° of $(41°, 47°)$ and $(-41°, -47°)$, respectively. With these state definitions, the left-to-right helix transition has an empirical rate of 9.1×10^{-3} ns^{-1}. In Fig. 5.1, we show the stationary distribution (π) and empirical committor projected on the collective variables (CVs) introduced in Lorpaiboon et al. [110]:

$$\xi_1 = \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6 + \gamma_7, \tag{5.33}$$

$$\xi_2 = \gamma_3 + \gamma_4 - \gamma_6 - \gamma_7. \tag{5.34}$$

The leftmost and rightmost states correspond to states A and B, respectively.

We use as inputs to the neural networks two sets of features with increasing expressivity: the CVs ξ_1 and ξ_2 ("CVs"), and the sines and cosines of the (ϕ, ψ) dihedral angles for residues 3–7 ("Dihedrals").

Trp-cage

Trp-cage is a designed 20-residue fast-folding protein that has been extensively studied both experimentally and computationally (see Strahan et al. [5] and references therein). Its folded structure consists of an α-helix (residues 2–9), a 3_{10}-helix (residues 11–14), and a polyproline II helix (residues 17–19), which form a cage around Trp6. We analyze a 208 µs trajectory of the K8A mutant (sequence: DAYAQWLADGGPSSGRPPPS) at 290 K, saved every 0.2 ns [39]. To generate training and validation data, the trajectory is split in half and treated as two independent segments.

We define three CVs: the root-mean-squared deviation (RMSD) of the C_α atoms to the experimental structure (PDB 2JOF [180]) of the α-helix (RMSD_1), the 3_{10}-helix (RMSD_2), and the α- and 3_{10}-helices ($\text{RMSD}_{1,2}$). Following Strahan et al. [5], we project the results onto RMSD_1 and RMSD_2. We define the unfolded state A as configurations with $\text{RMSD}_1 \geq 0.35\,\text{nm}$, $\text{RMSD}_2 \geq 0.2\,\text{nm}$, and $\text{RMSD}_{1,2} \geq 0.4\,\text{nm}$ and the folded state B as configurations with $\text{RMSD}_1 \leq 0.05\,\text{nm}$, $\text{RMSD}_2 \leq 0.05\,\text{nm}$, and $\text{RMSD}_{1,2} \leq 0.15\,\text{nm}$. With these state definitions, the empirical rate is $7.7 \times 10^{-2}\ \mu\text{s}^{-1}$. We show the stationary distribution and empirical committor projected onto the CVs in Fig. 5.2. The unfolded state A is in the upper right, and the folded state B is in the lower left.

We use as inputs to the neural networks three sets of features with increasing expressivity: the three RMSDs used to define the states ("RMSDs"), sines and cosines of the (ϕ, ψ) backbone dihedral angles ("Dihedrals"), and all pairwise distances between C_α atoms ("Distances").

Villin

The fast-folding 35-residue villin headpiece subdomain, hereafter referred to as villin, is one of the most well-studied protein folding models both experimentally and computationally (see Wang et al. [181] and references therein). In the folded state, its secondary structure consists of three α-helices spanning residues 3–10, 14–19, and 22–32; these helices pack around a hydrophobic core centered on residues Phe6, Phe10, and Phe17. We study the K65nL/N68H/K70nL mutant (sequence: LSDEDFKAVFGMTRSAFANLPLWnLQQHLnL KEKGLF), where nL is norleucine, which is engineered to fold more rapidly. The dataset consists of a single 125 µs trajectory, saved every 0.2 ns [39]. To generate training and validation data, the trajectory is split in half and treated as two independent segments.

We define three CVs: the RMSD of the C_α atoms to the experimental structure (PDB 2F4K [182]) of helices 1 and 2 ($RMSD_{1,2}$), helices 2 and 3 ($RMSD_{2,3}$), and helices 1, 2, and 3 ($RMSD_{1,2,3}$). The unfolded state A is defined as configurations with $RMSD_{1,2} \geq 0.4\,\text{nm}$, $RMSD_{2,3} \geq 0.4\,\text{nm}$, and $RMSD_{1,2,3} \geq 0.5\,\text{nm}$ and the folded state B is defined as configurations with $RMSD_{1,2} \leq 0.1\,\text{nm}$, $RMSD_{2,3} \leq 0.1\,\text{nm}$ and $RMSD_{1,2,3} \leq 0.15\,\text{nm}$. With these state definitions, unfolded to folded transitions occur at a rate of $3.7 \times 10^{-1}\ \mu\text{s}^{-1}$. We show the stationary distribution and empirical committor projected onto $RMSD_{1,2}$ and $RMSD_{2,3}$ in Fig. 5.3. The unfolded state A is in the upper right, and the folded state B is in the lower left.

We use as inputs to the neural networks three sets of features with increasing expressivity: the three RMSDs ("RMSDs"), sines and cosines of the (ϕ, ψ) backbone dihedral angles ("Dihedrals"), and all pairwise distances between C_α atoms ("Distances").

5.3.3 Committor

We examine the behavior of committors predicted using the VCN and EVCN loss functions at different lag times. We compute committors for individual conformations in the sampled trajectories and then visualize the results in two ways. First, we plot the average committor for structures in a bin defined by a pair of CV values. Second, we partition the predicted committor into 20 uniformly spaced bins over $[0, 1]$, and for each bin, we plot the mean empirical committor against the mean predicted committor. In the latter set of plots, which we term "reliability diagrams" in line with the machine learning literature (which also calls them "calibration curves"), each point is from a single model. Perfect alignment between the predicted and empirical committors corresponds to points lying along the diagonal (shown in black).

We first look at AIB_9 (Fig. 5.1). At the shortest lag time (1 ps), CV projections of the predicted committor exhibit a checkered pattern reflecting the boundaries of the intermediate states. The reliability diagrams show substantial noise, but the means follow the diagonal. These artifacts arise because the VCN and EVCN losses are both relatively insensitive to slowly decaying modes, encouraging the committor to be flat within intermediate states at the expense of transition states between them. At the intermediate lag time (100 ps), the committor becomes qualitatively accurate for both loss functions. At the longest lag time (10 ns), the committor for EVCN remains accurate, while the committor for VCN appears flattened in the transition region and exhibits large jumps near the boundaries of states A and B. These behaviors are reflected in the reliability diagrams. We can explain this behavior by considering $\mathbb{E}[\hat{u}(X_0)\hat{u}(X_\tau)] \approx \mathbb{E}[\hat{u}(X_0)]\,\mathbb{E}[\hat{u}(X_\tau)]$ as $\tau \to \infty$. With this approximation, (5.22) is minimized when $u(x) = \mathbb{1}_B(x)$ for $x \notin D$ and $u(x) = \mathbb{E}[\mathbb{1}_B(X_0)]/\,\mathbb{E}[\mathbb{1}_{D^c}(X_0)]$ otherwise.

Next, we look at Trp-cage (Fig. 5.2). At the 1 ns lag time, both loss functions incorrectly predict the committor of the intermediate state above the folded basin in the CV projection to be near zero. Reflecting this, the reliability diagram has most points above the diagonal,

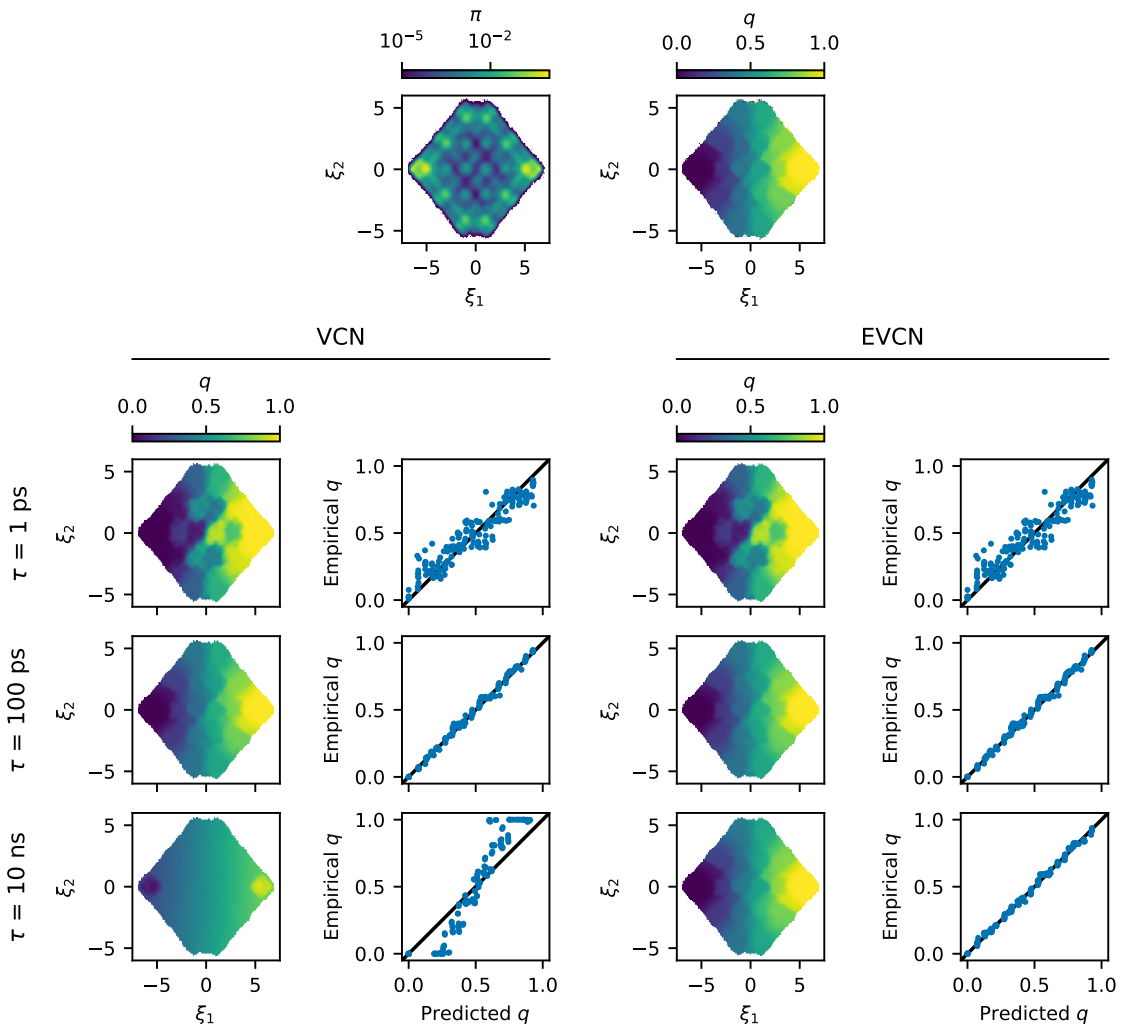

Figure 5.1: CV projections and reliability diagrams for AIB_9. (top left) CV projection of the stationary distribution. (top right) CV projection of the empirical committor. (left columns of VCN and EVCN) CV projections of the predicted committors. (right columns of VCN and EVCN) Reliability diagrams for the predicted committors. All results shown are obtained with the dihedral neural-network inputs.

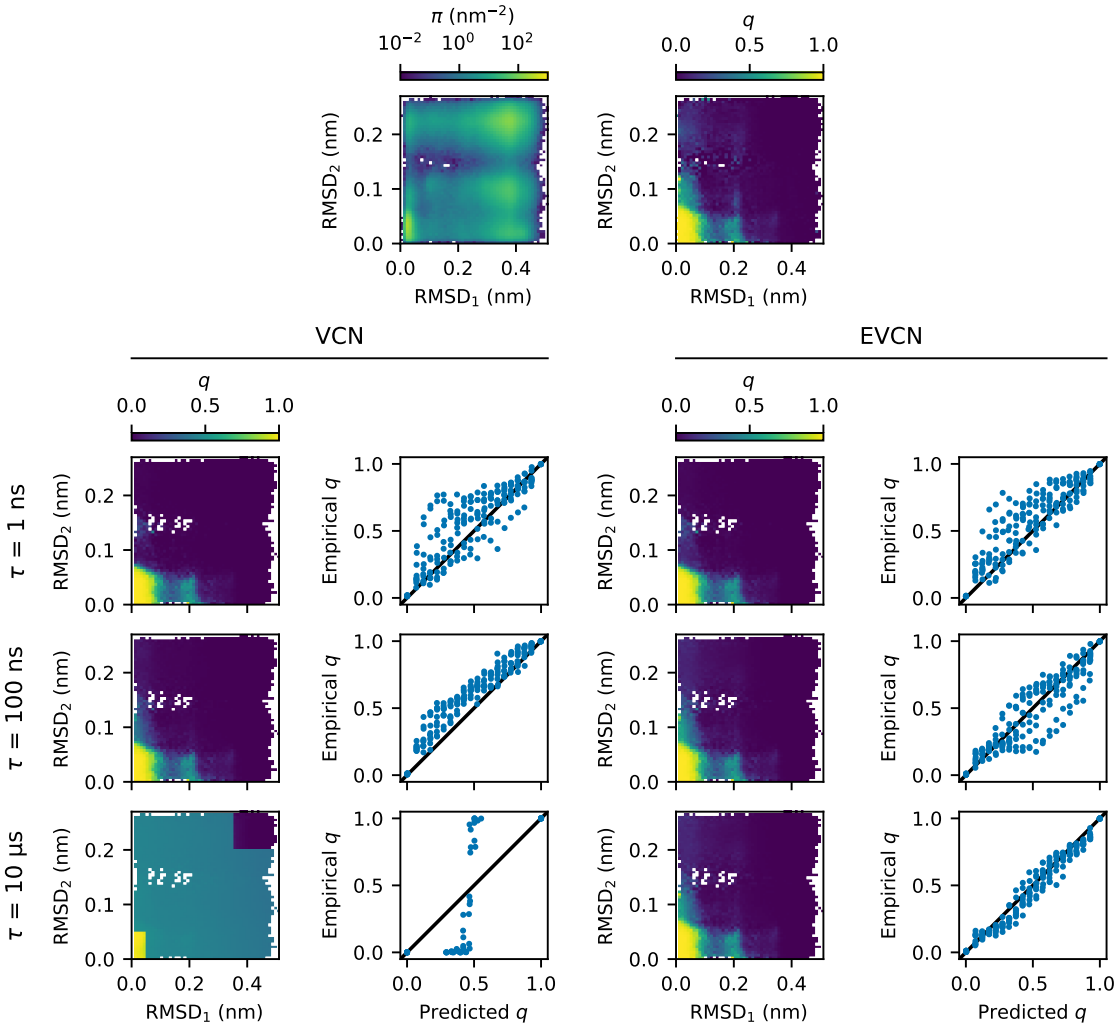

Figure 5.2: CV projections and reliability diagrams for Trp-cage. (top left) CV projection of the stationary distribution. (top right) CV projection of the empirical committor. (left columns of VCN and EVCN) CV projections of the predicted committors. (right columns of VCN and EVCN) Reliability diagrams for the predicted committors. All results shown are obtained with the dihedral neural-network inputs.

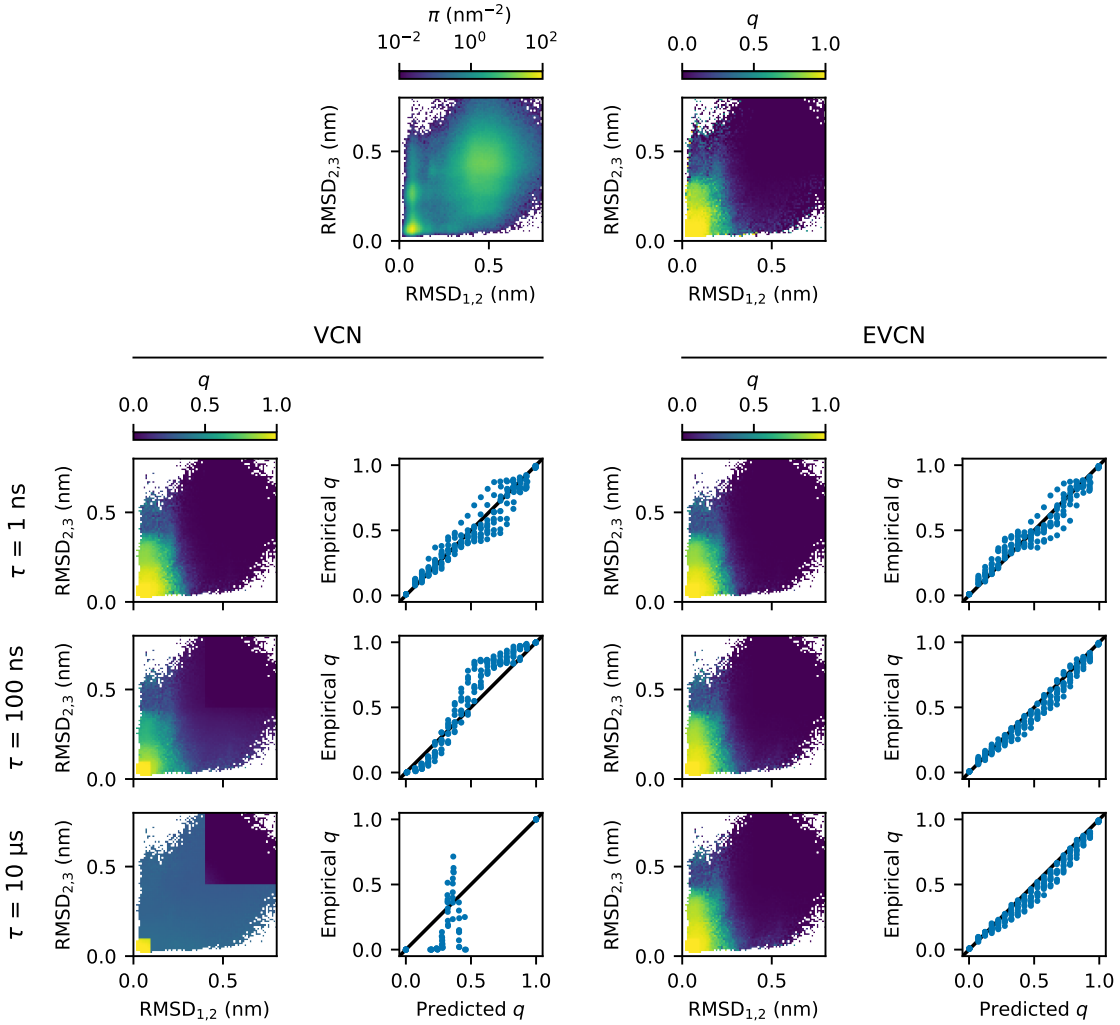

Figure 5.3: CV projections and reliability diagrams for villin. (top left) CV projection of the stationary distribution. (top right) CV projection of the empirical committor. (left columns of VCN and EVCN) CV projections of the predicted committors. (right columns of VCN and EVCN) Reliability diagrams for the predicted committors. All results shown are obtained with the dihedral neural-network inputs.

where the empirical committor is greater than the predicted committor. At the 100 ns lag time, both methods resolve the intermediate state but the VCN predictions are lower (darker CV projection). Correspondingly, the reliability diagram for VCN has more points above the diagonal. For EVCN, on the other hand, the points are noisy but do not show obvious bias. With a 10 µs lag time, VCN breaks down. The CV projection shows that the predicted committor is flat in D, and the reliability diagram shows that all the predicted values (other than 0 in A and 1 in B) are clustered near a single value. On the other hand, the CV projection of the EVCN result remains accurate, and the reliability diagram follows the diagonal with less noise than the 100 ns lag time.

Villin (Fig. 5.3) exhibits similar behavior to the previous systems. For the 1 ns lag time, both loss functions yield similar, good, predictions. However, at the 100 ns lag time, the EVCN prediction shows clear improvement over the VCN prediction, which is already beginning to flatten in the transition region. In the reliability diagrams, EVCN predictions are near the diagonal, while VCN predictions visibly deviate. At the 10 µs lag time, the VCN prediction is again clustered around a single value, while the EVCN prediction remains accurate.

Committor hyperparameters			Transition rate hyperparameters					
			VCN (5.22)			EVCN (5.19)		
			1 ps	100 ps	10 ns	1 ps	100 ps	10 ns
CVs	VCN	1 ps	7.0×10^{-2}	1.3×10^{-2}	6.8×10^{-3}	7.0×10^{-2}	1.3×10^{-2}	9.1×10^{-3}
		100 ps	8.7×10^{-2}	1.2×10^{-2}	6.7×10^{-3}	8.7×10^{-2}	1.2×10^{-2}	9.1×10^{-3}
		10 ns	8.2	1.1×10^{-1}	5.3×10^{-3}	8.2	1.8×10^{-1}	1.1×10^{-2}
	EVCN	1 ps	7.0×10^{-2}	1.3×10^{-2}	6.8×10^{-3}	7.0×10^{-2}	1.3×10^{-2}	9.1×10^{-3}
		100 ps	8.7×10^{-2}	1.2×10^{-2}	6.7×10^{-3}	8.7×10^{-2}	1.2×10^{-2}	9.1×10^{-3}
		10 ns	9.8×10^{-2}	1.2×10^{-2}	6.7×10^{-3}	9.8×10^{-2}	1.2×10^{-2}	9.1×10^{-3}
Dihedrals	VCN	1 ps	2.1×10^{-2}	1.5×10^{-2}	7.7×10^{-3}	2.1×10^{-2}	1.5×10^{-2}	9.6×10^{-3}
		100 ps	2.8×10^{-2}	9.9×10^{-3}	6.7×10^{-3}	2.8×10^{-2}	1.0×10^{-2}	9.1×10^{-3}
		10 ns	9.4	1.2×10^{-1}	5.3×10^{-3}	9.4	2.0×10^{-1}	1.1×10^{-2}
	EVCN	1 ps	2.1×10^{-2}	1.5×10^{-2}	7.7×10^{-3}	2.1×10^{-2}	1.5×10^{-2}	9.6×10^{-3}
		100 ps	2.7×10^{-2}	9.9×10^{-3}	6.7×10^{-3}	2.7×10^{-2}	9.9×10^{-3}	9.1×10^{-3}
		10 ns	4.5×10^{-2}	1.0×10^{-2}	6.7×10^{-3}	4.5×10^{-2}	1.0×10^{-2}	9.1×10^{-3}

Table 5.1: Predicted transition rates (ns^{-1}) for AIB_9, for different committor and transition rate estimator hyperparameters. The empirical transition rate if 9.1×10^{-3} ns^{-1}.

5.3.4 Transition rate

In Tables 5.1, 5.2, and 5.3, we examine the transition rates predicted by the VCN and EVCN loss functions with various lag times. The loss function and lag time can affect the transition rates through both training the neural network for the committor and application of the rate formula given the trained committor. To distinguish these effects, we evaluate the rates using loss functions and lag times (columns) that are independent of those used to obtain the committors (rows). Unsurprisingly, more accurate committors yield more accurate transition rates. Still, even for a given committor (row), the loss function and lag time used to predict the rate (column) noticeably affect the result. As in the previous section, rates are overestimated at short lag times and approach their correct values as lag times increase. At long lag times, the VCN loss predicts rates that go to zero (approximately proportional to $1/\tau$, as can be seen by substituting a constant committor value for $x \in D$, consistent with the flattening observed above), whereas the EVCN loss predicts rates that plateau near the true value. The choice of input features has a larger impact at shorter lag times. Less expressive input features—such as the CVs for AIB_9 and the RMSDs for Trp-cage and villin—tend to produce larger overestimates of the transition rate.

We examine the estimator in (5.29) in Tables 5.4, 5.5, and 5.6. We observe that using

Committor hyperparameters			Transition rate hyperparameters					
			VCN			EVCN		
			1 ns	100 ns	10 µs	1 ns	100 ns	10 µs
RMSDs	VCN	1 ns	6.7×10^{-1}	8.3×10^{-2}	1.4×10^{-2}	6.9×10^{-1}	1.2×10^{-1}	7.7×10^{-2}
		100 ns	9.1×10^{-1}	6.8×10^{-2}	1.4×10^{-2}	9.8×10^{-1}	1.0×10^{-1}	7.7×10^{-2}
		10 µs	1.8×10^{1}	3.2×10^{-1}	8.8×10^{-3}	2.7×10^{1}	7.7×10^{-1}	8.6×10^{-2}
	EVCN	1 ns	6.8×10^{-1}	8.4×10^{-2}	1.4×10^{-2}	6.9×10^{-1}	1.2×10^{-1}	7.7×10^{-2}
		100 ns	9.3×10^{-1}	6.8×10^{-2}	1.4×10^{-2}	9.6×10^{-1}	1.0×10^{-1}	7.7×10^{-2}
		10 µs	9.6×10^{-1}	7.2×10^{-2}	1.4×10^{-2}	9.9×10^{-1}	1.0×10^{-1}	7.7×10^{-2}
Dihedrals	VCN	1 ns	3.7×10^{-1}	5.9×10^{-2}	1.4×10^{-2}	4.1×10^{-1}	9.8×10^{-2}	7.7×10^{-2}
		100 ns	5.1×10^{-1}	5.4×10^{-2}	1.4×10^{-2}	6.3×10^{-1}	8.9×10^{-2}	7.7×10^{-2}
		10 µs	1.7×10^{1}	3.4×10^{-1}	8.9×10^{-3}	2.7×10^{1}	8.4×10^{-1}	8.7×10^{-2}
	EVCN	1 ns	3.6×10^{-1}	5.5×10^{-2}	1.4×10^{-2}	4.0×10^{-1}	9.4×10^{-2}	7.7×10^{-2}
		100 ns	5.0×10^{-1}	5.5×10^{-2}	1.5×10^{-2}	5.3×10^{-1}	8.5×10^{-2}	7.7×10^{-2}
		10 µs	6.0×10^{-1}	5.6×10^{-2}	1.5×10^{-2}	6.3×10^{-1}	8.6×10^{-2}	7.7×10^{-2}
Distances	VCN	1 ns	2.9×10^{-1}	5.9×10^{-2}	1.5×10^{-2}	3.4×10^{-1}	8.9×10^{-2}	7.7×10^{-2}
		100 ns	4.5×10^{-1}	5.0×10^{-2}	1.4×10^{-2}	5.9×10^{-1}	8.8×10^{-2}	7.7×10^{-2}
		10 µs	1.8×10^{1}	3.3×10^{-1}	8.8×10^{-3}	2.7×10^{1}	7.8×10^{-1}	8.6×10^{-2}
	EVCN	1 ns	3.3×10^{-1}	6.2×10^{-2}	1.5×10^{-2}	3.5×10^{-1}	8.9×10^{-2}	7.7×10^{-2}
		100 ns	4.4×10^{-1}	5.3×10^{-2}	1.5×10^{-2}	4.9×10^{-1}	8.5×10^{-2}	7.7×10^{-2}
		10 µs	4.7×10^{-1}	5.4×10^{-2}	1.4×10^{-2}	5.5×10^{-1}	8.7×10^{-2}	7.7×10^{-2}

Table 5.2: Predicted transition rates (μs^{-1}) for Trp-cage, for different committor and transition rate estimator hyperparameters. The empirical transition rate if 7.7×10^{-2} μs^{-1}.

Committor hyperparameters			Transition rate hyperparameters					
			VCN			EVCN		
			1 ns	100 ns	10 µs	1 ns	100 ns	10 µs
RMSDs	VCN	1 ns	1.6	3.1×10^{-1}	1.6×10^{-2}	1.6	4.1×10^{-1}	3.9×10^{-1}
		100 ns	2.7	2.8×10^{-1}	1.3×10^{-2}	3.6	4.6×10^{-1}	3.9×10^{-1}
		10 µs	1.7×10^{1}	4.0×10^{-1}	9.4×10^{-3}	2.6×10^{1}	9.5×10^{-1}	3.9×10^{-1}
	EVCN	1 ns	1.6	3.1×10^{-1}	1.6×10^{-2}	1.6	4.1×10^{-1}	3.9×10^{-1}
		10 µs	1.9	3.3×10^{-1}	1.7×10^{-2}	1.9	4.0×10^{-1}	3.9×10^{-1}
		100 ns	1.9	3.3×10^{-1}	1.7×10^{-2}	1.9	4.0×10^{-1}	3.9×10^{-1}
Dihedrals	VCN	1 ns	1.1	3.3×10^{-1}	1.7×10^{-2}	1.2	3.9×10^{-1}	3.9×10^{-1}
		100 ns	2.0	2.9×10^{-1}	1.4×10^{-2}	2.5	4.4×10^{-1}	3.9×10^{-1}
		10 µs	1.8×10^{1}	4.3×10^{-1}	9.4×10^{-3}	2.9×10^{1}	1.0	4.0×10^{-1}
	EVCN	1 ns	1.2	3.3×10^{-1}	1.7×10^{-2}	1.2	3.9×10^{-1}	3.9×10^{-1}
		100 ns	1.9	3.4×10^{-1}	1.7×10^{-2}	1.9	4.0×10^{-1}	3.9×10^{-1}
		10 µs	2.0	3.4×10^{-1}	1.7×10^{-2}	2.1	4.0×10^{-1}	3.9×10^{-1}
Distances	VCN	1 ns	8.8×10^{-1}	3.1×10^{-1}	1.6×10^{-2}	9.2×10^{-1}	3.8×10^{-1}	3.9×10^{-1}
		100 ns	1.8	2.8×10^{-1}	1.4×10^{-2}	2.3	4.4×10^{-1}	3.9×10^{-1}
		10 µs	1.9×10^{1}	4.3×10^{-1}	9.4×10^{-3}	3.0×10^{1}	1.0	4.0×10^{-1}
	EVCN	1 ns	9.0×10^{-1}	3.1×10^{-1}	1.7×10^{-2}	9.3×10^{-1}	3.8×10^{-1}	3.9×10^{-1}
		100 ns	1.4	3.2×10^{-1}	1.7×10^{-2}	1.4	3.9×10^{-1}	3.9×10^{-1}
		10 µs	1.5	3.3×10^{-1}	1.7×10^{-2}	1.5	3.9×10^{-1}	3.9×10^{-1}

Table 5.3: Predicted transition rates (ns^{-1}) for villin, for different committor and transition rate estimator hyperparameters. The empirical transition rate if 3.7×10^{-1} μs^{-1}.

Committor hyperparameters			Transition rate hyperparameters					
			(5.29) with VCN			(5.29) with EVCN		
			1 ps, 100 ps	1 ps, 10 ns	100 ps, 10 ns	1 ps, 100 ps	1 ps, 10 ns	100 ps, 10 ns
CVs	VCN	1 ps	1.2×10^{-2}	6.8×10^{-3}	6.7×10^{-3}	1.2×10^{-2}	9.1×10^{-3}	9.1×10^{-3}
		100 ps	1.1×10^{-2}	6.7×10^{-3}	6.6×10^{-3}	1.1×10^{-2}	9.1×10^{-3}	9.1×10^{-3}
		10 ns	2.4×10^{-2}	4.5×10^{-3}	4.3×10^{-3}	9.7×10^{-2}	1.0×10^{-2}	9.4×10^{-3}
	EVCN	1 ps	1.2×10^{-2}	6.8×10^{-3}	6.7×10^{-3}	1.2×10^{-2}	9.1×10^{-3}	9.1×10^{-3}
		100 ps	1.1×10^{-2}	6.7×10^{-3}	6.7×10^{-3}	1.1×10^{-2}	9.1×10^{-3}	9.1×10^{-3}
		10 ns	1.1×10^{-2}	6.7×10^{-3}	6.6×10^{-3}	1.1×10^{-2}	9.1×10^{-3}	9.1×10^{-3}
Dihedrals	VCN	1 ps	1.5×10^{-2}	7.7×10^{-3}	7.6×10^{-3}	1.5×10^{-2}	9.6×10^{-3}	9.6×10^{-3}
		100 ps	9.7×10^{-3}	6.7×10^{-3}	6.6×10^{-3}	9.8×10^{-3}	9.1×10^{-3}	9.1×10^{-3}
		10 ns	2.5×10^{-2}	4.3×10^{-3}	4.1×10^{-3}	1.1×10^{-1}	1.0×10^{-2}	9.4×10^{-3}
	EVCN	1 ps	1.5×10^{-2}	7.7×10^{-3}	7.6×10^{-3}	1.5×10^{-2}	9.6×10^{-3}	9.6×10^{-3}
		100 ps	9.7×10^{-3}	6.7×10^{-3}	6.7×10^{-3}	9.8×10^{-3}	9.1×10^{-3}	9.1×10^{-3}
		10 ns	9.9×10^{-3}	6.7×10^{-3}	6.6×10^{-3}	1.0×10^{-2}	9.1×10^{-3}	9.1×10^{-3}

Table 5.4: Transition rates (ns^{-1}) predicted using (5.29) for AIB_9, for different committor and transition rate estimator hyperparameters. The empirical transition rate if 9.1×10^{-3} ns^{-1}.

Committor hyperparameters			Transition rate hyperparameters					
			(5.29) with VCN			(5.29) with EVCN		
			1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs	1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs
RMSDs	VCN	1 ns	7.7×10^{-2}	1.4×10^{-2}	1.3×10^{-2}	1.1×10^{-1}	7.7×10^{-2}	7.7×10^{-2}
		100 ns	5.9×10^{-2}	1.4×10^{-2}	1.3×10^{-2}	9.5×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		10 µs	1.5×10^{-1}	7.1×10^{-3}	5.6×10^{-3}	5.1×10^{-1}	8.3×10^{-2}	7.9×10^{-2}
	EVCN	1 ns	7.8×10^{-2}	1.4×10^{-2}	1.3×10^{-2}	1.1×10^{-1}	7.7×10^{-2}	7.7×10^{-2}
		100 ns	5.9×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	9.2×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		10 µs	6.3×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	9.4×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
Dihedrals	VCN	1 ns	5.6×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	9.5×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		100 ns	4.9×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	8.4×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		10 µs	1.7×10^{-1}	7.2×10^{-3}	5.6×10^{-3}	5.7×10^{-1}	8.4×10^{-2}	7.9×10^{-2}
	EVCN	1 ns	5.2×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	9.1×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		100 ns	5.0×10^{-2}	1.5×10^{-2}	1.4×10^{-2}	8.0×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		10 µs	5.1×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	8.0×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
Distances	VCN	1 ns	5.7×10^{-2}	1.5×10^{-2}	1.4×10^{-2}	8.6×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		100 ns	4.6×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	8.3×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		10 µs	1.5×10^{-1}	7.1×10^{-3}	5.6×10^{-3}	5.1×10^{-1}	8.3×10^{-2}	7.9×10^{-2}
	EVCN	1 ns	5.9×10^{-2}	1.5×10^{-2}	1.4×10^{-2}	8.6×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		100 ns	4.9×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	8.1×10^{-2}	7.7×10^{-2}	7.7×10^{-2}
		10 µs	4.9×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	8.3×10^{-2}	7.7×10^{-2}	7.7×10^{-2}

Table 5.5: Transition rates (μs^{-1}) predicted using (5.29) for Trp-cage, for different committor and transition rate estimator hyperparameters. The empirical transition rate if 7.7×10^{-2} μs^{-1}.

Committor hyperparameters			Transition rate hyperparameters					
			(5.29) with VCN			(5.29) with EVCN		
			1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs	1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs
RMSDs	VCN	1 ns	2.9×10^{-1}	1.5×10^{-2}	1.3×10^{-2}	4.0×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		100 ns	2.5×10^{-1}	1.3×10^{-2}	1.0×10^{-2}	4.3×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		10 µs	2.4×10^{-1}	7.8×10^{-3}	5.5×10^{-3}	7.0×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
	EVCN	1 ns	3.0×10^{-1}	1.6×10^{-2}	1.3×10^{-2}	4.0×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		100 ns	3.2×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		10 µs	3.1×10^{-1}	1.6×10^{-2}	1.3×10^{-2}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
Dihedrals	VCN	1 ns	3.2×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		100 ns	2.7×10^{-1}	1.4×10^{-2}	1.2×10^{-2}	4.2×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		10 µs	2.5×10^{-1}	7.6×10^{-3}	5.2×10^{-3}	7.5×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
	EVCN	1 ns	3.2×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.8×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		100 ns	3.2×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		10 µs	3.2×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
Distances	VCN	1 ns	3.0×10^{-1}	1.6×10^{-2}	1.3×10^{-2}	3.8×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		100 ns	2.6×10^{-1}	1.4×10^{-2}	1.1×10^{-2}	4.2×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		10 µs	2.5×10^{-1}	7.5×10^{-3}	5.1×10^{-3}	7.4×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
	EVCN	1 ns	3.0×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.8×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		100 ns	3.1×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.8×10^{-1}	3.9×10^{-1}	3.9×10^{-1}
		10 µs	3.2×10^{-1}	1.7×10^{-2}	1.4×10^{-2}	3.8×10^{-1}	3.9×10^{-1}	3.9×10^{-1}

Table 5.6: Transition rates (ns^{-1}) predicted using (5.29) for villin, for different committor and transition rate estimator hyperparameters. The empirical transition rate if 3.7×10^{-1} μs^{-1}.

(5.29) with EVCN consistently yields more accurate rate estimates than (5.19) by itself. Rates predicted using the VCN loss (5.22) or its counterpart with (5.29) are not variational and tend to underestimate the rate at all but the shortest lag times. For both VCN and EVCN, the choice of the longer of the two lag times in (5.29) appears to matter more than the choice of the shorter of the two lag times.

