Physics-informed deep learning for protein dynamics

By

Bojun Liu

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Chemistry)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: Dec 11th, 2025

The dissertation is approved by the following members of the Final Oral Committee:
Xuhui Huang (Advisor), Professor, Chemistry
J. R. Schmidt, Professor, Chemistry
Reid C. Van Lehn, Associate Professor, Chemical & Biological Engineering
Sharon Y. Li, Associate Professor, Computer Sciences

To my dearest parents and grandparents, I love you deeply; you are the whole meaning of my life.

Acknowledgments

"What I am, you helped me to be."

– Anonymous

I could not have reached this point without the guidance and support of the many people who have helped me throughout this journey.

First and foremost, I want to express my deepest gratitude to my advisor, Dr. Xuhui Huang. Throughout my PhD, he consistently held me to the highest standards and expectations that, while challenging, helped me grow tremendously as a researcher. At the same time, whenever I faced difficulties, he always encouraged me and provided unwavering support. I still vividly remember our first meeting, when he placed his trust in me and welcomed me into his group. At that moment, I could not have imagined how profoundly joining Xuhui's group would change the course of my life.

I am also grateful to all my committee members, Dr. J. R. Schmidt, Dr. Reid C. Van Lehn, and Dr. Sharon Y. Li, for making the time to attend my defense. They have continued to offer me valuable suggestions and guidance that have enriched my dissertation and improved my research works.

During my Ph.D., I have been fortunate to work with an outstanding group of colleagues. Especially, I want to thank Dr. Siqin Cao, the most senior member of our group, who has always been willing to provide me with scientific mentorship as well as thoughtful advice in my personal life; Dr. Mingyi Xue, whose collaboration and insightful discussions have been a constant source of support and inspiration throughout my research; and Dr. Yunrui Qiu, my old friend since undergraduate years, who has always been there with valuable guidance, support, and encouragement. I would

also like to thank Dr. Ilona Unarta, Andrew Kai-hei Yik, Eshani Goonetilleke, and Yuzhe Zhang for their warmth and care during my difficult times, as well as Yichong Lao, Michael O'Connor, Michael Kalin, Zige Liu, Chengwei Dong, Jordan Boysen and Bo Zhu for their discussions and support.

I also want to thank my collaborators, Dr. Sharon Y. Li and Dr. Xuefeng Du from the Department of Computer Sciences. Their pioneering work has greatly inspired my own research. In addition, their lecture on modern deep learning, where Sharon served as the lecturer and Xuefeng as the teaching assistant, was the best graduate course I have ever taken and has significantly influenced my future research.

It is not only the people I met during my Ph.D. who have shaped me. I am deeply grateful to Dr. Zhonghuai Hou for kindly recognizing and supporting me when I first attempted to shift my research from experimental chemistry to theoretical chemistry as an undergraduate. I also want to thank my best friend, Dr. Banlan Chi, who has been my closest confidant and strongest supporter throughout these years. In addition, I want to thank my high school teachers, particularly my Chemistry Olympiad mentors, whose instruction and mentorship fostered a deep appreciation for chemistry and a lasting commitment to learning.

Last, I want to express my deepest gratitude and love to my family. Their unconditional love is the foundation of everything I am and everything I have accomplished. They are my greatest source of strength, my constant inspiration, and truly the most important part of my life. Without them, none of this would have been possible.

Contents

List of Figures

Abstract

While AlphaFold2 has revolutionized protein structure prediction, it does not fully address the protein folding problem. Proteins are inherently dynamic, and uncovering the mechanisms by which they switch among distinct shapes, known as conformational changes, is crucial for understanding many biological functions. Molecular dynamics (MD) simulations provide a robust tool to study protein dynamics at atomistic details; nevertheless, extracting mechanistic insight from high-dimensional trajectories and bridging the timescale gap between femtosecond integration steps and millisecond (or longer) biological processes remain major challenges.

To address this challenge, this thesis develops physics-informed deep learning frameworks that leverage principles from statistical mechanics together with modern deep learning to investigate protein dynamics. Specifically, we present GraphVAMP-nets, a geometric deep learning approach for kinetic modeling of multi-body systems, TS-DAR, an OOD-detection-inspired framework that learns meaningful hyperspherical latent representations and enables simultaneous identification of transition states across multiple free-energy barriers, and MEMnets, a deep learning method which extends Markovian dynamic models to the non-Markovian regime by explicitly incorporating memory effects for representation learning of kinetic data. Overall, this thesis opens new directions for understanding biomolecular mechanism and provides a pathway toward more effective biological interventions, including drug discovery. By integrating deep learning with physical principles, it demonstrates how elegant network design can extend applicability to complex problems in chemistry, while new theoretical insights from statistical mechanics can, in turn, guide the development of machine learning. Taken together, these developments bridge statistical mechanics and modern deep learning, establishing a unified perspective that is essential for uncovering the microscopic principles governing biomolecular systems.

List of Publications

1. Liu, B., Cao, S., Boysen, J.G., Xue, M., Huang, X. (2025). Memory kernel minimization-based neural networks for discovering slow collective variables of biomolecular dynamics. *Nature Computational Science*, 1-10.
2. Liu, B., Boysen, J. G., Unarta, I. C., Du, X., Li, Y., Huang, X. (2025). Exploring transition states of protein conformational changes via out-of-distribution detection in the hyperspherical latent space. *Nature Communications*, 16, 349.
3. Liu, B., Xue, M., Qiu, Y., Konovalov, K., O'Connor, M., Huang, X. (2023). GraphVAMPnets for uncovering slow collective variables of self-assembly dynamics. *The Journal of Chemical Physics*, 159, 094901.
4. Liu, B., Qiu, Y., Goonetilleke, E. C., Huang, X. (2022). Kinetic network models to study molecular self-assembly in the wake of machine learning. *MRS Bulletin*, 47, 958-966.
5. Xue, M., Liu, B., Cao, S., Huang, X. (2025). FeatureDock for protein-ligand docking guided by physicochemical feature-based local environment learning using transformer. *npj Drug Discovery*, 2, 4.
6. Goonetilleke, E. C., Liu, B., Wu, Y., O'Connor, M. S., Huang, X. (2025). A Practical Guide to Transition State Analysis in Biomolecular Simulations with TS-DAR. *The Journal of Physical Chemistry B*, 129, 47, 12133–12145.
7. Xue, M., Liu, B., Huang, X. (2025). Machine Learning in Chemistry: A Data Centred, Hands-on Introductory Machine Learning Course for Undergraduate Students. *ChemRxiv*.

8. Cao, S., Nüske, F., Liu, B., Soley, M., Huang, X. (2025). AMUSET-TICA: A Tensor-Based Approach for Identifying Slow Collective Variables in Biomolecular Dynamics. *Journal of Chemical Theory and Computation*, 21, 4855-4866.