5.3.5 Mean squared error

In Tables 5.7, 5.8, and 5.9, we show MSE estimates (mean of 10 models) for the predicted committor under various training hyperparameters. Focusing first on the column computed with (5.28), at short lag times, committors trained with the VCN and EVCN loss function have similar errors. Notably, more expressive features do not always yield lower error. For EVCN, increasing the lag time tends to reduce the MSE until it plateaus, potentially above zero if the input features fail to capture all relevant degrees of freedom. In contrast, for VCN, the MSE initially decreases with lag time, but then increases as the approximation breaks

Committor hyperparameters			Mean squared error hyperparameters						
			(5.28)	(5.30) with VCN			(5.30) with EVCN		
				1 ps, 100 ps	1 ps, 10 ns	100 ps, 10 ns	1 ps, 100 ps	1 ps, 10 ns	100 ps, 10 ns
CVs	VCN	1 ps	1.6×10^{-5}	5.8×10^{-5}	6.3×10^{-5}	5.8×10^{-4}	5.8×10^{-5}	6.1×10^{-5}	3.5×10^{-4}
		100 ps	-5.0×10^{-4}	7.6×10^{-5}	8.0×10^{-5}	5.0×10^{-4}	7.6×10^{-5}	7.8×10^{-5}	2.7×10^{-4}
		10 ns	1.9×10^{-2}	8.2×10^{-3}	8.2×10^{-3}	1.0×10^{-2}	8.1×10^{-3}	8.2×10^{-3}	1.7×10^{-2}
	EVCN	1 ps	1.6×10^{-5}	5.8×10^{-5}	6.3×10^{-5}	5.8×10^{-4}	5.8×10^{-5}	6.1×10^{-5}	3.5×10^{-4}
		100 ps	-4.9×10^{-4}	7.7×10^{-5}	8.1×10^{-5}	5.0×10^{-4}	7.7×10^{-5}	7.8×10^{-5}	2.7×10^{-4}
		10 ns	-5.0×10^{-4}	8.7×10^{-5}	9.2×10^{-5}	5.2×10^{-4}	8.7×10^{-5}	8.9×10^{-5}	2.9×10^{-4}
Dihedrals	VCN	1 ps	4.9×10^{-3}	6.4×10^{-6}	1.3×10^{-5}	7.1×10^{-4}	6.4×10^{-6}	1.2×10^{-5}	5.2×10^{-4}
		100 ps	-6.8×10^{-4}	1.8×10^{-5}	2.1×10^{-5}	3.2×10^{-4}	1.8×10^{-5}	1.9×10^{-5}	9.0×10^{-5}
		10 ns	2.2×10^{-2}	9.3×10^{-3}	9.3×10^{-3}	1.1×10^{-2}	9.2×10^{-3}	9.3×10^{-3}	1.9×10^{-2}
	EVCN	1 ps	4.9×10^{-3}	6.4×10^{-6}	1.3×10^{-5}	7.1×10^{-4}	6.4×10^{-6}	1.2×10^{-5}	5.2×10^{-4}
		100 ps	-6.7×10^{-4}	1.7×10^{-5}	2.0×10^{-5}	3.2×10^{-4}	1.7×10^{-5}	1.8×10^{-5}	8.8×10^{-5}
		10 ns	-7.1×10^{-4}	3.5×10^{-5}	3.9×10^{-5}	3.6×10^{-4}	3.5×10^{-5}	3.6×10^{-5}	1.3×10^{-4}

Table 5.7: Mean squared error in the predicted committor for AIB_9, for different committor and MSE estimator hyperparameters.

Committor hyperparameters			Mean squared error hyperparameters						
			(5.28)	(5.30) with VCN			(5.30) with EVCN		
				1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs	1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs
RMSDs	VCN	1 ns	4.8×10^{-3}	6.0×10^{-4}	6.6×10^{-4}	7.0×10^{-3}	5.8×10^{-4}	6.1×10^{-4}	3.9×10^{-3}
		100 ns	3.3×10^{-3}	8.6×10^{-4}	9.0×10^{-4}	5.4×10^{-3}	8.8×10^{-4}	9.0×10^{-4}	2.7×10^{-3}
		10 µs	9.0×10^{-2}	1.7×10^{-2}	1.8×10^{-2}	3.2×10^{-2}	2.7×10^{-2}	2.7×10^{-2}	6.9×10^{-2}
	EVCN	1 ns	4.8×10^{-3}	6.0×10^{-4}	6.6×10^{-4}	7.1×10^{-3}	5.8×10^{-4}	6.1×10^{-4}	3.9×10^{-3}
		100 ns	2.7×10^{-3}	8.7×10^{-4}	9.1×10^{-4}	5.4×10^{-3}	8.6×10^{-4}	8.8×10^{-4}	2.4×10^{-3}
		10 µs	2.8×10^{-3}	9.0×10^{-4}	9.5×10^{-4}	5.8×10^{-3}	8.9×10^{-4}	9.1×10^{-4}	2.7×10^{-3}
Dihedrals	VCN	1 ns	3.4×10^{-3}	3.1×10^{-4}	3.6×10^{-4}	4.5×10^{-3}	3.1×10^{-4}	3.3×10^{-4}	2.2×10^{-3}
		100 ns	1.1×10^{-3}	4.6×10^{-4}	4.9×10^{-4}	4.0×10^{-3}	5.5×10^{-4}	5.5×10^{-4}	1.2×10^{-3}
		10 µs	1.0×10^{-1}	1.7×10^{-2}	1.7×10^{-2}	3.3×10^{-2}	2.6×10^{-2}	2.7×10^{-2}	7.6×10^{-2}
	EVCN	1 ns	2.9×10^{-3}	3.1×10^{-4}	3.5×10^{-4}	4.1×10^{-3}	3.1×10^{-4}	3.2×10^{-4}	1.8×10^{-3}
		100 ns	6.5×10^{-4}	4.5×10^{-4}	4.9×10^{-4}	4.1×10^{-3}	4.5×10^{-4}	4.6×10^{-4}	8.2×10^{-4}
		10 µs	7.7×10^{-4}	5.5×10^{-4}	5.9×10^{-4}	4.2×10^{-3}	5.5×10^{-4}	5.6×10^{-4}	9.3×10^{-4}
Distances	VCN	1 ns	1.2×10^{-3}	2.3×10^{-4}	2.7×10^{-4}	4.5×10^{-3}	2.5×10^{-4}	2.6×10^{-4}	1.2×10^{-3}
		100 ns	1.2×10^{-3}	4.1×10^{-4}	4.4×10^{-4}	3.6×10^{-3}	5.0×10^{-4}	5.1×10^{-4}	1.1×10^{-3}
		10 µs	9.2×10^{-2}	1.8×10^{-2}	1.8×10^{-2}	3.2×10^{-2}	2.7×10^{-2}	2.7×10^{-2}	7.0×10^{-2}
	EVCN	1 ns	1.1×10^{-3}	2.7×10^{-4}	3.1×10^{-4}	4.7×10^{-3}	2.6×10^{-4}	2.7×10^{-4}	1.2×10^{-3}
		100 ns	6.6×10^{-4}	3.9×10^{-4}	4.2×10^{-4}	3.9×10^{-3}	4.0×10^{-4}	4.1×10^{-4}	8.9×10^{-4}
		10 µs	9.3×10^{-4}	4.2×10^{-4}	4.5×10^{-4}	4.0×10^{-3}	4.6×10^{-4}	4.7×10^{-4}	1.1×10^{-3}

Table 5.8: Mean squared error in the predicted committor for Trp-cage, for different committor and MSE estimator hyperparameters.

Committor hyperparameters			Mean squared error hyperparameters						
			(5.28)	(5.30) with VCN			(5.30) with EVCN		
				1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs	1 ns, 100 ns	1 ns, 10 µs	100 ns, 10 µs
RMSDs	VCN	1 ns	4.3×10^{-3}	1.3×10^{-3}	1.5×10^{-3}	2.9×10^{-2}	1.2×10^{-3}	1.2×10^{-3}	2.3×10^{-3}
		100 ns	9.7×10^{-3}	2.4×10^{-3}	2.7×10^{-3}	2.7×10^{-2}	3.2×10^{-3}	3.2×10^{-3}	7.6×10^{-3}
		10 µs	5.9×10^{-2}	1.6×10^{-2}	1.7×10^{-2}	4.0×10^{-2}	2.6×10^{-2}	2.6×10^{-2}	5.7×10^{-2}
	EVCN	1 ns	4.1×10^{-3}	1.3×10^{-3}	1.6×10^{-3}	3.0×10^{-2}	1.2×10^{-3}	1.2×10^{-3}	2.0×10^{-3}
		100 ns	3.6×10^{-3}	1.5×10^{-3}	1.8×10^{-3}	3.2×10^{-2}	1.5×10^{-3}	1.5×10^{-3}	1.5×10^{-3}
		10 µs	3.6×10^{-3}	1.6×10^{-3}	1.9×10^{-3}	3.2×10^{-2}	1.5×10^{-3}	1.5×10^{-3}	1.5×10^{-3}
Dihedrals	VCN	1 ns	2.5×10^{-3}	7.9×10^{-4}	1.1×10^{-3}	3.2×10^{-2}	7.8×10^{-4}	7.7×10^{-4}	5.4×10^{-4}
		100 ns	7.6×10^{-3}	1.7×10^{-3}	1.9×10^{-3}	2.8×10^{-2}	2.1×10^{-3}	2.1×10^{-3}	5.6×10^{-3}
		10 µs	6.7×10^{-2}	1.8×10^{-2}	1.8×10^{-2}	4.2×10^{-2}	2.8×10^{-2}	2.8×10^{-2}	6.4×10^{-2}
	EVCN	1 ns	2.3×10^{-3}	8.5×10^{-4}	1.2×10^{-3}	3.2×10^{-2}	8.2×10^{-4}	8.1×10^{-4}	4.4×10^{-4}
		100 ns	3.3×10^{-3}	1.6×10^{-3}	1.9×10^{-3}	3.2×10^{-2}	1.5×10^{-3}	1.5×10^{-3}	1.4×10^{-3}
		10 µs	3.5×10^{-3}	1.7×10^{-3}	2.0×10^{-3}	3.3×10^{-2}	1.7×10^{-3}	1.7×10^{-3}	1.6×10^{-3}
Distances	VCN	1 ns	1.4×10^{-3}	5.8×10^{-4}	8.7×10^{-4}	2.9×10^{-2}	5.4×10^{-4}	5.3×10^{-4}	-5.3×10^{-4}
		100 ns	6.8×10^{-3}	1.5×10^{-3}	1.7×10^{-3}	2.7×10^{-2}	1.9×10^{-3}	2.0×10^{-3}	4.8×10^{-3}
		10 µs	6.7×10^{-2}	1.9×10^{-2}	1.9×10^{-2}	4.3×10^{-2}	2.9×10^{-2}	3.0×10^{-2}	6.4×10^{-2}
	EVCN	1 ns	1.3×10^{-3}	6.0×10^{-4}	8.9×10^{-4}	3.0×10^{-2}	5.5×10^{-4}	5.4×10^{-4}	-6.2×10^{-4}
		100 ns	2.4×10^{-3}	1.0×10^{-3}	1.3×10^{-3}	3.1×10^{-2}	1.0×10^{-3}	1.0×10^{-3}	4.6×10^{-4}
		10 µs	2.4×10^{-3}	1.1×10^{-3}	1.4×10^{-3}	3.1×10^{-2}	1.1×10^{-3}	1.1×10^{-3}	4.3×10^{-4}

Table 5.9: Mean squared error in the predicted committor for villin, for different committor and MSE estimator hyperparameters.

down.

In Tables 5.7, 5.8, and 5.9, we also compare (5.30) with (5.28). The two estimators generally track each other, but (5.30) can differ from (5.28) by an order of magnitude when using VCN with long lag times. For villin, the approximation becomes negative for ECVN at long lag times, likely due to statistical noise. In contrast to (5.29), for a given committor (row), the choice of the shorter of the two lag times appears to have more impact on the MSE approximated by (5.30) than the choice of the longer of the two lag times.

5.4 Discussion

In this work, we derived an exact multiple-time-step expression for the transition rate and demonstrated, using models of molecular systems, that a variational scheme based on this expression yields accurate committors and transition rates with relatively little sensitivity to the choice of lag time. Our approach extends previous methods [158], which are valid only in the limit of a single-time-step lag time and break down at longer lag times.

We suggest several directions for improvement. For datasets composed of short trajectories, long lag times are inaccessible and so the method may struggle to capture slow modes. This may be addressed by using loss functions with greater sensitivity to these modes, such as (5.29). More advanced approaches incorporating memory, such as those proposed for slow modes in Liu et al. [183], may further improve accuracy.

Overfitting is also a challenge, and standard regularization techniques are often ineffective. We expect that integrating over lag times, as in Lorpaiboon et al. [9], may reduce overfitting and improve robustness. We are exploring regularization strategies that directly restrain the committor to depend on physical or learned CVs in a simple way.

Finally, the variational methods considered in the present study assume detailed balance and equilibrium sampling (or equilibrium reweighting). Theoretical [184] and empirical [5, 147] evidence indicates that data distributed according to the equilibrium distribution are not sufficient to accurately learn rare-event statistics. The schemes proposed in Strahan et al. [135, 147] allow both non-equilibrium data distributions and irreversible dynamics, and have shown promise on several benchmark problems. However, these schemes have tradeoffs relative to the scheme proposed in this article. The method proposed in Strahan et al. [147] requires a trajectory dataset with multiple short forward simulations for every initial configuration, and the method proposed in Strahan et al. [135] uses a training procedure that is not the gradient of any loss function when the data are non-equilibrium or the dynamics are irreversible, which can complicate training. Developing a method that combines the strengths of all of these approaches remains an open problem.

5.5 Derivation of the transition rate

In this subsection, we use transition path theory to derive expressions for the transition rate in terms of committors and finite-length trajectories. In brief, we first define the transition rate (Φ) in terms of the number of transitions within a time interval, then express Φ as an

average of changes in a reaction coordinate over single and multiple time steps.

We denote discrete time intervals by $[t, t'] = \{t, t+\epsilon, \ldots, t'\}$. A transition path is a trajectory segment $X_{[s,s']} = (X_s, X_{s+\epsilon}, \ldots, X_{s'})$ with $X_s \in A$, $X_{s'} \in B$, and $X_t \in D$ for $s < t < s'$. We define

$$H_{AB}(X_{[s,s']}) = \mathbb{1}_A(X_s) \prod_{t \in [s+\epsilon, s-\epsilon]} \mathbb{1}_D(X_t) \mathbb{1}_B(X_{s'}), \tag{5.35}$$

which is 1 if $X_{[s,s']}$ is a transition path and 0 otherwise. The number of transition paths from A to B within the trajectory segment $X_{[t,t']}$ is

$$N_{AB}(X_{[t,t']}) = \sum_{s \in [t, t'-\epsilon]} \sum_{s' \in [s+\epsilon, t']} H_{AB}(X_{[s,s']}), \tag{5.36}$$

and the transition rate Φ is the mean of N_{AB} per time:

$$\Phi = \lim_{T \to \infty} \frac{1}{2T} N_{AB}(X_{[-T,T]}). \tag{5.37}$$

We now express (5.37) in terms of a reaction coordinate ξ that satisfies $\xi(x) = 0$ for $x \in A$ and $\xi(x) = 1$ for $x \in B$. We start with the identity

$$H_{AB}(X_{[s,s']}) = H_{AB}(X_{[s,s']}) \sum_{t \in [s, s'-\epsilon]} (\xi(X_{t+\epsilon}) - \xi(X_t)), \tag{5.38}$$

which states that the total change in progress along a transition path is one. Mathematically, the sum telescopes, and $\xi(X_{s'}) - \xi(X_s) = 1$ for transition paths because $X_s \in A$ and $X_{s'} \in B$. We substitute this identity into the summand of (5.36) and interchange the sums, yielding

$$N_{AB}(X_{[t,t']}) = \sum_{t'' \in [t, t'-\epsilon]} \sum_{s \in [t, t'']} \sum_{s' \in [t''+\epsilon, t']} H_{AB}(X_{[s,s']})(\xi(X_{t''+\epsilon}) - \xi(X_{t''})). \tag{5.39}$$

We now set $t = -T$ and $t' = T$ and let $T \to \infty$; (5.37) becomes

$$\Phi = \lim_{T\to\infty} \frac{1}{2T} \sum_{t\in[-T,T-\epsilon]} F_{t,t+\epsilon}, \tag{5.40}$$

where $F_{t,t+\epsilon}$ is the reaction progress from time t to time $t+\epsilon$:

$$F_{t,t+\epsilon} = \sum_{s\in[-\infty,t]} \sum_{s'\in[t+\epsilon,\infty]} H_{AB}(X_{[s,s']})(\xi(X_{t+\epsilon}) - \xi(X_t)). \tag{5.41}$$

The standard transition path theory expression for the transition rate uses single-step trajectories. In that case, the reaction progress (5.41) can be expressed as

$$F_{t,t+\epsilon} = \mathbb{1}_A(X_{\bar{S}_t})\mathbb{1}_B(X_{S_{t+\epsilon}})(\xi(X_{t+\epsilon}) - \xi(X_t)), \tag{5.42}$$

using the identities

$$\mathbb{1}_A(X_{\bar{S}_t}) = \sum_{s\in[-\infty,t]} \mathbb{1}_A(X_s) \prod_{t'\in[s+\epsilon,t]} \mathbb{1}_D(X_{t'}), \tag{5.43}$$

$$\mathbb{1}_B(X_{S_t}) = \sum_{s\in[t,\infty]} \prod_{t'\in[t,s-\epsilon]} \mathbb{1}_D(X_{t'})\mathbb{1}_B(X_s). \tag{5.44}$$

Using (5.2) and (5.3), the expectation of (5.42) conditioned on single-step trajectories $X_{[t,t+\epsilon]} = (X_t, X_{t+\epsilon})$, denoted by $f(X_{[t,t+\epsilon]})$, is

$$f(X_{[t,t+\epsilon]}) = \bar{q}(X_t)q(X_{t+\epsilon})(\xi(X_{t+\epsilon}) - \xi(X_t)). \tag{5.45}$$

We can then use ergodicity to express (5.40) as

$$\Phi = \frac{1}{\epsilon}\,\mathbb{E}[f(X_{[0,\epsilon]})], \tag{5.46}$$

where the expectation is over paths sampled at equilibrium.

In previous work, we proposed and applied a multiple-step expression for the transition rate. [5, 185] There, we applied (5.9) and (5.10) to an average of (5.46):

$$\Phi = \frac{1}{\tau} \sum_{t \in [0, \tau - \epsilon]} \mathbb{E}[f(X_{[t,t+\epsilon]})] \tag{5.47}$$

$$= \frac{1}{\tau} \sum_{t \in [0, \tau - \epsilon]} \mathbb{E}[\bar{q}(X_{0 \vee \bar{S}_t}) q(X_{\tau \wedge S_{t+\epsilon}})(\xi(X_{t+\epsilon}) - \xi(X_t))]. \tag{5.48}$$

In this work, we derive an equivalent expression that depends on committors only at the endpoints. To this end, we note that the reaction progress from time t to time t' can be written as a sum over single steps:

$$F_{t,t'} = \sum_{t'' \in [t, t' - \epsilon]} F_{t'', t''+\epsilon}. \tag{5.49}$$

We then substitute (5.41) into (5.49), split the sums by whether the transition path starts

before time t and ends after time t', and interchange and telescope the sums. The result is

$$\begin{aligned}
F_{t,t'} = &\sum_{s\in[-\infty,t-\epsilon]}\sum_{s'\in[t'+\epsilon,\infty]} H_{AB}(X_{[s,s']})(\xi(X_{t'})-\xi(X_t)) \\
&+ \sum_{s\in[t,t']}\sum_{s'\in[t'+\epsilon,\infty]} H_{AB}(X_{[s,s']})(\xi(X_{t'})-0) \\
&+ \sum_{s\in[-\infty,t-\epsilon]}\sum_{s'\in[t,t']} H_{AB}(X_{[s,s']})(1-\xi(X_t)) \\
&+ \sum_{s\in[t,t'-\epsilon]}\sum_{s'\in[s+\epsilon,t']} H_{AB}(X_{[s,s']})(1-0) \qquad (5.50) \\
= &\mathbb{1}_A(X_{\bar{S}_t})\mathbb{1}_D(X_{t'\wedge S_t})\mathbb{1}_B(X_{S_{t'}})(\xi(X_{t'})-\xi(X_t)) \\
&+ \mathbb{1}_A(X_{t\vee\bar{S}_{t'}})\mathbb{1}_B(X_{S_{t'}})(\xi(X_t)-0) \\
&+ \mathbb{1}_A(X_{\bar{S}_t})\mathbb{1}_B(X_{t'\wedge S_t})(1-\xi(X_t)) \\
&+ N_{AB}(X_{[t,t']}), \qquad (5.51)
\end{aligned}$$

where we have used (5.43), (5.44), and

$$\mathbb{1}_A(X_{t\vee\bar{S}_{t'}}) = \sum_{s\in[t,t']}\mathbb{1}_A(X_s)\prod_{t''\in[s+\epsilon,t']}\mathbb{1}_D(X_{t''}), \qquad (5.52)$$

$$\mathbb{1}_B(X_{t'\wedge S_t}) = \sum_{s\in[t,t']}\prod_{t''\in[t,s-\epsilon]}\mathbb{1}_D(X_{t''})\mathbb{1}_B(X_s), \qquad (5.53)$$

$$\mathbb{1}_D(X_{t'\wedge S_t}) = \mathbb{1}_D(X_{t\vee\bar{S}_{t'}}) = \prod_{t''\in[t,t']}\mathbb{1}_D(X_{t''}). \qquad (5.54)$$

For times within the interval $[t,t']$, these indicator functions identify whether the system last exited A before time t', first entered B after time t, or was always in D, respectively. The

expectation of $F_{t,t'}$ conditioned on multiple-step trajectories $X_{[t,t']}$, denoted by $f(X_{[t,t']})$, is

$$\begin{aligned} f(X_{[t,t']}) = \bar{q}(X_t)\mathbb{1}_D(X_{t'\wedge S_t})q(X_{t'})(\xi(X_{t'}) - \xi(X_t)) \\ + \mathbb{1}_A(X_{t\vee \bar{S}_{t'}})q(X_{t'})(\xi(X_{t'}) - 0) \\ + \bar{q}(X_t)\mathbb{1}_B(X_{t'\wedge S_t})(1 - \xi(X_t)) \\ + N_{AB}(X_{[t,t']}), \end{aligned} \tag{5.55}$$

where we have used (5.2) and (5.3). By ergodicity, (5.40) is

$$\Phi = \frac{1}{\tau}\,\mathbb{E}[f(X_{[0,\tau]})]. \tag{5.56}$$

The key point is that transition paths observed in the time interval $[0, \tau]$ can be classified into four cases by their starting and ending times: starting before 0 and ending after τ, starting before 0 and ending within $[0, \tau]$, starting within $[0, \tau]$ and ending after τ, and both starting and ending within $[0, \tau]$. Each term in (5.13) corresponds to one of these cases and represents the product of the probability of observing that type of transition path and the progress it contributes within the time interval $[0, \tau]$.

5.6 Mode decomposition of the variational loss function

When detailed balance is satisfied, we can write a perturbation η to q as in (5.24) as a sum over eigenvalues $\{e^{-\tau/\sigma_k}\}$ and eigenfunctions $\{v_k\}$:

$$\mathbb{E}[\mathbb{1}_D(X_{\tau\wedge S_0})\eta(X_\tau) \mid X_0 = x] = \sum_k e^{-\tau/\sigma_k} v_k(x)\langle v_k, \eta\rangle, \tag{5.57}$$

where $\tau \geq 0$, $\langle v_k, \eta \rangle = \int v_k(x)\eta(x)\pi(x)\,\mathrm{d}x$, $\langle v_k, v_l \rangle = \delta_{kl}$, and δ_{kl} is the Kronecker delta. The estimated transition rate is then

$$\tilde{L}_\tau(q+\eta) = \Phi + \frac{1}{2\tau}\,\mathbb{E}[\eta(X_0)^2 + \eta(X_\tau)^2 - 2\mathbb{1}_D(X_{\tau\wedge S_0})\eta(X_0)\eta(X_\tau)] \tag{5.58}$$

$$= \Phi + \sum_k \frac{1 - e^{-\tau/\sigma_k}}{\tau}\langle v_k, \eta\rangle^2. \tag{5.59}$$

Since $\langle v_k, \eta\rangle^2 \geq 0$ and $(1 - e^{-\tau/\sigma_k})/\tau$ is a nonnegative, monotonically decreasing function of τ, the estimated transition rate decreases (and accuracy increases) with increasing τ for fixed η. For $\sigma_k \ll \tau$, $(1 - e^{-\tau/\sigma_k})/\tau \approx 1/\tau$, while for $\sigma_k \gg \tau$, $(1 - e^{-\tau/\sigma_k})/\tau \approx 1/\sigma_k$. Thus, the loss is similarly sensitive to components of η with decay times shorter than τ, and progressively less sensitive to those with longer decay times.

5.7 Mode decomposition of the mean squared error and alternative transition rate expression

As shown by the mode expansion in (5.59), transition rate estimates at short lag times tend to overestimate the true rate and can be improved by extrapolating the rate to infinite lag time. We do so linearly using the points $(1/\tau_1, \tilde{L}_{\tau_1}(u))$ and $(1/\tau_2, \tilde{L}_{\tau_2}(u))$:

$$\tilde{L}_{\tau_1,\tau_2}(u) = \tilde{L}_{\tau_1}(u) + \frac{\tilde{L}_{\tau_2}(u) - \tilde{L}_{\tau_1}(u)}{1/\tau_2 - 1/\tau_1}\Big(\frac{1}{\infty} - \frac{1}{\tau_1}\Big) \tag{5.60}$$

$$= \frac{\tau_1\tilde{L}_{\tau_1}(u) - \tau_2\tilde{L}_{\tau_2}(u)}{\tau_1 - \tau_2}. \tag{5.61}$$

Surprisingly, (5.61) is variational: it satisfies $\tilde{L}_{\tau_1,\tau_2}(u) \geq \Phi$ for all $\tau_1, \tau_2 \geq 0$. Using the mode expansion in (5.59), we find

$$\tilde{L}_{\tau_1,\tau_2}(q+\eta) = \Phi + \sum_k \frac{e^{-\tau_1/\sigma_k} - e^{-\tau_2/\sigma_k}}{\tau_2 - \tau_1}\langle v_k, \eta\rangle^2, \tag{5.62}$$

in which each coefficient of $\langle v_k, \eta \rangle^2$ is nonnegative. By the same approach, it can be shown that (5.61) is, in fact, the minimum variational affine combination of $\tilde{L}_{\tau_1}$ and $\tilde{L}_{\tau_2}$. To the best of our knowledge, the variational expression in (5.61) has not been proposed previously.

We now examine how $\tilde{L}_{\tau_1,\tau_2}(u)$ varies with τ_1 and τ_2. It is symmetric ($\tilde{L}_{\tau_1,\tau_2}(u) = \tilde{L}_{\tau_2,\tau_1}(u)$), decreases with increasing τ_1 or τ_2, and is bounded by

$$\tilde{L}_{\tau_2}(u) = \tilde{L}_{0,\tau_2} \geq \tilde{L}_{\tau_1,\tau_2}(u) \geq \tilde{L}_{\infty,\tau_2}(u) = \Phi. \tag{5.63}$$

To understand the behavior away from these bounds, we consider a representative case $\tilde{L}_{\tau,\tau}(u) = \mathrm{d}(\tau \tilde{L}_\tau(u)) \Big/ \mathrm{d}\tau$, which has a simple mode expansion:

$$\tilde{L}_{\tau,\tau}(q+\eta) = \Phi + \sum_k \frac{e^{-\tau/\sigma_k}}{\sigma_k} \langle v_k, \eta \rangle^2. \tag{5.64}$$

As τ increases, each coefficient of $\langle v_k, \eta \rangle^2$ exponentially decays from $1/\sigma_k$ to zero. Unlike $\tilde{L}_\tau$, $\tilde{L}_{\tau,\tau}$ is not monotonic in σ_k: the coefficient is zero at both $\sigma_k = 0$ and $\sigma_k = \infty$, and has a maximum of $1/(\tau e)$ at $\sigma_k = \tau$.

Similarly, we can use the mode expansion in (5.59) to understand the associated approximation of the MSE in (5.30). To this end, we write

$$\mathrm{MSE}_{\tau_1,\tau_2}(q+\eta) = \mathrm{MSE}(q+\eta) - \sum_k \frac{e^{-\tau_1/\sigma_k}/\tau_1 - e^{-\tau_2/\sigma_k}/\tau_2}{1/\tau_1 - 1/\tau_2} \langle v_k, \eta \rangle^2. \tag{5.65}$$

Note the mode expansion of the MSE is

$$\mathrm{MSE}(q+\eta) = \sum_k \langle v_k, \eta \rangle^2. \tag{5.66}$$

Because $0 \leq (e^{-\tau_1/\sigma_k}/\tau_1 - e^{-\tau_2/\sigma_k}/\tau_2)/(1/\tau_1 - 1/\tau_2) \leq 1$, this expression always underestimates the MSE in the limit of infinite data. It can be shown that (5.30) is the maximum linear

combination of $\tilde{L}_{\tau_1}$ and $\tilde{L}_{\tau_2}$ that is a lower bound for the MSE. More precisely, $\mathrm{MSE}_{\tau_1,\tau_2}$ is symmetric ($\mathrm{MSE}_{\tau_1,\tau_2}(u) = \mathrm{MSE}_{\tau_2,\tau_1}(u)$), increases with increasing τ_1 or τ_2, and is bounded by

$$0 = \mathrm{MSE}_{0,\tau_2}(u) \le \mathrm{MSE}_{\tau_1,\tau_2}(u) \le \mathrm{MSE}_{\infty,\tau_2}(u) = \tau_2(\tilde{L}_{\tau_2}(u) - \Phi) \le \mathrm{MSE}(u). \tag{5.67}$$

For τ_1 close to τ_2, rather than near 0 or ∞, we consider a representative case $\mathrm{MSE}_{\tau,\tau}(u) = \mathrm{d}\tilde{L}_\tau(u) \Big/ \mathrm{d}(1/\tau)$, which has a simple mode expansion:

$$\mathrm{MSE}_{\tau,\tau}(q+\eta) = \mathrm{MSE}(q+\eta) - \sum_k (1 + \tau/\sigma_k) e^{-\tau/\sigma_k} \langle v_k, \eta \rangle^2. \tag{5.68}$$

We thus see that $\mathrm{MSE}_{\tau,\tau}(q+\eta)$ performs well when $\tau \gg \sigma_k$, but performs poorly when $\tau \ll \sigma_k$, and therefore is insensitive to slowly decaying modes.

5.8 Connection to the mean squared residual

Another approach to solving for the committor is to minimize the mean squared residual (MSR) [147],

$$\mathrm{MSR}_\tau(u) = \int (u(x) - \mathbb{E}[\hat{u}(X_{\tau \wedge S_0}) \mid X_0 = x])^2 \mu(x)\, \mathrm{d}x, \tag{5.69}$$

where μ is an arbitrary distribution of initial states. This objective typically requires two or more independent simulations from each starting configuration x to avoid bias due to the double sampling problem [135, 147]. However, when the dynamics satisfy detailed balance, forward-in-time and backward-in-time conditional expectations are equal, and so (5.69) with $\mu = \pi$ can be expressed as

$$\mathrm{MSR}_\tau(u) = \mathbb{E}[(u(X_\tau) - \hat{u}(X_{2\tau \wedge S_\tau}))(u(X_\tau) - \hat{u}(X_{0 \vee \bar{S}_\tau}))], \tag{5.70}$$

because $X_{2\tau\wedge S_\tau}$ and $X_{0\vee\bar{S}_\tau}$ are independent samples conditioned on X_τ. The squared residual can be written as

$$(u(X_\tau)-\hat{u}(X_{2\tau\wedge S_\tau}))(u(X_\tau)-\hat{u}(X_{0\vee\bar{S}_\tau})) = g(X_{[0,\tau]},u)+g(X_{[\tau,2\tau]},u)-g(X_{[0,2\tau]},u), \quad (5.71)$$

allowing the MSR to be expressed in terms of the EVCN loss (5.19):

$$\mathrm{MSR}_\tau(u) = 2\tau(\tilde{L}_\tau(u) - \tilde{L}_{2\tau}(u)) = \mathrm{MSE}_{\tau,2\tau}(u). \quad (5.72)$$

As discussed in Appendix 5.7, $\mathrm{MSE}_{\tau,2\tau}(u)$ is variational, and thus $\mathrm{MSR}_\tau(u)$ is variational as well.

5.9 Further discussion of overfitting

The loss functions (5.22) and (5.19) can exhibit pathological overfitting when sampling is sparse, which is common for high-dimensional systems. When no configuration appears more than once in a dataset, highly expressive models and input features (e.g., those from Pengmei et al. [186]) can identify the trajectory and time index of each configuration, ignoring the actual dynamics.

In the worst-case scenario, the dataset consists of a single trajectory $X_{[0,\tau]}$ from each initial structure X_0, with no configuration appearing more than once. With highly expressive models and input features, the loss can be minimized independently for each trajectory. Under these conditions, the VCN loss is minimized at

$$\min_u L_\tau(u) = \frac{1}{2\tau}\,\mathbb{E}[\mathbb{1}_A(X_0)\mathbb{1}_B(X_\tau)+\mathbb{1}_B(X_0)\mathbb{1}_A(X_\tau)], \quad (5.73)$$

with $u(X_0) = u(X_\tau)$ if $X_0 \in D$ and $X_\tau \in D$; otherwise, $u(X_0) = \mathbb{1}_B(X_0) + \mathbb{1}_D(X_0)\mathbb{1}_B(X_\tau)$

and $u(X_\tau) = \mathbb{1}_B(X_\tau) + \mathbb{1}_D(X_\tau)\mathbb{1}_B(X_0)$. The EVCN loss is minimized at

$$\min_u \tilde{L}_\tau(u) = \frac{1}{2\tau}\,\mathbb{E}[N_{AB}(X_{[0,\tau]}) + N_{BA}(X_{[0,\tau]})], \tag{5.74}$$

with $u(X_0) = u(X_\tau)$ if $X_t \in D$ for all $t \in [0, \tau]$; otherwise, $u(X_0) = \mathbb{1}_B(X_{\tau \wedge S_0})$ and $u(X_\tau) = \mathbb{1}_B(X_{0 \vee \bar{S}_\tau})$. For both loss functions, when $u(X_0) = u(X_\tau)$, the output is otherwise unconstrained and can take arbitrary values without affecting the loss.

These issues can arise even with long trajectories if each configuration still appears only once. With a single long trajectory and a single-step lag time, minimizing the loss effectively reduces to solving for the committor along a 1D random walk on the time index. At long lag times, the EVCN loss is minimized with the empirical committor $u(X_t) = (\mathbb{1}_B(X_{S(t)}) + \mathbb{1}_B(X_{\bar{S}(t)}))/2$, which takes values in $\{0, 0.5, 1\}$.

To mitigate these problems, we treated the number of optimization steps as a hyperparameter and selected it based on validation performance. We also restricted the model's flexibility by using a simple architecture and less expressive input features. A more principled solution would be to explicitly regularize the model to prevent it from encoding the trajectory and time index, which we leave for future work.

CHAPTER 6

INEXACT SUBSPACE ITERATION

This chapter is adapted from J. Strahan et al., "Inexact iterative numerical linear algebra for neural network-based spectral estimation and rare-event prediction", J. Chem. Phys. **159**, 014110 (2023), with the permission of AIP Publishing.

Understanding dynamics in complex systems is challenging because there are many degrees of freedom, and those that are most important for describing events of interest are often not obvious. The leading eigenfunctions of the transition operator are useful for visualization, and they can provide an efficient basis for computing statistics such as the likelihood and average time of events (predictions). Here we develop inexact iterative linear algebra methods for computing these eigenfunctions (spectral estimation) and making predictions from a data set of short trajectories sampled at finite intervals. We demonstrate the methods on a low-dimensional model that facilitates visualization and a high-dimensional model of a biomolecular system. Implications for the prediction problem in reinforcement learning are discussed.

6.1 Introduction

Modern observational, experimental, and computational approaches often yield high-dimensional time series data (trajectories) for complex systems. In principle, these trajectories contain rich information about dynamics and, in particular, the infrequent events that are often most consequential. In practice, however, high-dimensional trajectory data are often difficult to parse for useful insight. The need for more efficient statistical analysis tools for trajectory data is critical, especially when the goal is to understand rare-events that may not be well represented in the data.

We consider dynamics that can be treated as Markov processes. A common starting point

for statistical analyses of Markov processes is the transition operator, which describes the evolution of function expectations. The eigenfunctions of the transition operator characterize the most slowly decorrelating features (modes) of the system [1, 9–11, 15]. These can be used for dimensionality reduction to obtain a qualitative understanding of the dynamics [187, 188], or they can be used as the starting point for further computations [5, 12, 14]. Similarly, prediction functions, which provide information about the likelihood and timing of future events as a function of the current state, are defined through linear equations of the transition operator [4, 5].

A straightforward numerical approach to obtaining these functions is to convert the transition operator to a matrix by projecting onto a finite basis for Galerkin approximation [4, 5, 8, 10, 11, 17, 175, 189]. The performance of such a linear approximation depends on the choice of basis [4, 5, 189], and previous work often resorts to a set of indicator functions on a partition of the state space (resulting in a Markov state model or MSM [175]) for lack of a better choice. While Galerkin approximation has yielded many insights [185, 190], the limited expressivity of the basis expansion has stimulated interest in (nonlinear) alternatives.

In particular, artificial neural networks can be harnessed to learn eigenfunctions of the transition operator and prediction functions from data [9, 19, 20, 147, 157, 191–194]. However, existing approaches based on neural networks suffer from various drawbacks. As discussed in Lorpaiboon et al. [9], their performance can often be very sensitive to hyperparameters, requiring extensive tuning and varying with random initialization. Many use loss functions that are estimated against the stationary distribution [157, 167, 169, 170, 172, 173], so that metastable states contribute most heavily, which negatively impacts performance [147, 167]. Assumptions about the dynamics (e.g., microscopic reversibility) limit applicability. In Strahan et al. [147] we introduced an approach that overcomes the issues above, but it uses multiple trajectories from each initial condition; this limits the approach to analysis of simulations and moreover requires specially prepared data sets.

The need to compute prediction functions from observed trajectory data also arises in reinforcement learning. There the goal is to optimize an expected future reward (the prediction function) over a policy (a Markov process). For a fixed Markov process, the prediction problem in reinforcement learning is often solved by temporal difference (TD) methods, which allow the use of arbitrary ensembles of trajectories without knowledge of the details of the underlying dynamics [195]. TD methods have a close relationship with an inexact form of power iteration, which, as we describe, can perform poorly on rare-event related problems.

Motivated by this relationship, as well as by an inexact power iteration scheme previously proposed for approximating the stationary probability distribution of a Markov process using trajectory data [196], we propose a computational framework for spectral estimation and rare-event prediction based on inexact iterative numerical linear algebra. Our framework includes an inexact Richardson iteration for the prediction problem, as well as an extension to inexact subspace iteration for the prediction and spectral estimation problems. The theoretical properties of exact subspace iteration suggest that eigenfunctions outside the span of the approximation will contribute significantly to the error of our inexact iterative schemes [197]. Consistent with this prediction, we demonstrate that learning additional eigenvalues and eigenfunctions simultaneously through inexact subspace iteration accelerates convergence dramatically relative to inexact Richardson and power iteration in the context of rare events. While we assume the dynamics can be modeled by a Markov process, we do not require knowledge of their form or a specific underlying model. The method shares a number of further advantages with the approach discussed in Strahan et al. [147] without the need for multiple trajectories from each initial condition in the data set. This opens the door to treating a wide range of observational, experimental, and computational data sets.

The remainder of the chapter is organized as follows. In Section 6.2, we describe the quantities that we seek to compute in terms of linear operators. In Sections 6.3 and 6.4, we

introduce an inexact subspace iteration algorithm that we use to solve for these quantities. Section 6.5 illustrates how the loss function can be tailored to the known properties of the desired quantity. Section 6.6 summarizes the two test systems that we use to illustrate our methods: a two-dimensional potential, for which we can compute accurate reference solutions, and a molecular example that is high-dimensional but still sufficiently tractable that statistics for comparison can be computed from long trajectories. In Section 6.7, we explain the details of the invariant subspace iteration and then demonstrate its application to our two examples. Lastly, Section 6.8 details how the subspace iteration can be modified to compute prediction functions and compares the effect of different loss functions, as well as the convergence properties of power iteration and subspace iteration. We conclude with implications for reinforcement learning.

6.2 Spectral estimation and the prediction problem

We have two primary applications in mind in this article. First, we would like to estimate the *dominant eigenfunctions and eigenvalues* of the transition operator $\mathcal{T}^t$ for a Markov process $X^t \in \mathbb{R}^d$, defined as

$$\mathcal{T}^t f(x) = \mathbb{E}_x \left[f(X^t) \right], \tag{6.1}$$

where f is an arbitrary real-valued function and the subscript x indicates the initial condition $X^0 = x$. The transition operator encodes how expectations of functions evolve in time. The right eigenfunctions of $\mathcal{T}^t$ with the largest eigenvalues characterize the most slowly decorrelating features (modes) of the Markov process [1, 9–11].

Our second application is to compute *prediction functions* of the general form

$$u(x) = \mathbb{E}_x \left[\Psi(X^T) + \sum_{t=0}^{T-1} \Gamma(X^t) \right], \tag{6.2}$$

where T is the first time $X^t \notin D$ from some domain D, and Ψ and Γ are functions associated with the escape event (rewards in reinforcement learning). Prototypical examples of prediction functions that appear in our numerical results are the mean first passage time (MFPT) from each x:

$$m(x) = \mathbb{E}_x\left[T\right] \tag{6.3}$$

and the committor:

$$q(x) = \mathbb{E}_x\left[\mathbb{1}_B(X^T)\right] = \mathbb{P}_x\left[X^T \in B\right], \tag{6.4}$$

where A and B are disjoint sets ("reactant" and "product" states) and $D = (A \cup B)^{\mathsf{c}}$. The MFPT is important for estimating rates of transitions between regions of state space, while the committor can serve as a reaction coordinate [108, 148, 173, 198] and as a key ingredient in transition path theory statistics [3, 147, 151]. For any $\tau > 0$, the prediction function $u(x)$ satisfies the linear equation

$$\left(\mathcal{I} - \mathcal{S}^\tau\right) u(x) = \mathbb{E}_x\left[\sum_{t=0}^{(\tau\wedge T)-1} \Gamma(X^t)\right] \tag{6.5}$$

for $x \in D$, with boundary condition

$$u(x) = \Psi(x) \tag{6.6}$$

for $x \notin D$. In (6.5), $\mathcal{I}$ is the identity operator and

$$\mathcal{S}^\tau f(x) = \mathbb{E}_x\left[f(X^{\tau\wedge T})\right] \tag{6.7}$$

is the "stopped" transition operator [5]. We use the notation $\tau \wedge T = \min\{\tau, T\}$. The parameter τ is known as the lag time. While it is an integer corresponding to a number of integration steps, in our numerical examples, we often express it in terms of equivalent time

units.

Our specific goal is to solve both the eigenproblem and the prediction problem for X^t in high dimensions and without direct access to a model governing its evolution. Instead, we have access to trajectories of X^t of a fixed, finite duration τ. A natural and generally applicable approach to finding an approximate solution to the prediction problem is to attempt to minimize the residual of (6.5) over parameters θ of a neural network $u_\theta(x)$ representing $u(x)$. For example, we recently suggested an algorithm that minimizes the residual norm

$$\frac{1}{2}\left\|\left(\mathcal{I}-\mathcal{S}^\tau\right)u_\theta - r\right\|_\mu^2, \tag{6.8}$$

where $r(x)$ is the right-hand side of (6.5) and μ is an arbitrary distribution of initial conditions X^0 (boundary conditions were enforced by an additional term) [147]. The norm $\|\cdot\|_\mu$ is defined by $\|f\|_\mu^2 = \langle f, f\rangle_\mu$, where $\langle f, g\rangle_\mu = \int f(x)g(x)\mu(dx)$ for arbitrary functions f and g. The gradient of the residual norm in (6.8) with respect to neural-network parameters θ can be written

$$\left\langle\left(\mathcal{I}-\mathcal{S}^\tau\right)u_\theta - r, \nabla_\theta u_\theta\right\rangle_\mu - \left\langle\left(\mathcal{I}-\mathcal{S}^\tau\right)u_\theta - r, \mathcal{S}^\tau\nabla_\theta u_\theta\right\rangle_\mu \tag{6.9}$$

Given a data set of initial conditions $\{X_j^0\}_{j=1}^n$ drawn from μ and a single sample trajectory $\{X_j^t\}_{t=0}^\tau$ of X^t for each X_j^0, the first term in the gradient (6.9) can be approximated without bias as

$$\left\langle\left(\mathcal{I}-\mathcal{S}^\tau\right)u_\theta - r, \nabla_\theta u_\theta\right\rangle_\mu \approx \frac{1}{n}\sum_{j=1}^n \left(u_\theta(X_j^0) - u_\theta(X_j^{\tau\wedge T_j}) - \sum_{t=0}^{(\tau\wedge T_j)-1}\Gamma(X_j^t)\right)\nabla_\theta u_\theta(X_j^0) \tag{6.10}$$

where T_j is the first time $X_j^t \notin D$.

In contrast, unbiased estimation of the second term in (6.9) requires access to two independent trajectories of X^t for each sample initial condition since it is quadratic in $\mathcal{S}^\tau$ [147, 195]. In the reinforcement learning community, TD methods were developed for the

purpose of minimizing a loss of a very similar form to (6.8), and they perform a "semigradient" descent by following only the first term in (6.9) [195]. As in the semigradient approximation, the algorithms proposed in this chapter only require access to inner products of the form $\langle f, \mathcal{A}g\rangle_\mu$ for an operator $\mathcal{A}$ related to $\mathcal{T}^\tau$ or $\mathcal{S}^\tau$, and they avoid terms that are non-linear in $\mathcal{A}$. As we explain, such inner products can be estimated using trajectory data. However, rather than attempting to minimize the residual directly by an approximate gradient descent, we view the eigenproblem and prediction problems through the lens of iterative numerical linear algebra schemes.

6.3 An inexact power iteration

To motivate the iterative numerical linear algebra point of view, observe that the first term in (6.9) is the exact gradient with respect to θ' of the loss

$$\frac{1}{2}\left\|u_{\theta'} - \mathcal{S}^\tau u_\theta - r\right\|_\mu^2, \tag{6.11}$$

evaluated at $\theta' = \theta$. The result of many steps of gradient descent (later, "inner iterations") on this loss with θ held fixed can then be viewed as an inexact Richardson iteration for (6.5), resulting in a sequence

$$u_{\theta^{s+1}} \approx \mathcal{S}^\tau u_{\theta^s} + r, \tag{6.12}$$

where, for each s, u_{θ^s} is a parametrized neural-network approximation of the solution to (6.5). To unify our discussion of the prediction and spectral estimation problems, it is helpful to observe that (6.12) can be recast as an equivalent inexact power iteration:

$$\bar{u}_{\theta^{s+1}} \approx \mathcal{A}\bar{u}_{\theta^s} \tag{6.13}$$

where

$$\bar{u}_\theta = \begin{pmatrix} 1 \\ u_\theta \end{pmatrix} \quad \text{and} \quad \mathcal{A} = \begin{bmatrix} 1 & 0 \\ r & \mathcal{S}^\tau \end{bmatrix}. \tag{6.14}$$

Note that $(1, u)^\top$ is the dominant eigenfunction of $\mathcal{A}$ and has eigenvalue equal to 1.

Wen et al. [196] introduced an inexact power iteration to compute the stationary probability measure of $\mathcal{T}^\tau$, i.e., its dominant left eigenfunction. As those authors note, an inexact power update such as (6.13) can be enforced using a variety of loss functions. In our setting, the L^2_μ norm in (6.11) can be replaced by any other measure of the difference between $u_{\theta'}$ and $\mathcal{S}^\tau u_\theta + r$, as long as an unbiased estimator of its gradient with respect to θ' is available. This flexibility is discussed in more detail in Section 6.5, and we exploit it in applications presented later in this article. For now, we focus instead on another important implication of this viewpoint: the flexibility in the form of the iteration itself.

We will see that when the spectral gap of $\mathcal{A}$, the difference between its largest and second largest eigenvalues, is small, the inexact power iteration (or the equivalent Richardson iteration) described above fails to reach an accurate solution. The largest eigenvalue of $\mathcal{A}$ in (6.14) is 1 and its next largest eigenvalue is the dominant eigenvalue of $\mathcal{S}^\tau$. For rare-event problems the difference between these two eigenvalues is expected to be very small. Indeed, when X^0 is drawn from the quasi-stationary distribution of X^t in D, the logarithm of the largest eigenvalue of $\mathcal{S}^\tau$ is exactly $-\tau/\,\mathbb{E}[T]$ [199]. Thus, when the mean exit time from D is large, we can expect the spectral gap of $\mathcal{A}$ in (6.14) to be very small. In this case, classical convergence results tell us that exact power iteration converges slowly, with the largest contributions to the error coming from the eigenfunctions of $\mathcal{S}^\tau$ with largest magnitude eigenvalues [197]. Iterative schemes that approximate multiple dominant eigenfunctions simultaneously, such as subspace iteration and Krylov methods, can converge much more rapidly [197]. In the next section, we introduce an inexact form of subspace iteration to address this shortcoming. Beyond dramatically improving performance on the prediction

problem for rare-events when applied with $\mathcal{A}$ in (6.14), it can also be applied with $\mathcal{A} = \mathcal{T}^\tau$ to compute multiple dominant eigenfunctions and values of $\mathcal{T}^\tau$ itself.

6.4 An inexact subspace iteration

Our inexact subspace iteration for the k dominant eigenfunctions/values of $\mathcal{A}$ proceeds as follows. Let $\{\varphi^a_{\theta^s}\}_{a=1}^k$ be a sequence of k basis functions parametrized by θ^s (these can be scalar or vector valued functions depending on the form of $\mathcal{A}$). We can represent this basis as the vector valued function

$$U_{\theta^s} = \left(\varphi^1_{\theta^s}, \varphi^2_{\theta^s}, \ldots, \varphi^k_{\theta^s}\right). \tag{6.15}$$

Then, we can obtain a new set of k basis functions by approximately applying $\mathcal{A}$ to each of the components of U_{θ^s}:

$$U_{\theta^{s+1}} K^{s+1} \approx \mathcal{A}\, U_{\theta^s} \tag{6.16}$$

where K^{s+1} is an invertible, upper-triangular $k \times k$ matrix that does not change the span of the components of $U_{\theta^{s+1}}$ but is included to facilitate training. One way the approximate application of $\mathcal{A}$ can be accomplished is by minimizing

$$\frac{1}{2}\sum_{a=1}^{k}\left\|\sum_{b=1}^{k}\varphi^b_\theta K_{ba} - \mathcal{A}\,\varphi^a_{\theta^s}\right\|^2_\mu \tag{6.17}$$

over θ and K with θ^s held fixed.

The eigenvalues and eigenfunctions of $\mathcal{A}$ are then approximated by solving the finite-dimensional generalized eigenproblem

$$C^\tau W = C^0 W \Lambda \tag{6.18}$$

where

$$C^{\tau}_{ab} = \langle \varphi^a_{\theta^s}, \mathcal{A}\varphi^b_{\theta^s} \rangle_\mu \tag{6.19}$$

$$C^{0}_{ab} = \langle \varphi^a_{\theta^s}, \varphi^b_{\theta^s} \rangle_\mu, \tag{6.20}$$

each inner product is estimated using trajectory data, and W and Λ are $k \times k$ matrices. The matrix Λ is diagonal, and the a-th eigenvalue λ_a of $\mathcal{A}$ is approximated by Λ_{aa}; the corresponding eigenfunction v_a is approximated by $\sum_{b=1}^{k} W_{ab}\, \varphi^b_{\theta^s}$.

Even when sampling is not required to estimate the matrices in (6.19) and (6.20), the numerical rank of C^{τ} becomes very small as the eigenfunctions become increasingly aligned with the single dominant eigenfunction. To overcome this issue, we apply an orthogonalization step between iterations (or every few iterations). Just as the matrices C^0 and C^{τ} can be estimated using trajectory data, the orthogonalization procedure can also be implemented approximately using data.

Finally, in our experiments we find it advantageous to damp large parameter fluctuations during training by mixing the operator $\mathcal{A}$ with a multiple of the identity, i.e., we perform our inexact subspace iteration on the operator $(1-\alpha_s)\mathcal{I} + \alpha_s \mathcal{A}$ in place of $\mathcal{A}$ itself. This new operator has the same eigenfunctions as $\mathcal{A}$. In our experiments, decreasing the parameter α_s as the number of iterations increases results in better generalization properties of the final solution and helps ensure convergence of the iteration. For our numerical experiments we use

$$\alpha_s = \begin{cases} 1 & s < \sigma \\ 1/\sqrt{s+1-\sigma} & s \geq \sigma \end{cases} \tag{6.21}$$

where σ is a user chosen hyperparameter that sets the number of iterations performed before damping begins.

The details, including estimators for all required inner products, in the case of the

eigenproblem ($\mathcal{A} = \mathcal{T}^\tau$) are given in Section 6.7 and Algorithm 4. For the prediction problem with $\mathcal{A}$ as in (6.14), they are given in Section 6.8 and Algorithm 5.

6.5 Alternative loss functions

As mentioned above, the inexact application of the operator $\mathcal{A}$ can be accomplished by minimizing loss functions other than (6.17). The key requirement for a loss function in the present study is that $\mathcal{A}$ appears in its gradient only through terms of the form $\langle f, \mathcal{A}g\rangle_\mu$ for some functions f and g, so that the gradient can be estimated using trajectory data. As a result, we have flexibility in the choice of loss and, in turn, the representation of u. In particular, we consider the representation $u_\theta = \zeta(z_\theta)$, where ζ is an increasing function, and z_θ is a function parameterized by a neural network. An advantage of doing so is that the function ζ can restrict the output values of u_θ to some range. For example, when computing a probability such as the committor, a natural choice is the sigmoid function $\zeta(x) = (1+e^{-x})^{-1}$.

Our goal is to train a sequence of parameter values so that u_{θ^s} approximately follows a subspace iteration, i.e., so that $\zeta(z_{\theta^{s+1}}) \approx \mathcal{A}u_{\theta^s}$. To this end, we minimize with respect to θ the loss function

$$\mathbb{E}_{X^0\sim\mu}\left[V(z_\theta) - z_\theta \mathcal{A}u_{\theta^s}\right], \tag{6.22}$$

where V is an antiderivative of ζ, and θ^s is fixed. The subscript $X^0 \sim \mu$ in this expression indicates that X^0 is drawn from μ. Note that, as desired, $\mathcal{A}$ appears in the gradient of (6.22) only in an inner product of the required form, and an approximate minimizer, θ^{s+1}, of this loss satisfies $\zeta(z_{\theta^{s+1}}) \approx \mathcal{A}u_{\theta^s}$. This general form of loss function is adapted from variational expressions for the divergence of two probability measures [145, 196].

The L^2_μ loss in (6.17), which we use in several of our numerical experiments, corresponds to the choice $\zeta(x) = x$ and $V(x) = x^2/2$. The choice of $\zeta(x) = (1 + e^{-x})^{-1}$ mentioned above corresponds to $V(x) = \log(1 + e^x)$; we refer to the loss in (6.22) with this choice of V as the

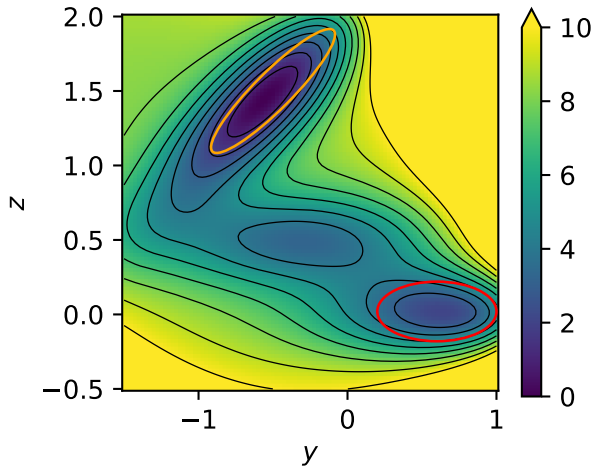

Figure 6.1: Müller-Brown potential energy surface. The orange and red ovals indicate the locations of states A and B respectively when computing predictions. Contour lines are drawn every 1 β^{-1}.

"softplus" loss. We note that in the context of committor function estimation, in the limit of infinite lag time the "softplus" loss corresponds directly to the maximum log-likelihood loss (for independent Bernoulli random variables) that defines the logistic regression estimate of the probability of reaching B before A. Logistic regression has previously been used in conjunction with data sets of trajectories integrated all the way until reaching A or B to estimate the committor function [154–156, 200–202].

6.6 Test problems

We illustrate our methods with two well-characterized systems that exemplify features of molecular transitions. In this section, we provide key details of these systems.

6.6.1 *Müller-Brown potential*

The first system is defined by the Müller-Brown potential [79] (Figure 6.1):

$$V_{\mathrm{MB}}(y,z) = \frac{1}{20}\sum_{i=1}^{4} C_i \exp[a_i(y-y_i)^2 + b_i(y-y_i)(z-z_i) + c_i(z-z_i)^2]. \tag{6.23}$$

The two-dimensional nature of this model facilitates visualization. The presence of multiple minima and saddlepoints that are connected by a path that does not align with the coordinate axes makes the system challenging for both sampling and analysis methods. In Sections 6.7.1 and 6.8.1, we use $C_i = \{-200, -100, -170, 15\}$, $a_i = \{-1, -1, -6.5, 0.7\}$, $b_i = \{0, 0, 11, 0.6\}$, $c_i = \{-10, -10, -6.5, 0.7\}$, $y_i = \{1, -0.27, -0.5, -1\}$, $z_i = \{0, 0.5, 1.5, 1\}$. In Section 6.8.2, we tune the parameters to make transitions between minima rarer; there, the parameters are $C_i = \{-250, -150, -170, 15\}$, $a_i = \{-1, -3, -6.5, 0.7\}$, $b_i = \{0, 0, 11, 0.6\}$, $c_i = \{-10, -30, -6.5, 0.7\}$, $y_i = \{1, -0.29, -0.5, -1\}$, $z_i = \{0, 0.5, 1.5, 1\}$. For prediction, we analyze transitions between the upper left minimum (A) and the lower right minimum (B) in Figure 6.1; these states are defined as

$$\begin{aligned} A &= \{y, z : 6.5(y+0.5)^2 - 11(y+0.5)(z-1.5) + 6.5(z-1.5)^2 < 0.3\} \\ B &= \{y, z : (y-0.6)^2 + 0.5(z-0.02)^2 < 0.2\}. \end{aligned} \tag{6.24}$$

To generate a data set, we randomly draw 50,000 initial conditions X_j^0 uniformly from the region

$$\Omega = \{y, z : -1.5 < y < 1.0,\ -0.5 < z < 2.5,\ V_{\mathrm{MB}}(y,z) < 12\} \tag{6.25}$$

and, from each of these initial conditions, generate one trajectory according to the discretized overdamped Langevin dynamics

$$X_j^t = X_j^{t-1} - \delta_t \nabla V_{\mathrm{MB}}(X_j^{t-1}) + \sqrt{\delta_t\, 2\beta^{-1}}\, \xi_j^t \tag{6.26}$$

where $1 \leq t \leq \tau$, the ξ_j^t are independent standard Gaussian random variables, and the timestep is $\delta_t = 0.001$. We use an inverse temperature of $\beta = 2$, and we vary τ as indicated below (note that τ is an integer number of steps, but we present our results for the Müller-Brown model in terms of the dimensionless scale used for δ_t immediately above). For each test, we use independent random numbers and redraw the initial conditions because it is faster to generate the trajectories than to read them from disk in this case. All error bars are computed from the empirical standard deviation over 10 replicate data sets.

To validate our results, we compare the neural-network results against grid-based references, computed as described in Section S4 of the Supplementary Material of Thiede et al. [4] and the Appendix of Lorpaiboon et al. [203] (our notation here follows the former more closely). The essential idea is that the terms in the infinitesimal generator of the transition operator can be approximated on a grid to second order in the spacing by expanding both the probability for transitions between sites and a test function. To this end, we define the transition matrix for neighboring sites $x = (y, z)$ and $x' = (y \pm \delta_x, z)$ or $(y, z \pm \delta_x)$ on a grid with spacings δ_x:

$$P(x'|x) = \frac{1}{1 + e^{\beta[V(x') - V(x)]}} \tag{6.27}$$

(this definition differs from that in Thiede et al. [4] by a factor of $1/4$, and we scale $P - I$, where I is the identity matrix, accordingly to set the time units below). The diagonal entry $P(x|x)$ is fixed to make the transition matrix row-stochastic. We use $\delta_x = 0.0125$; the grid is a rectangle with $y \in [-1.5, 1.0]$, and $z \in [-0.5, 2.0]$. The reference invariant subspaces are computed by diagonalizing the matrix $2(P - I)/\beta\delta_x^2$ with a sparse eigensolver; we use scipy.sparse.linalg. We obtain the reference committor $\hat{q}_+$ for grid sites in $(A \cup B)^\mathsf{c}$ by solving

$$\mathrm{diag}(\hat{\mathbb{1}}_{(A\cup B)^\mathsf{c}})(P - I)\hat{q}_+ = -\mathrm{diag}(\hat{\mathbb{1}}_{(A\cup B)^\mathsf{c}})(P - I)\hat{\mathbb{1}}_B \tag{6.28}$$

and for grid sites in $A \cup B$ by setting $\hat{q}_+ = \hat{\mathbb{1}}_B$. Here, $\hat{q}_+$ and $\hat{\mathbb{1}}_B$ are vectors corresponding

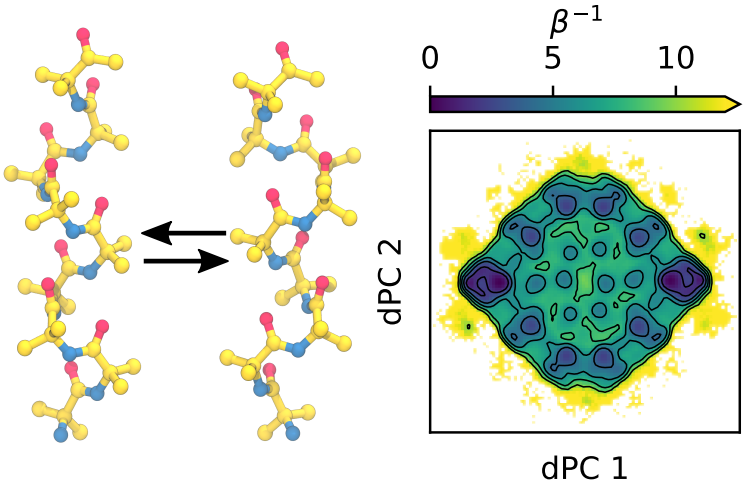

Figure 6.2: Helix-to-helix transition of AIB_9. (left) Left- and right-handed helices, which we use as states A and B, respectively, when computing predictions. Carbon, nitrogen, and oxygen atoms are shown in yellow, blue, and red, respectively; hydrogen atoms are omitted. (right) Potential of mean force constructed from the histogram of value pairs of the first two dihedral angle principal components; data are from the 20 trajectories of 5 μs that we use to construct reference statistics (see text). The left-handed helix corresponds to the left-most basin, and the right-handed helix corresponds to the right-most basin. Contour lines are drawn every 2 β^{-1} corresponding to a temperature of 300 K.

to the functions evaluated at the grid sites.

6.6.2 AIB_9 helix-to-helix transition

The second system is a peptide of nine α-aminoisobutyric acids (AIB_9; Figure 6.2). Because AIB is achiral around its α-carbon atom, AIB_9 can form left- and right-handed helices with equal probabilities, and we study the transitions between these two states. This transition was previously studied using MSMs and long unbiased molecular dynamics simulations [140, 141]. AIB_9 poses a stringent test due to the presence of many metastable intermediate states.

The states are defined in terms of the internal ϕ and ψ dihedral angles. We classify an amino acid as being in the "left" state if its dihedral angle values are within a circle of radius 25° centered at $(41°, 43°)$, that is

$$(\phi - 41°)^2 + (\psi - 43°)^2 \leq (25°)^2.$$

Amino acids are classified as being in the "right" state using the same radius, but centered instead at $(-41°, -43°)$. States A and B are defined by the amino acids at sequence positions 3–7 being all left or all right, respectively. We do not use the two residues on each end of AIB_9 in defining the states as these are typically more flexible [141]. The states can be resolved by projecting onto dihedral angle principal components (dPCs; Figure 6.2, right) as described previously [146].

Following a procedure similar to that described in Perez et al. [141], we generate a data set of short trajectories. From each of the 691 starting configurations in Perez et al. [141], we simulate 10 trajectories of duration 20 ns with initial velocities drawn randomly from a Maxwell-Boltzmann distribution at a temperature of 300 K. The short trajectory data set thus contains 6,910 trajectories, corresponding to a total sampling time of 138.2 μs. We use a timestep of 4 fs together with a hydrogen mass repartitioning scheme [144], and configurations are saved every 40 ps. We employ the AIB parameters from Forcefield_NCAA [143] and the GBNeck2 implicit-solvent model [145]. Simulations are performed with the Langevin integrator in OpenMM 7.7.0 [142] using a friction coefficient of $1\,\mathrm{ps}^{-1}$. To generate a reference for comparison to our results, we randomly select 20 configurations from the data set above and, from each, run a simulation of 5 μs with the same simulation parameters. For all following tests on AIB_9, batches consist of pairs of frames separated by τ drawn randomly with replacement from the short trajectories (i.e., from all possible such pairs in the data set). From each frame, we use only the atom positions because the momenta decorrelate within a few picoseconds, which is much shorter than the lag times that we consider. However, in principle, the momenta could impact the prediction functions [138] and be used as neural-network inputs as well.

6.7 Spectral estimation

In this section, we provide some further numerical details for the application of our method to spectral estimation and demonstrate the method on the test problems. For our subspace iteration with $\mathcal{A} = \mathcal{T}^\tau$, we require estimators for inner products of the form $\langle f, \mathcal{T}^\tau g\rangle_\mu$. For example, the gradient of the loss function (6.17) involves inner products of the form

$$\left\langle \nabla_\theta \varphi_\theta^a, \mathcal{T}^\tau \varphi_\theta^b \right\rangle_\mu . \tag{6.29}$$

For these, we use the unbiased data-driven estimator

$$\langle f, \mathcal{T}^\tau g\rangle_\mu \approx \frac{1}{n} \sum_{j=1}^{n} f(X_j^0) g(X_j^\tau). \tag{6.30}$$

As discussed in Section 6.4, applying the operator $\mathcal{T}^\tau$ repeatedly causes each basis function to converge to the dominant eigenfunction and leads to numerical instabilities. To avoid this, we orthogonalize the outputs of the networks with a QR decomposition at the end of each subspace iteration by constructing the matrix $\Phi_{ia} = \varphi_\theta^a(X_i^0)$ and then computing the factorization $\Phi = QR$ where Q is an $n \times k$ matrix with orthogonal columns and R is an upper triangular $k \times k$ matrix. Finally, we set $\tilde{\varphi}_s^a = \sum_{b=1}^k \varphi_\theta^b \, (R^{-1}N)_{ba}$, where N is a diagonal matrix with entries equal to the norms of the columns of Φ (before orthogonalization). To ensure that the networks remain well-separated (i.e., the eigenvalues of C^0 remain away from zero), we penalize large off-diagonal entries of K by adding to the loss

$$\gamma_1 \|K - \mathrm{diag}(K)\|_{\mathrm{F}}^2, \tag{6.31}$$

where γ_1 allows us to tune the strength of this term relative to others, and $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. We control the scale of each network output using the strategy from Wen

et al. [196]; that is, we add to the loss a term of the form

$$\gamma_2 \sum_{a=1}^{k} \left[2\nu_a \left(\frac{1}{n} \sum_{j=1}^{n} \varphi_\theta^a(X_j^0)^2 - 1 \right) - \nu_a^2 \right], \tag{6.32}$$

where we have introduced the conjugate variables ν_a which we maximize with gradient ascent (or similar optimization). In general, our numerical experiments suggest that it is best to keep γ_1 and γ_2 relatively small. We find that stability of the algorithm over many subspace iterations is improved if the matrix K is set at its optimal value before each inner loop. To do this, we set

$$K_{1:i,i} = \arg\min_c \frac{1}{n} \sum_{j=1}^{n} \left(\sum_{a=1}^{i} \varphi_\theta^a(X_j^0) c_a - \tilde{\varphi}_s^a(X_j^\tau) \right)^2 + \gamma_2 \sum_{a=1}^{i-1} c_a^2. \tag{6.33}$$

The above minimization can be solved with linear least squares. Finally, we note that in practice any optimizer can be used for the inner iteration steps, though the algorithm below implements stochastic gradient descent. In this work, we use Adam [204] for all numerical tests. We summarize our procedure for spectral estimation in Algorithm 4.

6.7.1 Müller-Brown model

Table 6.1: Parameter choices used in this work

	Spectral Estimation		**Committor**			**MFPT**
Hyperparameter	Müller-Brown	AIB_9	Müller-Brown	Modified Müller-Brown	AIB_9	AIB_9
subspace dimension k	3	5	1	2, 1[1]	1	5
input dimensionality	2	174	2	2	174	174
hidden layers	6	6	6	6	6	6
hidden layer width	64	128	64	64	150	150
hidden layer nonlinearity	CeLU	CeLU	ReLU	ReLU	ReLU	ReLU
output layer nonlinearity	none	tanh	sigmoid/none	none	none	none
outer iterations S	10	100	100	$4 + 10^1$	100	300
inner iterations M	5000	2000	200	5000	2000	1000
σ	2	50	50	0	50	0
batch size B	2000	1024	5000	2000	1024	2000
learning rate η	0.001	0.0001	0.001	0.001	0.001	0.001
γ_1	0.15	0.001	-	-	-	0.1
γ_2	0.01	0.01	-	-	-	10
loss for φ_θ^1	L_μ^2	L_μ^2	L_μ^2/softplus	softplus	softplus	L_μ^2
loss for φ_θ^a for $a > 1$	L_μ^2	L_μ^2	-	L_μ^2	-	L_μ^2

Algorithm 4 Inexact subspace iteration (with L^2_μ loss) for spectral estimation

Require: Subspace dimension k, transition data $\{X_j^0, X_j^\tau\}_{j=1}^n$, batch size B, learning rate η, number of subspace iterations S, number of inner iterations M, regularization parameters γ_1 and γ_2

Initialize $\{\varphi_\theta^a\}_{a=1}^k$ and $\{\tilde{\varphi}_1^a\}_{a=1}^k$
for $s = 1 \dots S$ **do**
  for $m = 1 \dots M$ **do**
    Sample a batch of data $\{X_j^0, X_j^\tau\}_{j=1}^B$
    $\hat{\mathcal{L}}_1 \leftarrow \frac{1}{B}\sum_{j=1}^B \sum_{a=1}^k \big[\frac{1}{2}(\sum_{b=1}^k \varphi_\theta^b(X_j^0)K_{ba})^2 - \sum_{b=1}^k \varphi_\theta^b(X_j^0)K_{ba}\big\{\alpha_s\tilde{\varphi}_s^a(X_j^\tau) + (1-\alpha_s)\tilde{\varphi}_s^a(X_j^0)\big\}\big]$
    $\hat{\mathcal{L}}_K \leftarrow \gamma_1 \|K - \mathrm{diag}(K)\|_{\mathrm{F}}^2$
    $\hat{\mathcal{L}}_{\mathrm{norm}} \leftarrow \gamma_2 \sum_{a=1}^k (2\nu_a(\frac{1}{B}\sum_{j=1}^B(\varphi_\theta^a(X_j^0)^2) - 1) - \nu_a^2)$
    $\hat{\mathcal{L}} \leftarrow \hat{\mathcal{L}}_1 + \hat{\mathcal{L}}_K + \hat{\mathcal{L}}_{\mathrm{norm}}$
    $\theta \leftarrow \theta - \eta\nabla_\theta\hat{\mathcal{L}}$
    $K \leftarrow K - \eta\,\mathrm{triu}(\nabla_K\hat{\mathcal{L}})$
    $\nu_a \leftarrow \nu_a + \eta\nabla_{\nu_a}\hat{\mathcal{L}}$
  end for
  Compute the matrix $\Phi_{ia} = \varphi_\theta^a(X_i^0)$ ▷ $\Phi \in \mathbb{R}^{n\times k}$
  Compute diagonal matrix $N_{aa}^2 = \sum_i \varphi_\theta^a(X_i^0)^2$
  Compute QR-decomposition $\Phi = QR$ ▷ $Q \in \mathbb{R}^{n\times k}$ and $R \in \mathbb{R}^{k\times k}$
  $\tilde{\varphi}_s^a \leftarrow \sum_{b=1}^k \varphi_\theta^b\,(R^{-1}N)_{ba}$
  $K_{1:i,i} \leftarrow \arg\min_c \frac{1}{n}\sum_{j=1}^n \Big(\sum_{a=1}^i \varphi_\theta^a(X_j^0)c_a - \tilde{\varphi}_s^a(X_j^\tau)\Big)^2 + \gamma_2 \sum_{a=1}^{i-1} c_a^2$
end for
Compute the matrices $C_{ab}^t = \frac{1}{n}\sum_{j=1}^n \tilde{\varphi}_s^a(X_j^0)\tilde{\varphi}_s^b(X_j^t)$ for $t = 0, \tau$ ▷ $C^t \in \mathbb{R}^{k\times k}$
Solve the generalized eigenproblem $C^\tau W = C^0 W\Lambda$ for W and Λ
return eigenvalues Λ, eigenfunctions $v_a = \sum_{b=1}^k W_{ab}\tilde{\varphi}_s^b$

As a first test of our method, we compute the $k = 3$ dominant eigenpairs for the Müller-Brown model. Since we know that the dominant eigenfunction of the transition operator is the constant function $v_1(y, z) = 1$ with eigenvalue $\lambda_1 = 1$, we directly include this function in the basis as a non-trainable function, i.e. $\varphi_\theta^1(y, z) = 1$. To initialize $\tilde{\varphi}_1^a$ for each $a > 1$, we choose a standard Gaussian vector $(Y^a, Z^a) \in \mathbb{R}^2$, and set $\tilde{\varphi}_1^a(y, z) = y\, Y^a + z\, Z^a$. This ensures that the initial basis vectors are well-separated and the first QR step is numerically stable. Here and in all subsequent Müller-Brown tests, batches of trajectories are drawn from the entire data set with replacement. Other hyperparameters are listed in Table 6.1.

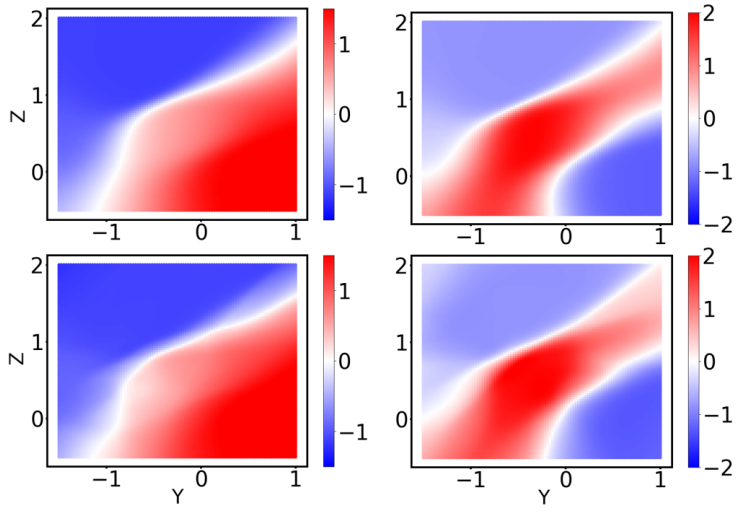

Figure 6.3: First two non-trivial eigenfunctions of the Müller-Brown model. (top) Grid-based reference. (bottom) Neural network subspace after ten subspace iteration steps, computed with $\tau = 300$ steps (i.e., $0.3\,\delta_t^{-1}$).

Figure 6.3 shows that we obtain good agreement between the estimate produced by the inexact subspace iteration in Algorithm 4 and reference eigenfunctions. Figure 6.4 (upper panels) shows how the corresponding eigenvalues vary with lag time; again there is good agreement with the reference. Furthermore, there is a significant gap between λ_3 and λ_4, indicating that a three-dimensional subspace captures the dynamics of interest for this system.

We compare the subspace that we obtain from our method with that from an MSM constructed from the same amount of data by using k-means to cluster the configurations into

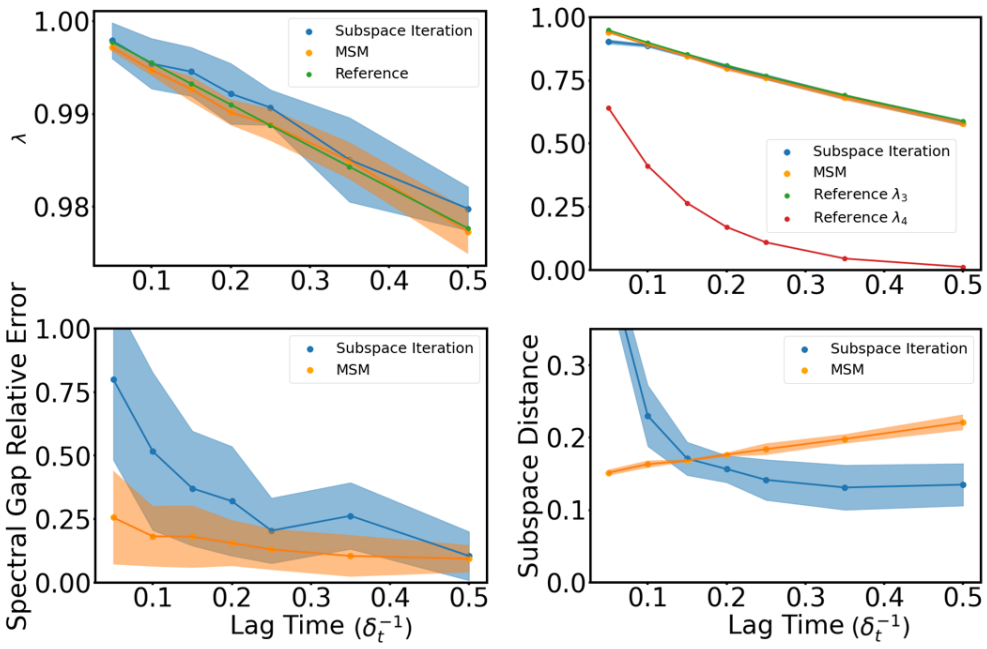

Figure 6.4: Spectral estimation as a function of lag time (in units of δ_t^{-1}) for the Müller-Brown model. (top left) Second eigenvalue. (top right) Third and fourth eigenvalues; only the reference fourth eigenvalue is shown to illustrate the spectral gap. (bottom left) Relative error in the first spectral gap (i.e., $1 - \lambda_2$). (bottom right) Subspace distance between estimated and reference three-dimensional invariant subspaces.