9. Liu, Z., Cao, S., Liu, B., Huang, X. (2025). Revealing Competing Kinetic Pathways in Amphiphilic Pt(II) Complex Self-Assembly via Deep Learning with Graph Neural Networks. *Aggregate*, e70201.

10. Qiu, Y., O'Connor, M. S., Xue, M., Liu, B., Huang, X. (2023). An Efficient Path Classification Algorithm Based on Variational Autoencoder to Identify Metastable Path Channels for Complex Conformational Changes. *Journal of Chemical Theory and Computation*, 19, 4728-4742.

11. Liu, Z., Kalin, M. L., Liu, B., Cao, S., Huang, X. (2023). Kinetic network models to elucidate aggregation dynamics of aggregation-induced emission systems. *Aggregate*, 5, e422.

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Protein Structure Determination since 1960s

Proteins are fundamental biomolecules that orchestrate nearly all aspects of life [20, 104]. They act as enzymes that catalyze biochemical reactions, molecular machines that transmit signals, structural scaffolds that maintain cellular integrity, and regulatory components that control gene expression and metabolism. Given their central role in sustaining cellular function and driving virtually every biological process, elucidating how proteins perform their tasks and respond to their environment remains a central goal in molecular biology and biochemistry.

To uncover the molecular basis of protein function, scientists began investigating their three-dimensional structures more than six decades ago, recognizing that a protein's function is intrinsically determined by its structure. The pioneering determination of the myoglobin and hemoglobin structures in the late 1950s marked the birth of structural biology, revealing the first atomic-level insights into how molecular architecture underlies biological activity [65, 112]. Since then, the field has expanded dramatically, and experimental techniques such as X-ray crystallography [58], nuclear magnetic resonance (NMR) spectroscopy [159], and cryo-electron microscopy (cryo-EM) [3] have revolutionized our ability to determine protein structures with near-atomic precision.

Despite these experimental breakthroughs, determining protein structures at atomic resolution remains time-consuming, technically demanding, and costly. Many proteins are difficult to crystallize, unstable in isolation, or exist as heterogeneous complexes that resist high-resolution characterization. As a result, structural coverage across the proteome remains incomplete [6]. In recent years, the rapid development of artificial intelligence (AI) has transformed this landscape. Data-driven models trained on large structural databases have demonstrated remarkable ability to predict protein folds directly from amino acid sequences. In particular, AlphaFold2 [63] and RoseTTAFold [2] have achieved near-experimental accuracy in native structure prediction, marking a major milestone in computational structural biology and dramatically accelerating the pace of protein discovery. However, proteins are not characterized by a single native structure but by a range of distinct structural states associated with their functions. Recent efforts have therefore sought to extend these AI-based frameworks to represent structural heterogeneity in protein systems. For example, approaches such as AlphaFold-Multimer [44], AlphaFold-MSA clustering [154], DiG [169] and BioEmu [79] are developed to capture alternative conformations and infer the structural diversity of proteins from amino acid sequences.

1.1.2 Conformational Changes and Molecular Dynamics Simulation

While AlphaFold-inspired foundation models have revolutionized protein structure prediction, the *protein folding problem* [37] remains far from fully resolved. Proteins are inherently dynamic, and elucidating the mechanisms underlying their conformational changes is essential for understanding their biological functions. Many biological processes, such as molecular recognition, allosteric regulation, and enzymatic catalysis, are driven by conformational changes that enable proteins to perform specific tasks in a regulated and efficient manner. Such intrinsic flexibility highlights the importance of viewing proteins as dynamic ensembles rather than static structures, where conformational fluctuations and transitions are fundamental to their functional mechanisms.

To investigate protein conformational changes, molecular dynamics (MD) simulations [99] provide a physically grounded framework for exploring biomolecular behavior at atomic resolution. Unlike those AI-based models that depend primarily on evolutionary information of protein sequences and are purely data-driven, MD simulations are rooted in the fundamental principles of statistical mechanics. They compute the time evolution of atomic motions by integrating Newton's equations of motion based on the potential energy function, and simulate trajectories that sample the configurational space accessible under a certain thermodynamic ensemble. Within this framework, the probability of observing a specific conformation is determined by its Boltzmann weight, allowing one to derive equilibrium properties as ensemble averages over the simulated configurations. More importantly, the time-resolved nature of MD trajectories enables the characterization of dynamic properties, such as transition pathways, mean first passage time, and relaxation timescales of different dynamic patterns. Owing to their high spatial and temporal resolution, MD simulations can provide atomistic insights into the detailed mechanisms of protein motions that underlie relevant biological processes.

1.1.3 Kinetic Modeling for Protein Dynamics

Although MD simulations provide high-resolution, time-resolved trajectories of atomic motions, significant challenges remain in directly leveraging these simulations to elucidate the protein dynamics. First, raw MD trajectories evolve in a high-dimensional configurational space consisting of the three-dimensional coordinates of thousands to millions of atoms. As a result, it is not straightforward to interpret the underlying kinetic mechanisms or to extract thermodynamic and kinetic properties from such complex data. Second, MD simulations are computationally demanding due to the intrinsic separation of timescales in biomolecular systems. The integration time step is typically on the order of one to two femtoseconds to resolve the fastest bond and angle vibrations, yet biologically relevant conformational changes often occur on millisecond or longer. Bridging this timescale gap requires propagating billions to trillions of integration steps, leading to substantial computational cost even on modern high-performance computing architectures [85].

To overcome these limitations, kinetic modeling frameworks such as Markov state models (MSMs) [73, 89, 116, 153] have been developed to extract relevant thermodynamic and kinetic properties and to predict long-timescale dynamics by integrating ensembles of short, parallel MD trajectories of protein conformational changes. The central idea of MSMs is to coarse-grain the continuous dynamics in the high-dimensional configurational space into a series of discrete transitions among a finite number of metastable states separated by free-energy barriers. The transition probabilities between these states can be estimated directly by counting the observed transitions in the MD data. In this way, MSMs circumvent the need for a single, sufficiently long trajectory to sample all states, instead reconstructing global kinetics from multiple shorter simulations. Once the transition probability matrix is obtained, thermodynamic quantities, such as the equilibrium distribution, and kinetic observables, such as transition pathways and mean first-passage times, can be computed. Furthermore, the master-equation formalism allows the propagation of these state-to-state transitions to predict the long-time evolution of the system.