400 states and counting the transitions between clusters. This is a very fine discretization for this system, and the MSM is sufficiently expressive to yield eigenfunctions in good agreement with the reference. The relative error of $1 - \lambda_2$ is comparable for the two methods (Figure 6.4, lower left). To compare two finite dimensional subspaces, $\mathcal{U}$ and $\mathcal{V}$, we define the subspace distance as [1]

$$d(\mathcal{U}, \mathcal{V}) = \left\| (I - P_{\mathcal{U}}) P_{\mathcal{V}} \right\|_{\mathrm{F}}, \tag{6.34}$$

where $P_{\mathcal{U}}$ and $P_{\mathcal{V}}$ denote the orthogonal projections onto $\mathcal{U}$ and $\mathcal{V}$, respectively, and $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. Figure 6.4 (lower right) shows the subspace distances from the reference as functions of lag time. We see that the inexact subspace iteration better approximates the three-dimensional dominant eigenspace for moderate to long lag times, even though the eigenvalues are comparable.

6.7.2 AIB_9

For the molecular test system, we compute the dominant five-dimensional subspace. We compare the inexact subspace iteration in Algorithm 4 with MSMs constructed on dihedral angles ("dihedral MSM") and on Cartesian coordinates ("Cartesian MSM"). We expect the dihedral MSM to be accurate given that the dynamics of AIB_9 are well-described by the backbone dihedral angles [140, 141], and we thus use it as a reference. It is constructed by clustering the sine and cosine of each of the backbone dihedral angles (ϕ and ψ) for the nine residues (for a total of $2 \times 2 \times 9 = 36$ input features) into 1000 clusters using k-means and counting transitions between clusters. The Cartesian MSM is constructed by similarly counting transitions between 1000 clusters from the k-means algorithm, but the clusters are based on the Cartesian coordinates of all non-hydrogen atoms after aligning the backbone atoms of the trajectories, for a total of 174 input features. Because of the difficulty of clustering high-dimensional data, we expect the Cartesian MSM basis to give poor results. The neural network for the inexact subspace iteration receives the same 174 Cartesian coordinates as input features. We choose to use Cartesian coordinates rather than dihedral angles as inputs because it requires the network to identify nontrivial representations for describing the dynamics.

As in the Müller-Brown example, we use $\varphi_\theta^1 = 1$ and a random linear combination of coordinate functions to initialize $\tilde{\varphi}_1^a$ for $a > 1$. Other hyperparameters are listed in Table 6.1. With these choices, the neural-network training typically requires about 20 minutes on a single NVIDIA A40 GPU; this is much longer than the time required for diagonalization of the 1000×1000 MSM transition matrix, which is nearly instantaneous. However, the time for neural-network training is negligible compared with the time to generate the data set, which is the same for both approaches.

Taking the dihedral MSM as a reference, the Cartesian MSM systematically underestimates the eigenvalues (Figure 6.5). The subspace iteration is very accurate for the first four

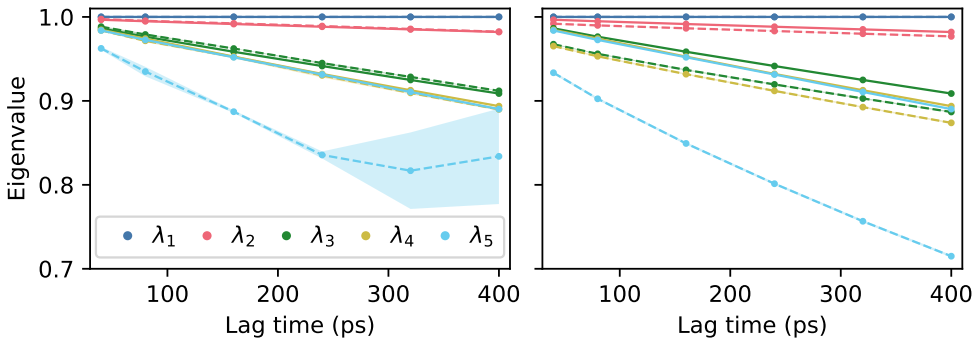

Figure 6.5: First five eigenvalues of the transition operator for AIB_9 as a function of lag time. (left) Comparison between eigenvalues computed using the dihedral MSM with 1000 clusters (solid lines) and the inexact subspace iteration (dashed lines). The shading indicates standard deviations over five trained networks for the subspace iteration. (right) Comparison between a dihedral MSM (solid lines) and Cartesian MSMs with 1000 clusters (dashed lines). The standard deviations for the Cartesian MSMs over five random seeds for k-means clustering are too narrow to be seen.

eigenvalues but the estimates for the fifth are low and vary considerably from run to run. A very small gap between λ_4 and λ_5 may contribute to the difficulty in estimating λ_5. In Figure 6.6, we plot the first two non-trivial eigenfunctions (v_2 and v_3), which align with the axes of the dPC projection. The eigenfunction v_2 corresponds to the transition between the left- and right-handed helices; the eigenfunction v_3 is nearly orthogonal to v_2 and corresponds to transitions between intermediate states. It is challenging to visualize the remaining two eigenfunctions by projecting onto the first two dPCs because v_4 and v_5 are orthogonal to v_2 and v_3. The estimates for v_2 are in qualitative agreement for all lag times tested (Figure 6.6 shows results for τ corresponding to 40 ps), but the subspace iteration results are less noisy for the shortest lag times. Moreover, the estimate for v_3 from subspace iteration agrees more closely with that from the dihedral MSM than does the estimate for v_3 from the Cartesian MSM. The subspace distance for v_2 and v_3 between the subspace iteration and the dihedral MSM is 0.947, compared with a value of 0.969 for the subspace distance between the two MSMs. Together, our results indicate that the neural networks are able to learn the leading eigenfunctions and eigenvalues of the transition operator (dynamical modes) of this system

despite being presented with coordinates that are not the natural ones for describing the dynamics.

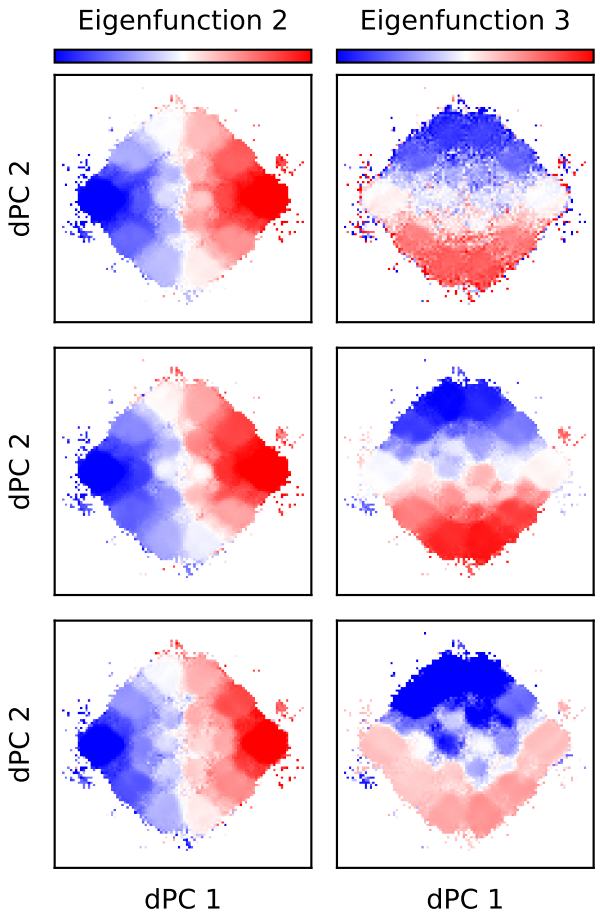

Figure 6.6: First two non-trivial eigenfunctions of AIB_9 projected onto the first two dPCs (i.e., averaged for bins in the two-dimensional space shown). (top) MSM constructed on sine and cosine of dihedral angles with 1000 clusters and lag time corresponding to 40 ps. (middle) Inexact subspace iteration using Cartesian coordinates and the same lag time. (bottom) MSM constructed on Cartesian coordinates with 1000 clusters and the same lag time.

6.8 Prediction

Inexact subspace iteration for $\mathcal{A}$ in (6.14) is equivalent to performing the inexact Richardson iteration in (6.12) on the first basis function φ_θ^1 and then performing an inexact subspace iteration for the operator $\mathcal{S}^\tau$ on the rest of the basis functions. The iteration requires unbiased

estimators of the forms

$$\langle f, \mathcal{S}^\tau g \rangle_\mu \approx \frac{1}{n} \sum_{j=1}^{n} f(X_j^0) g(X_j^{\tau \wedge T_j}) \tag{6.35}$$

and

$$\langle f, r \rangle_\mu \approx \frac{1}{n} \sum_{j=1}^{n} f(X_j^0) \sum_{t=0}^{(\tau \wedge T_j)-1} \Gamma(X_j^t), \tag{6.36}$$

where T_j is the first time X_j^t enters D^{c} and r is the right-hand side of (6.5), as previously.

The Richardson iterate, φ_θ^1, must satisfy the boundary condition $\varphi_\theta^1(x) = \Psi(x)$ for $x \notin D$. The other basis functions should satisfy $\varphi_\theta^a(x) = 0$ for $x \notin D$. In practice, we enforce the boundary conditions by explicitly setting $\varphi_\theta^1(x) = \Psi(x)$ and $\varphi_\theta^a(x) = 0$ for $a > 1$ when $x \notin D$.

When the boundary condition is zero, as for the MFPT, we find an approximate solution of the form

$$u_\theta = \sum_{a=1}^{k} w_a \varphi_\theta^a \tag{6.37}$$

by solving the k-dimensional linear system

$$\left(C^0 - C^\tau \right) w = p \tag{6.38}$$

where, for $a, b \geq 1$,

$$C_{ab}^t = \langle \varphi_\theta^a, \mathcal{S}^t \varphi_\theta^b \rangle_\mu \tag{6.39}$$

for $t = 0, \tau$, and

$$p_a = \left\langle \varphi_\theta^a, \mathbb{E}_x \left[\rho(X) \right] \right\rangle_\mu. \tag{6.40}$$

In (6.40), we introduce the notation

$$\rho(X) = \sum_{t=0}^{(\tau \wedge T)-1} \Gamma(X^t) \tag{6.41}$$

for use in Algorithm 5.

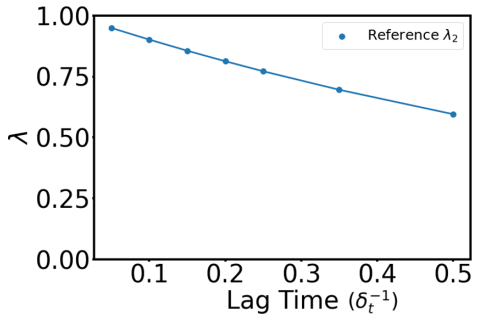

Figure 6.7: First eigenvalue of $\mathcal{S}^\tau$ (second of $\mathcal{A}$ in (6.14)) for the Müller-Brown model as a function of lag time (in units of δ_t^{-1}). The gap between this eigenvalue and the dominant eigenvalue, which is one, determines the rate of convergence of the Richardson iteration.

When the boundary condition is non-zero, as for the committor, we restrict (6.38) to a $(k-1)$-dimensional linear system by excluding the indices $a = 1$ and $b = 1$ in (6.39) and (6.40) and setting

$$\rho(X) = \varphi_\theta^1(X^{\tau\wedge T}) - \varphi_\theta^1(X^0) + \sum_{t=0}^{(\tau\wedge T)-1} \Gamma(X^t). \tag{6.42}$$

In this case the corresponding approximate solution is

$$u_\theta = \varphi_\theta^1 + \sum_{a=2}^{k} w_a \varphi_\theta^a. \tag{6.43}$$

This approximate solution corresponds to the one given by dynamical Galerkin approximation [4, 5] with the basis $\{\varphi_\theta^a\}_{a=2}^k$ and a "guess" function of φ_θ^1.

When the boundary conditions are zero, the orthogonalization procedure and the matrix K are applied to all basis functions as in Section 6.7. When the boundary conditions are non-zero, the orthogonalization procedure is only applied to the basis functions $\{\varphi_\theta^a\}_{a=2}^k$, and $K_{a1} = I_{a1}$, the $a1$ element of the identity matrix. We summarize our procedure for prediction in Algorithm 5.

Algorithm 5 Inexact subspace iteration (with L^2_μ loss) for prediction functions

Require: Subspace dimension k, stopped transition data $\{X_j^0, X_j^{\tau\wedge T_j}\}_{j=1}^n$, reward data $\{\rho(X_j)\}_{j=1}^n$, batch size B, learning rate η, number of subspace iterations S, number of inner iterations M, regularization parameters γ_1 and γ_2

Initialize $\{\varphi_\theta^a\}_{a=1}^k$ and $\{\tilde{\varphi}_1^a\}_{a=1}^k$
for $s = 1 \dots S$ **do**
  for $m = 1 \dots M$ **do**
    Sample a batch of data $\{X_j^0, X_j^{\tau\wedge T_j}\}_{j=1}^B$, $\{\rho(X_j)\}_{j=1}^B$
    $\hat{\mathcal{L}}_1 \leftarrow \frac{1}{B}\sum_{j=1}^B \sum_{a=1}^k \big[\frac{1}{2}(\sum_{b=1}^k \varphi_\theta^b(X_j^0)K_{ba})^2 - \alpha_s\big(\sum_{b=1}^k \varphi_\theta^b K_{ba}(\tilde{\varphi}_s^a(X_j^{\tau\wedge T_j}) + \rho(X_j)I_{a1})\big)\big]$
    $\hat{\mathcal{L}}_2 \leftarrow -\frac{1}{B}\sum_{j=1}^B \sum_{a=1}^k (1-\alpha_s)\sum_{b=1}^k \varphi_\theta^b K_{ba}\tilde{\varphi}_s^a(X_j^0)$
    $\hat{\mathcal{L}}_K \leftarrow \gamma_1 \|K - \mathrm{diag}(K)\|_F^2$
    $\hat{\mathcal{L}}_{\mathrm{norm}} \leftarrow \gamma_2 \sum_{a=2}^k (2\nu_a(\frac{1}{B}\sum_{j=1}^B(\varphi_\theta^a(X_j^0)^2) - 1) - \nu_a^2)$
    $\hat{\mathcal{L}} \leftarrow \hat{\mathcal{L}}_1 + \hat{\mathcal{L}}_2 + \hat{\mathcal{L}}_K + \hat{\mathcal{L}}_{\mathrm{norm}}$
    $\theta \leftarrow \theta - \eta\nabla_\theta\hat{\mathcal{L}}$
    $K \leftarrow K - \eta\,(\mathrm{triu}(\nabla_K\hat{\mathcal{L}}))$
    $\nu_a \leftarrow \nu_a + \eta\nabla_{\nu_a}\hat{\mathcal{L}}$
  end for
  if $\Psi(x) = 0$ **then**
    Compute the matrix $\Phi_{ia} = \varphi_\theta^a(X_i^0)$ ▷ $\Phi \in \mathbb{R}^{n\times k}$
  else
    Compute the matrix $\Phi_{ia} = \varphi_\theta^a(X_i^0)$ for $a > 1$ ▷ $\Phi \in \mathbb{R}^{n\times(k-1)}$
  end if
  Compute QR-decomposition $\Phi = QR$
  Compute diagonal matrix $N_{aa}^2 = \sum_i \varphi_\theta^a(X_i^0)^2$
  $\tilde{\varphi}_s^a \leftarrow \sum_{b=1}^k \varphi_\theta^b\,(R^{-1}N)_{ba}$ ▷ if $\Psi(x) = 0$ exclude $a = 1$
  $K_{1:i,i} \leftarrow \arg\min_c \frac{1}{n}\sum_{j=1}^n \Big(\sum_{a=1}^i \varphi_\theta^a(X_j^0)c_a - \tilde{\varphi}_s^a(X_j^{\tau\wedge T_j})\Big)^2 + \gamma_2\sum_{a=1}^{i-1} c_a^2$
end for
Compute the matrix $C_{ab}^t = \frac{1}{n}\sum_{j=1}^n \varphi_\theta^a(X_j^0)\varphi_\theta^b(X_j^t)$ for $t = 0, \tau\wedge T_j$ ▷ $C^t \in \mathbb{R}^{k\times k}$
Compute the vector $p_a = \frac{1}{n}\sum_{j=1}^n \varphi_\theta^a(X_j^0)\rho(X_j)$
Solve linear system $(C^0 - C^\tau)w = p$ ▷ if $\Psi(x) = 0$ enforce $w_1 = 1$
return $u = \sum_{a=1}^k w_a\varphi_\theta^a$

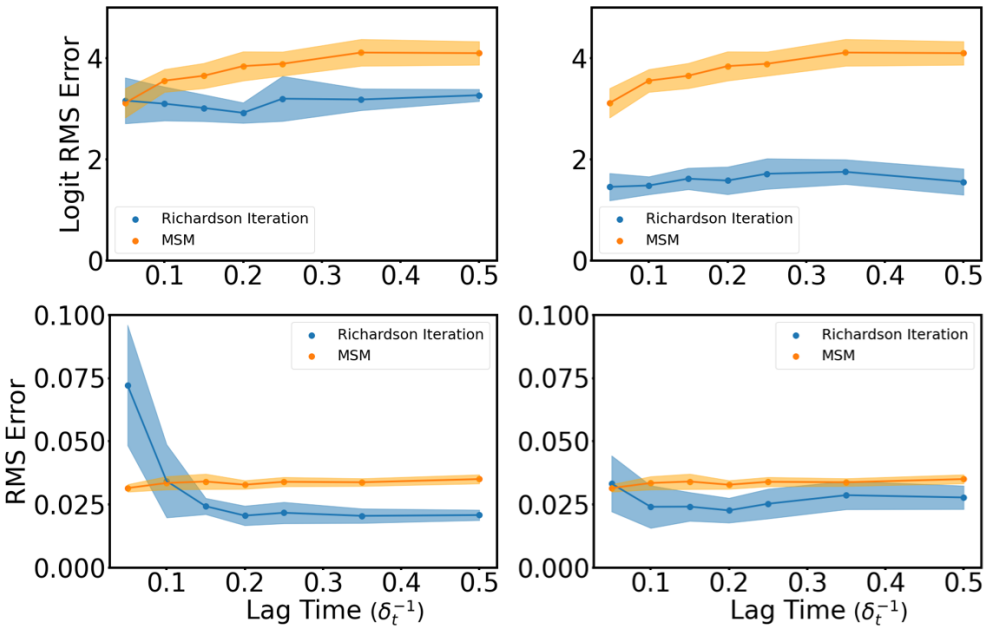

Figure 6.8: Committor prediction for the Müller-Brown system as a function of lag time (in units of δ_t^{-1}). (left) Comparison of the inexact Richardson iteration using the L^2_μ loss and an MSM with 400 states. (right) Same comparison using the softplus loss in place of the L^2_μ loss.

6.8.1 Müller-Brown committor

In this section, we demonstrate the use of our method for prediction by estimating the committor for the Müller-Brown model with a shallow intermediate basin at $(-0.25, 0.5)$ (Figure 6.1). Here the sets A and B are defined as in Eq. (6.24) and T is the time of first entrance to $D^{\mathsf{c}} = A \cup B$. In this case, a one-dimensional subspace iteration (i.e., $k = 1$ in Algorithm 5) appears sufficient to accurately solve the prediction problem. Figure 6.7 shows the largest eigenvalue of the stopped transition operator $\mathcal{S}^\tau$ (the second largest of $\mathcal{A}$ in (6.14)) computed from our grid-based reference scheme (Section 6.6.1). Richardson iteration should converge geometrically in this eigenvalue [197], and so, for moderate lag times, we can expect our method to converge in a few dozen iterations. To initialize the algorithm we choose $\tilde{\varphi}_1^1 = \mathbb{1}_B$. All other hyperparameters are listed in Table 6.1.

We compare the estimate of the committor from our approach with that from an MSM constructed from the same amount of data by using k-means to cluster the configurations

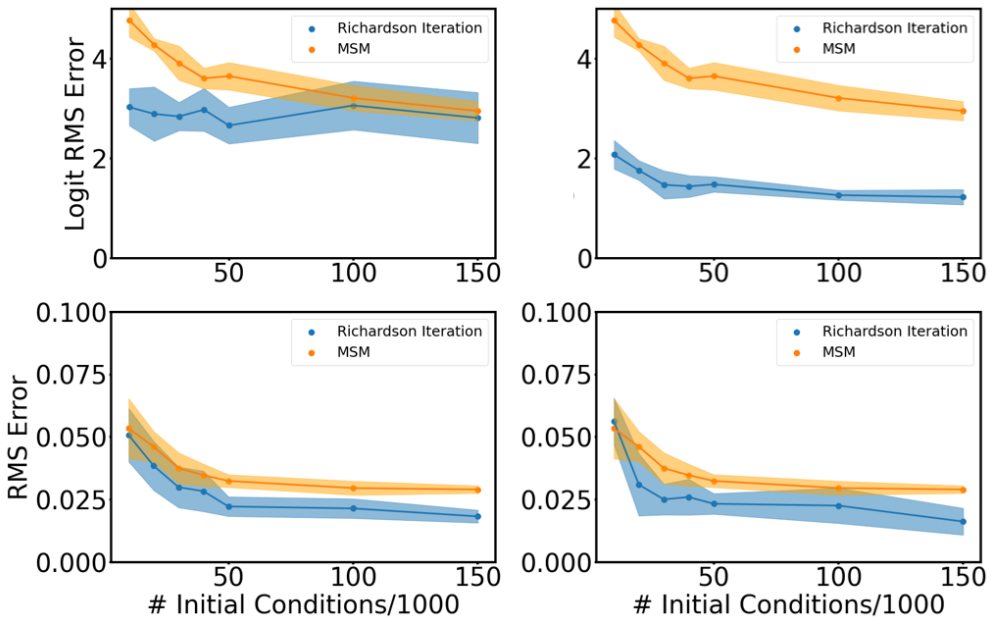

Figure 6.9: Committor prediction for the Müller-Brown as a function of number of initial conditions for a fixed lag time of $\tau = 0.1\delta_t^{-1}$. (left) Comparison of inexact Richardson iteration using the L^2_μ loss and an MSM with 400 states. (right) Same comparison using the softplus loss in place of the L^2_μ loss.

outside A and B into 400 states and counting the transitions between clusters. In addition to the root mean square error (RMSE) for the committor itself, we show the RMSE of

$$\text{logit}_\varepsilon(q) = \log\left(\frac{\varepsilon + q}{1 + \varepsilon - q}\right) \tag{6.44}$$

for points outside A and B. This function amplifies the importance of values close to zero and one. We include ε because we want to assign only a finite penalty if the procedure estimates q to be exactly zero or one; we use $\varepsilon = e^{-20}$.

Results as a function of lag time are shown in Figure 6.8. We see that the Richardson iterate is more accurate than the MSM for all but the shortest lag times. When using the L^2_μ loss, the results are comparable, whereas the softplus loss allows the Richardson iterate to improve the RMSE of the logit function in (6.44) with no decrease in performance with respect to the RMSE of the committor. Results as a function of the size of the data set are

shown in Figure 6.9 for a fixed lag time of $\tau = 0.1\delta_t^{-1}$. The Richardson iterate generally does as well or better than the MSM. Again, the differences are more apparent in the RMSE of the logit function in (6.44). By that measure, the Richardson iterate obtained with both loss functions is significantly more accurate than the MSM for small numbers of trajectories. The softplus loss maintains an advantage even for large numbers of trajectories.

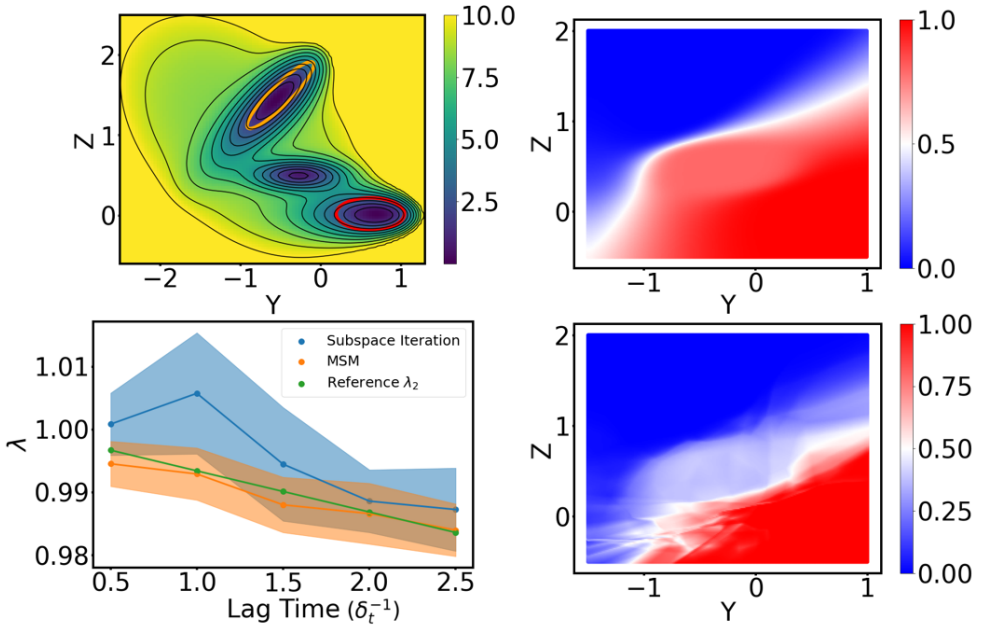

Figure 6.10: Richardson iteration for the committor converges slowly for a Müller-Brown potential with a deepened intermediate. (top left) Potential energy surface, with states A and B indicated. Contour lines are drawn every 1 β^{-1}. (top right) Reference committor. (bottom left) Dominant eigenvalue as a function of lag time (in units of δ_t^{-1}) from an MSM with 400 states, subspace iteration, and the grid-based reference. (bottom right) Example of the Richardson iteration after 400 iterations. Note the overfitting artifacts and lack of convergence near the intermediate state.

6.8.2 Accelerating convergence by incorporating eigenfunctions

As discussed in Section 6.3, we expect Richardson iteration to converge slowly when the largest eigenvalue of $\mathcal{S}^\tau$, λ_1, is close to 1. More precisely, the number of iterations required to reach convergence should scale with $-1/\log\lambda_1 = \mathbb{E}[T]/\tau$, the mean escape time from the quasi-stationary distribution to the boundary of D divided by the lag time. With this in mind, we can expect inexact Richardson iteration for the Müller-Brown to perform poorly if we deepen the intermediate basin at $(-0.25, 0.5)$ as in Figure 6.10 (top left). Again, the sets A and B are defined as in (6.24), and T is the time of first entrance to $D^\mathsf{c} = A \cup B$. In this case, $-1/\log\lambda_1$ is on the order of 100 for the lag times we consider and, as expected, inexact Richardson iteration converges slowly (Figure 6.10, bottom left). Estimates of the committor by inexact Richardson iteration do not reach the correct values even after hundreds

of iterations (Figure 6.10, bottom right).

We now show that convergence can be accelerated dramatically by incorporating additional eigenfunctions of $\mathcal{S}^\tau$ (i.e., $k > 1$ in Algorithm 5). For the Müller-Brown model with a deepened intermediate basin, the second eigenvalue of $\mathcal{S}^\tau$ is of order 10^{-4} for a lag time of $\tau = 1000$ steps or $1\,{\delta_t}^{-1}$ (while the first is near one as discussed above). We therefore choose $k = 2$ with $\tilde{\varphi}_1^2$ initialized as a random linear combination of coordinate functions as in previous examples. We run the subspace iteration for four iterations, compute the committor as a linear combination of the resulting functions, and then refine this result with a further ten Richardson iterations (i.e., $k = 1$ with the starting vector as the output of the $k = 2$ subspace iteration). To combine the functions, we use a linear solve which incorporates memory (Algorithm 6) [7, 136]. We find that the use of memory improves the data-efficiency substantially for poorly conditioned problems. For our tests here, we use three memory kernels, corresponding to $\tau_{\text{mem}} = \lfloor \tau/4 \rfloor$.

The bottom row of Figure 6.11 illustrates the idea of the subspace iteration. The second eigenfunction (Figure 6.11, center) is peaked at the intermediate. As a result, the two neural-network functions linearly combined by the Galerkin approach with memory can yield a good result for the committor (Figure 6.11, bottom right). Figure 6.12 compares the RMSE for the committor and the RMSE for the logit in (6.44) for Algorithm 5 with $k = 1$ (pure Richardson iteration) and $k = 2$ (incorporating the first non-trivial eigenfunction), and an MSM with 400 states. We see that the Richardson iteration suffers large errors at all lag times; as noted previously, this error is mainly in the vicinity of the intermediate. The MSM cannot accurately compute the small probabilities, but does as well as the subspace iteration in terms of RMSE.

Algorithm 6 Memory-corrected linear solve for predictions

Require: Stopped transition data $\{X_j^0, X_j^{1\wedge T_j}, \ldots, X_j^{\tau\wedge T_j}\}_{j=1}^n$, guess function h, reward data $\{\rho(X_j)\}_{j=1}^n$, basis set $\{f^a\}_{a=1}^k$, lag between memory kernels τ_{mem}.

for $s = 0 \ldots (\tau/\tau_{\text{mem}})$ **do**
  Initialize the matrix C^s with zeros ▷ $C^s \in \mathbb{R}^{(k+1)\times(k+1)}$
  $C^s_{11} \leftarrow 1$
  for $a = 2 \ldots k$ **do**
    $C^s_{a1} \leftarrow \frac{1}{n}\sum_{j=1}^n f^a(X_j^0)\rho(X_j^{s\tau_{\text{mem}}\wedge T_j})$
    for $b = 2 \ldots k$ **do**
      $C^s_{ab} \leftarrow \frac{1}{n}\sum_{j=1}^n f^a(X_j^0) f^b(X_j^{s\tau_{\text{mem}}\wedge T_j})$
    end for
  end for
end for
$A \leftarrow C^1 - C^0$
for $s = 0 \ldots (\tau/\tau_{\text{mem}}) - 2$ **do**
  $M^s \leftarrow C^{s+2} - C^{s+1} - A(C^0)^{-1}C^{s+1} - \sum_{j=0}^s M^j (C^0)^{-1} C^{s-j}$
end for
$A_{\text{mem}} \leftarrow A + \sum_{s=0}^{(\tau/\tau_{\text{mem}})-2} M^s$
Solve $A_{\text{mem}} w = w$
return $u = h + \sum_{a=2}^k w_a f^a$

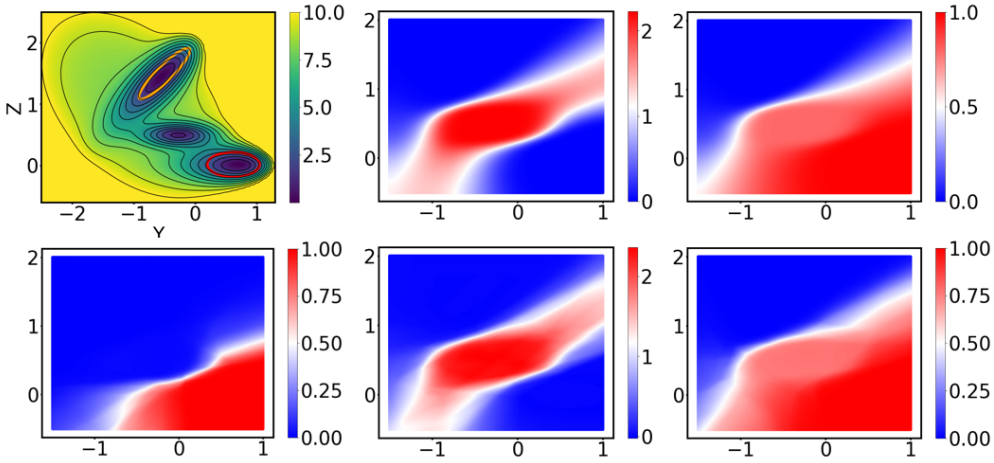

Figure 6.11: Illustration of the subspace iteration for the Müller-Brown committor. (top left) Modified Müller-Brown potential. (top center) Reference second eigenfunction. (top right) Reference committor. (bottom left) Neural-network Richardson iterate after four iterations. (bottom center) First non-dominant eigenfunction obtained from the neural network after four iterations. (bottom right) Committor resulting from linear combination of the Richardson iterate and second eigenfunction. Results shown are for $\tau = 1000$ steps (i.e., $1\,{\delta_t}^{-1}$).

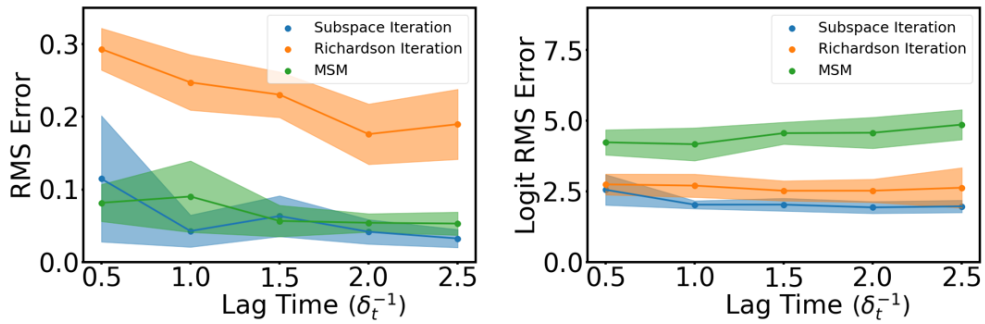

Figure 6.12: Committor for the Müller-Brown potential with deepened intermediate as a function of lag time (in units of ${\delta_t}^{-1}$). (left) Comparison of RMSE for subspace iteration as described above, Richardson iteration (as in Section 6.8.1 but instead with 500 subspace iterations), and an MSM with 400 states. (right) RMSE of the logit function in (6.44).

6.8.3 AIB_9 prediction results

As an example of prediction in a high-dimensional system, we compute the committor for the transition between the left- and right-handed helices of AIB_9 using the inexact Richardson iteration scheme ($k = 1$ in Algorithm 5) with the softplus loss function. Specifically, for this committor calculation T is the time of first entrance to $D^{\mathsf{c}} = A \cup B$ with A and B defined in Section 6.6.2. As before, we initialize $\tilde{\varphi}_1^1 = \mathbb{1}_B$.

To validate our results, we use the 5 μs reference trajectories to compute an empirical committor as a function of the neural network outputs, binned into intervals:

$$\bar{q}(s) = \mathbb{P}\left[X^T \in B \;\middle|\; u_\theta(X^0) \in [s, s + \Delta s]\right] \tag{6.45}$$

for $s \in [0, 1 - \Delta s]$. Here, we use $\Delta s = 0.05$. The overall error in the committor estimate is defined as

$$q \text{ error} = \left(\Delta s \sum_{n=0}^{1/\Delta s - 1} \left[\bar{q}(n\Delta s) - n\Delta s\right]^2\right)^{1/2}. \tag{6.46}$$

While this measure of error can only be used when the data set contains trajectories of long enough duration to reach D^{c}, it has the advantage that it does not depend on the choice of projection that we use to visualize the results.

Results for the full data set with τ corresponding to 400 ps are shown in Figure 6.13. The projection on the principal components is consistent with the symmetry of the system, and the predictions show good agreement with the empirical committors. As τ decreases, the results become less accurate (Figure 6.14, top left); at shorter lag times we would expect further increases in the error. We also examine the dependence of the results on the size of the data set by subsampling the short trajectories and then training neural networks on the reduced set of trajectories (Figure 6.14, top right). We find that the performance steadily drops as the number of trajectories is reduced and degrades rapidly for the data sets subsampled more than 20-fold (Figure 6.14, bottom), corresponding to about 7 μs of total sampling.

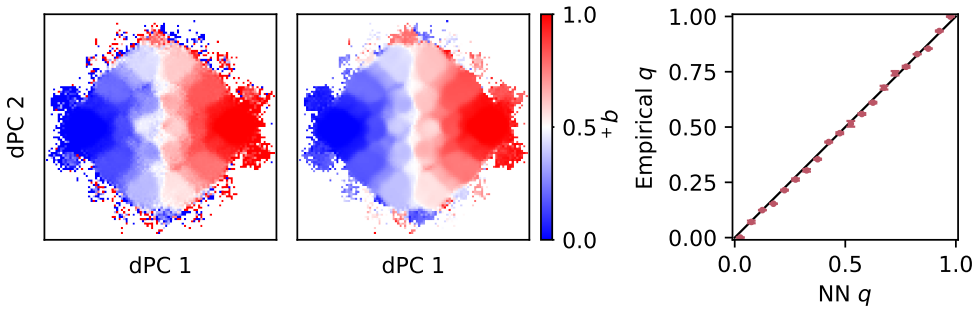

Figure 6.13: AIB_9 committor for the transition between left- and right-handed helices. (left) Averages of $\mathbb{1}_B(X^T)$ for initial conditions in bins in the first two dPCs computed from 20 long (5 μs) trajectories. (middle) Averages of representative neural-network committors trained on the data set of 6,910 short (20 ns) trajectories; τ corresponds to 400 ps. (right) Comparison between empirical committors (as defined in (6.45)) and the neural-network committors (trained as for the middle panel). Error bars indicate standard deviations over ten different initializations of the neural-network parameters.

Finally, we compute the MFPT to reach the right-handed helix using the same data set. For the MFPT calculation T is the time of first entrance to $D^{\mathsf{c}} = B$. Note that the time of first entrance to B includes long dwell times in A and is expected to be much larger than the time of first entrance to $A \cup B$.

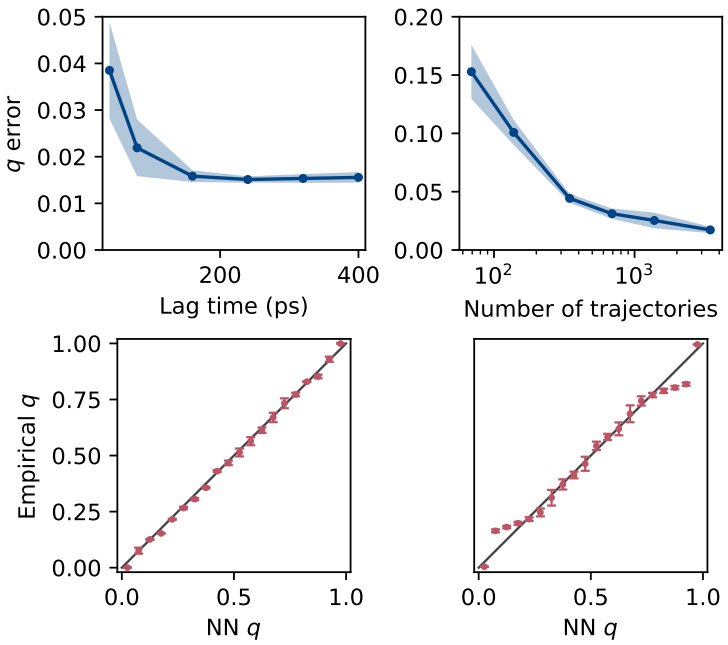

Figure 6.14: AIB_9 committor for the transition between left- and right-handed helices, as functions of lag time (in ps) and number of initial conditions. (top left) Error in the committor as a function of lag time (in ps). Shading indicates the standard deviation over ten different initializations of the neural-network parameters. (top right) Error in the committor as a function of the number of initial conditions with τ corresponding to 160 ps. Shading indicates the standard deviation over ten different random samples of the trajectories. (bottom) Comparison between empirical committors and neural-network committors trained on data sets with (left) 1/2 and (right) 1/20 of the short trajectories. Error bars indicate standard deviations over ten random samples of the trajectories.

We compare against an empirical estimate of the MFPT defined by

$$\bar{m}(s) = \mathbb{E}\left[T \middle| u_\theta(X^0) \in [s, s + \Delta s]\right] \tag{6.47}$$

for $s \in [0, m_{\max} - \Delta s]$ where $\Delta s = 3$ and $m_{\max} = 57$ ns. Overall error is defined analogously to Eq. (6.46).

In Figure 6.15, we show the MFPT obtained from Algorithm 5 with $k = 5$ and the L^2_μ loss function. Initially we set $\tilde{\varphi}^1_1$ equal to an arbitrary positive function (we use $5\mathbb{1}_A$) and $\tilde{\varphi}^a_s$ for $a > 1$ to a random linear combination of coordinate functions. In Figure 6.16 we examine the convergence of the MFPT from the left-handed helix to the right-handed helix for the

MFPT computed with $k = 1$ (pure Richardson iteration) and $k = 5$. The horizontal line indicates a MFPT of about 56 ns estimated from the long reference trajectories. We see that the algorithm with $k = 5$ converges much faster (note the differing scales of the horizontal axes) and yields accurate results at all lag times other than the shortest shown. The need to choose $k > 1$ for this MFPT calculation is again consistent with theoretical convergence behavior of exact subspace iteration. Because the typical time of first entrance to B from points in A is very large, we expect the dominant eigenvalue of $\mathcal{S}^\tau$ to be very near to one when $D = B^\mathsf{c}$. In contrast, the committor calculation benefits from the fact that the time of first entrance to $A \cup B$ is much shorter, implying a smaller dominant eigenvalue of $\mathcal{S}^\tau$ when $D = (A \cup B)^\mathsf{c}$.

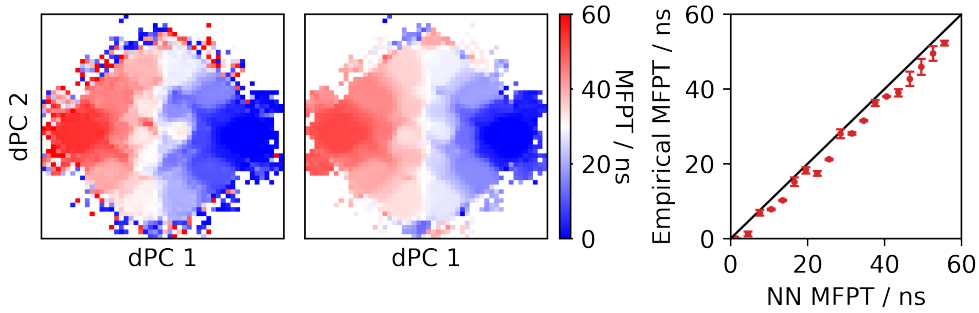

Figure 6.15: AIB_9 MFPT to the right-handed helix. (left) Averages of the time to next reach B for initial conditions in bins in the first two dPCs computed from 20 long (5 μs) trajectories. (middle) Averages of representative neural-network committors trained on the data set of 6,910 short (20 ns) trajectories; τ corresponds to 400 ps. (right) Comparison between empirical committors (as defined in (6.47)) and the neural-network committors (trained as for the middle panel). Error bars indicate standard deviations over ten different initializations of the neural-network parameters.

6.9 Conclusions

In this work we have presented a method for spectral estimation and rare-event prediction from short-trajectory data. The key idea is that we use the data as the basis for an inexact subspace iteration. For the test systems that we considered, the method not only outperforms

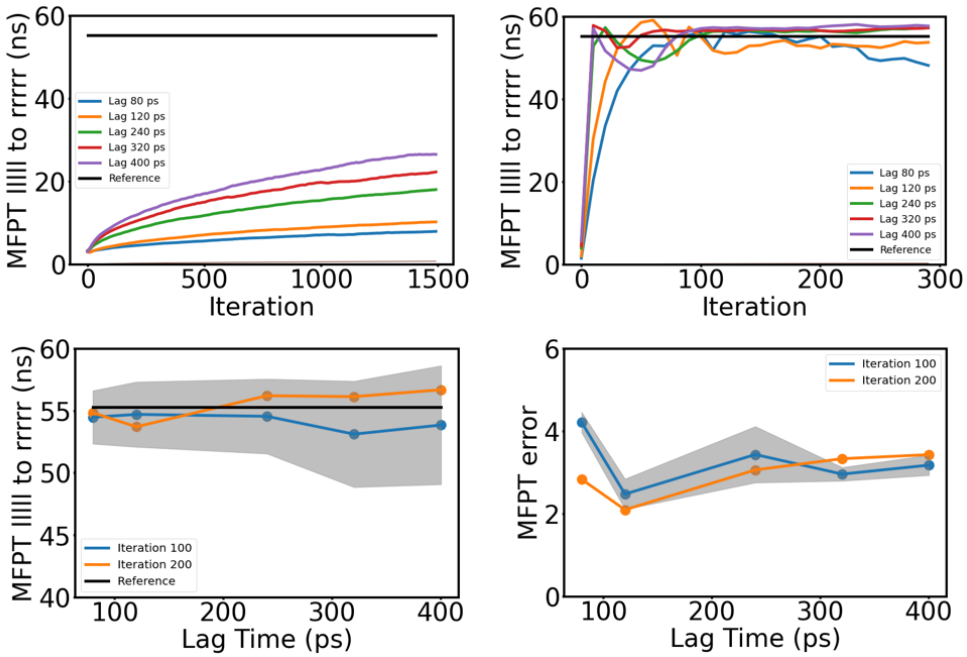

Figure 6.16: MFPT between left- and right-handed helices for the AIB_9 system. (top left) Convergence of Richardson iteration. The llllll to rrrrr MFPT is computed by averaging the richardson iteration result over each frame of each of the long reference trajectories in the llllll state. (top right) Convergence of a five-dimensional subspace iteration. (bottom left) MFPT after 100 and 200 subspace iterations as a function of lag time. Shading indicates standard deviations over ten different initializations of the neural-network parameters. (bottom right) Overall error in MFPT. To obtain the results shown in this figure, we first use the short-trajectory dataset to train neural networks to predict the MFPT; we then use these networks with fixed parameters to evaluate the MFPT for all structures in the long reference trajectories and average the results for those structures in the left-handed helix state. The horizontal lines in the top panels are obtained from averaging the time to the right-handed helix for the same structures.

high-resolution MSMs, but it can be tuned through the choice of loss function to compute committor probabilities accurately near the reactants, transition states, and products. Other than the Markov assumption, our method requires no knowledge of the underlying model and puts no restrictions on its dynamics.

As discussed in prior neural-network based prediction work [147, 167], our method is sensitive to the quality and distribution of the initial sampling data. However, our work shares with Strahan et al. [147] the major advantage of allowing the use of arbitrary inner

products. This enables adaptive sampling of the state space [147, 164] and—together with the features described above—the application to observational and experimental data, for which the stationary distribution is generally unavailable.

In the present work, we focused on statistics of transition operators, but our method can readily be extended to solve problems involving their adjoint operators as well. By this means, we can obtain the stationary distribution as well as the backward committor. The combination of forward and backward predictions allows the analysis of transition paths using transition path theory without needing to generate full transition paths [3, 151, 203] and has been used to understand rare transition events in molecular dynamics [5, 8, 76, 185, 190, 205] and geophysical flows [165, 206–209]. We leave these extensions to future work.

In cases in which trajectories that reach the reactant and product states are available, it would be interesting to compare our inexact iterative schemes against schemes for committor approximation based on logistic regression and related approaches [108, 109, 154–156, 200–202]. These schemes are closely related to what is called "Monte-Carlo" approximation in reinforcement learning [195], and also to the inexact Richardson iteration that we propose here with $\tau \to \infty$.

We have seen that temporal difference (TD) learning, a workhorse for prediction in reinforcement learning, is closely related to an inexact form of Richardson iteration. Variants like TD(λ), have similar relationships to inexact iterative schemes. As we showed, subspace iteration is a natural way of addressing slow convergence. We thus anticipate that our results have implications for reinforcement learning, particularly in scenarios in which the value function depends on the occurrence of some rare-event. Finally, we note that our framework can be extended to the wider range of iterative numerical linear algebra algorithms. In particular, Krylov or block Krylov subspace methods may offer further acceleration. In fact, very recently an approach along these lines was introduced for value function estimation in reinforcement learning [210].

CHAPTER 7

AUGMENTED TRANSITION PATH THEORY

This chapter is adapted from C. Lorpaiboon et al., "Augmented transition path theory for sequences of events", J. Chem. Phys. **157**, 094115 (2022), with the permission of AIP Publishing.

Transition path theory provides a statistical description of the dynamics of a reaction in terms of local spatial quantities. In its original formulation, it is limited to reactions that consist of trajectories flowing from a reactant set A to a product set B. We extend the basic concepts and principles of transition path theory to reactions in which trajectories exhibit a specified sequence of events and illustrate the utility of this generalization on examples.

7.1 Introduction

Many reactions studied today proceed through competing pathways. Understanding such reactions relies on being able to assess the relative importance of the competing pathways and how they contribute to overall rates. When the pathways are well separated, they can be treated independently, often by traditional theories that assume a well-defined activated complex (transition state) and a simple form for the underlying (free) energy landscape governing the dynamics [211, 212]. However, when the (observed) dynamics are stochastic, the pathways of reactions often overlap in configuration space. Approaches that treat competing pathways in a unified fashion are thus needed.

To this end, here, we build on transition path theory (TPT) [3, 73, 74, 138, 151]. The core idea of transition path theory is that the statistics of the ensemble of reactive trajectories can be related to quantities that are local in space: probability currents of reactive trajectories (henceforth, reactive currents) and committors. These quantities enable TPT to go beyond traditional theories by providing information about mechanisms. Reactive currents quantify

flows in phase space. Committors, which are probabilities of reaching one metastable state before another, by definition characterize the progress of stochastic reactions [148].

In its traditional formulation, TPT focuses on transitions between two metastable states. In the present chapter, we extend TPT to compute statistics for sequences of events, and we show how this significantly expands its applicability. Our work builds on but goes beyond previous studies. It is closely related to history-augmented Markov state models, in which states are labeled based on the last metastable state visited [37]. Separating the ensemble of reactive trajectories using these labels enables rates to be computed from the flux into a metastable state [213, 214], as well as reactive currents and committors from the underlying trajectories [205]. Our approach generalizes the labeling strategy to sequences of arbitrary numbers of states and allows specification of not just past events but also future ones.

Our work also has connections to that of Koltai and co-workers, who extended TPT to allow trajectories to leave and enter the region connecting the metastable states to analyze trajectory segments from satellite data for drifters in the ocean [207]. Specifically, they redefined committors to exclude trajectories that were not wholly within this region and noted that this approach could be used to exclude trajectories that pass through selected states. They also considered computing statistics for trajectories beginning and ending in specific portions of metastable states. Both of these developments allow statistics to be computed for subsets of reactive trajectories based on the states that they visit.

We present our work as follows. First, in Sections 7.2.1 and 7.2.2, we review how TPT expresses path statistics in terms of spatially and temporally local quantities using committors. In Section 7.2.3, we present a motivating example in which this is not possible within the existing framework, but it can be made possible by augmenting the stochastic process with labels that account for sequences of events. This is the key idea of the chapter. While this idea is straightforward, formulating the theory in full generality requires some technical development, and readers may wish to skim Sections 7.2.4 and 7.4 initially, focusing on

the brief summaries in the first paragraphs of Sections 7.2.4 and 7.4. In Section 7.2.4, we discuss conditions of consistency and Markovianity that must be satisfied for TPT to apply to the augmented process. We also generalize committors and integrals based on them. In Section 7.3, we review the most commonly computed TPT statistics and show how they can be computed in our augmented TPT framework. In Section 7.4, we introduce a procedure for constructing the augmented process from pairs of successive time points rather than full trajectory segments, and we show how processes can be composed to construct more complex ones. We summarize the operational procedure in Section 7.5. Then, in Section 7.6, we illustrate our approach on two systems with multiple pathways and intermediates. In Section 7.7, possible extensions and numerical strategies for treating more complicated systems are discussed. In Appendix 7.8.1, we provide a method for calculating augmented TPT statistics using a finite difference scheme. Code implementing this method is available at `github.com/dinner-group/atpt`.

7.2 Framework

In this section, we review TPT to show how it casts statistics for reactive trajectories in terms of local quantities. Then we present an example that cannot be treated within the traditional TPT framework and show how it can be treated by introducing an augmented process. The essential idea is that the augmented process accounts for the order of events. The challenge in implementing this idea is that, for a finite-length trajectory segment, we generally do not know the events that occur before and after it.

For clarity, we present our results in terms of a discrete-time Markov process X_t with time step ϵ, but our results generalize to continuous-time processes in the limit $\epsilon \to 0$. We denote the time interval $r, r+\epsilon, \ldots, s$ by $r : s$ and a trajectory segment on this time interval by $X_{r:s} = (X_r, X_{r+\epsilon}, \ldots, X_s)$. For conciseness, we denote an infinite trajectory $X_{-\infty:\infty}$ by X.

7.2.1 Ensemble of Reactive Trajectories

In both traditional and augmented TPT, statistics are computed over the ensemble of reactive trajectories. In this section, we define this ensemble and integrals over it. Here, we focus on traditional TPT, but the framework generalizes to augmented TPT immediately once we define the augmented process in Section 7.2.4.

Traditional TPT considers a reaction from a set A to a set B via trajectories that cross a region D. In anticipation of our augmented framework, we allow $D \subseteq (A \cup B)^c$ as in Miron et al. [207]. We consider a trajectory $X_{r:s}$ to be reactive if its first time point X_r is in the reactant set A, its last time point X_s is in the product set B, and all intervening time points $X_{r+\epsilon:s-\epsilon}$ are in the region D. Mathematically, we implement this definition through the indicator function

$$\omega(X_{r:s}) = \mathbb{1}_{A\times D\times\cdots\times D\times B}(X_{r:s}), \tag{7.1}$$

where

$$\mathbb{1}_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{otherwise,} \end{cases} \tag{7.2}$$

and $S_1 \times \cdots \times S_n = \{(x_1, \ldots, x_n) \mid x_1 \in S_1, \ldots, x_n \in S_n\}$ is the n-fold Cartesian product.

Given (7.1), we define the integral over the ensemble of reactive trajectories to be

$$\mathbb{I}^X_\omega[\eta] = \lim_{T\to\infty} \frac{1}{2T}\, \mathbb{I}^X\Big[\sum_{\substack{r=-T:T-\epsilon \\ s=r+\epsilon:T}} \omega(X_{r:s})\eta(X_{r:s})\Big], \tag{7.3}$$

where $\mathbb{I}^X[f(X)]$ is the integral of $f(X)$ over the distribution of infinite trajectories X, which we denote using the superscript X. When X_t is a stationary ergodic process and $\mathbb{I}^X$ is the expectation $\mathbb{E}^X$ over the distribution of infinite trajectories, as in traditional TPT, we can compute $\mathbb{I}^X_\omega[\eta]$ from a single infinite trajectory and so $\mathbb{I}^X$ can be omitted; however, this is not necessarily true for time-dependent processes, as in Helfmann et al. [215], or for augmented

processes, as in this work. As X_t is a Markov process, we can compute this integral by sampling configurations X_{-T} from the distribution of states at time $-T$ and propagating until time T. The prefactor $1/(2T)$ ensures that $\mathbb{I}_\omega^X[\eta]$ gives consistent results across different trajectory lengths $2T$. We can then calculate expectations over the ensemble of reactive trajectories as

$$\mathbb{E}_\omega^X[\eta] = \mathbb{I}_\omega^X[\eta] / \mathbb{I}_\omega^X[1], \tag{7.4}$$

where the normalization factor $\mathbb{I}_\omega^X[1]$ is the expected number of reactive trajectories which start (or end) per unit time. The integral $\mathbb{I}_\omega^X[\eta]$ thus yields statistics that can be used to characterize and compare reaction pathways.

7.2.2 Transition Path Theory

In general, the ensemble of reactive trajectories can only be meaningfully interpreted through its statistics. Although these statistics can be computed directly from the ensemble of reactive trajectories, TPT enables them to be computed from other data as well by expressing them in terms of spatially and temporally local quantities.

TPT specifically considers functions that can be written as

$$\eta(X_{r:s}) = \sum_{t=r:s-\epsilon} \gamma(X_{t:t+\epsilon})\epsilon, \tag{7.5}$$

where $\gamma(X_{t:t+\epsilon})$ is a function of successive time points X_t and $X_{t+\epsilon}$. In this case, substituting (7.5) into (7.3) and exchanging the order of the sums yields

$$\mathbb{I}_\omega^X[\eta] = \lim_{T\to\infty} \frac{\epsilon}{2T} \sum_{t=-T:T-\epsilon} \mathbb{I}^X\Big[\sum_{\substack{r=-T:t \\ s=t+\epsilon:T}} \omega(X_{r:s})\gamma(X_{t:t+\epsilon})\Big] \tag{7.6}$$

$$= \mathbb{I}^{X,t}\Big[\sum_{\substack{r=-\infty:t \\ s=t+\epsilon:\infty}} \omega(X_{r:s})\gamma(X_{t:t+\epsilon})\Big], \tag{7.7}$$

where from (7.6) to (7.7) we have taken the limit $T \to \infty$ and performed a time average over t, which we denote by the superscript t. That is,

$$\mathbb{I}^{X,t}[f(X,t)] = \lim_{T\to\infty} \frac{\epsilon}{2T} \sum_{t=-T:T-\epsilon} \mathbb{I}^{X}[f(X,t)]. \tag{7.8}$$

We can then factor

$$\sum_{\substack{r=-\infty:t \\ s=t+\epsilon:\infty}} \omega(X_{r:s}) = \mathbb{1}_A(X_{S_-(t)})\mathbb{1}_B(X_{S_+(t+\epsilon)}), \tag{7.9}$$

where

$$S_-(t) = \max\{t' \leq t \mid X_{t'} \in D^c\} \tag{7.10}$$

$$S_+(t) = \min\{t' \geq t \mid X_{t'} \in D^c\} \tag{7.11}$$

are the last exit time from D^c and the first entrance time to D^c, respectively. Equation (7.9) results from the identities

$$\sum_{r=-\infty:t} \mathbb{1}_{A\times D\times\cdots\times D}(X_{r:t}) = \mathbb{1}_A(X_{S_-(t)}), \tag{7.12}$$

$$\sum_{s=t:\infty} \mathbb{1}_{D\times\cdots\times D\times B}(X_{t:s}) = \mathbb{1}_B(X_{S_+(t)}). \tag{7.13}$$

We arrive at (7.13) by observing that only one term in the sum can be nonzero: because D and B are disjoint, $\mathbb{1}_{D\times\cdots\times D\times B}(X_{t:s})$ can be nonzero only when $s = S_+(t)$ is the first time $t' \geq t$ that $X_{t'} \notin D$. Similar logic applies for (7.12).

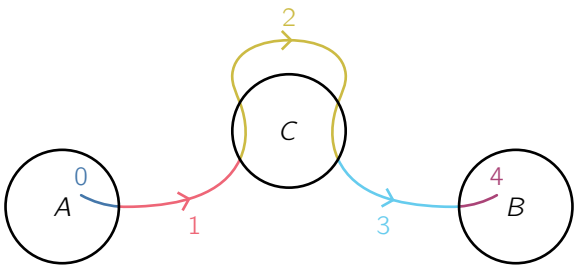

Figure 7.1: Values of Y_t at each point of a reactive trajectory described by (7.19).

Consequently, for a Markov process, (7.7) can be expressed in terms of only local quantities:

$$\mathbb{I}_\omega^X[\eta] = \mathbb{I}^{X,t}[\mathbb{1}_A(X_{S_-(t)})\mathbb{1}_B(X_{S_+(t+\epsilon)})\gamma(X_{t:t+\epsilon})] \tag{7.14}$$

$$= \mathbb{I}^{X,t}[q_-(X_t)q_+(X_{t+\epsilon})\gamma(X_{t:t+\epsilon})], \tag{7.15}$$

where we have defined the backward and forward committors respectively as

$$q_-(X_t) = \mathbb{E}^X[\mathbb{1}_A(X_{S_-(t)}) \mid X_t], \tag{7.16}$$

$$q_+(X_t) = \mathbb{E}^X[\mathbb{1}_B(X_{S_+(t)}) \mid X_t]. \tag{7.17}$$

The backward committor $q_-(X_t)$ is the probability that X_t last came from A rather than $(A \cup D)^c$, and the forward committor $q_+(X_t)$ is the probability that X_t will go to B before $(B \cup D)^c$.

The main result of this section is (7.15). The advantage of (7.15) over (7.3) is that the former involves only statistics that are local in space and time. This aids in the interpretation of these statistics, and it enables their estimation from short trajectories $X_{t:t+\epsilon}$, thus eliminating the need for trajectories that actually cross from A to B.

7.2.3 A Motivating Reaction

To motivate our augmented framework, we consider a reaction with an intermediate state C and compute statistics only for reactive trajectories that proceed through the intermediate. The function that selects trajectories of interest is

$$\omega(X_{r:s}) = \sum_{\substack{t_1=r+\epsilon:s-\epsilon \\ t_2=t_1:s-\epsilon}} [\mathbb{1}_{A\times D\times\cdots\times D}(X_{r:t_1-\epsilon}) \\ \mathbb{1}_{C\times(C\cup D)\times\cdots\times(C\cup D)\times C}(X_{t_1:t_2}) \\ \mathbb{1}_{D\times\cdots\times D\times B}(X_{t_2+\epsilon:s})], \tag{7.18}$$

where $D = (A \cup B \cup C)^c$, and the sum allows for any t_1 and t_2 satisfying $r < t_1 \leq t_2 < s$. The sum searches for times t_1 and t_2, which are the first and last times that the trajectory is in C. Because determining t_1 and t_2 requires a search over the entire trajectory $X_{r:s}$, we cannot factor $\omega(X_{r:s})$ as in (7.9).

Now suppose that, for each reactive trajectory segment $X_{r:s}$ in the infinite trajectory X, we have a process Y_t with

$$Y_t = \begin{cases} 0 & \text{if } t \leq r, \\ 1 & \text{if } r < t < t_1, \\ 2 & \text{if } t_1 \leq t \leq t_2, \\ 3 & \text{if } t_2 < t < s, \\ 4 & \text{if } s \leq t. \end{cases} \tag{7.19}$$

An example reactive trajectory labeled with Y_t is shown in Figure 7.1. We can apply TPT on the augmented process $Z_t = (X_t, Y_t)$ because we can write (7.18) in the form of (7.1) as

$$\omega(Z_{r:s}) = \mathbb{1}_{(A\times\{0\})\times D'\times\cdots\times D'\times(B\times\{4\})}(Z_{r:s}), \tag{7.20}$$

where $D' = (D \times \{1\}) \cup ((C \cup D) \times \{2\}) \cup (D \times \{3\})$.

This approach suggests a general strategy. We identify events—in this case, the first time t_1 and last time t_2 that X_t is in C—and define a process Y_t which labels these events. Then, we define reactive trajectories on the augmented state space using (7.1). So long as Z_t satisfies the assumptions behind TPT, we can express statistics using local quantities in the same manner as in (7.15). In Section 7.2.4, we discuss conditions for this to be the case, allowing for the possibility that an infinite trajectory has multiple labelings (e.g., to account for multiple finite reactive segments).

7.2.4 Augmented Transition Path Theory

In this section, we introduce a function $\Omega(Y \mid X)$ for constructing an ensemble of trajectories augmented with labels from the distribution of trajectories X. We first consider the case of infinite length trajectories $Z = (X, Y)$ and then the case of finite length trajectories $Z_{r:s} = (X_{r:s}, Y_{r:s})$, to which one is limited in practice. We discuss two conditions that must hold for our framework. First, $\Omega(Y_{r:s} \mid X_{r:s})$ must be consistent with $\Omega(Y \mid X)$. Second, Z_t must be Markovian. Later, in Section 7.4, we detail a specific construction of Y from X which requires examining only successive pairs of time points $X_{t:t+\epsilon}$ and $Y_{t:t+\epsilon}$.

We now present our augmented framework. We replace X_t with the augmented process $Z_t = (X_t, Y_t)$, where Y_t augments X_t with information about past and future events. Often, a single Y is associated with each infinite trajectory X because the latter contains full information about the past and future of any X_t. However, cases arise in which multiple Y can be associated with a given infinite trajectory X. For example, in the motivating reaction above, we define a Y for each reactive trajectory segment in X (i.e., we consider multiple r and s). It is thus necessary to consider a distribution of Y, and we compute integrals over the distribution of infinite trajectories Z as

$$\mathbb{I}^Z[f(Z)] = \mathbb{I}^X\Big[\int \Omega(Y \mid X) f(Z)\, \mathrm{d}Y\Big], \tag{7.21}$$

where $\Omega(Y \mid X)$ is the distribution of Y (and thus Z) for a given infinite trajectory X.

This immediately yields analogues of (7.15), (7.16), and (7.17):

$$\mathbb{I}_{\omega}^{Z}[\eta] = \mathbb{I}^{Z,t}[q_{-}(Z_t)q_{+}(Z_{t+\epsilon})\gamma(Z_{t:t+\epsilon})], \tag{7.22}$$

$$q_{-}(Z_t) = \mathbb{E}^{Z}[\mathbb{1}_{A}(Z_{S_{-}(t)}) \mid Z_t], \tag{7.23}$$

$$q_{+}(Z_t) = \mathbb{E}^{Z}[\mathbb{1}_{B}(Z_{S_{+}(t)}) \mid Z_t], \tag{7.24}$$

where sets A, B, and D are now defined on the augmented state space, and $S_{-}(t)$ and $S_{+}(t)$ are now on the augmented process.

However, we cannot yet evaluate these integrals and expectations because Y_t and thus each time point Z_t depends on the infinite trajectory X. Instead, we must convert integrals over Z to integrals over X using

$$\mathbb{I}^{Z}[f(Z_{r:s})] = \mathbb{I}^{X}\Big[\int \Omega(Y_{r:s} \mid X_{r:s}) f(Z_{r:s}) \,\mathrm{d}Y_{r:s}\Big], \tag{7.25}$$

where the weight of $Y_{r:s}$ (and thus $Z_{r:s}$) given the trajectory segment $X_{r:s}$ is

$$\Omega(Y_{r:s} \mid X_{r:s}) = \mathbb{E}^{X}\Big[\iint \Omega(Y \mid X) \,\mathrm{d}Y_{-\infty:r-\epsilon} \,\mathrm{d}Y_{s+\epsilon:\infty} \;\Big|\; X_{r:s}\Big]. \tag{7.26}$$

When a single Y is associated with each infinite trajectory X, $\Omega(Y_{r:s} \mid X_{r:s})$ is the probability of $Y_{r:s}$ given $X_{r:s}$ and so $\int \Omega(Y_{r:s} \mid X_{r:s}) \,\mathrm{d}Y_{r:s} = 1$; this is not true in the general case. Equation (7.26) is the first requirement of our augmented framework: we must be able to convert integrals involving $\Omega(Y \mid X)$, which depend on the infinite trajectory X, to those involving $\Omega(Y_{r:s} \mid X_{r:s})$, which depend only on the finite trajectory $X_{r:s}$.

Using (7.25), we can then write (7.22), (7.23), and (7.24) as

$$\mathbb{I}^Z_\omega[\eta] = \mathbb{I}^{X,t}\Big[\int \Omega(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) q_-(Z_t) q_+(Z_{t+\epsilon}) \gamma(Z_{t:t+\epsilon}) \,\mathrm{d}Y_{t:t+\epsilon}\Big], \tag{7.27}$$

$$q_-(Z_t) = \frac{\mathbb{E}^X[\int \Omega(Y_{-\infty:t} \,|\, X_{-\infty:t}) \mathbb{1}_A(Z_{S_-(t)}) \,\mathrm{d}Y_{-\infty:t-\epsilon} \mid X_t]}{\mathbb{E}^X[\int \Omega(Y_{-\infty:t} \,|\, X_{-\infty:t}) \,\mathrm{d}Y_{-\infty:t-\epsilon} \mid X_t]} \tag{7.28}$$

$$= \frac{\mathbb{E}^X[\int \Omega(Y_{-\infty:t} \,|\, X_{-\infty:t}) \mathbb{1}_A(Z_{S_-(t)}) \,\mathrm{d}Y_{-\infty:t-\epsilon} \mid X_t]}{\Omega(Y_t \,|\, X_t)}, \tag{7.29}$$

$$q_+(Z_t) = \frac{\mathbb{E}^X[\int \Omega(Y_{t:\infty} \,|\, X_{t:\infty}) \mathbb{1}_B(Z_{S_+(t)}) \,\mathrm{d}Y_{t+\epsilon:\infty} \mid X_t]}{\mathbb{E}^X[\int \Omega(Y_{t:\infty} \,|\, X_{t:\infty}) \,\mathrm{d}Y_{t+\epsilon:\infty} \mid X_t]} \tag{7.30}$$

$$= \frac{\mathbb{E}^X[\int \Omega(Y_{t:\infty} \,|\, X_{t:\infty}) \mathbb{1}_B(Z_{S_+(t)}) \,\mathrm{d}Y_{t+\epsilon:\infty} \mid X_t]}{\Omega(Y_t \,|\, X_t)}. \tag{7.31}$$

For $q_-(Z_t)$ and $q_+(Z_t)$, we excluded Y_t from the variables over which we integrate because we conditioned on it. We note that when $\omega(Z_{r:s})$ has no dependence on $Y_{r:s}$ [i.e., $\omega(Z_{r:s}) = \omega(X_{r:s})$] and there is a one-to-one correspondence between X and Z [i.e., $\int \Omega(Y \,|\, X) \,\mathrm{d}Y = 1$], we can recover the traditional TPT committors as

$$q_-(X_t) = \int \Omega(Y_t \,|\, X_t) q_-(Z_t) \,\mathrm{d}Y_t, \tag{7.32}$$

$$q_+(X_t) = \int \Omega(Y_t \,|\, X_t) q_+(Z_t) \,\mathrm{d}Y_t. \tag{7.33}$$

In traditional TPT, X_t must be a Markov process so that, from (7.14) to (7.15), we could take expectations of $\mathbb{1}_A(X_{S_-(t)})$ and $\mathbb{1}_B(X_{S_+(t+\epsilon)})$ to obtain committors $q_-(X_t)$ and $q_+(X_{t+\epsilon})$. For the augmented process Z_t to be similarly treatable, we also require it to be a Markov process. This requirement may be surprising because Y_t can depend on the future of X_t. This can be understood by observing that, for the augmented process, the probability distribution of $X_{t+\epsilon}$ depends on both X_t and Y_t. For example, for $q_+(Z_t)$ in (7.24), the distribution of $X_{t+\epsilon:\infty}$ conditioned on Z_t is not the same as that of $X_{t+\epsilon:\infty}$ conditioned on X_t alone, since Y_t specifies that X_t must undergo certain events in the future.

Since X_t and Z_t are Markov processes, we can factor the path probabilities $\mathbb{P}^X[X_{r:s}]$ and $\mathbb{P}^Z[Z_{r:s}]$ of the original and augmented processes:

$$\mathbb{P}^X[X_{r:s}] = \mathbb{P}^X[X_r] \prod_{t=r:s-\epsilon} \mathbb{P}^X[X_{t+\epsilon} \mid X_t] \tag{7.34}$$

$$= \mathbb{P}^X[X_r] \prod_{t=r:s-\epsilon} \frac{\mathbb{P}^X[X_{t:t+\epsilon}]}{\mathbb{P}^X[X_t]}, \tag{7.35}$$

$$\mathbb{P}^Z[Z_{r:s}] = \mathbb{P}^Z[Z_r] \prod_{t=r:s-\epsilon} \mathbb{P}^Z[Z_{t+\epsilon} \mid Z_t] \tag{7.36}$$

$$= \mathbb{P}^Z[Z_r] \prod_{t=r:s-\epsilon} \frac{\mathbb{P}^Z[Z_{t:t+\epsilon}]}{\mathbb{P}^Z[Z_t]}. \tag{7.37}$$

As above, the superscripts indicate distributions of infinite trajectories. Thus, for example,

$$\mathbb{P}^X[X_{r:s}] = \iint \mathbb{P}^X[X] \, \mathrm{d}X_{-\infty:r-\epsilon} \, \mathrm{d}X_{s+\epsilon:\infty} \tag{7.38}$$

is the probability of observing $X_{r:s}$ with all possible semi-infinite segments $X_{-\infty:r-\epsilon}$ and $X_{s+\epsilon:\infty}$ before and after $r:s$, respectively; $\mathbb{P}^X[X]$ is the probability of a specific infinite trajectory X. The probability distribution of Z is

$$\mathbb{P}^Z[Z] = \Omega(Y \mid X) \, \mathbb{P}^X[X]/c \tag{7.39}$$

with $c = \int \Omega(Y \mid X) \, \mathbb{P}^X[X] \, \mathrm{d}Z$. Therefore, from (7.26),

$$\Omega(Y_{r:s} \mid X_{r:s}) = \frac{\iint \Omega(Y \mid X) \, \mathbb{P}^X[X] \, \mathrm{d}Z_{-\infty:r-\epsilon} \, \mathrm{d}Z_{s+\epsilon:\infty}}{\mathbb{P}^X[X_{r:s}]} \tag{7.40}$$

$$= \frac{\iint c \, \mathbb{P}^Z[Z] \, \mathrm{d}Z_{-\infty:r-\epsilon} \, \mathrm{d}Z_{s+\epsilon:\infty}}{\mathbb{P}^X[X_{r:s}]} \tag{7.41}$$

$$= c \frac{\mathbb{P}^Z[Z_{r:s}]}{\mathbb{P}^X[X_{r:s}]}. \tag{7.42}$$

To compute $\Omega(Y_{r:s} \,|\, X_{r:s})$, we divide (7.37) by (7.35) and then apply (7.42):

$$\Omega(Y_{r:s} \,|\, X_{r:s}) = \Omega(Y_r \,|\, X_r) \prod_{t=r:s-\epsilon} \frac{\Omega(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})}{\Omega(Y_t \,|\, X_t)}. \tag{7.43}$$

This factorization is the second requirement of our augmented framework: we must be able to construct $\Omega(Y \,|\, X)$, which depends on the infinite trajectory X, from $\Omega(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$, which can only depend on pairs of successive time points $X_{t:t+\epsilon}$.

7.3 Reactive Statistics

In this section, we discuss TPT statistics that provide information about mechanisms. These include committors, the reactive flux, the reactive density, the reactive current, and expectations over reactive trajectories that they enable computing. We present expressions for augmented TPT in the form of (7.22), which can be evaluated using (7.27). The corresponding expressions for traditional TPT can be obtained by replacing Z_t with X_t. The statistics are normalized so that different reactions that are specified through different $\omega(Z_{r:s})$ but calculated from the same distribution of infinite trajectories X are directly comparable.

We note that augmented TPT is useful even for reactions which can be described using traditional TPT [i.e., $\omega(Z_{r:s}) = \omega(X_{r:s})$]. The augmented process allows reaction mechanisms to be resolved in more detail, since committors and other statistics can be calculated on points which depend on both past and future behaviors of trajectories. Furthermore, the addition of past and future information enables the calculation of statistics with $\eta(X_{r:s})$ no longer restricted to the form in (7.5).

Several of the statistics that we discuss yield quantities on points v in a collective variable (CV) space θ, that is, a space of functions of a subset of the coordinates. We indicate this using the subscript θ. We express these statistics on a CV space rather than the state space of Z_t because, for complex systems, it is often the case that the full state space contains

variables that are irrelevant to understanding the reaction. This is particularly true for the augmented state space, which must contain the information required to select reactive trajectories using (7.1) and compute statistics using (7.5), both of which rely on Y_t to obtain past or future information. Nevertheless, the theory holds for the choice $\theta(Z_t) = Z_t$.