Despite their success [5, 14, 16, 26–28, 32, 45, 59, 66, 71, 72, 76, 96, 114, 133, 134, 136, 147, 148], traditional MSMs face fundamental challenges when applied to complex biomolecular systems. Constructing an MSM hinges on partitioning the configurational landscape into kinetically meaningful states, yet defining such states in a continuous, high-dimensional space remains inherently ambiguous and system dependent. In practice, this process involves a series of heuristic operations, including feature selection, dimensionality reduction, clustering, and lumping, all of which require substantial domain expertise and subjective judgment, often introducing bias and inconsistency into model quality. The challenge becomes even more pronounced in multi-body systems, where the relevant motions involve collective rearrangements that are invariant under permutations of identical particles and cannot be adequately represented by simple geometric descriptors such as pairwise distances. At the same time, MSMs have inherent methodological limitations. For example, MSMs emphasize transitions among metastable states while neglecting short-lived transition states that are critical for elucidating reaction mechanisms and identifying rate-limiting steps. In addition, the underlying Markovian assumption in MSMs that future dynamics depend solely on the current state at a fixed lag time can break down in large systems with memory effects, leading to distorted predictions of long-timescale dynamics. Collectively, these limitations highlight the need for next-generation modeling frameworks, capable of learning meaningful latent representations, resolving transition states, and capturing memory effects beyond the Markovian regime of protein dynamics.

1.2 Contributions and Roadmap

1.2.1 Major Advances

Given the pressing importance of understanding protein conformational changes and the existing challenges in both AI-based and physics-driven computational approaches, this thesis develops physics-informed deep learning frameworks that combine the rigor of statistical mechanics with the flexibility of modern deep learning to investigate protein dynamics. Specifically, my major contributions include developing a geometric deep learning framework that generalizes traditional kinetic modeling from single-protein systems to complex multi-body assemblies [90]; introducing an out-of-distribution (OOD) detection-inspired deep learning framework that enables consistent characterization across multiple levels of resolution, from coarse-grained metastable states to high-resolution transient transition states in biomolecular dynamics [87]; and formulating a new theoretical framework grounded in statistical mechanics that extends conventional Markovian kinetic models to the non-Markovian regime by explicitly incorporating memory effects for representation learning of time-series data [88].

The research outcomes [87–90] of this thesis have been published in leading journals across chemistry, physics, and AI, including *Nature Communications*, *Nature Computational Science*, *Journal of Chemical Physics*, *MRS Bulletin*, etc. These works have generated substantial interest in both the computational chemistry and machine learning communities, influencing ongoing developments in kinetic modeling, representation learning, trustworthy AI and drug discovery. My work demonstrates how principled, theory-informed deep learning can bridge traditional boundaries between chemistry, physics, and computer science, paving the way toward a new era of AI-driven discovery in the molecular sciences.

1.2.2 Structure of the Thesis

The remainder of this thesis is organized as follows.

To begin with, **chapter 2** (Markov State Models for Kinetic Modeling of Biomolecular Dynamics) provides a comprehensive overview of MSMs as a foundation for kinetic modeling in biomolecular systems. It introduces the theoretical principles underlying MSMs and the pipeline of their practical construction, and discusses their historical development, advantages, and inherent limitations when applied to complex biological processes.

Then, **chapter 3** (Learning Geometry from Symmetry — *GraphVAMPnets for Uncovering Slow Collective Variables of Multi-Body Systems*) presents the design and implementation of the GraphVAMPnets framework [90]. This chapter elaborates on its neural network architecture, symmetry-preserving design, and variational optimization scheme for capturing slow dynamics, demonstrating how GraphVAMPnets extract kinetic geometries from molecular symmetry and extend traditional MSMs to efficiently model the dynamics of multi-body systems.

Next, **chapter 4** (Revealing Transition from Uncertainty — *Exploring Transition States of Protein Conformational Changes via Out-of-Distribution Detection in a Hyperspherical Latent Space*) introduces the TS-DAR, a novel deep learning framework that can simutaneously and automatically identify all transition states across multiple free energy barrier in a latent hypersphere [87]. TS-DAR leverages the concept of OOD detection from trustworthy AI, recognizing that transition states can be viewed as OOD samples exhibiting a distributional shift relative to the in-distribution data representing metastable states. This work is the first to translate the idea of OOD detection into the domain of protein dynamics through quantitative uncertainty measures, illustrating how a simple yet principled architectural design can extend the applicability of deep learning in chemistry. Furthermore, **section 4.5** presents AutoDAR (Autonomous Dynamics Analysis and Routing), an agentic extension that integrates TS-DAR with LLM-based multi-agent orchestration to support user-friendly, natural-language-driven kinetic modeling workflows. AutoDAR also provides a foundation for closed-loop modeling and adaptive sampling, in which TS-DAR-derived OOD scores can be used to automatically select informative conformations to initiate new simulations and iteratively improve rare-event sampling.

The previous chapters focus on Markovian frameworks, while **chapter 5** (Introduction to Non-Markovian Dynamic Modeling of Protein Dynamics) provides a conceptual bridge from Markovian to non-Markovian modeling regimes. It revisits the generalized master equation (GME) formalism as a theoretical basis for incorporating memory effects into coarse-grained dynamics, and discusses how non-Markovian frameworks can more accurately capture long-timescale dynamics in biomolecular systems.

Last, **chapter 6** (Deriving Simplicity from Memory —*Memory Kernel Minimization-based Neural Networks for Uncovering Slow Collective Variables of Biomolecular Dynamics*) proposes the MEMnets framework [88]. Building upon the GME formalism introduced in the previous chapter, MEMnets establish a theoretical framework for quantitatively evaluating the total memory effects embedded in the dynamics of latent continuous space and formulate a neural approach that minimizes these memory effects to identify optimal collective variables, yielding a concise and physically interpretable representation of time-series data. This chapter demonstrates how explicitly modeling memory effects can lead to simpler yet more accurate dynamical descriptions and provides a principled pathway toward bridging statistical mechanics and deep learning.

Chapter 2

Markov State Models for Kinetic Modeling of Biomolecular Dynamics

2.1 Introduction

The dynamics of biomolecular conformational changes play an important role in many biological functions. Molecular dynamics (MD) simulations provide a powerful computational approach that complements experimental techniques in studying such processes. By integrating Newton's equations of motion, MD simulations generate atomistic trajectories that, in principle, contain all the information required to describe the underlying molecular dynamics. However, extracting interpretable kinetic models from the high-dimensional and noisy data produced by MD remains a major challenge. Moreover, the characteristic timescales of biologically relevant events (often milliseconds or longer) remain difficult to access routinely with all-atom MD simulations.

Over the past two decades, Markov state models (MSMs) [73, 89, 116, 153] have emerged as a powerful and conceptually elegant framework for studying biomolecular dynamics [5, 14, 16, 26–28, 32, 45, 59, 66, 71, 72, 76, 96, 114, 133, 134, 136, 147, 148]. By partitioning the conformational space into metastable states and estimating the transition probabilities between them, MSMs provide a discrete-state representation of the system's dynamics that is both interpretable and amenable to quantitative analysis. MSMs enable the computation of equilibrium populations, mean-first-passage times, and dominant kinetic pathways directly from simulation data, establishing a rigorous connection between microscopic sampling and macroscopic thermodynamics and kinetics.

Despite their success, MSMs also face intrinsic challenges when applied to complex biological systems. Constructing a reliable MSM requires defining an appropriate set of features, selecting relevant collective variables (CVs), and discretizing a continuous free energy landscape into kinetically meaningful states. Each of these steps introduces potential sources of ambiguity and may compromise model accuracy.

This chapter provides an overview of the theoretical foundation and practical implementation of MSMs for kinetic modeling of biomolecular dynamics. We begin by introducing the operator-theoretic formulation of molecular kinetics and its discrete approximations that lead to the master equation framework. We then summarize the standard workflow for MSM construction—from feature selection and dimensionality reduction to clustering and kinetic lumping—and discuss how each step affects model fidelity. Finally, we highlight the strengths and limitations of conventional MSMs, motivating the development of advanced approaches such as geometric, reliable and non-Markovian deep learning frameworks presented in the following chapters.

2.2 Theoretical Basis

2.2.1 Transfer Operator in Full Configuration Space

The theoretical foundation of molecular dynamics can be cast in an operator-theoretic form that links microscopic equations of motion to the time evolution of probability densities and observables. Let $\mathbf{x} \in \Omega$ denote the configurational coordinates of the system and $p_t(\mathbf{x})$ the probability density at time t. For Hamiltonian (Liouville) or stochastic dynamics (Fokker–Planck /Langevin), the evolution over a finite lag time τ can be written as the action of the *forward operator* $\mathcal{P}_\tau$:

$$p_{t+\tau}(\mathbf{x}) = \mathcal{P}_\tau \circ p_t(\mathbf{x}) = \int_\Omega p(\mathbf{y},\mathbf{x};\tau)\, p_t(\mathbf{y})\, d\mathbf{y} \tag{2.1}$$

where $\mathcal{P}_\tau$ transports the ensemble densities forward by τ and thereby encapsulates all dynamical information, and $p(\mathbf{y},\mathbf{x};\tau)$ denotes the transition density, the conditional probability density of being $\mathbf{x}$ at time $t+\tau$, given $\mathbf{y}$ at time t.

In equilibrium systems, there always exists a stationary distribution $\pi(\mathbf{x})$, which satisfies $\int_\Omega p(\mathbf{y},\mathbf{x};\tau)\,\pi(\mathbf{y})\,d\mathbf{y} = \pi(\mathbf{x})$. Equivalently, one may normalize densities by stationary distribution and define the *transfer operator* $\mathcal{T}_\tau := M_\pi^{-1}\mathcal{P}_\tau M_\pi$, where $(M_\pi u)(\mathbf{x}) = \pi(\mathbf{x})u(\mathbf{x})$, so that

$$(\mathcal{T}_\tau u)(\mathbf{x}) \;=\; \frac{1}{\pi(\mathbf{x})}\int_\Omega p(\mathbf{y},\mathbf{x};\tau)\,u(\mathbf{y})\,\pi(\mathbf{y})\,d\mathbf{y} \tag{2.2}$$

On the dual side, the *Koopman* (*backward*) *operator* $\mathcal{K}_\tau$ acts on observables via

$$(\mathcal{K}_\tau g)(\mathbf{x}) \;=\; \mathbb{E}[\,g(\mathbf{x}_{t+\tau})\,|\,\mathbf{x}_t = \mathbf{x}\,] \tag{2.3}$$

Under microscopic reversibility (detailed balance with respect to π), $\mathcal{T}_\tau$ is self-adjoint to $\mathcal{K}_\tau$, admitting a real spectrum $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots$ with orthonormal eigenfunctions $\{\psi_i\}$.

Given the above definition, there are several important properties about $\mathcal{T}_\tau$ as in the following.

- The eigenfunctions $\{\psi_i\}$ of $\mathcal{T}_\tau$ construct a complete Hilbert space, where the inner product is weighted by the stationary distribution $\pi(\mathbf{x})$. The first eigenpair is $(\lambda_1, \psi_1) = (1, \mathbf{1})$, and the rest of nontrivial eigenpairs characterize relaxation processes with implied timescales $t_i = -\tau/\ln\lambda_i$.
- $\mathcal{T}_\tau$ satisfies the *Chapman–Kolmogorov* (CK) relation, i.e., the semigroup property $\mathcal{T}_{\tau_1+\tau_2} = \mathcal{T}_{\tau_1}\mathcal{T}_{\tau_2}$, and the dynamics can be propagated to arbitrary long times, i.e., $u_{t+k\tau}(\mathbf{x}) = (\mathcal{T}_\tau)^k \circ u_t(\mathbf{x})$.

Therefore, for an arbitrary dynamic process governed by $\mathcal{T}_\tau$, it can be decomposed as a linear combination of the orthogonal eigenfunctions $\{\psi_i\}$ (or called dynamic modes) of $\mathcal{T}_\tau$:

$$\begin{aligned} u_{t+k\tau}(\mathbf{x}) &= (\mathcal{T}_\tau)^k \circ u_t(\mathbf{x}) \\ &= \sum_{i=1} \lambda_i^k \langle u_t, \psi_i\rangle_\pi \psi_i(\mathbf{x}) \\ &= 1 + \sum_{i=2} \exp(-\frac{k\tau}{t_i})\langle u_t, \psi_i\rangle_\pi \psi_i(\mathbf{x}) \end{aligned} \tag{2.4}$$

where u_t is normalized so that $\langle u_t, \psi_1 \rangle_\pi = 1$, and the first term reduces to 1.

This decomposition shows how any normalized distribution relaxes to equilibrium as time evolves: the nontrivial modes ($\{\psi_i\}_{i\geq 2}$) decay exponentially with rates set by their implied timescales $t_i = -\tau/\ln\lambda_i$. Modes with larger eigenvalues (equivalently, larger t_i) decay more slowly and thus dominate long-time behavior. When a spectral gap separates a few leading nontrivial eigenfunctions from the remainder, these modes collectively capture the dominant slow processes and furnish a low-dimensional kinetic manifold. Their level sets also align with metastable regions, whereas changes in sign or magnitude delineate transition patterns.

2.2.2 Transition Probability Matrix and the Master Equation

The above operator-theoretic formulation introduces the *transfer operator* $\mathcal{T}_\tau$ that governs the dynamics in the full-configuration space and encapsulates the full dynamic information of the system. However, it's untractable to build a kinetic model that operates at such a high-dimensional continuous space. In practice, since there are typically a small number of slowly decaying modes that dominate long-time behavior in protein dynamics, we can pass from the continuous-space operator to a finite-state approximation by partitioning Ω into disjoint sets $\{S_1, \ldots, S_n\}$ and tracking state-occupancy probabilities

$$p_i(t) = \int_{S_i} p_t(\mathbf{x})\, d\mathbf{x}, \qquad \pi_i = \int_{S_i} \pi(\mathbf{x})\, d\mathbf{x}, \qquad i = 1, \ldots, n \tag{2.5}$$