7.3.1 Reactive Flux

The reactive flux $R = \mathbb{I}_\omega^Z[1]$ is the expected number of reactive trajectories which start (or end) per unit time. We can express the reactive flux in the form of (7.22) by choosing $\gamma(Z_{t:t+\epsilon})$ so that $\eta(Z_{r:s}) = 1$ when $Z_{r:s}$ is reactive. Such choices of $\gamma(Z_{t:t+\epsilon})$ include $\mathbb{1}_A(Z_t)/\epsilon$ and $\mathbb{1}_B(Z_{t+\epsilon})/\epsilon$, which are nonzero only when $Z_{t:t+\epsilon}$ is the first or last step of the reactive trajectory, respectively. Consequently, we can compute the reactive flux using

$$R = \mathbb{I}^{Z,t}[\mathbb{1}_A(Z_t)q_+(Z_{t+\epsilon})/\epsilon] \tag{7.44}$$

$$= \mathbb{I}^{Z,t}[q_-(Z_t)\mathbb{1}_B(Z_{t+\epsilon})/\epsilon], \tag{7.45}$$

where we have applied the identities $\mathbb{1}_A(Z_t)q_-(Z_t) = \mathbb{1}_A(Z_t)$ and $\mathbb{1}_B(Z_t)q_+(Z_t) = \mathbb{1}_B(Z_t)$. Equation (7.44) counts the number of trajectories that exit A in the time interval ϵ and then react; (7.45) is the analogue for trajectories entering B.

The reactive flux is of interest not only in its own right but also for calculating expectations over reactive trajectories:

$$\mathbb{E}_\omega^Z[\eta] = \mathbb{I}_\omega^Z[\eta]/\,\mathbb{I}_\omega^Z[1] = \mathbb{I}_\omega^Z[\eta]/R. \tag{7.46}$$

For example, the duration $N(Z_{r:s}) = s - r$ of a trajectory can be expressed in the form of (7.5) with $\gamma(Z_{t:t+\epsilon}) = 1$, and so the expected duration of a reactive trajectory is

$$\mathbb{E}_\omega^Z[N] = \mathbb{I}^{Z,t}[q_-(Z_t)q_+(Z_{t+\epsilon})]/R. \tag{7.47}$$

7.3.2 Reactive Density

The reactive density is the distribution of configurations which belong to reactive trajectories. For a point v in the CV space θ, the reactive density $\rho_\theta(v)$ is the probability that $\theta(Z_t) = v$ and Z_t is part of a reactive trajectory. Equivalently, it is the expected fraction of time an infinite trajectory spends reactive at v. It can be expressed in the form of (7.22) as

$$\rho_\theta(v) = \mathbb{I}^{Z,t}\left[q_-(Z_t)q_+(Z_{t+\epsilon})\frac{\delta_v(\theta(Z_t)) + \delta_v(\theta(Z_{t+\epsilon}))}{2}\right], \tag{7.48}$$

where δ is the Dirac delta function. When computing the expectation, $\delta_v(\theta(Z_t))$ selects the points Z_t with $\theta(Z_t) = v$. The term $(\delta_v(\theta(Z_t)) + \delta_v(\theta(Z_{t+\epsilon})))/2$ corresponds to assuming that half of the time of each step $Z_{t:t+\epsilon}$ is spent in Z_t and half of the time is spent in $Z_{t+\epsilon}$.

In turn, the reactive density can be used to evaluate (7.22) when $\gamma(Z_{t:t+\epsilon}) = (f(\theta(Z_t)) + f(\theta(Z_{t+\epsilon})))/2$ is a path-independent function on the CV space:

$$\mathbb{I}^Z_\omega[\eta] = \int \rho_\theta(v) f(v)\,\mathrm{d}v. \tag{7.49}$$

For instance, the expected fraction of time an infinite trajectory spends reactive can be obtained by setting $f(v) = 1$, so that

$$\mathbb{I}^Z_\omega[N] = \int \rho_\theta(v)\,\mathrm{d}v = \mathbb{I}^{Z,t}[q_-(Z_t)q_+(Z_{t+\epsilon})], \tag{7.50}$$

where we have assumed the distribution of trajectories X to be a probability distribution, so that $\mathbb{I}^X[1] = 1$.

We note that when the CV space θ is contained in the CV space θ', i.e., we can write $\theta(Z_t) = \zeta(\theta'(Z_t))$ for some $\zeta(v')$, we can calculate $\rho_\theta(v)$ by projecting $\rho_{\theta'}(v')$ onto θ:

$$\rho_\theta(v) = \int \delta_v(\zeta(v'))\rho_{\theta'}(v')\,\mathrm{d}v'. \tag{7.51}$$

We can do the same for functions $f(v')$ defined on the CV space θ'. We calculate $A_\theta[f](v)$, the expected value of $f(v')$ at a point v in the CV space θ, conditioned on trajectories passing through that point being reactive, as

$$A_\theta[f](v) = \frac{\int \delta_v(\zeta(v'))\rho_{\theta'}(v')f(v')\,\mathrm{d}v'}{\rho_\theta(v)}. \tag{7.52}$$

We emphasize that $f(v')$ can use Y_t to obtain information from the past and future of X_t, and so (7.52) is significantly more powerful than its traditional TPT counterpart. For instance, we can calculate the conditional mean first passage time, the expected time it takes for Z_t to hit B given that Z_t is part of a reactive trajectory, using (7.52) as discussed further in Section 7.3.5, whereas in traditional TPT we would need to employ a Feynman–Kac formula (e.g., see Finkel et al. [189]).

7.3.3 Reactive Current

The reactive current $J_\theta(v)$ through a point v in the CV space θ is the net flow of reactive trajectories within θ through v. It can be expressed in the form of (7.22) as

$$J_\theta(v) = \mathbb{I}^{Z,t}\left[q_-(Z_t)q_+(Z_{t+\epsilon})\frac{\delta_v(\theta(Z_t)) + \delta_v(\theta(Z_{t+\epsilon}))}{2}\frac{\theta(Z_{t+\epsilon}) - \theta(Z_t)}{\epsilon}\right]. \tag{7.53}$$

Conceptually, for each pair of successive time points $Z_{t:t+\epsilon}$ that is part of a reactive trajectory, we compute the numerical derivative $(\theta(Z_{t+\epsilon}) - \theta(Z_t))/\epsilon$ and then split it equally between Z_t and $Z_{t+\epsilon}$. In fact, in the limit $\epsilon \to 0$, when $\theta(Z_t)$ is differentiable, (7.53) becomes

$$J_\theta(v) = \mathbb{I}^{Z,t}\left[\mathbb{1}_A(Z_{S_-(t)})\mathbb{1}_B(Z_{S_+(t)})\delta_v(\theta(Z_t))\frac{\mathrm{d}\theta(Z_t)}{\mathrm{d}t}\right], \tag{7.54}$$

which is the time derivative of $\theta(Z_t)$ integrated over the distribution of reactive trajectories Z_t with $\theta(Z_t) = v$.

It can be useful to compute the reactive current along the gradient of a function $f(v)$,

$$J_\theta[f](v) = \mathbb{I}^{Z,t}\left[q_-(Z_t)q_+(Z_{t+\epsilon})\frac{\delta_v(\theta(Z_t)) + \delta_v(\theta(Z_{t+\epsilon}))}{2}\frac{f(\theta(Z_{t+\epsilon})) - f(\theta(Z_t))}{\epsilon}\right]. \quad (7.55)$$

In the limit $\epsilon \to 0$, for differentiable $f(v)$, (7.55) is $J_\theta[f](v) = J_\theta(v) \cdot \nabla_\theta f(v)$ which we can derive by observing that the finite differences in (7.53) and (7.55) are $\mathrm{d}\theta(Z_t)/\mathrm{d}t$ and $\mathrm{d}f(\theta(Z_t))/\mathrm{d}t$ in this limit, respectively, and by the chain rule $\mathrm{d}f(\theta(Z_t))/\mathrm{d}t = \mathrm{d}\theta(Z_t)/\mathrm{d}t \cdot \nabla_\theta f(\theta(Z_t))$.

Like the reactive density, we can calculate $J_\theta(v)$ and $J_\theta[f](v)$ by projecting $J_{\theta'}(v')$ and $J_{\theta'}[f \circ \zeta](v')$ onto θ:

$$J_\theta(v) = \int \delta_v(\zeta(v'))J_{\theta'}[\zeta](v')\,\mathrm{d}v', \quad (7.56)$$

$$J_\theta[f](v) = \int \delta_v(\zeta(v'))J_{\theta'}[f \circ \zeta](v')\,\mathrm{d}v', \quad (7.57)$$

where $(f \circ \zeta)(v') = f(\zeta(v'))$.

7.3.4 Committors

The committors $q_-(Z_t)$ and $q_+(Z_t)$ are defined on the state space of Z_t, which makes them useful for calculating other statistics but can make them hard to interpret. To address this issue, we can treat the committors as reaction coordinates, and project them onto a CV space θ as $A_\theta[q_-](v)$ and $A_\theta[q_+](v)$. These quantities have a physical interpretation. For instance, $A_\theta[q_+](v)$ is the probability that a trajectory starting at a point Z_t that is drawn from configurations with $\theta(Z_t) = v$ in the ensemble of reactive trajectories, enters B when it first leaves D. We note that, unlike most other reactive statistics, $A_\theta[q_-](v)$ and $A_\theta[q_+](v)$ with $\theta(Z_t) = X_t$ are not independent of the choice of Y_t, even when the same ensemble of reactive trajectories is selected because the likelihood that a trajectory contributes positively to the committor and the likelihood that it is reactive are correlated.

7.3.5 Conditional Mean First and Last Passage Times

The first passage time to the product is the time it takes for a trajectory starting at time t to reach the product B, at time $S_+(t)$. It can be expressed as

$$f(Z_t') = Y_t' = \begin{cases} Y_{t+\epsilon}' + \epsilon & \text{if } Z_t \notin B, \\ 0 & \text{otherwise,} \end{cases} \tag{7.58}$$

where $Z_t' = (Z_t, Y_t')$. This increments Y_t' by ϵ for each time step backward in time when $Z_t \notin B$, and sets $Y_t' = 0$ when $Z_t \in B$. The conditional mean first passage time, $m_+(Z_t)$, is the expected first passage time to the product for a point Z_t that is part of a reactive trajectory. This statistic, and higher moments of the first passage time distribution, are useful for real-time forecasting, e.g., of weather [189]. To compute the conditional mean first passage time, we take the conditional expectation of (7.58) with respect to the reactive density using (7.52), i.e.,

$$m_+(Z_t) = A_{\theta'}[f](Z_t), \tag{7.59}$$

where $\theta'(Z_t') = Z_t$. We can also calculate more general statistics on the distribution of the first passage time. For instance, the conditional variance of the first passage time to the product is $A_{\theta'}[f^2](Z_t) - (A_{\theta'}[f](Z_t))^2$. Likewise, the last passage time from the reactant is the time it takes for a trajectory ending at time t to come from the reactant A, at time $S_-(t)$, conditioned on Z_t being part of a reactive trajectory, and can be expressed as

$$g(Z_t'') = Y_t'' = \begin{cases} Y_{t-\epsilon}'' + \epsilon & \text{if } Z_t \notin A, \\ 0 & \text{otherwise,} \end{cases} \tag{7.60}$$

where $Z''_t = (Z_t, Y''_t)$. The conditional mean last passage time from the reactant is then

$$m_-(Z_t) = A_{\theta''}[g](Z_t), \tag{7.61}$$

where $\theta''(Z''_t) = Z_t$. These statistics can be projected onto points v on a CV space $\theta(Z_t)$ as $A_\theta[m_-](v)$ and $A_\theta[m_+](v)$.

7.4 Construction of the Augmented Process

In this section, we describe a particularly useful way to define the augmented process. Namely, we decompose $\Omega(Y \,|\, X)$ into a product over functions of successive time points, $\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$. We show that $\Omega(Y_{r:s} \,|\, X_{r:s})$ so defined satisfies the required properties (7.26) and (7.43) by construction. We then demonstrate the construction of complex augmented processes from simpler ones by composition. Finally, we show how this machinery applies to the motivating reaction.

7.4.1 Decomposition of Ω

To apply our augmented framework, we need to construct the augmented process and in turn the distribution of infinite trajectories Z such that $\Omega(Y_{r:s} \,|\, X_{r:s})$ satisfies (7.26) and (7.43). One way to do so is to define $\Omega(Y \,|\, X)$, calculate $\Omega(Y_{r:s} \,|\, X_{r:s})$ from $\Omega(Y \,|\, X)$ using (7.26), and then verify that (7.43) holds. In this section, we present an alternative approach. We specify the augmented process through a function of pairs of successive time points, $\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$, and then use it to calculate $\Omega(Y_{r:s} \,|\, X_{r:s})$. This procedure satisfies (7.26) and (7.43) by construction.

We start by using the pair structure of (7.43) to factor

$$\Omega(Y \,|\, X) = \prod_{t=-\infty:\infty} \kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}). \tag{7.62}$$

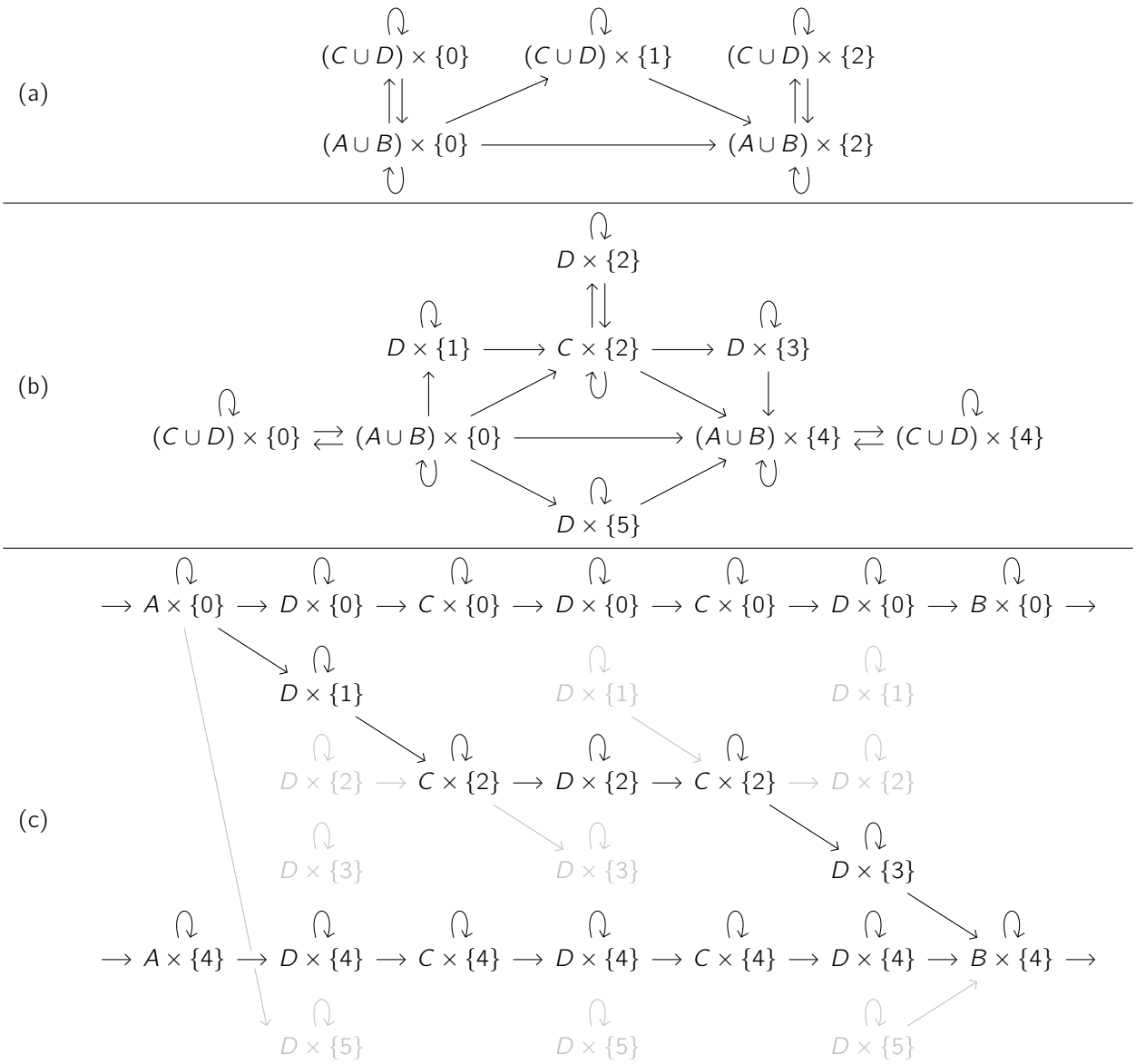

Figure 7.2: Construction of the augmented process for a reaction with an intermediate. (a) Possible transitions of the augmented process defined by (7.74). (b) Possible transitions of the augmented process defined by (7.79). Each arrow from one set to another indicates that Y_t may transition to $Y_{t+\epsilon}$ when Z_t is in the first set and $Z_{t+\epsilon}$ is in the second set. (c) Determination of $Y_{r:s}$ for the reactive trajectory $X_{r:s}$ from Figure 7.1. The black elements indicate trajectories $Y_{r:s}$ that satisfy $\kappa(Y_{r:s} \,|\, X_{r:s}) = 1$. The gray elements indicate pairs $Y_{t:t+\epsilon}$ that satisfy $\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = 1$ but do not belong to any $Y_{r:s}$ with $\kappa(Y_{r:s} \,|\, X_{r:s})$.

It follows immediately that

$$\Omega(Y \,|\, X) = \kappa(Y_{-\infty:r} \,|\, X_{-\infty:r})\kappa(Y_{r:s} \,|\, X_{r:s})\kappa(Y_{s:\infty} \,|\, X_{s:\infty}), \tag{7.63}$$

where

$$\kappa(Y_{r:s} \,|\, X_{r:s}) = \prod_{t=r:s-\epsilon} \kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}). \tag{7.64}$$

This cleanly separates terms which depend on the past $X_{-\infty:r}$ and future $X_{s:\infty}$ from those which depend on the trajectory segment $X_{r:s}$.

Using (7.63), we can compute (7.26) as

$$\Omega(Y_{r:s} \,|\, X_{r:s}) = k_-(Y_r \,|\, X_r)\kappa(Y_{r:s} \,|\, X_{r:s})k_+(Y_s \,|\, X_s), \tag{7.65}$$

where we have defined

$$k_-(Y_t \,|\, X_t) = \mathbb{E}^X\Big[\int \kappa(Y_{-\infty:t} \,|\, X_{-\infty:t})\, \mathrm{d}Y_{-\infty:t-\epsilon} \;\Big|\; X_t\Big], \tag{7.66}$$

$$k_+(Y_t \,|\, X_t) = \mathbb{E}^X\Big[\int \kappa(Y_{t:\infty} \,|\, X_{t:\infty})\, \mathrm{d}Y_{t+\epsilon:\infty} \;\Big|\; X_t\Big]. \tag{7.67}$$

We describe how to calculate $k_-(Y_t \,|\, X_t)$ and $k_+(Y_t \,|\, X_t)$ in Section 7.5. The case $r = s = t$ is the weight of Y_t given X_t:

$$\Omega(Y_t \,|\, X_t) = k_-(Y_t \,|\, X_t)k_+(Y_t \,|\, X_t). \tag{7.68}$$

We can verify that the resulting Z_t is a Markov process by substituting (7.65) and (7.68) into (7.43).

7.4.2 Building Augmented Processes by Composition

The factorization in (7.62) has a number of advantages over (7.43). One is that it facilitates deriving useful expressions for treating multiple augmented processes. If we have $\Omega(Y \,|\, X)$ of the form

$$\Omega(Y \,|\, X) = \prod_n \Omega(Y^{(n)} \,|\, X^{(n)}), \tag{7.69}$$

where n labels different augmented processes, we can factor both sides by (7.62) to obtain

$$\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = \prod_n \kappa(Y^{(n)}_{t:t+\epsilon} \,|\, X^{(n)}_{t:t+\epsilon}), \tag{7.70}$$

which involves only successive time points. Each of the terms $\kappa(Y^{(n)}_{t:t+\epsilon} \,|\, X^{(n)}_{t:t+\epsilon})$ defines a process $Y^{(n)}_t$ using information from the original process X_t and other processes $Y^{(m)}_t$, which we denote as $X^{(n)}_t$. We can use this to combine multiple augmented processes which may be defined independently or hierarchically.

For example, consider the augmented process $Z^\omega_t = (X_t, Y^\omega_t)$, where Y^ω_t is used to define pathways and is defined using $\kappa(Y^\omega_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$. To compute statistics on the first and last passage times, we can augment this process with the augmented processes in (7.58) and (7.60) by defining

$$\kappa(Y'_{t:t+\epsilon} \,|\, Z^\omega_{t:t+\epsilon}) = \begin{cases} \delta_{Y'_t}(Y'_{t+\epsilon} + \epsilon) & \text{if } Z^\omega_t \notin B, \\ \delta_{Y'_t}(0) & \text{otherwise,} \end{cases} \tag{7.71}$$

$$\kappa(Y''_{t:t+\epsilon} \,|\, Z^\omega_{t:t+\epsilon}) = \begin{cases} \delta_{Y''_{t+\epsilon}}(Y''_t + \epsilon) & \text{if } Z^\omega_{t+\epsilon} \notin B, \\ \delta_{Y''_{t+\epsilon}}(0) & \text{otherwise.} \end{cases} \tag{7.72}$$

The combined process $Y_t = (Y^\omega_t, Y'_t, Y''_t)$ is then specified by

$$\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = \kappa(Y^\omega_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) \kappa(Y'_{t:t+\epsilon} \,|\, Z^\omega_{t:t+\epsilon}) \kappa(Y''_{t:t+\epsilon} \,|\, Z^\omega_{t:t+\epsilon}). \tag{7.73}$$

This example furthermore shows how (7.62) allows forward-in-time and backward-in-time augmented processes to be treated in a unified manner and combined, which is not straightforward with (7.43).

7.4.3 Augmented Process for the Motivating Reaction

We can also use (7.70) to construct augmented processes by combining simpler augmented processes. Here, we detail a possible construction of the augmented process (7.19) as a composite of three augmented processes. First, we define an augmented process $Z_t^{(0)} = (X_t, Y_t^{(0)})$ that selects all reactive trajectories regardless of pathway:

$$\begin{aligned}\kappa(Y_{t:t+\epsilon}^{(0)} \,|\, X_{t:t+\epsilon}) = [&\mathbb{1}_{\{0\}\times\{0\}}(Y_{t:t+\epsilon}^{(0)}) \\ &+ \mathbb{1}_{((C\cup D)\times\{1\})\times((C\cup D)\times\{1\})}(Z_{t:t+\epsilon}^{(0)}) \\ &+ \mathbb{1}_{\{2\}\times\{2\}}(Y_{t:t+\epsilon}^{(0)}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times((C\cup D)\times\{1\})}(Z_{t:t+\epsilon}^{(0)}) \\ &+ \mathbb{1}_{((C\cup D)\times\{1\})\times((A\cup B)\times\{2\})}(Z_{t:t+\epsilon}^{(0)}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times((A\cup B)\times\{2\})}(Z_{t:t+\epsilon}^{(0)})].\end{aligned} \tag{7.74}$$

For a reactive trajectory $X_{r:s}$ from A to B, $Y_t^{(0)}$ splits the infinite trajectory X into three parts: time points $X_{-\infty:r}$ before the reaction ($Y_t^{(0)} = 0$), time points $X_{r+\epsilon:s-\epsilon}$ during the reaction ($Y_t^{(0)} = 1$), and time points $X_{s:\infty}$ after the reaction ($Y_t^{(0)} = 2$). We list the possible transitions of this augmented process in Figure 7.2(a). The nodes are sets in which $Z_t^{(0)}$ may belong; an arrow from one set to another indicates that $Y_t^{(0)}$ may transition to $Y_{t+\epsilon}^{(0)}$ when $Z_t^{(0)}$ is in the first set and $Z_{t+\epsilon}^{(0)}$ is in the second set. For instance, the fourth term in (7.74) corresponds to the arrow from $(A \cup B) \times \{0\}$ to $(C \cup D) \times \{1\}$.

Next, we employ additional augmented processes to find the first and last times t_1 and t_2

that the trajectory is in C. Using (7.74), we define the processes

$$Y_t^{(1)} = \begin{cases} Y_{t-\epsilon}^{(1)} & \text{if } X_t \in D \text{ and } Y_t^{(0)} = 1, \\ 1 & \text{if } X_t \in C \text{ and } Y_t^{(0)} = 1, \\ 0 & \text{if } Y_t^{(0)} \in \{0, 2\}, \end{cases} \tag{7.75}$$

$$Y_t^{(2)} = \begin{cases} Y_{t+\epsilon}^{(2)} & \text{if } X_t \in D \text{ and } Y_t^{(0)} = 1, \\ 1 & \text{if } X_t \in C \text{ and } Y_t^{(0)} = 1, \\ 0 & \text{if } Y_t^{(0)} \in \{0, 2\}. \end{cases} \tag{7.76}$$

During the reaction (i.e., $Y_t^{(0)} = 1$), $Y_t^{(1)} = 1$ for times $t \geq t_1$ and $Y_t^{(2)} = 1$ for times $t \leq t_2$. We can write (7.75) and (7.76) as

$$\begin{aligned} \kappa(Y_{t:t+\epsilon}^{(1)} \mid Z_{t:t+\epsilon}^{(0)}) = [&\mathbb{1}_{D\times\{1\}\times\{(0,0),(1,1)\}}(X_{t+\epsilon}, Y_{t+\epsilon}^{(0)}, Y_{t:t+\epsilon}^{(1)}) \\ &+ \mathbb{1}_{C\times\{1\}\times\{1\}}(X_{t+\epsilon}, Y_{t+\epsilon}^{(0)}, Y_{t+\epsilon}^{(1)}) \\ &+ \mathbb{1}_{\{0,2\}\times\{0\}}(Y_{t+\epsilon}^{(0)}, Y_{t+\epsilon}^{(1)})], \end{aligned} \tag{7.77}$$

$$\begin{aligned} \kappa(Y_{t:t+\epsilon}^{(2)} \mid Z_{t:t+\epsilon}^{(0)}) = [&\mathbb{1}_{D\times\{1\}\times\{(0,0),(1,1)\}}(X_t, Y_t^{(0)}, Y_{t:t+\epsilon}^{(2)}) \\ &+ \mathbb{1}_{C\times\{1\}\times\{1\}}(X_t, Y_t^{(0)}, Y_t^{(2)}) \\ &+ \mathbb{1}_{\{0,2\}\times\{0\}}(Y_t^{(0)}, Y_t^{(2)})]. \end{aligned} \tag{7.78}$$

We can then combine (7.74), (7.77), and (7.78) using (7.70):

$$\begin{aligned}\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = [&\mathbb{1}_{\{0\}\times\{0\}}(Y_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{1\})\times(D\times\{1\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((C\cup D)\times\{2\})\times((C\cup D)\times\{2\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{3\})\times(D\times\{3\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{\{4\}\times\{4\}}(Y_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{5\})\times(D\times\{5\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times(D\times\{1\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times(C\times\{2\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{1\})\times(C\times\{2\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(C\times\{2\})\times(D\times\{3\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(C\times\{2\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{3\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times(D\times\{5\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{5\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon})],\end{aligned} \tag{7.79}$$

where $Z_t = (X_t, Y_t)$, and to match (7.19), we have merged $Y_t^{(0)}$, $Y_t^{(1)}$, and $Y_t^{(2)}$ into

$$Y_t = \begin{cases} 0 & \text{if } (Y_t^{(0)}, Y_t^{(1)}, Y_t^{(2)}) = (0,0,0), \\ 1 & \text{if } (Y_t^{(0)}, Y_t^{(1)}, Y_t^{(2)}) = (1,0,1), \\ 2 & \text{if } (Y_t^{(0)}, Y_t^{(1)}, Y_t^{(2)}) = (1,1,1), \\ 3 & \text{if } (Y_t^{(0)}, Y_t^{(1)}, Y_t^{(2)}) = (1,1,0), \\ 4 & \text{if } (Y_t^{(0)}, Y_t^{(1)}, Y_t^{(2)}) = (2,0,0), \\ 5 & \text{if } (Y_t^{(0)}, Y_t^{(1)}, Y_t^{(2)}) = (1,0,0). \end{cases} \tag{7.80}$$

We show the possible transitions of this augmented process in Figure 7.2(b).

In Figure 7.2(c), we illustrate the determination of $Y_{r:s}$ from $\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$ for the trajectory $X_{r:s}$ in Figure 7.1. For each time point X_t, we list the possible values of Y_t from (7.79). We then stitch together trajectories by connecting successive time-point pairs $Z_{t:t+\epsilon}$ and following the arrows in Figure 7.2(b). The black elements in Figure 7.2(c) indicate step pairs $Y_{t:t+\epsilon}$ in trajectories $Y_{r:s}$ that satisfy $\kappa(Y_{r:s} \,|\, X_{r:s}) = 1$, and the gray elements indicate step pairs $Y_{t:t+\epsilon}$ that satisfy $\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = 1$ but do not belong to any $Y_{r:s}$ that satisfies $\kappa(Y_{r:s} \,|\, X_{r:s}) = 1$.

There is one Y associated with each reactive trajectory segment in X. The diagonal path in Figure 7.2(c) represents one such reactive trajectory segment and can be selected using augmented TPT. The horizontal paths are collections of augmented processes corresponding to times before each future reaction ($Y_t = 0$) and after each past reaction ($Y_t = 4$).

7.5 Algorithm

In this section, we summarize the operational aspects of the method. For the numerical examples that we consider in the present chapter, we evaluate integrals of the form in (7.3)

using a finite difference approximation, which we detail in Appendix 7.8.1; more complex systems can be treated by extending the approach in Thiede et al. [4] and Strahan et al. [5], which we leave for future work. In terms of the finite difference approximation, the algorithm for evaluating these statistics is as follows:

1. Define $\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$, $\omega(Z_{r:s})$, and $\gamma(Z_{t:t+\epsilon})$ for the statistic of interest.

2. Compute $k_-(Y_t \,|\, X_t)$ and $k_+(Y_t \,|\, X_t)$, which account for weights associated with the past and future segments of trajectories. To this end, we express (7.66) and (7.67) as

$$k_-(Y_t \,|\, X_t) = \mathbb{E}^X\Big[\int \kappa(Y_{t-\epsilon:t} \,|\, X_{t-\epsilon:t}) k_-(Y_{t-\epsilon} \,|\, X_{t-\epsilon}) \,\mathrm{d}Y_{t-\epsilon} \;\Big|\; X_t\Big], \tag{7.81}$$

$$k_+(Y_t \,|\, X_t) = \mathbb{E}^X\Big[\int \kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) k_+(Y_{t+\epsilon} \,|\, X_{t+\epsilon}) \,\mathrm{d}Y_{t+\epsilon} \;\Big|\; X_t\Big], \tag{7.82}$$

   and solve these equations using (7.108) and (7.109).

3. Compute $\Omega(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon})$ and $\Omega(Y_t \,|\, X_t)$ by (7.65) and (7.68), respectively.

4. Compute $q_-(Z_t)$ and $q_+(Z_t)$. To this end, we express the committors as solutions to boundary value problems. For $Z_t \in D$,

$$q_-(Z_t) = \mathbb{E}^Z[q_-(Z_{t-\epsilon}) \mid Z_t] \tag{7.83}$$

$$= \mathbb{E}^X\Big[\int \Omega(Y_{t-\epsilon:t} \,|\, X_{t-\epsilon:t}) q_-(Z_{t-\epsilon}) \,\mathrm{d}Y_{t-\epsilon} \;\Big|\; X_t\Big] \Big/ \Omega(Y_t \,|\, X_t), \tag{7.84}$$

$$q_+(Z_t) = \mathbb{E}^Z[q_+(Z_{t+\epsilon}) \mid Z_t] \tag{7.85}$$

$$= \mathbb{E}^X\Big[\int \Omega(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) q_+(Z_{t+\epsilon}) \,\mathrm{d}Y_{t+\epsilon} \;\Big|\; X_t\Big] \Big/ \Omega(Y_t \,|\, X_t). \tag{7.86}$$

   For $Z_t \notin D$, $q_-(Z_t) = \mathbb{1}_A(Z_t)$ and $q_+(Z_t) = \mathbb{1}_B(Z_t)$. Above, (7.83) and (7.85) result from applying the identities (7.12) and (7.13) to the definitions (7.16) and (7.17), and (7.84) and (7.86) follow in turn from (7.29) and (7.31). We solve (7.84) and (7.86) using (7.108) and (7.109).

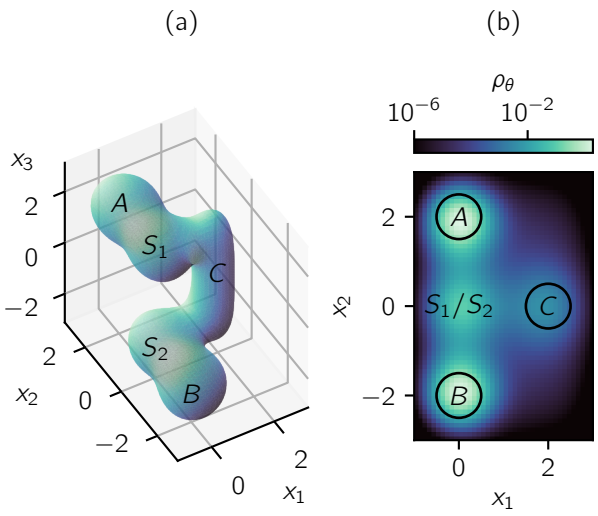

Figure 7.3: Reaction through an intermediate. (a) $U(x) = -3$ isosurface of (7.88). (b) Marginal distribution on the CV space (x_1, x_2).

5. Evaluate (7.27) using (7.110).

7.6 Numerical Examples

In this section, we demonstrate our augmented framework on simple examples that make the limitations of traditional TPT apparent. The examples we consider employ overdamped Langevin dynamics on a potential $U(x)$ and satisfy the Fokker–Planck equation

$$\frac{\partial \mathbb{P}^X[X_t]}{\partial t} = \nabla \cdot (\mathbb{P}^X[X_t] \nabla U(X_t)) + \nabla^2 \mathbb{P}^X[X_t]. \tag{7.87}$$

We calculate all statistics using a quadrature scheme adapted from Thiede et al. [4], which we detail in Appendix 7.8.1.

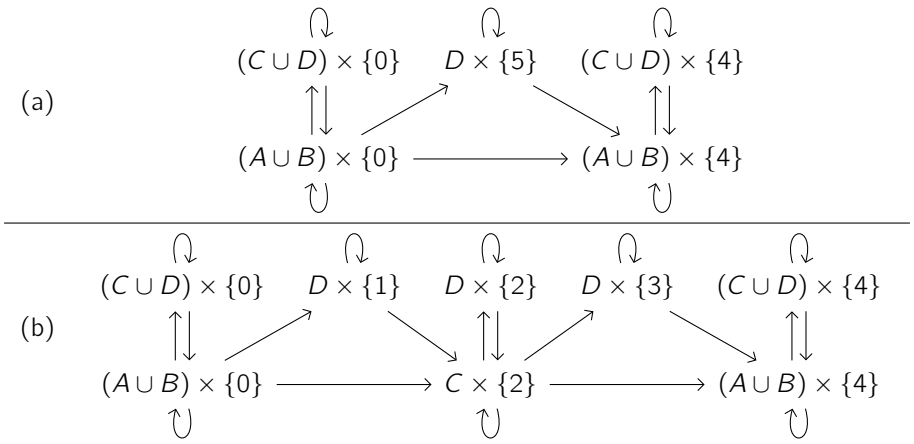

Figure 7.4: Augmented processes for reaction through an intermediate. Each arrow from one set to another indicates that Y_t may transition to $Y_{t+\epsilon}$ when Z_t is in the first set and $Z_{t+\epsilon}$ is in the second set. (a) Possible transitions of the augmented process (7.91) for the uncatalyzed pathway. (b) Possible transitions of the augmented process (7.93) for the catalyzed pathway.

7.6.1 Reaction through an Intermediate

In our first example, we demonstrate the use of augmented TPT to resolve individual reaction steps. We consider a reaction through an intermediate with the three-dimensional potential

$$\begin{aligned} U(x) = 5\Big[&\Big(\frac{x_1-1}{2}\Big)^4 + \Big(\frac{x_2}{3}\Big)^4 + \Big(\frac{x_3}{3}\Big)^4 - e^{-(x_1-2)^2-x_2^2} \\ &- 3e^{-x_1^2-(x_2-2)^2-(x_3-2)^2} - 2e^{-x_1^2-x_2^2-(x_3-2)^2} \\ &- 3e^{-x_1^2-(x_2+2)^2-(x_3+2)^2} - 2e^{-x_1^2-x_2^2-(x_3+2)^2}\Big], \end{aligned} \tag{7.88}$$

where $x = (x_1, x_2, x_3)$. We visualize the $U(x) = -3$ isosurface and the probability density on the CV space $\theta(x) = (x_1, x_2)$ in Figure 7.3.

Our reaction of interest is described by the indicator function

$$\omega(X_{r:s}) = \mathbb{1}_{A\times(C\cup D)\times\cdots\times(C\cup D)\times B}(X_{r:s}), \tag{7.89}$$

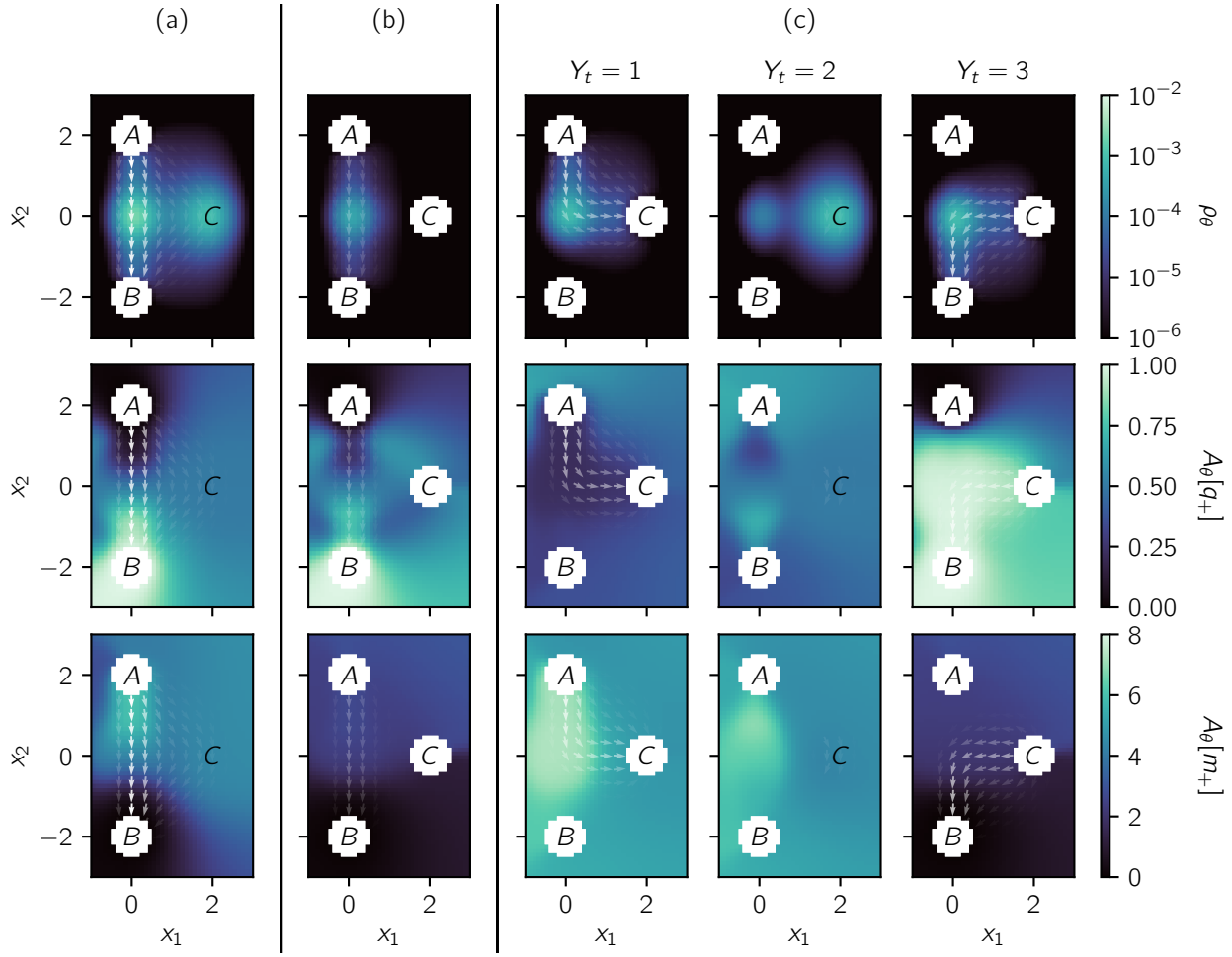

Figure 7.5: Reactive statistics for the reaction (7.89). The reactive current (vector field) is plotted over each of the other statistics (color scale): the forward committor $A_\theta[q_+]$, the reactive density ρ_θ, and the conditional mean first passage time to the product $A_\theta[m_+]$. The magnitude of the reactive current is represented by the opacity of the arrows, which are comparable for the full reaction and all reaction steps. Augmented TPT statistics are computed using the augmented process (7.93). (a) Reactive statistics computed using traditional TPT. (b) Reactive statistics for the uncatalyzed pathway computed using augmented TPT. (c) Reactive statistics for the catalyzed pathway computed using augmented TPT.

where we have defined the reactant A, product B, and intermediate C to be

$$\begin{aligned} A &= \{x \mid x_1^2 + (x_2 - 2)^2 \leq 0.5^2\}, \\ B &= \{x \mid x_1^2 + (x_2 + 2)^2 \leq 0.5^2\}, \\ C &= \{x \mid (x_1 - 2)^2 + x_2^2 \leq 0.5^2\}, \end{aligned} \tag{7.90}$$

and $D = (A \cup B \cup C)^c$. This reaction represents, for instance, a catalyzed reaction where the interaction of the substrate with the catalyst (represented by x_1) and a substrate internal coordinate (represented by x_2) can be observed while the status of the reaction (represented by x_3) cannot. The observable variables form the CV space (x_1, x_2), and the sets A, B, and C are defined on this CV space.

There are two pathways in this reaction: uncatalyzed and catalyzed. In the uncatalyzed pathway, the system transitions from the reactant A to S_1, then crosses directly to S_2 before entering the product B. In the catalyzed pathway, instead of directly crossing from S_1 to S_2, the system transitions from S_1 into the intermediate C and then to S_2.

We select trajectories that react through each pathway by applying augmented TPT to the reaction. We define an augmented process for each of the pathways by including only the terms from (7.79) that are involved in that pathway. For the uncatalyzed pathway, we remove all reactive trajectories that visit the intermediate C by removing all terms that

contain $Y_t \in \{1, 2, 3\}$ from (7.79), yielding

$$\begin{aligned}\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = [&\mathbb{1}_{\{0\}\times\{0\}}(Y_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{5\})\times(D\times\{5\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{\{4\}\times\{4\}}(Y_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times(D\times\{5\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{5\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon})],\end{aligned} \tag{7.91}$$

and then select reactive trajectories using

$$\omega(Z_{r:s}) = \mathbb{1}_{(A\times\{0\})\times(D\times\{5\})\times\cdots\times(D\times\{5\})\times(B\times\{4\})}(Z_{r:s}). \tag{7.92}$$

For the catalyzed pathway, we retain only reactive trajectories that pass through the intermediate C by removing all terms that contain $Y_t \in \{5\}$ as well as the direct transition from $(A \cup B) \times \{0\}$ to $(A \cup B) \times \{4\}$, which does not pass through C. This yields the augmented

process

$$\begin{aligned}\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = [&\mathbb{1}_{\{0\}\times\{0\}}(Y_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{1\})\times(D\times\{1\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((C\cup D)\times\{2\})\times((C\cup D)\times\{2\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{3\})\times(D\times\{3\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{\{4\}\times\{4\}}(Y_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times(D\times\{1\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{((A\cup B)\times\{0\})\times(C\times\{2\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{1\})\times(C\times\{2\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(C\times\{2\})\times(D\times\{3\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(C\times\{2\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon}) \\ &+ \mathbb{1}_{(D\times\{3\})\times((A\cup B)\times\{4\})}(Z_{t:t+\epsilon})].\end{aligned} \tag{7.93}$$

We then select reactive trajectories using

$$\omega(Z_{r:s}) = \mathbb{1}_{(A\times\{0\})\times D'\times\cdots\times D'\times(B\times\{4\})}(Z_{r:s}), \tag{7.94}$$

where $D' = (D \times \{1\}) \cup ((C \cup D) \times \{2\}) \cup (D \times \{3\})$. We show the possible transitions of (7.91) in Figure 7.4(a) and (7.93) in Figure 7.4(b).

Our goal is to visualize the mechanism of the reaction in the CV space (x_1, x_2) and quantify the relative rates of the two pathways. To this end, we examine four reactive statistics: the reactive density ρ_θ, the reactive current J_θ, the forward committor $A_\theta[q_+]$, and the conditional mean first passage time to the product $A_\theta[m_+]$.

We plot the reactive statistics from traditional TPT in Figure 7.5(a). The reactive density ρ_θ reveals that reactive trajectories spend much of their time around $(x_1, x_2) = (0, 0)$ and $(x_1, x_2) = (2, 0)$, which is consistent with the presence of intermediates S_1/S_2 and C. The

reactive current J_θ (vector field) suggests that the reaction is dominated by the uncatalyzed pathway, although a significant fraction does react through the catalyzed pathway. The forward committor $A_\theta[q_+]$ changes rapidly around $(x_1, x_2) = (0, 0)$, suggesting the presence of a bottleneck, corresponding to direct crossing from S_1 to S_2. It is almost uniform around C, suggesting the presence of the intermediate C. The conditional mean first passage time $A_\theta[m_+]$ can be interpreted in the same way as the forward committor; however, it allows us to visualize the order in which states are visited more clearly. The region below A has a higher value of $A_\theta[m_+]$ than the region around C, which suggests that reactive trajectories usually visit the former before the latter. Together, these reactive statistics suggest a cohesive picture. The reaction is dominated by the uncatalyzed pathway, which has a bottleneck around $(x_1, x_2) = (0, 0)$. Reactive trajectories may leave this pathway before the bottleneck into the catalyzed pathway, which has an intermediate C, and return after the bottleneck (i.e., they appear to circumvent S_1/S_2).

Some of the results from traditional TPT are misleading. For example, traditional TPT suggests that the uncatalyzed pathway is dominant, yet the total reactive flux from A to B is 3.0×10^{-4}, while the reactive flux for trajectories that visit C is 2.2×10^{-4}, i.e., 73% of trajectories go through the intermediate. This results from the restriction of the observed coordinates to (x_1, x_2); in the full state space (x_1, x_2, x_3), traditional TPT is capable of correctly resolving the two pathways. However, we note that even given (x_1, x_2, x_3), traditional TPT cannot calculate dynamical statistics for the ensemble of trajectories that react through a particular pathway. Augmented TPT provides a solution to the overlap issue and enables the calculation of reactive statistics for individual steps of each pathway.

We first analyze the uncatalyzed pathway, which was wrongly suggested by traditional TPT to be the dominant pathway. Reactive statistics for this pathway are shown in Figure 7.5(b). As we would expect, the reactive density ρ_θ shows that reactive trajectories spend much of their time around S_1/S_2, and the reactive current J_θ suggests that reactive trajectories flow

directly from A to S_1/S_2 to B, without any notable deviation to the vicinity of C. On the uncatalyzed pathway, the forward committor $A_\theta[q_+]$ changes rapidly around $(x_1, x_2) = (0, 0)$ due to the transition from S_1 to S_2. Off the uncatalyzed pathway, the forward committor has a higher value closer to A and a lower value closer to B. This is surprising and results from slight differences in the reactive density at different values of x_3. The conditional mean first passage time $A_\theta[m_+]$ rapidly decreases near S_1/S_2, suggesting the same single bottleneck.

We now analyze the catalyzed pathway through the intermediate C, which dominates the rate. We select trajectories that react through this pathway using (7.94). Augmented TPT enables us to split the pathway into individual steps, and so resolve the structure of each reaction step. The first step ($Y_t = 1$) starts when the reactive trajectory leaves the reactant A and ends when it first enters the intermediate C. The second step ($Y_t = 2$) starts at the first time the reactive trajectory enters C, and ends at the last time the reactive trajectory leaves C. The third step ($Y_t = 3$) starts when the reactive trajectory last leaves the intermediate C and ends when it enters the product B. As we now explain, separating the catalyzed pathway into these steps leads to reactive statistics that lead to a different interpretation than those from traditional TPT.

In the first step, the reactive density ρ_θ and reactive current J_θ clearly show that most reactive trajectories flow through an intermediate near $(x_1, x_2) = (0, 0)$, in this case S_1, rather than a more direct path from A to C, as suggested by the reactive current from traditional TPT. Likewise, in the last step, they show that most reactive trajectories flow through an intermediate near $(x_1, x_2) = (0, 0)$, in this case S_2. The absence of any significant reactive current in the second step, along with the high reactive density near C, suggests that reactive trajectories predominantly remain in C during this step, with a few trajectories transitioning back and forth to S_1/S_2, where there is a lower value of reactive density. The reactive current from traditional TPT is misleading because the flows from S_1 to C and from C to S_2 cancel each other, since S_1 and S_2 overlap in the CV space.

The forward committor $A_\theta[q_+]$ on the catalyzed pathway is uniformly low in the first step and uniformly high in the third step. This suggests that the main bottleneck occurs in the second step, where $A_\theta[q_+] \approx 0.5$ around C. The abrupt changes between steps suggest that the dynamics of the variables not captured within the CV space are influential in determining whether the reaction occurs. For the second step, we note that the low value of $A_\theta[q_+]$ below A and high value above B reflect the full three-dimensional potential (Figure 7.3(a)). In traditional TPT, the high value of $A_\theta[q_+]$ in the first step and the low value in the third step cancel, giving rise to the apparent rapid change near $(x_1, x_2) = (0, 0)$.

In the first step, $A_\theta[m_+]$ decreases from $(x_1, x_2) = (0, 0)$ to C, suggesting the presence of a bottleneck between intermediates S_1 to C. The same holds for the second step, suggesting that if the system crosses back to an intermediate near $(x_1, x_2) = (0, 0)$, it needs to overcome the same bottleneck to return to C. In the third step, $A_\theta[m_+]$ decreases from $(x_1, x_2) = (0, 0)$ to B, marking the transition from S_2 to B. We note that this decrease occurs at a slightly lower value than in the uncatalyzed pathway. This separation of the two bottlenecks lies in contrast with $A_\theta[m_+]$ from traditional TPT, where the superposition of the two bottlenecks creates an apparent bottleneck near $(x_1, x_2) = (0, 0)$, which conflates the dynamics of the catalyzed and uncatalyzed pathways.

Overall, we see that the statistics from traditional TPT qualitatively resemble a superposition of those for the uncatalyzed pathway and those associated with the second step of the catalyzed pathway, in which the system is mainly localized at C. The important contributions from the first and third steps of the catalyzed pathway mask each other in traditional TPT.

7.6.2 Reaction with Multiple Pathways

For our second example, we demonstrate the use of augmented TPT to separate pathways which overlap in the CV space. We consider overdamped Langevin dynamics on the three-

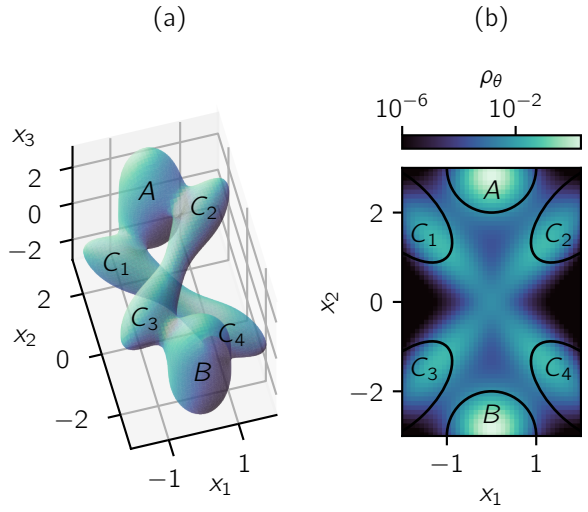

Figure 7.6: Reaction with multiple pathways. (a) $U(x) = -3$ isosurface of (7.95). (b) Marginal distribution on the CV space (x_1, x_2).

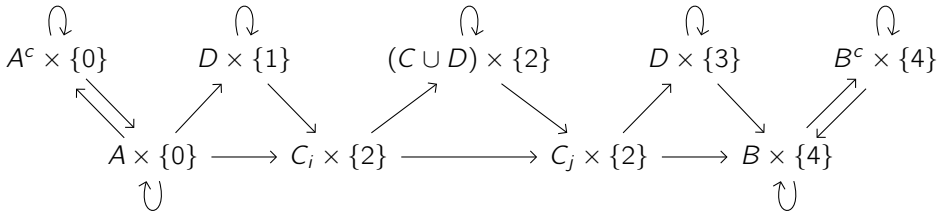

Figure 7.7: Possible transitions for the augmented process of the reaction with multiple pathways, defined by (7.99).

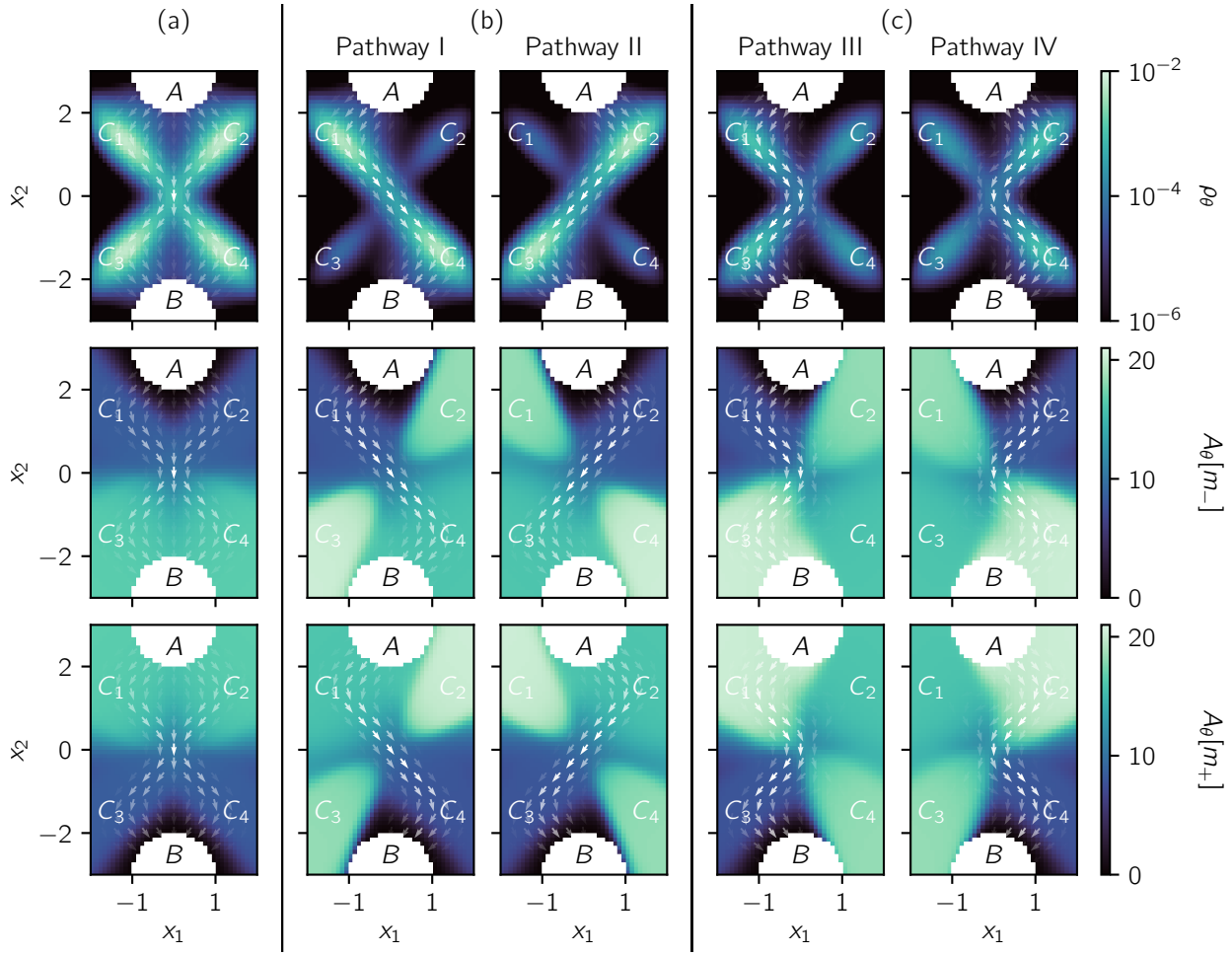

Figure 7.8: Reactive statistics for the reaction with multiple pathways. The reactive current (vector field) is plotted over each of the other statistics. Note that the magnitude of the reactive current is represented by the opacity of the arrows, and that they are normalized differently for each reaction and pathway. The other reactive statistics (color scale) are the reactive density ρ_θ, the conditional mean last passage time from the reactant $A_\theta[m_-]$, and the conditional mean first passage time to the product $A_\theta[m_+]$. (a) Reactive statistics from traditional TPT. (b) Reactive statistics from augmented TPT for major pathways I and II. (c) Reactive statistics from augmented TPT for minor pathways III and IV.

dimensional potential

$$\begin{aligned} U(x) = 5\Big[&\Big(\frac{x_1}{2}\Big)^4 + \Big(\frac{x_2}{3}\Big)^4 + \Big(\frac{x_3}{2}\Big)^4 + e^{-x_2^2} \\ &- 3e^{-x_1^2-(x_2-3)^2} - 2e^{-(x_1-x_2)^2+(x_3-1)^2} \\ &- 3e^{-x_1^2-(x_2+3)^2} - 2e^{-(x_1+x_2)^2-(x_3+1)^2}\Big], \end{aligned} \tag{7.95}$$

where $x = (x_1, x_2, x_3)$. As previously, these dynamics satisfy the Fokker–Planck equation in (7.87). The $U(x) = -3$ isosurface for this potential is shown in Figure 7.6(a), and the probability distribution on the (x_1, x_2) coordinates, which we use as CVs, is shown in Figure 7.6(b). We define the reactant A, product B, and intermediates C_1, C_2, C_3, and C_4 to be

$$\begin{aligned} A &= \{x \mid x_1^2 + (x_2 - 3)^2 \le 1\}, \\ B &= \{x \mid x_1^2 + (x_2 + 3)^2 \le 1\}, \\ C_1 &= \{x \mid (x_1 + x_2)^2 + (x_1 - x_2 + 4)^2/4 \le 1\}, \\ C_2 &= \{x \mid (x_1 + x_2 - 4)^2/4 + (x_1 - x_2)^2 \le 1\}, \\ C_3 &= \{x \mid (x_1 + x_2 + 4)^2/4 + (x_1 - x_2)^2 \le 1\}, \\ C_4 &= \{x \mid (x_1 + x_2)^2 + (x_1 - x_2 - 4)^2/4 \le 1\}. \end{aligned} \tag{7.96}$$

We also define the sets $C = C_1 \cup C_2 \cup C_3 \cup C_4$ and $D = (A \cup B \cup C)^c$. The reaction of interest is specified through the indicator function

$$\omega(X_{r:s}) = \mathbb{1}_{A\times(C\cup D)\times\cdots\times(C\cup D)\times B}(X_{r:s}). \tag{7.97}$$

We use the intermediates to define pathways. This is advantageous because it is not possible to divide the space into regions corresponding to different pathways owing to overlap. We define four pathways: major pathways I and II, and minor pathways III and IV. We define pathway I to first hit intermediate $C_i = C_1$ after leaving A and last hit intermediate

$C_j = C_4$ before hitting B. We likewise define pathway II with $(C_i, C_j) = (C_2, C_3)$, pathway III with $(C_i, C_j) = (C_1, C_3)$, and pathway IV with $(C_i, C_j) = (C_2, C_4)$.

To define means of selecting these pathways, we label a trajectory before the reaction with $Y_t = 0$ and after the reaction with $Y_t = 4$. When the reaction is in process, we label times before the reactive trajectory first enters C_i with $Y_t = 1$, after the reactive trajectory last exits C_j with $Y_t = 3$, and between those times with $Y_t = 2$. Then, we select pathways using

$$\omega(Z_{r:s}) = \mathbb{1}_{(A\times\{0\})\times D'\cdots D'\times(B\times\{4\})}(Z_{r:s}) \tag{7.98}$$

where $D' = (D \times \{1\}) \cup ((C \cup D) \times \{2\}) \cup (D \times \{3\})$. This choice of Y_t corresponds to

$$\begin{aligned}
\kappa(Y_{t:t+\epsilon} \,|\, X_{t:t+\epsilon}) = [&\mathbb{1}_{\{0\}\times\{0\}}(Y_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(D\times\{1\})\times(D\times\{1\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{((C\cup D)\times\{2\})\times((C\cup D)\times\{2\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(D\times\{3\})\times(D\times\{3\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{\{4\}\times\{4\}}(Y_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(A\times\{0\})\times(D\times\{1\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(A\times\{0\})\times(C_i\times\{2\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(D\times\{1\})\times(C_i\times\{2\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(C_j\times\{2\})\times(D\times\{3\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(C_j\times\{2\})\times(B\times\{4\})}(Z_{t:t+\epsilon}) \\
&+ \mathbb{1}_{(D\times\{3\})\times(B\times\{4\})}(Z_{t:t+\epsilon})].
\end{aligned} \tag{7.99}$$

The first five terms of (7.99) denote the sets in which X_t may be for each of the labels $\{0, 1, 2, 3, 4\}$, and the remaining six terms describe the permitted transitions between the labels. We represent (7.99) visually in Figure 7.7. For instance, the seventh term,

$\mathbb{1}_{(A\times\{0\})\times(C_i\times\{2\})}(Z_{t:t+\epsilon})$, corresponds to the single timestep transition from A to C_i and associates this with a change in the label from $Y_t = 0$ to $Y_t = 2$.

We determine the reactive flux associated with each pathway and compare it with the total reactive flux. For the reaction specified by (7.97), the total reactive flux is 7.5×10^{-4}. Each of the major pathways has a reactive flux of 2.9×10^{-4}, which is 39% of the total reactive flux. Each of the minor pathways has a reactive flux of 6.4×10^{-4}, which is 8.5% of the total reactive flux. These four pathways thus give rise to 95% of the total reactive flux, and so are representative of the majority of reactive trajectories. The remaining 5% results from trajectories that do not conform to these pathways (e.g., ones that pass through only a single intermediate).

In Figure 7.8, we plot four reactive statistics: the reactive density ρ_θ, the reactive current J_θ, the conditional mean last passage time from the reactant $A_\theta[m_-]$, and the conditional mean first passage time to the product $A_\theta[m_+]$.

Reactive statistics from traditional TPT are shown in Figure 7.8(a). From the reactive density ρ_θ, we observe that reactive trajectories spend most of their time in the X-shaped region that connects the intermediates. The reactive current J_θ suggests that the majority of the reactive trajectories flow from the reactant A to either C_1 or C_2, then to either C_3 or C_4 via $(x_1, x_2) = (0, 0)$, and lastly to the product B. Importantly, even if we consider the full state space (x_1, x_2, x_3), we cannot determine the relative weights of these four possible pathways, because the pathways are composed of segments that belong to multiple pathways (e.g., the first half of pathway III overlaps with pathway I and the second half of pathway III overlaps with pathway II) and the transitions through $(x_1, x_2, x_3) = (0, 0, 0)$ along pathways III and IV occur in opposite directions. Other quantities calculated using traditional TPT have the same issue. Both $A_\theta[m_-]$ and $A_\theta[m_+]$ are unable to distinguish between the pathways and only indicate the presence of a bottleneck between $C_1 \cup C_2$ and $C_3 \cup C_4$.

In Figure 7.8(b), we visualize the reactive statistics for the major pathways. As pathway

I and pathway II are mirror images of one another, we discuss only pathway I. The reactive current J_θ clearly shows that the system transitions directly between on-pathway intermediates, from A to C_1 to C_4 to B. The reactive density ρ_θ corroborates this picture, with relatively little density in C_2 and C_3 compared to C_1 and C_4. We observe that $A_\theta[m_-]$ and $A_\theta[m_+]$ are highest near C_2 and C_3, suggesting that these configurations are dynamically disconnected from the main flow of the reactive trajectories. The transition from C_1 to C_4 is accompanied by an abrupt increase in $A_\theta[m_-](v)$ and an abrupt decrease in $A_\theta[m_+]$, which shows that a transition bottleneck is traversed. We note that the sharp increase in $A_\theta[m_-]$ from A to C_1 and the sharp decrease in $A_\theta[m_+]$ from C_4 to B imply the presence of bottlenecks between each of these pairs of states.

The minor pathways in Figure 7.8(c) result from trajectories that switch between the major pathways. Since the two minor pathways are mirror images of each other, we discuss only pathway III, which involves a switch from pathway I to pathway II. In contrast with the major pathways, the reactive density ρ_θ indicates that the system is likely to visit off-pathway intermediates C_2 and C_4 on its way from C_1 to C_3. The conditional mean last passage time $A_\theta[m_-]$ from the reactant is nearly identical to that of pathway I, with higher values near C_2 and C_3 and lower values near C_1 and C_4. However, the conditional mean first passage time $A_\theta[m_+]$ to the product is nearly identical to that of pathway II. In conjunction with ρ_θ, the slight increase in $A_\theta[m_-]$ from C_1 to C_4 suggests that these intermediates readily interconvert, and similarly for the slight increase in $A_\theta[m_+]$ from C_3 to C_2. The larger change in $A_\theta[m_-]$ and $A_\theta[m_+]$ between $C_2 \cup C_3$ and $C_1 \cup C_4$ implies a bottleneck between $C_2 \cup C_3$ and $C_1 \cup C_4$. This bottleneck is significant because the transition from C_1 to C_3 must occur for this pathway. As with the major pathways, there are also bottlenecks between A and C_1, and C_3 and B.

7.7 Discussion

In this chapter, we introduced an augmented process that labels sequences of events. This process enabled us to write statistics that depend on knowledge of past and future events in terms of quantities that are local in time and in turn to extend the TPT framework. We demonstrated how this framework can be used to separate statistics of competing pathways in reactions with intermediates to reveal features of mechanisms that are not apparent from traditional TPT analyses. Our framework can also be used to treat new classes of reactions that are not amenable to TPT analyses. For instance, reactions with the same reactant and product states, such as cycles of oscillators and excitable systems, can be handled using augmented TPT but not traditional TPT.

Our framework generalizes a previous extension of TPT [207] and history augmented approaches for computing rates [37, 213, 214] and reactive statistics [205]. The augmented process that we introduce is distinct from that in Helfmann et al. [215], in which the state space is expanded to include a time variable to treat time-dependent processes, including transient relaxations and systems with periodically varying dynamics. As a result, the two approaches can be combined to treat sequences of events of finite-time processes.

Our focus here was on establishing the conceptual framework for augmented TPT, and the examples that we showed were sufficiently simple that the Fokker–Planck equations defining their dynamics could be numerically integrated in the variables by quadrature. The dynamics of models with larger numbers of variables must instead be sampled through simulations that generate stochastic realizations of trajectories (i.e., the dynamics of the variables are numerically integrated in time). Because, like traditional TPT, the framework casts statistics in terms of quantities that are local in time, we can extend methods that compute reactive statistics from short trajectories [4, 5, 205]. Such efforts are underway.

7.8 Appendix

7.8.1 Finite Difference Scheme

For a time-reversible drift-diffusion process

$$\frac{\partial \, \mathbb{P}^X[X_t]}{\partial t} = \nabla \cdot (\mathbb{P}^X[X_t] \nabla U(X_t)) + \nabla^2 \, \mathbb{P}^X[X_t] \tag{7.100}$$

with stationary distribution $\pi(X_t) \propto \exp(-U(X_t))$, the infinitesimal generator can be used to compute expectations forward-in-time as

$$\frac{\partial \, \mathbb{E}^X[f(X_{t+\epsilon}) \mid X_t]}{\partial \epsilon} = -\nabla U(X_t) \cdot \nabla f(X_t) + \nabla^2 f(X_t). \tag{7.101}$$

To evaluate expectations by quadrature, we adapt the finite difference scheme from Thiede et al. [4], which we reproduce here. We approximate (7.100) as a discrete time Markov jump process with time step ϵ on a grid with uniform spacing Δ. For a small change Δ_i in the direction of the i^{th} coordinate with magnitude Δ, we substitute $-\nabla U(X_t) = \nabla \pi(X_t)/\pi(X_t)$ and then make the approximation

$$\begin{aligned}
&\frac{\mathbb{E}[f(X_{t+\epsilon}) \mid X_t] - f(X_t)}{\epsilon} \\
&\quad \approx \frac{1}{2} \sum_i \frac{(\pi(X_t + \Delta_i) - \pi(X_t))/\Delta}{(\pi(X_t + \Delta_i) + \pi(X_t))/2} \left[\frac{f(X_t + \Delta_i) - f(X_t)}{\Delta} \right] \\
&\qquad + \frac{1}{2} \sum_i \frac{(\pi(X_t) - \pi(X_t - \Delta_i))/\Delta}{(\pi(X_t) + \pi(X_t - \Delta_i))/2} \left[\frac{f(X_t) - f(X_t - \Delta_i)}{\Delta} \right] \\
&\qquad + \sum_i \frac{f(X_t + \Delta_i) + f(X_t - \Delta_i) - 2f(X_t)}{\Delta^2}.
\end{aligned} \tag{7.102}$$

Alternatively, we can write

$$\begin{aligned}
&\frac{\mathbb{E}^X[f(X_{t+\epsilon}) \mid X_t] - f(X_t)}{\epsilon} \\
&= \frac{1}{\epsilon}\left(\frac{\int \mathbb{P}^X[X_{t:t+\epsilon}] f(X_{t+\epsilon})\,\mathrm{d}X_{t+\epsilon}}{\mathbb{P}^X[X_t]} - f(X_t)\right) && (7.103)\\
&\approx \frac{1}{\epsilon}\Big[\Big(P(X_t, X_t) f(X_t) && (7.104)\\
&\qquad + \sum_i P(X_t, X_t + \Delta_i) f(X_t + \Delta_i) \\
&\qquad + \sum_i P(X_t, X_t - \Delta_i) f(X_t - \Delta_i)\Big)\Big/\pi(X_t) - f(X_t)\Big],
\end{aligned}$$

where $P(X_{t:t+\epsilon})$ represents the approximation of $\mathbb{P}^X[X_{t:t+\epsilon}]$ on the grid. Above, the first equality follows from the definition of conditional expectation, and the second assumes that all transitions within time Δ are to neighboring grid points.

By matching terms between (7.104) and (7.102), the only nonzero entries of $P(x, x')$ are

$$P(x, x + \Delta_i) = \frac{2\epsilon}{\Delta^2}\frac{1}{1/\pi(x) + 1/\pi(x + \Delta_i)}, \tag{7.105}$$

$$P(x, x - \Delta_i) = \frac{2\epsilon}{\Delta^2}\frac{1}{1/\pi(x) + 1/\pi(x - \Delta_i)}, \tag{7.106}$$

$$P(x, x) = \pi(x) - \sum_i [P(x, x + \Delta_i) + P(x, x - \Delta_i)]. \tag{7.107}$$

Then, we can use the above expressions to estimate expectations using

$$\mathbb{E}^X[f(X_{t-\epsilon:t}) \mid X_t] \approx \frac{\sum_{X_{t-\epsilon}} P(X_{t-\epsilon:t}) f(X_{t-\epsilon:t})}{\pi(X_t)}, \tag{7.108}$$

$$\mathbb{E}^X[f(X_{t:t+\epsilon}) \mid X_t] \approx \frac{\sum_{X_{t+\epsilon}} P(X_{t:t+\epsilon}) f(X_{t:t+\epsilon})}{\pi(X_t)}, \tag{7.109}$$

$$\mathbb{E}^X[f(X_{t:t+\epsilon})] \approx \sum_{X_{t:t+\epsilon}} P(X_{t:t+\epsilon}) f(X_{t:t+\epsilon}), \tag{7.110}$$

where the sums are over points on the grid.

CHAPTER 8

CONCLUSIONS AND OUTLOOK

In this dissertation, we have developed several frameworks for estimating dynamical observables from ensembles of short simulations. Chapters 2–6 introduced methods that are more accurate or robust. Specifically, Chapters 2 and 4 demonstrated that using multiple time lags improves dynamical observable estimation, leading to better data efficiency and reduced sampling requirements. Chapters 3 and 5 use finite time lags to approximate Markovian behavior better than previous formulations with infinitesimal time lags, while Chapter 4 lifts the Markov assumption entirely by incorporating memory. Chapter 7 proposes a framework that shows how to introduce auxilliary variables to compute path- (and thus history-) dependent statistics.

Chapters 2, 5, and 6, explored deep learning approaches for estimating slow modes and prediction functions. Despite their flexibility, neural networks still depend on carefully chosen input features. An important future goal is to learn dynamical observables directly from Cartesian coordinates, without user intervention. As an initial attempt, in Pengmei et al. [186], we pretrained graph neural networks on small-molecule configurations and showed that their embeddings can be used as input features for learning slow modes in macromolecules. However, these features performed poorly for prediction functions with both exact VCN and inexact subspace iteration. This failure resulted from the fact that small changes in molecular configuration can lead to large, nonlinear changes in these features, making it difficult for the model to generalize. Further progress requires developing transferable features that are data-efficient and interpretable.

Neural networks are also prone to overfitting, particularly since both variational and iterative schemes rely on temporal differences: they self-consistently minimize the difference between a learned observable and its time-lagged expectation, typically estimated from a single trajectory segment. With limited and highly correlated data, repeated training on the

same segments allows the model to memorize trajectories rather capture physical features. For variational methods, early stopping based on cross-validation with the loss function helps mitigate overfitting, though artifacts remain. The iterative method, lacking a loss function or a meaningful validation metric, is even more susceptible. Further tests show that, other than early stopping, traditional regularization methods—including dropout, batch normalization, and weight decay—fail to prevent overfitting. When the input features are well-behaved, adding noise or penalizing the input gradient can help. However, when the input features are poorly behaved (e.g., from pretrained graph neural networks), a more effective strategy is to regularize the output distribution directly by penalizing it complexity, for instance using the Wasserstein k-variance.

We believe that the inexact subspace iteration of Chapter 6 holds the most promise, given its theoretical applicability to general datasets and observables, but it remains practically challenging. Training is often unstable, with results oscillating or diverging, and our attempts to learn backward-in-time prediction functions or ATPT statistics have largely failed for this reason. The absence of a validation metric exacerbates overfitting, as early stopping is not applicable. Regularizing the output, as mentioned above, offers a partial solution. In ongoing work, we adapt techniques from deep reinforcement learning to improve both stability and validation.

Prediction functions and TPT statistics depend on defining metastable states, which requires prior knowledge of the system. In Chapter 7, we discussed selecting reaction steps and pathways, which again requires defining metastable and intermediate states. Poor state definitions can yield useless results. An important future direction is to learn states and pathways directly from dynamics. Our preliminary work proposes optimizing state definitions by minimizing transition rates, with penalties to avoid excessively small states. A natural next step is to learn to separate pathways, for example by identifying mutually exclusive intermediates.

REFERENCES

[1] R. J. Webber, E. H. Thiede, D. Dow, A. R. Dinner, and J. Weare, "Error bounds for dynamical spectral estimation", SIAM J. Math. Data Sci. **3**, 225–252 (2021).

[2] S. Schultze and H. Grubmüller, "Time-lagged independent component analysis of random walks and protein dynamics", J. Chem. Theory Comput. **17**, 5766–5776 (2021).

[3] E. Vanden-Eijnden, "Transition path theory", in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, edited by M. Ferrario, G. Ciccotti, and K. Binder, Lecture Notes in Physics (Springer, Berlin, Heidelberg, 2006), pp. 453–493.

[4] E. H. Thiede, D. Giannakis, A. R. Dinner, and J. Weare, "Galerkin approximation of dynamical quantities using trajectory data", J. Chem. Phys. **150**, 244111 (2019).

[5] J. Strahan, A. Antoszewski, C. Lorpaiboon, B. P. Vani, J. Weare, and A. R. Dinner, "Long-time-scale predictions from short-trajectory data: a benchmark analysis of the Trp-cage miniprotein", J. Chem. Theory Comput. **17**, 2948–2963 (2021).

[6] A. J. Chorin, O. H. Hald, and R. Kupferman, "Optimal prediction with memory", Phys. Nonlinear Phenom. **166**, 239–257 (2002).

[7] S. Cao, A. Montoya-Castillo, W. Wang, T. E. Markland, and X. Huang, "On the advantages of exploiting memory in Markov state models for biomolecular dynamics", J. Chem. Phys. **153**, 014105 (2020).