The state approximation of $\mathcal{T}_\tau$, i.e., the τ-lag transition probability matrix (TPM) $\mathbf{T}(\tau) = [T_{ij}(\tau)]$, is defined by averaging the transition density over the sources /target sets:

$$T_{ij}(\tau) = \frac{1}{\pi_i} \int_{S_i} \int_{S_j} p(\mathbf{y}, \mathbf{x}; \tau)\, \pi(\mathbf{y})\, d\mathbf{x}\, d\mathbf{y}, \qquad T_{ij}(\tau) \geq 0, \quad \sum_{j=1}^{n} T_{ij}(\tau) = 1 \tag{2.6}$$

With TPM, the coarse-grained dynamics then satisfies the discrete-time *master equation* under the Markovian approximation (see Chapter 5 for the detailed derivation of a general form based on projection operator formalism):

$$\mathbf{p}(t+\tau)^\top = \mathbf{p}(t)^\top \mathbf{T}(\tau), \qquad \boldsymbol{\pi}^\top \mathbf{T}(\tau) = \boldsymbol{\pi}^\top \tag{2.7}$$

i.e., $\mathbf{T}(\tau)$ is row-stochastic and preserves the stationary distribution $\boldsymbol{\pi}$.

Estimation of TPM and the detailed balance constraint. With the above definitions, the central idea of MSMs is to decompose the continuous MD trajectories into discrete transitions among several metastable states and to estimate the TPM simply by counting the transitions between them. With TPM, one can propagate long-timescale dynamics based on master equation and extract thermodynamic/kinetic properties of the system. This approach avoids the need for a single very long, globally equilibrated trajectory: many short MD simulations that are locally equilibrated can be stitched together to provide sufficient statistics for the transitions at lag τ, greatly reducing computational cost. In addition, this coarse-grained, state-based dynamic representation significantly enhances the interpretability of the corresponding biomolecular dynamic system.

In detail, we can directly estimate the TPM from MD data. Once we have the disjoint sets $\{S_1, \ldots, S_n\}$ and a chosen lag time τ, we can first compute the transition count matrix (TCM), i.e., $\mathbf{C}(\tau) = [C_{ij}(\tau)]$, using sliding windows over all trajectories

$$C_{ij}(\tau) = \sum_t \mathbf{1}\{\mathbf{x}_t \in S_i,\ \mathbf{x}_{t+\tau} \in S_j\} \tag{2.8}$$

Then, we can estimate the empirical TPM by row normalization

$$\widehat{T}_{ij}(\tau) = \frac{C_{ij}(\tau)}{\sum_j C_{ij}(\tau)} \tag{2.9}$$

For equilibrium systems, microscopic reversibility implies that the transition probabilities should satisfy the *detailed balance condition*,

$$\pi_i T_{ij}(\tau) = \pi_j T_{ji}(\tau), \qquad \forall\, i, j \tag{2.10}$$

which ensures that $\mathbf{T}(\tau)$ is reversible with respect to the stationary distribution $\boldsymbol{\pi}$

and thus preserves thermodynamic consistency. Enforcing detailed balance not only reduces statistical noise but also improves the robustness of estimated kinetics when sampling is limited or uneven.

In practice, there are two common strategies for constructing a reversible TPM from the transition counts. The first is the *transpose symmetrization* (or row–column averaging) strategy, where the reversible TPM $\widehat{\mathbf{T}}(\tau)$ is symmetrized to satisfy Eq. (2.10) by

$$T_{ij}^{\mathrm{sym}}(\tau) \;=\; \frac{1}{2}\left[\widehat{T}_{ij}(\tau) + \frac{\pi_j}{\pi_i}\widehat{T}_{ji}(\tau)\right] \tag{2.11}$$

with π_i estimated from state populations, e.g., $\pi_i = \sum_j C_{ij} / \sum_{k,l} C_{kl}$. This simple symmetrization enforces reversibility by construction and performs well when transition counts are sufficiently large.

A more statistically rigorous alternative is the *maximum-likelihood estimation* (MLE) approach under detailed balance constraints. In this formulation, the reversible TPM $\mathbf{T}(\tau)$ given stationary distribution $\boldsymbol{\pi}$ are obtained by maximizing the log-likelihood of observing the transition counts (see Ref. [142] for more general cases),

$$\mathcal{L}(\mathbf{T}) = \sum_{i,j} C_{ij}(\tau)\ln T_{ij}(\tau) \tag{2.12}$$

subject to

$$\sum_j T_{ij}(\tau) = 1, \quad \pi_i T_{ij}(\tau) = \pi_j T_{ji}(\tau), \quad T_{ij}(\tau) \ge 0.$$

This constrained optimization can be efficiently solved using iterative algorithms [116], yielding a reversible MSM that is statistically optimal given the available data. Compared with the symmetrization method, the MLE approach generally provides smoother estimates and more accurate long-timescale predictions, particularly when sampling is sparse or when certain transitions are rarely observed. Together, these reversible estimation strategies ensure that the constructed MSM adheres to equilibrium thermodynamics and forms a consistent bridge to the continuous reversible dynamics discussed before.

Validation of Markovianity: the Chapman–Kolmogorov test. After estimating a reversible TPM, it is essential to verify whether the chosen lag time τ yields a

Markovian model, i.e., whether the dynamics between states can indeed be described by a memoryless process at this temporal resolution. The standard diagnostic tool for this purpose is the *Chapman–Kolmogorov (CK) test*, which examines the consistency of transition probabilities across different multiples of the lag time.

For an ideal Markov process, the TPMs at different lag times should satisfy the Chapman–Kolmogorov equation

$$\mathbf{T}(k\tau) = [\mathbf{T}(\tau)]^k, \qquad k = 1,2,\dots, \tag{2.13}$$

meaning that a transition over $k\tau$ can be represented as k successive transitions of length τ. In practice, $\mathbf{T}(k\tau)$ can be directly estimated from MD data using the same procedure as for $\mathbf{T}(\tau)$, while $[\mathbf{T}(\tau)]^k$ can be computed via matrix multiplication. The degree of agreement between the two provides a quantitative measure of the Markovianity of the chosen model.

A common way to visualize this test is to compare the evolution of state populations predicted by the model, $\mathbf{p}(t_0)^\top[\mathbf{T}(\tau)]^k$, with the corresponding empirical populations observed at time $t_0 + k\tau$. Alternatively, one may compare the implied timescales $t_i(\tau) = -\tau/\ln\lambda_i(\tau)$ as a function of τ: for a truly Markovian model, these timescales should remain constant with respect to the lag time. In both cases, systematic deviations indicate that the selected lag time is too short and memory effects remain significant. The CK test therefore provides a practical criterion for choosing an appropriate lag time τ and validating the internal consistency of the MSM before proceeding to further kinetic or thermodynamic analyses.