[8] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, "Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations", Proc. Natl. Acad. Sci. **106**, 19011–19016 (2009).

[9] C. Lorpaiboon, E. H. Thiede, R. J. Webber, J. Weare, and A. R. Dinner, "Integrated variational approach to conformational dynamics: a robust strategy for identifying eigenfunctions of dynamical operators", J. Phys. Chem. B **124**, 9354–9364 (2020).

[10] F. Noé and F. Nuske, "A variational approach to modeling slow processes in stochastic dynamical systems", Multiscale Model. Simul. **11**, 635–655 (2013).

[11] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, "Variational approach to molecular kinetics", J. Chem. Theory Comput. **10**, 1739–1752 (2014).

[12] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, "Identification of slow molecular order parameters for Markov model construction", J. Chem. Phys. **139**, 015102 (2013).

[13] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations", Phys. Rev. Lett. **72**, 3634 (1994).

[14] C. R. Schwantes and V. S. Pande, "Improvements in Markov state model construction reveal many non-native interactions in the folding of NTL9", J. Chem. Theory Comput. **9**, 2000–2009 (2013).

[15] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, "Data-driven model reduction and transfer operator approximation", J. Nonlinear Sci. **28**, 985–1010 (2018).

[16] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard, "A direct approach to conformational dynamics based on hybrid Monte Carlo", J. Comput. Phys. **151**, 146–168 (1999).

[17] W. C. Swope, J. W. Pitera, and F. Suits, "Describing protein folding kinetics by molecular dynamics simulations. 1. Theory", J. Phys. Chem. B **108**, 6571–6581 (2004).

[18] W. C. Swope, J. W. Pitera, F. Suits, M. Pitman, M. Eleftheriou, B. G. Fitch, R. S. Germain, A. Rayshubski, T. C. Ward, Y. Zhestkov, et al., "Describing protein folding kinetics by molecular dynamics simulations. 2. Example applications to alanine dipeptide and a β-hairpin peptide", J. Phys. Chem. B **108**, 6582–6594 (2004).

[19] A. Mardt, L. Pasquali, H. Wu, and F. Noé, "VAMPnets for deep learning of molecular kinetics", Nat. Commun. **9**, 1–11 (2018).

[20] W. Chen, H. Sidky, and A. L. Ferguson, "Nonlinear discovery of slow molecular modes using state-free reversible VAMPnets", J. Chem. Phys. **150**, 214114 (2019).

[21] H. Takano and S. Miyashita, "Relaxation modes in random spin systems", J. Phys. Soc. Jpn. **64**, 3688–3698 (1995).

[22] O. Kallenberg, *Foundations of modern probability* (Springer Science & Business Media, 2006).

[23] T. Eisner, B. Farkas, M. Haase, and R. Nagel, *Operator theoretic aspects of ergodic theory*, Vol. 272 (Springer, 2015).

[24] H. Hirao, S. Koseki, and H. Takano, "Molecular dynamics study of relaxation modes of a single polymer chain", J. Phys. Soc. Jpn. **66**, 3399–3405 (1997).

[25] V. S. Pande, K. Beauchamp, and G. R. Bowman, "Everything you wanted to know about Markov state models but were afraid to ask", Methods San Diego Calif **52**, 99–105 (2010).

[26] E. Vanden-Eijnden, "Transition path theory", in *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, edited by G. R. Bowman, V. S. Pande, and F. Noé, Advances in Experimental Medicine and Biology (Springer Netherlands, Dordrecht, 2014), pp. 91–100.

[27] F. Noé and J.-H. Prinz, "Analysis of Markov models", in *An introduction to Markov state models and their application to long timescale molecular simulation, vol. 797 of advances in experimental medicine and biology*, edited by G. R. Bowman, V. S. Pande, and F. Noé (Springer, 2014) Chap. 6.

[28] B. G. Keller, S. Aleksic, and L. Donati, "Markov state models in drug design", in *Biomolecular simulations in drug discovery*, edited by F. L. Gervasio and V. Spiwok (Wiley-VCH, 2019) Chap. 4.

[29] F. Vitalini, F. Noé, and B. Keller, "A basis set for peptides for the variational approach to conformational kinetics", J. Chem. Theory Comput. **11**, 3992–4004 (2015).

[30] L. Boninsegna, G. Gobbo, F. Noé, and C. Clementi, "Investigating molecular kinetics by variationally optimized diffusion maps", J. Chem. Theory Comput. **11**, 5947–5960 (2015).

[31] C. R. Schwantes, R. T. McGibbon, and V. S. Pande, "Perspective: Markov models for long-timescale biomolecular dynamics", J. Chem. Phys. **141**, 09b201 (2014).

[32] H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé, "Variational Koopman models: slow collective variables and molecular kinetics from short off-equilibrium simulations", J. Chem. Phys. **146**, 154104 (2017).

[33] R. T. McGibbon and V. S. Pande, "Variational cross-validation of slow dynamical modes in molecular kinetics", J. Chem. Phys. **142**, 124105 (2015).

[34] Y. Naritomi and S. Fuchigami, "Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: the case of domain motions", J. Chem. Phys. **134**, 065101 (2011).

[35] B. E. Husic and V. S. Pande, "Note: MSM lag time cannot be used for variational model selection", J. Chem. Phys. **147**, 176101 (2017).

[36] H. Wu, J.-H. Prinz, and F. Noé, "Projected metastable Markov processes and their estimation with observable operator models", J. Chem. Phys. **143**, 144101 (2015).

[37] E. Suárez, J. L. Adelman, and D. M. Zuckerman, "Accurate estimation of protein folding and unfolding times: beyond Markov state models", J. Chem. Theory Comput. **12**, 3473–3481 (2016).

[38] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints", SIAM J. Matrix Anal. Appl. **20**, 303–353 (1998).

[39] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, "How fast-folding proteins fold", Science **334**, 517–520 (2011).

[40] J. C. McKnight, D. S. Doering, P. T. Matsudaira, and P. S. Kim, "A thermostable 35-residue subdomain within villin headpiece", J. Mol. Biol. **260**, 126–134 (1996).

[41] J. Kubelka, W. A. Eaton, and J. Hofrichter, "Experimental tests of villin subdomain folding simulations", J. Mol. Biol. **329**, 625–630 (2003).

[42] Y. Duan and P. A. Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution", Science **282**, 740–744 (1998).

[43] H. Sidky, W. Chen, and A. L. Ferguson, "High-resolution Markov state models for the dynamics of Trp-cage miniprotein constructed over slow folding modes identified by state-free reversible VAMPnets", J. Phys. Chem. B **123**, 7999–8009 (2019).

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization", in Int. Conf. Learn. Represent. (2019).

[45] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms", IEEE Trans. Audio Electroacoustics **15**, 70–73 (1967).

[46] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, "GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers", SoftwareX **1–2**, 19–25 (2015).

[47] S. Páll, M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl, "Tackling exascale software challenges in molecular dynamics simulations with GROMACS", in Int. Conf. Exascale Appl. Softw. (2014), pp. 3–27.

[48] S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, et al., "GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit", Bioinformatics **29**, 845–854 (2013).

[49] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, "GROMACS: fast, flexible, and free", J. Comput. Chem. **26**, 1701–1718 (2005).

[50] E. Lindahl, B. Hess, and D. Van Der Spoel, "GROMACS 3.0: a package for molecular simulation and trajectory analysis", Mol. Model. Annu. **7**, 306–317 (2001).

[51] H. J. Berendsen, D. van der Spoel, and R. van Drunen, "GROMACS: a message-passing parallel molecular dynamics implementation", Comput. Phys. Commun. **91**, 43–56 (1995).

[52] A. D. MacKerell Jr, N. Banavali, and N. Foloppe, "Development and current status of the CHARMM force field for nucleic acids", Biopolym. Orig. Res. Biomol. **56**, 257–265 (2000).

[53] H. J. Berendsen, J. P. Postma, W. F. van Gunsteren, and J. Hermans, "Interaction models for water in relation to protein hydration", in *Intermolecular forces* (Springer, 1981), pp. 331–342.

[54] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, "A smooth particle mesh ewald method", J. Chem. Phys. **103**, 8577–8593 (1995).

[55] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije, "LINCS: a linear constraint solver for molecular simulations", J. Comput. Chem. **18**, 1463–1472 (1997).

[56] S. Miyamoto and P. A. Kollman, "Settle: an analytical version of the SHAKE and RATTLE algorithm for rigid water models", J. Comput. Chem. **13**, 952–962 (1992).

[57] M. Parrinello and A. Rahman, "Polymorphic transitions in single crystals: a new molecular dynamics method", J. Appl. Phys. **52**, 7182–7190 (1981).

[58] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, "PyEMMA 2: a software package for estimation, validation, and analysis of Markov models", J. Chem. Theory Comput. **11**, 5525–5542 (2015).

[59] G. A. Huber and S. Kim, "Weighted-ensemble Brownian dynamics simulations for protein association reactions.", Biophys. J. **70**, 97–110 (1996).

[60] A. Dickson and A. R. Dinner, "Enhanced sampling of nonequilibrium steady states", Annu. Rev. Phys. Chem. **61**, 441–459 (2010).

[61] N. Guttenberg, A. R. Dinner, and J. Weare, "Steered transition path sampling", J. Chem. Phys. **136**, 234103 (2012).

[62] D. Aristoff, J. M. Bello-Rivas, and R. Elber, "A mathematical framework for exact milestoning", Multiscale Model. Simul. **14**, 301–322 (2016).

[63] A. R. Dinner, J. C. Mattingly, J. O. B. Tempkin, B. V. Koten, and J. Weare, "Trajectory stratification of stochastic dynamics", SIAM Rev. **60**, 909–938 (2018).

[64] R. J. Allen, D. Frenkel, and P. R. ten Wolde, "Simulating rare events in equilibrium or nonequilibrium stochastic systems", J. Chem. Phys. **124**, 024102 (2006).

[65] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data", J. Fluid Mech. **656**, 5–28 (2010).

[66] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, "Markov models of molecular kinetics: generation and validation", J. Chem. Phys. **134**, 174105 (2011).

[67] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: extending dynamic mode decomposition", J. Nonlinear Sci. **25**, 1307–1346 (2015).

[68] B. E. Husic and V. S. Pande, "Markov state models: from an art to a science", J. Am. Chem. Soc. **140**, 2386–2396 (2018).

[69] F. Nüske, R. Schneider, F. Vitalini, and F. Noé, "Variational tensor approach for approximating the rare-event kinetics of macromolecular systems", J. Chem. Phys. **144**, 054105 (2016).

[70] J.-H. Prinz, J. D. Chodera, and F. Noé, "Spectral rate theory for two-state kinetics", Phys. Rev. X **4**, 011020 (2014).

[71] P. Del Moral, *Feynman–Kac formulae* (Springer, 2004).

[72] I. Karatzas and S. Shreve, *Brownian motion and stochastic calculus*, Vol. 113 (Springer Science & Business Media, 2012).

[73] W. E. and E. Vanden-Eijnden, "Towards a theory of transition paths", J. Stat. Phys. **123**, 503 (2006).

[74] P. Metzner, C. Schütte, and E. Vanden-Eijnden, "Transition path theory for Markov jump processes", Multiscale Model. Simul. **7**, 1192–1219 (2009).

[75] Y. Liu, D. P. Hickey, S. D. Minteer, A. Dickson, and S. Calabrese Barton, "Markov-state transition path analysis of electrostatic channeling", J. Phys. Chem. C **123**, 15284–15292 (2019).

[76] Y. Meng, D. Shukla, V. S. Pande, and B. Roux, "Transition path theory analysis of c-Src kinase activation", Proc. Natl. Acad. Sci. **113**, 9193–9198 (2016).

[77] R. Elber, J. M. Bello-Rivas, P. Ma, A. E. Cardenas, and A. Fathizadeh, "Calculating iso-committor surfaces as optimal reaction coordinates with milestoning", Entropy **19**, 219 (2017).

[78] R. R. Coifman and S. Lafon, "Diffusion maps", Appl. Comput. Harmon. Anal., Special Issue: Diffusion Maps and Wavelets **21**, 5–30 (2006).

[79] K. Müller and L. D. Brown, "Location of saddle points and minimum energy paths by a constrained simplex optimization procedure", Theor. Chim. Acta **53**, 75–93 (1979).

[80] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers, "Atomic-level characterization of the structural dynamics of proteins", Science **330**, 341–346 (2010).

[81] L. Qiu, S. A. Pabit, A. E. Roitberg, and S. J. Hagen, "Smaller and faster: the 20-residue Trp-cage protein folds in 4 μs", J. Am. Chem. Soc. **124**, 12952–12953 (2002).

[82] J. Juraszek and P. G. Bolhuis, "Sampling the multiple folding mechanisms of Trp-cage in explicit solvent", Proc. Natl. Acad. Sci. **103**, 15859–15864 (2006).

[83] J. Juraszek and P. G. Bolhuis, "Rate constant and reaction coordinate of Trp-cage folding in explicit water", Biophys. J. **95**, 4246–4257 (2008).

[84] F. Marinelli, F. Pietrucci, A. Laio, and S. Piana, "A kinetic model of Trp-cage folding from multiple biased molecular dynamics simulations", PLoS Comput Biol **5**, e1000452 (2009).

[85] N.-j. Deng, W. Dai, and R. M. Levy, "How kinetics within the unfolded state affects protein folding: an analysis based on Markov state models and an ultra-long MD trajectory", J. Phys. Chem. B **117**, 12787–12799 (2013).

[86] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden, "Markov state models based on milestoning", J. Chem. Phys. **134**, 204105 (2011).

[87] M. Weber, K. Fackeldey, and C. Schütte, "Set-free Markov state model building", J. Chem. Phys. **146**, 124133 (2017).

[88] R. Zwanzig, *Nonequilibrium statistical mechanics* (Oxford University Press, 2001).

[89] N. Guttenberg, J. F. Dama, M. G. Saunders, G. A. Voth, J. Weare, and A. R. Dinner, "Minimizing memory as an objective for coarse-graining", J. Chem. Phys. **138**, 094111 (2013).

[90] M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, and M. Parrinello, "PLUMED: a portable plugin for free-energy calculations with molecular dynamics", Comput. Phys. Commun. **180**, 1961–1972 (2009).

[91] G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, and G. Bussi, "PLUMED 2: new feathers for an old bird", Comput. Phys. Commun. **185**, 604–613 (2014).

[92] T. P. consortium, "Promoting transparency and reproducibility in enhanced molecular simulations", Nat. Methods **16**, 670–673 (2019).

[93] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus, "All-atom empirical potential for molecular modeling and dynamics studies of proteins", J. Phys. Chem. B **102**, 3586–3616 (1998).

[94] R. B. Best, X. Zhu, J. Shim, P. E. M. Lopes, J. Mittal, M. Feig, and A. D. MacKerell, "Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain χ_1 and χ_2 dihedral angles", J. Chem. Theory Comput. **8**, 3257–3273 (2012).

[95] J. Huang, S. Rauscher, G. Nawrocki, T. Ran, M. Feig, B. L. de Groot, H. Grubmüller, and A. D. MacKerell, "CHARMM36m: an improved force field for folded and intrinsically disordered proteins", Nat Methods **14**, 71–73 (2017).

[96] J. W. Neidigh, R. M. Fesinmeyer, and N. H. Andersen, "Designing a 20-residue protein", Nat. Struct. Biol. **9**, 425–430 (2002).

[97] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water", J. Chem. Phys. **79**, 926–935 (1983).

[98] S. Jo, T. Kim, V. G. Iyer, and W. Im, "CHARMM-GUI: a web-based graphical user interface for CHARMM", J. Comput. Chem. **29**, 1859–1865 (2008).

[99] J. Lee, X. Cheng, J. M. Swails, M. S. Yeom, P. K. Eastman, J. A. Lemkul, S. Wei, J. Buckner, J. C. Jeong, Y. Qi, S. Jo, V. S. Pande, D. A. Case, C. L. Brooks, A. D. MacKerell, J. B. Klauda, and W. Im, "CHARMM-GUI input generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM simulations using the CHARMM36 additive force field", J. Chem. Theory Comput. **12**, 405–413 (2016).

[100] M. Marchi and P. Ballone, "Adiabatic bias molecular dynamics: a method to navigate the conformational space of complex molecular systems", J. Chem. Phys. **110**, 3697–3702 (1999).

[101] S. Park, T. Kim, and W. Im, "Transmembrane helix assembly by window exchange umbrella sampling", Phys. Rev. Lett. **108**, 108102 (2012).

[102] Y. Sugita, A. Kitao, and Y. Okamoto, "Multidimensional replica-exchange method for free-energy calculations", J. Chem. Phys. **113**, 6042–6051 (2000).

[103] E. H. Thiede, B. Van Koten, J. Weare, and A. R. Dinner, "Eigenvector method for umbrella sampling enables error analysis", J. Chem. Phys. **145**, 084115 (2016).

[104] A. Antoszewski, C.-J. Feng, B. P. Vani, E. H. Thiede, L. Hong, J. Weare, A. Tokmakoff, and A. R. Dinner, "Insulin dissociates by diverse mechanisms of coupled unfolding and unbinding", J. Phys. Chem. B **124**, 5571–5587 (2020).

[105] A. R. Dinner, E. H. Thiede, B. V. Koten, and J. Weare, "Stratification as a general variance reduction method for Markov Chain Monte Carlo", SIAMASA J. Uncertain. Quantif., 1139–1188 (2020).

[106] S. B. Kim, C. J. Dsilva, I. G. Kevrekidis, and P. G. Debenedetti, "Systematic characterization of protein folding pathways using diffusion maps: application to Trp-cage miniprotein", J. Chem. Phys. **142**, 085101 (2015).

[107] E. Thiede, B. Van Koten, and J. Weare, "Sharp entrywise perturbation bounds for Markov chains", SIAM J. Matrix Anal. Appl. **36**, 917–941 (2015).

[108] A. Ma and A. R. Dinner, "Automatic method for identifying reaction coordinates in complex systems", J. Phys. Chem. B **109**, 6769–6779 (2005).

[109] J. Hu, A. Ma, and A. R. Dinner, "A two-step nucleotide-flipping mechanism enables kinetic discrimination of DNA lesions by AGT", Proc. Natl. Acad. Sci. **105**, 4615–4620 (2008).

[110] C. Lorpaiboon, S. C. Guo, J. Strahan, J. Weare, and A. R. Dinner, "Accurate estimates of dynamical statistics using memory", J. Chem. Phys. **160**, 084108 (2024).

[111] W. Liebert and H. G. Schuster, "Proper choice of the time delay for the analysis of chaotic time series", Phys. Lett. A **142**, 107–111 (1989).

[112] H. Arbabi and I. Mezić, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator", SIAM J. Appl. Dyn. Syst. **16**, 2096–2126 (2017).

[113] S. Das and D. Giannakis, "Delay-coordinate maps and the spectra of Koopman operators", J. Stat. Phys. **175**, 1107–1145 (2019).

[114] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, "Time-delay observables for Koopman: theory and applications", SIAM J. Appl. Dyn. Syst. **19**, 886–917 (2020).

[115] H. Mori, "Transport, collective motion, and Brownian motion", Prog. Theor. Phys. **33**, 423–455 (1965).

[116] R. Zwanzig, "From classical dynamics to continuous time random walks", J. Stat. Phys. **30**, 255–262 (1983).

[117] M. Berkowitz, J. D. Morgan, D. J. Kouri, and J. A. McCammon, "Memory kernels from molecular dynamics", J. Chem. Phys. **75**, 2462–2463 (1981).

[118] M. Berkowitz, J. D. Morgan, and J. A. McCammon, "Generalized Langevin dynamics simulations with arbitrary time-dependent memory kernels", J. Chem. Phys. **78**, 3256–3261 (1983).

[119] A. Perico, R. Pratolongo, K. F. Freed, R. W. Pastor, and A. Szabo, "Positional time correlation function for one-dimensional systems with barrier crossing: memory function corrections to the optimized Rouse–Zimm approximation", J. Chem. Phys. **98**, 564–573 (1993).

[120] K. S. Kostov and K. F. Freed, "Mode coupling theory for calculating the memory functions of flexible chain molecules: Influence on the long time dynamics of oligoglycines", J. Chem. Phys. **106**, 771–783 (1997).

[121] K. S. Kostov and K. F. Freed, "Long-time dynamics of met-enkephalin: comparison of theory with Brownian dynamics simulations", Biophys. J. **76**, 149–163 (1999).

[122] H. Lei, N. A. Baker, and X. Li, "Data-driven parameterization of the generalized Langevin equation", Proc. Natl. Acad. Sci. **113**, 14183–14188 (2016).

[123] F. Grogan, H. Lei, X. Li, and N. A. Baker, "Data-driven molecular modeling with the generalized Langevin equation", J. Comput. Phys. **418**, 109633 (2020).

[124] B. J. Berne, J. P. Boon, and S. A. Rice, "On the calculation of autocorrelation functions of dynamical variables", J. Chem. Phys. **45**, 1086–1096 (1966).

[125] J.-P. Boon and S. A. Rice, "Memory effects and the autocorrelation function of a dynamical variable", J. Chem. Phys. **47**, 2480–2490 (1967).

[126] C. Ayaz, L. Tepper, F. N. Brünig, J. Kappler, J. O. Daldrop, and R. R. Netz, "Non-Markovian modeling of protein folding", Proc. Natl. Acad. Sci. **118**, e2023856118 (2021).

[127] O. F. Lange and H. Grubmüller, "Collective Langevin dynamics of conformational motions in proteins", J. Chem. Phys. **124**, 214903 (2006).

[128] H. S. Lee, S.-H. Ahn, and E. F. Darve, "The multi-dimensional generalized Langevin equation for conformational motion of proteins", J. Chem. Phys. **150**, 174113 (2019).

[129] Y. T. Lin, Y. Tian, D. Livescu, and M. Anghel, "Data-driven learning for the Mori–Zwanzig formalism: a generalization of the Koopman learning framework", SIAM J. Appl. Dyn. Syst. **20**, 2558–2601 (2021).

[130] H. Vroylandt, L. Goudenège, P. Monmarché, F. Pietrucci, and B. Rotenberg, "Likelihood-based non-Markovian models from molecular dynamics", Proc. Natl. Acad. Sci. **119**, e2117586119 (2022).

[131] D. Aristoff, M. Johnson, and D. Perez, "Arbitrarily accurate, nonparametric coarse graining with Markov renewal processes and the Mori–Zwanzig formulation", AIP Adv. **13**, 095131 (2023).

[132] Y. T. Lin, Y. Tian, D. Perez, and D. Livescu, "Regression-based projection for learning Mori–Zwanzig operators", SIAM J. Appl. Dyn. Syst. **22**, 2890–2926 (2023).

[133] S. Cao, Y. Qiu, M. L. Kalin, and X. Huang, "Integrative generalized master equation: a method to study long-timescale biomolecular dynamics via the integrals of memory kernels", J. Chem. Phys. **159**, 134106 (2023).

[134] A. J. Dominic III, T. Sayer, S. Cao, T. E. Markland, X. Huang, and A. Montoya-Castillo, "Building insightful, memory-enriched models to capture long-time biochemical processes from short-time simulations", Proc. Natl. Acad. Sci. **120**, e2221048120 (2023).

[135] J. Strahan, S. C. Guo, C. Lorpaiboon, A. R. Dinner, and J. Weare, "Inexact iterative numerical linear algebra for neural network-based spectral estimation and rare-event prediction", J. Chem. Phys. **159**, 014110 (2023).

[136] E. Darve, J. Solomon, and A. Kia, "Computing generalized langevin equations and generalized Fokker–Planck equations", Proc. Natl. Acad. Sci. **106**, 10884–10889 (2009).

[137] J. Cerrillo and J. Cao, "Non-Markovian dynamical maps: numerical processing of open quantum trajectories", Phys. Rev. Lett. **112**, 110401 (2014).

[138] P. Metzner, C. Schütte, and E. Vanden-Eijnden, "Illustration of transition path theory on a collection of simple examples", J. Chem. Phys. **125**, 084110 (2006).

[139] P. Reimann, G. J. Schmid, and P. Hänggi, "Universal equivalence of mean first-passage time and Kramers rate", Phys. Rev. E **60**, R1–r4 (1999).

[140] S. Buchenberg, N. Schaudinnus, and G. Stock, "Hierarchical biomolecular dynamics: picosecond hydrogen bonding regulates microsecond conformational transitions", J. Chem. Theory Comput. **11**, 1330–1336 (2015).

[141] A. Perez, F. Sittel, G. Stock, and K. Dill, "MELD-path efficiently computes conformational transitions, including multiple and diverse paths", J. Chem. Theory Comput. **14**, 2109–2116 (2018).

[142] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: rapid development of high performance algorithms for molecular dynamics", PLOS Comput. Biol. **13**, edited by R. Gentleman, e1005659 (2017).

[143] G. A. Khoury, J. Smadbeck, P. Tamamis, A. C. Vandris, C. A. Kieslich, and C. A. Floudas, "Forcefield_NCAA: ab initio charge parameters to aid in the discovery and design of therapeutic proteins and peptides with unnatural amino acids and their application to complement inhibitors of the compstatin family", ACS Synth. Biol. **3**, 855–869 (2014).

[144] C. W. Hopkins, S. Le Grand, R. C. Walker, and A. E. Roitberg, "Long-time-step molecular dynamics through hydrogen mass repartitioning", J. Chem. Theory Comput. **11**, 1864–1874 (2015).

[145] H. Nguyen, D. R. Roe, and C. Simmerling, "Improved generalized Born solvent model parameters for protein simulations", J. Chem. Theory Comput. **9**, 2020–2034 (2013).

[146] F. Sittel, T. Filk, and G. Stock, "Principal component analysis on a torus: theory and application to protein dynamics", J. Chem. Phys. **147**, 244101 (2017).

[147] J. Strahan, J. Finkel, A. R. Dinner, and J. Weare, "Predicting rare events using neural networks and short-trajectory data", J. Comput. Phys. **488**, 112152 (2023).

[148] R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, and E. S. Shakhnovich, "On the transition coordinate for protein folding", J. Chem. Phys. **108**, 334–350 (1998).

[149] G. Hummer, "From transition paths to transition states and rate coefficients", J. Chem. Phys. **120**, 516–523 (2004).

[150] W. E, W. Ren, and E. Vanden-Eijnden, "Transition pathways in complex systems: reaction coordinates, isocommittor surfaces, and transition tubes", Chem. Phys. Lett. **413**, 242–247 (2005).

[151] W. E and E. Vanden-Eijnden, "Transition-path theory and path-finding algorithms for the study of rare events", Annu. Rev. Phys. Chem. **61**, 391–420 (2010).

[152] A. M. Berezhkovskii and A. Szabo, "Diffusion along the splitting/commitment probability reaction coordinate", J. Phys. Chem. B **117**, 13115–13119 (2013).

[153] P. V. Banushkina and S. V. Krivov, "Optimal reaction coordinates", WIREs Comput. Mol. Sci. **6**, 748–763 (2016).

[154] B. Peters and B. L. Trout, "Obtaining reaction coordinates by likelihood maximization", J. Chem. Phys. **125**, 054108 (2006).

[155] B. Peters, G. T. Beckham, and B. L. Trout, "Extensions to the likelihood maximization approach for finding reaction coordinates", J. Chem. Phys. **127**, 034109 (2007).

[156] H. Jung, R. Covino, A. Arjun, C. Leitold, C. Dellago, P. G. Bolhuis, and G. Hummer, "Machine-guided path sampling to discover mechanisms of molecular self-organization", Nat. Comput. Sci. **3**, 334–345 (2023).

[157] H. Li, Y. Khoo, Y. Ren, and L. Ying, "A semigroup method for high dimensional committor functions based on neural network", in Proc. 2nd Math. Sci. Mach. Learn. Conf. (Apr. 2022), pp. 598–618.

[158] H. Chen, B. Roux, and C. Chipot, "Discovering reaction pathways, slow variables, and committor probabilities with machine learning", J. Chem. Theory Comput. **19**, 4414–4426 (2023).

[159] D. Aristoff, M. Johnson, G. Simpson, and R. J. Webber, "The fast committor machine: interpretable prediction with kernels", J. Chem. Phys. **161**, 084113 (2024).

[160] L. Evans, M. K. Cameron, and P. Tiwary, "Computing committors via Mahalanobis diffusion maps with enhanced sampling data", J. Chem. Phys. **157**, 214107 (2022).

[161] L. Evans, M. K. Cameron, and P. Tiwary, "Computing committors in collective variables via Mahalanobis diffusion maps", Appl. Comput. Harmon. Anal. **64**, 62–101 (2023).

[162] Z. Song, M. K. Cameron, and H. Yang, "A finite expression method for solving high-dimensional committor problems", SIAM J. Sci. Comput. **47**, C1–C21 (2025).

[163] A. R. Mitchell and G. M. Rotskoff, "Committor guided estimates of molecular transition rates", J. Chem. Theory Comput. **20**, 9378–9393 (2024).

[164] D. Lucente, J. Rolland, C. Herbert, and F. Bouchet, "Coupling rare event algorithms with data-based learned committor functions using the analogue Markov chain", J. Stat. Mech. Theory Exp. **2022**, 083201 (2022).

[165] D. Lucente, C. Herbert, and F. Bouchet, "Committor functions for climate phenomena at the predictability margin: the example of El Niño-Southern Oscillation in the Jin and Timmermann model", J. Atmos. Sci. **79**, 2387–2400 (2022).

[166] V. Jacques-Dumas, R. M. van Westen, F. Bouchet, and H. A. Dijkstra, "Data-driven methods to estimate the committor function in conceptual ocean models", Nonlinear Process. Geophys. **30**, 195–216 (2023).

[167] G. M. Rotskoff, A. R. Mitchell, and E. Vanden-Eijnden, "Active importance sampling for variational objectives dominated by rare events: consequences for optimization and generalization", in Proc. 2nd Math. Sci. Mach. Learn. Conf. Vol. 145, edited by J. Bruna, J. Hesthaven, and L. Zdeborova, Proceedings of Machine Learning Research (Aug. 2022), pp. 757–780.

[168] P. Kang, E. Trizio, and M. Parrinello, "Computing the committor with the committor to study the transition state ensemble", Nat. Comput. Sci. **4**, 451–460 (2024).

[169] Y. Khoo, J. Lu, and L. Ying, "Solving for high-dimensional committor functions using artificial neural networks", Res. Math. Sci. **6**, 1 (2018).

[170] Q. Li, B. Lin, and W. Ren, "Computing committor functions for the study of rare events using deep learning", J. Chem. Phys. **151**, 054112 (2019).

[171] Y. Chen, J. Hoskins, Y. Khoo, and M. Lindsey, "Committor functions via tensor networks", J. Comput. Phys. **472**, 111646 (2023).

[172] B. Roux, "String method with swarms-of-trajectories, mean drifts, lag time, and committor", J. Phys. Chem. A **125**, 7558–7571 (2021).

[173] B. Roux, "Transition rate theory, spectral analysis, and reactive paths", J. Chem. Phys. **156**, 134111 (2022).

[174] Z. He, C. Chipot, and B. Roux, "Committor-consistent variational string method", J. Phys. Chem. Lett. **13**, 9263–9271 (2022).

[175] G. R. Bowman, V. S. Pande, and F. Noé, eds., *An introduction to Markov state models and their application to long timescale molecular simulation*, Vol. 797, Advances in experimental medicine and biology (Springer Netherlands, Dordrecht, 2014).

[176] K. Everett, L. Xiao, M. Wortsman, A. A. Alemi, R. Novak, P. J. Liu, I. Gur, J. Sohl-Dickstein, L. P. Kaelbling, J. Lee, and J. Pennington, *Scaling exponents across parameterizations and optimizers*, July 2024.

[177] S. Ioffe and C. Szegedy, *Batch normalization: accelerating deep network training by reducing internal covariate shift*, Mar. 2015.

[178] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer normalization*, July 2016.

[179] S. Kumar, H. Marklund, and B. V. Roy, *Maintaining plasticity in continual learning via regenerative regularization*, Oct. 2024.

[180] B. Barua, J. C. Lin, V. D. Williams, P. Kummler, J. W. Neidigh, and N. H. Andersen, "The Trp-cage: optimizing the stability of a globular miniprotein", Protein Eng. Des. Sel. **21**, 171–185 (2008).

[181] E. Wang, P. Tao, J. Wang, and Y. Xiao, "A novel folding pathway of the villin headpiece subdomain HP35", Phys. Chem. Chem. Phys. **21**, 18219–18226 (2019).

[182] J. Kubelka, T. K. Chiu, D. R. Davies, W. A. Eaton, and J. Hofrichter, "Sub-microsecond protein folding", J. Mol. Biol. **359**, 546–553 (2006).

[183] B. Liu, S. Cao, J. G. Boysen, M. Xue, and X. Huang, "Memory kernel minimization-based neural networks for discovering slow collective variables of biomolecular dynamics", Nat. Comput. Sci., 1–10 (2025).

[184] X. Cheng and J. Weare, "The surprising efficiency of temporal difference learning for rare event prediction", in Thirty-Eighth Annu. Conf. Neural Inf. Process. Syst. (2024).

[185] A. Antoszewski, C. Lorpaiboon, J. Strahan, and A. R. Dinner, "Kinetics of phenol escape from the insulin R_6 hexamer", J. Phys. Chem. B **125**, 11637–11649 (2021).

[186] Z. Pengmei, C. Lorpaiboon, S. C. Guo, J. Weare, and A. R. Dinner, "Using pretrained graph neural networks with token mixers as geometric featurizers for conformational dynamics", J. Chem. Phys. **162**, 044107 (2025).

[187] R. T. McGibbon, B. E. Husic, and V. S. Pande, "Identification of simple reaction coordinates from complex dynamics", J. Chem. Phys. **146**, 044109 (2017).

[188] L. Busto-Moner, C.-J. Feng, A. Antoszewski, A. Tokmakoff, and A. R. Dinner, "Structural ensemble of the insulin monomer", Biochemistry **60**, 3125–3136 (2021).

[189] J. Finkel, R. J. Webber, E. P. Gerber, D. S. Abbot, and J. Weare, "Learning forecasts of rare stratospheric transitions from short simulations", Mon. Weather Rev. **149**, 3647–3669 (2021).

[190] S. C. Guo, R. Shen, B. Roux, and A. R. Dinner, "Dynamics of activation in the voltage-sensing domain of ci-VSP", bioRxiv (2022).

[191] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis", in Proc. 30th Int. Conf. Mach. Learn. (2013), pp. 1247–1255.

[192] C. Wehmeyer and F. Noé, "Time-lagged autoencoders: deep learning of slow collective variables for molecular kinetics", J. Chem. Phys. **148**, 241703 (2018).

[193] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics", Nat. Commun. **9**, 4950 (2018).

[194] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, "Unsupervised learning methods for molecular simulation data", Chem. Rev. **121**, 9722–9758 (2021).

[195] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition (The MIT Press, Cambridge, Massachusetts, 2018).

[196] J. Wen, B. Dai, L. Li, and D. Schuurmans, "Batch stationary distribution estimation", in Proc. 37th Int. Conf. Mach. Learn. ICML'20 (2020), pp. 10203–10213.

[197] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. (The Johns Hopkins University Press, 1996).

[198] S. V. Krivov, "On reaction coordinate optimality", J. Chem. Theory Comput. **9**, 135–146 (2013).

[199] P. Collett, S. Martinez, and J. San Martin, *Quasi-stationary distributions*, Probability and its Applications (Springer Berlin Heidelberg, Oct. 2012).

[200] H. Jung, R. Covino, and G. Hummer, "Artificial intelligence assists discovery of reaction coordinates and mechanisms from molecular dynamics simulations", ArXiv Prepr. ArXiv190104595 (2019).

[201] A. Chattopadhyay, E. Nabizadeh, and P. Hassanzadeh, "Analog forecasting of extreme-causing weather patterns using deep learning", J. Adv. Model. Earth Syst. **12**, e2019MS001958 (2020).

[202] G. Miloshevich, B. Cozian, P. Abry, P. Borgnat, and F. Bouchet, "Probabilistic forecasts of extreme heatwaves using convolutional neural networks in a regime of lack of data", Phys. Rev. Fluids **8**, 040501 (2023).

[203] C. Lorpaiboon, J. Weare, and A. R. Dinner, "Augmented transition path theory for sequences of events", J. Chem. Phys. **157**, 094115 (2022).

[204] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization", ArXiv Prepr. ArXiv14126980 (2014).

[205] B. P. Vani, J. Weare, and A. R. Dinner, "Computing transition path theory quantities with trajectory stratification", J. Chem. Phys. **157**, 034106 (2022).

[206] J. Finkel, D. S. Abbot, and J. Weare, "Path properties of atmospheric transitions: illustration with a low-order sudden stratospheric warming model", J. Atmos. Sci. **77**, 2327–2347 (2020).

[207] P. Miron, F. J. Beron-Vera, L. Helfmann, and P. Koltai, "Transition paths of marine debris and the stability of the garbage patches", Chaos **31**, 033101 (2021).

[208] J. Finkel, E. P. Gerber, D. S. Abbot, and J. Weare, "Revealing the statistics of extreme events hidden in short weather forecast data", AGU Adv. **4**, e2023AV000881 (2023).

[209] J. Finkel, R. J. Webber, E. P. Gerber, D. S. Abbot, and J. Weare, "Data-driven transition path analysis yields a statistical understanding of sudden stratospheric warming events in an idealized model", J. Atmos. Sci. **80**, 519–534 (2023).

[210] E. Xia and M. Wainwright, "Krylov–Bellman boosting: super-linear policy evaluation in general state spaces", in Proc. 26th Int. Conf. Artif. Intell. Stat. Vol. 206, edited by F. Ruiz, J. Dy, and J.-W. van de Meent, Proceedings of Machine Learning Research (Apr. 2023), pp. 9137–9166.

[211] P. Hänggi, P. Talkner, and M. Borkovec, "Reaction-rate theory: fifty years after Kramers", Rev. Mod. Phys. **62**, 251–341 (1990).

[212] B. Peters, *Reaction rate theory and rare events* (Elsevier, 2017).

[213] E. Vanden-Eijnden and M. Venturoli, "Exact rate calculations by trajectory parallelization and tilting", J. Chem. Phys. **131**, 044120 (2009).

[214] A. Dickson, A. Warmflash, and A. R. Dinner, "Separating forward and backward pathways in nonequilibrium umbrella sampling", J. Chem. Phys. **131**, 154104 (2009).

[215] L. Helfmann, E. Ribera Borrell, C. Schütte, and P. Koltai, "Extending transition path theory: periodically driven and finite-time dynamics", J. Nonlinear Sci. **30**, 3321–3366 (2020).