Relation to the transfer operator. The discrete transition probability matrix $\mathbf{T}(\tau)$ can be interpreted as a finite-state approximation to the transfer operator $\mathcal{T}_\tau$ introduced in Section 2.2.1. Specifically, while $\mathcal{T}_\tau$ governs the time evolution of probability densities in the full continuous configuration space Ω, $\mathbf{T}(\tau)$ represents its Galerkin projection onto the finite-dimensional subspace spanned by the indicator functions of the metastable sets $\{S_i\}_{i=1}^n$. In other words, $\mathbf{T}(\tau)$ captures how the probability mass is transferred between these coarse-grained regions under the dynamics generated by $\mathcal{T}_\tau$:

$$T_{ij}(\tau) \;=\; \frac{1}{\pi_i}\left\langle \mathbf{1}_{S_i}, \mathcal{T}_\tau \mathbf{1}_{S_j} \right\rangle_\pi \tag{2.14}$$

This projection establishes a rigorous connection between the transfer operator formalism and the discrete MSM representation. The eigenfunctions of $\mathcal{T}_\tau$ describe the slow dynamic modes of the full system, while the corresponding eigenvectors of $\mathbf{T}(\tau)$ provide their discrete approximations in the coarse-grained space. Consequently, the implied timescales computed from $\mathbf{T}(\tau)$ approximate the relaxation times associated with the slowest eigenmodes of $\mathcal{T}_\tau$. As the state decomposition $\{S_i\}$ becomes increasingly fine and the sampling converges, the MSM approaches the exact spectral properties of the transfer operator.

Thus, the MSM can be viewed as a discretized projection of the transfer operator, offering a finite yet physically interpretable representation of molecular kinetics. This correspondence forms the theoretical foundation linking operator theory, Markov modeling, and spectral analysis of molecular dynamics.

2.3 Recommended Protocol for MSMs construction

The preceding sections established the theoretical foundation of MSMs [116], showing how the transfer operator $\mathcal{T}_\tau$ that governs dynamics in the full configuration space can be approximated by a discrete transition probability matrix $\mathbf{T}(\tau)$, whose spectral properties encode the slow dynamic modes of the system. However, the accuracy and interpretability of such a model critically depend on how the continuous molecular configurations are discretized into a finite set of states. In practice, the choice of representation and partitioning scheme directly determines whether the resulting MSM can faithfully approximate the true spectral space of $\mathcal{T}_\tau$ and thus reproduce the underlying long-timescale dynamics.

Constructing an MSM therefore requires a careful and systematic procedure [73, 89]. A well-designed MSM should resolve kinetically meaningful conformational states while preserving the relevant kinetic properties of the underlying dynamics. To this end, MSM construction in practice typically follows a standardized workflow comprising four major stages:

1. **Feature selection:** choosing physically meaningful descriptors (such as dihedral angles, contact maps, or pairwise distances) that capture the essential degrees of freedom governing conformational changes;

2. **Dimensionality reduction:** projecting the high-dimensional feature space onto a lower-dimensional manifold that preserves the slow collective motions, often using methods such as time-lagged independent component analysis (tICA) or other variational approaches;
3. **Clustering:** discretizing the reduced space into a large number of microstates and provide sufficient resolution of the dynamic landscape;
4. **Kinetic lumping:** coarse-graining microstates into macrostates or metastable basins to enhance interpretability and reduce model complexity while retaining the dominant kinetic behavior.

Together, these steps constitute the practical pipeline for building reliable and physically grounded MSMs. The following subsections elaborate on each stage in detail, discussing their theoretical rationale, methodological choices, and practical considerations for constructing robust models of biomolecular dynamics.

2.3.1 Feature Selection

The first step in constructing an MSM is the selection of appropriate molecular descriptors, or *features*, that characterize the relevant conformational degrees of freedom of the system. These features define the mapping from the continuous atomic coordinates $\mathbf{x}_t \in \Omega$ to a lower-dimensional representation $\mathbf{f}(\mathbf{x}_t) \in \mathbb{R}^d$, which serves as the basis for all subsequent analysis. In essence, feature selection determines what aspects of molecular motion are considered dynamically relevant and thus sets the stage for an effective discretization of the configurational space.

A well-chosen feature set should satisfy several criteria. First, it should be *physically meaningful*, i.e., the selected descriptors should correspond to structural variables that govern conformational transitions of interest. Second, the features should be *invariant* to irrelevant transformations, such as overall translation and rotation, ensuring that the equivalent conformations are represented identically. Third, they should be *sufficiently expressive* to distinguish kinetically distinct conformations while avoiding redundant or irrelevant high-frequency motions.

In practice, the choice of features depends on the molecular system and the type of motion being investigated. For small peptides or loop rearrangements, backbone torsion angles often provide a compact and interpretable description. For larger proteins or multi-domain biomolecules, pairwise distances between selected residues, inter-domain center-of-mass separations, or contact maps are commonly employed to capture both local flexibility and large-scale conformational rearrangements. In more complex biomolecular assemblies such as protein–ligand complexes, additional features such as hydrogen-bond networks, salt bridges, or solvent-accessible surface areas may be included to reflect the relevant biochemical interactions.

From a theoretical perspective, feature selection defines the subspace in which the transfer operator $\mathcal{T}_\tau$ will be approximated. The closer this feature space aligns with the true slow eigenspace of $\mathcal{T}_\tau$, the better the resulting MSM can reproduce the long-time dynamics of the full system. Conversely, inadequate or poorly chosen features may merge kinetically distinct configurations or emphasize irrelevant fluctuations, leading to inaccurate transition models.

Modern molecular modeling increasingly benefits from automated and data-driven feature extraction methods, such as graph-based representations that encode atomic connectivity, or symmetry-preserving descriptors that maintain physical invariances. In addition, recently developed approaches, spectral-oASIS [86], can automatically identify optimal subset of features as those that can reproduce the time-correlation matrix of the full feature set upon reconstruction and capture the slow dynamics; and MoSAIC [36], introduces a geometry-based selection method that can cluster candidate features based on their geometric correlations in molecular dynamics. These approaches extend traditional handcrafted features while retaining interpretability and physical consistency. Regardless of the specific choice, careful feature selection remains the cornerstone of any reliable MSM construction, as it determines the resolution and fidelity with which the subsequent dimensionality reduction and state decomposition steps can capture the relevant conformational dynamics of the system.

2.3.2 Dimensionality Reduction

After selecting physically meaningful molecular descriptors, the next step is to identify a small set of collective variables that best represent the slow, large-scale dynamics of the system. Since the raw feature space $\mathbf{f}(\mathbf{x}_t) \in \mathbb{R}^d$ is often high-dimensional and contains both fast and slow degrees of freedom, direct clustering or state decomposition in this space is inefficient and may lead to noisy or kinetically meaningless partitions. Therefore, dimensionality reduction serves to extract a low-dimensional manifold that captures the dominant dynamic modes while filtering out high-frequency fluctuations.

The fundamental objective of this step is to approximate the dominant eigenspace of the transfer operator $\mathcal{T}_\tau$ within the chosen feature basis. Ideally, one seeks a set of coordinates $\boldsymbol{\xi}(\mathbf{x}_t)$ that approximates the leading eigenfunctions of $\mathcal{T}_\tau$, as these modes govern the relaxation toward equilibrium and define the system's metastable structure (Section 2.2.1).

In practice, several linear and nonlinear methods based on *variational approach* [108] have been developed for this purpose. Among the most widely used are time-lagged independent component analysis (tICA) [117, 127] and the variational approach for Markov processes (VAMP) [156, 158]. tICA seeks linear projections of the input features that maximize the normalized autocorrelation at lag time τ, providing an approximation to the leading eigenfunctions of $\mathcal{T}_\tau$ under the assumption of reversibility. VAMP generalizes this framework by optimizing the variational score of the Koopman operator, allowing for both reversible and nonreversible dynamics. In both formulations, the resulting components are ordered by their associated timescales, and only the slowest dynamic modes are retained to form the reduced kinetic subspace.

Beyond these linear approaches, recent developments in deep learning have enabled nonlinear generalizations that can capture more complex dynamical manifolds. Time-lagged autoencoders [155], VAMPnets [98], and SRVs [23] employ neural networks to learn nonlinear transformations that preserve slow kinetic information, often leading to more accurate representations of biomolecular dynamics. These methods remain grounded in the same variational principle that underlies tICA and VAMP, sharing the same theoretical connection to the transfer or Koopman operator formalism.

In summary, dimensionality reduction links the original high-dimensional molecular descriptors to the lower-dimensional space on which Markov models are built. By projecting the data onto a space spanned by the dominant slow modes, this step greatly improves the statistical efficiency and physical interpretability of subsequent clustering and kinetic lumping, while ensuring that the discretized model remains faithful to the slow dynamics of the underlying system.

2.3.3 Clustering

After obtaining a reduced representation of the molecular dynamics, the next step is to partition this space into a large number of discrete regions, or *microstates*, by applying a clustering algorithm. This discretization transforms continuous trajectories into symbolic sequences of state indices, forming the basis for constructing the transition probability matrix $\mathbf{T}(\tau)$. The principal goal of this step is not to perform coarse-graining, but rather to create a *high-resolution approximation* of the transfer operator $\mathcal{T}_\tau$, in which the dynamics are finely resolved across the configurational landscape.

From a theoretical perspective, clustering can be viewed as a Galerkin discretization of $\mathcal{T}_\tau$ using indicator functions over the clusters $\{S_i\}_{i=1}^n$ as basis functions. Increasing the number of microstates enhances the representational capacity of this discrete space, allowing the model to more accurately approximate the true eigenfunctions of $\mathcal{T}_\tau$. In this sense, a microstate MSM acts as a high-dimensional numerical representation of the transfer operator: although individual states may not be directly interpretable, the ensemble of many small regions collectively provides a faithful reconstruction of the system's slow dynamic modes. This fine-grained representation then serves as the foundation for the subsequent kinetic lumping step, which extracts physically meaningful macrostates from the underlying microstate dynamics.

In practice, clustering is usually performed in the space of the slow coordinates obtained from dimensionality reduction (e.g., tICA or VAMP components), where the relevant kinetic structures are most clearly separated. Common algorithms include k-means [130] and k-centers [49, 168], which are both distance-based methods: k-means tends to place centers in regions of high data density, whereas k-centers emphasizes a more uniform coverage of the space by minimizing the maximum distance between any point and its nearest center. For systems with irregularly shaped or nonuniformly sampled manifolds, density-based methods such as DBSCAN [43] or density-peak clustering [121] can adaptively identify clusters that follow the underlying data distribution.

A key practical consideration is the number of clusters n. Too few clusters produce an overly coarse discretization that fails to resolve important kinetic variations, whereas too many may lead to sparse transition statistics and numerical noise. The optimal choice typically balances resolution and statistical reliability and can be assessed by examining the convergence of implied timescales or the stability of dominant eigenvectors as n varies.

In summary, clustering provides a fine-grained discretization of the reduced configuration space, enabling the construction of a microstate MSM that offers a detailed, high-resolution approximation of the underlying molecular kinetics. This microstate model, while not directly interpretable, faithfully preserves the system's dominant spectral structure and thereby paves the way for the subsequent kinetic lumping step, in which metastable macrostates and their transition networks are identified.

2.3.4 Kinetic Lumping

Once a high-resolution microstate MSM has been constructed, the next step is to coarse-grain it into a smaller number of kinetically meaningful macrostates that capture the dominant slow processes of the system. This procedure, known as *kinetic lumping*, transforms the detailed but often noisy microstate model into a simplified representation that reveals the metastable organization of the molecular dynamics while preserving its essential kinetic characteristics.

From the perspective of the transfer operator $\mathcal{T}_\tau$, kinetic lumping can be regarded as a second-level approximation that groups microstates whose probability densities evolve coherently under $\mathcal{T}_\tau$. Ideally, each macrostate corresponds to a metastable basin associated with a nearly constant sign structure of the leading eigenfunctions of $\mathcal{T}_\tau$ or, equivalently, of its discrete counterpart $\mathbf{T}(\tau)$. The goal is to construct a partition of the n microstates into $m \ll n$ macrostates, typically encoded in a partition matrix $\mathbf{A}$, such that the coarse-grained transition matrix obtained by projecting $\mathbf{T}(\tau)$ with respect to the stationary distribution reproduces the dominant eigenstructures of the original dynamics.

Several spectral methods have been developed to perform this coarse-graining optimally. Among them, the Perron cluster cluster analysis (PCCA [33, 34] and its improved variant PCCA+ [124]) remains the most widely used. PCCA+ identifies metastable macrostates by analyzing the subspace spanned by the leading eigenvectors of $\mathbf{T}(\tau)$ and assigning microstates based on the membership functions derived from these eigenvectors. This approach ensures that each macrostate is associated with a set of microstates that interconvert rapidly within the cluster but only slowly between clusters, thereby maximizing kinetic coherence while maintaining spectral fidelity.

By separating the slow inter-macrostate transition modes from the fast intra-macrostate dynamics, kinetic lumping substantially improves the model's interpretability and reduces statistical noise, enabling the visualization and quantitative analysis of long-timescale behavior in terms of a small number of metastable basins. To determine an appropriate number of macrostates, one can inspect the separation of implied timescales in the microstate MSM.

In summary, kinetic lumping bridges the gap between the detailed microstate representation and the physically interpretable macrostate model. By projecting the fine-grained MSM onto a reduced basis aligned with the dominant eigenfunctions of $\mathcal{T}_\tau$, this step completes the MSM construction pipeline, yielding a compact, interpretable, and dynamically consistent description of biomolecular dynamics.

2.4 Discussions

The framework presented above establishes a rigorous foundation for constructing Markov State Models that bridge microscopic dynamics and macroscopic kinetic observables. Once a reliable transition probability matrix $\mathbf{T}(\tau)$ is obtained and validated, a wide range of theoretical analyzes can be performed to elucidate the mechanisms of biomolecular conformational changes. Among these, Transition Path Theory (TPT) [42, 106] provides a particularly powerful tool for characterizing the reactive pathways between two metastable basins, such as folded and unfolded or ligand-bound and unbound states. By conducting a TPT analysis, key kinetic quantities can be computed, including committor probabilities, flux networks, and mean first-passage times, enabling a quantitative description of the reaction mechanisms of the system.

In parallel with advances in the theoretical analysis of MSMs, recent developments in deep learning have led to end-to-end frameworks that integrate the entire modeling pipeline into a single differentiable architecture. Among these, VAMPnets represent a milestone. Based on the VAMP theory, VAMPnets employ neural networks to learn nonlinear transformations of molecular configurations from input geometric features to metastable state assignments by optimizing a variational score associated with the approximation of slowest dynamic modes. In doing so, they unify feature selection, dimensionality reduction, state decomposition, and kinetic modeling into a single end-to-end pipeline. The resulting models can automatically discover kinetically meaningful order parameters or metastable states without manual preprocessing, thereby providing a data-driven yet physically grounded approach to uncovering molecular kinetics.

Although the MSM framework and its deep learning-based extensions, such as VAMPnets [98], have proven highly effective for elucidating protein conformational changes, several challenges persist. One of the most pressing issues lies in accurately representing high-dimensional, multi-body systems in which geometric and permutation symmetries play a central role. In such systems—ranging from protein complexes and supramolecular assemblies to colloidal and polymeric aggregates—the dynamic behavior emerges from collective motions among many identical or symmetry-related components. Conventional MSMs rely on fixed feature representations, such as pairwise distances or internal coordinates, that do not inherently preserve these symmetries. Consequently, equivalent configurations may be mapped to distinct points in the feature space, introducing artificial degeneracy and fragmenting the kinetic landscape. Similarly, VAMPnets, while more flexible through their neural-network parameterization, still operate on vectorized inputs lacking explicit awareness of molecular geometry and relational structure. These limitations hinder the faithful identification of slow collective modes and metastable states that arise from coordinated multi-body rearrangements. Overcoming these representational challenges requires models that intrinsically encode geometric relations and respect the chemical symmetries of the system.

In addition, two broader issues are worth emphasizing. First, standard MSM analyses focus on transitions among metastable states and therefore provide a coarse-grained view of mechanisms; resolving transition-state ensembles and high-resolution barrier-crossing structures generally requires complementary approaches, which are discussed in later chapters. Second, the Markovian assumption underlying $\mathbf{T}(\tau)$ is not guaranteed after discretization: projecting continuous dynamics onto a finite state representation can induce memory. Although using a sufficiently long lag time can, in principle, alleviate this violation, in complex biomolecular systems the required lag time often becomes comparable to—or even exceeds—the available MD trajectory length, rendering the approximation impractical. How such non-Markovian effects arise, how they can be diagnosed, and how they can be treated systematically will be addressed in subsequent chapters.

The next chapter builds upon this foundation by addressing this representational limitation. We introduce GraphVAMPnets [90], a graph-based neural architecture that learns slow collective variables directly from molecular configurations. By encoding physical invariances and relational information in the learning process, GraphVAMPnets provide a unified framework to uncover the intrinsic kinetic geometry of complex biomolecular systems. This development represents a natural evolution from the theoretical principles outlined in this chapter toward more expressive and physically consistent learning-based models for molecular dynamics.

Chapter 3

Learning Geometry from Symmetry – *GraphVAMPnets for Uncovering Slow Collective Variables of Multi-Body Systems*

3.1 Introduction

.

Understanding the dynamics of multi-body systems, where numerous interacting particles, residues, or molecular subunits collectively form complex structures, is a fundamental goal across chemistry, biology, and materials science [21, 38, 47, 53, 80, 170]. Such processes often proceed through a hierarchy of intermediate configurations governed not only by thermodynamic stability but also by kinetic control. While free-energy landscapes provide valuable thermodynamic insight, they may fail to capture the true dynamical nature of these multi-body transformations, which are often dominated by rare events, transient intermediates, and cooperative rearrangements. A kinetic perspective is therefore essential to reveal mechanistic pathways and rate-determining steps, ultimately enabling the rational design of functional materials and facilitating the understanding of many relevant biological processes.

Molecular dynamics (MD) simulations offer atomistic access to these processes,

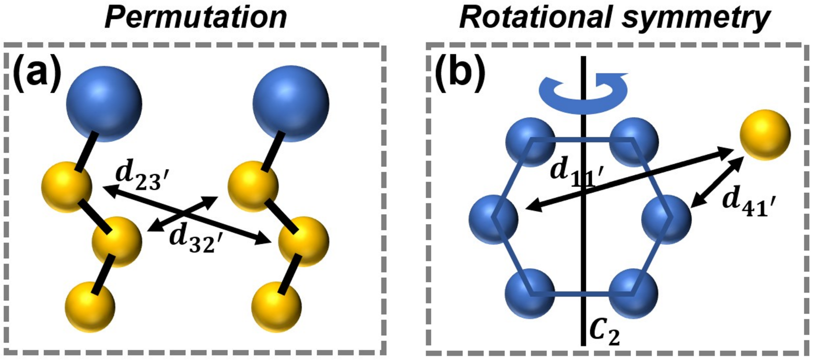


Figure 3.1: Illustration of challenges in feature selection for multi-body systems. *(a) Permutation-invariance: the monomers in multi-body system are usually identical. If two monomers exchange their positions, the structure is the same but the defined pairwise distances $d_{23'}$ and $d_{32'}$ will change. (b) Rotational symmetry-invariance: the monomers in multi-body system could be rotationally symmetric. If the hexagonal ring rotates along its C_2 axis, the structure is identical but the defined pairwise distances $d_{11'}$ and $d_{41'}$ will change.*

but their raw trajectories alone rarely yield a clear statistical picture of how distinct configurations interconvert. Kinetic network models (KNMs), and in particular, Markov state models (MSMs) [73, 89, 116, 153], address this challenge by statistically combining many MD trajectories to infer a global dynamical model. As discussed in Chapter 2, an MSM can approximate the transfer operator governing the time evolution of molecular ensembles, allowing one to extract kinetic pathways, transition rates, and mean first-passage times. However, the construction of such models for multi-body systems remains far from trivial, as it critically depends on identifying suitable representations that capture the relevant slow dynamics while respecting the underlying physical symmetries.

Two major challenges arise in this context [89]. (1) Symmetry invariance: multi-body systems often consist of identical or geometrically symmetric components—such as identical monomers in self-assembly, equivalent residues in protein oligomers, or indistinguishable ions in condensed phases. A permutation of identical subunits or a rigid-body rotation of a symmetric structure should not change its representation, yet commonly used feature sets, such as pairwise distances or Cartesian coordinates, violate these invariances (Figure 3.1). This leads to redundant or inconsistent descriptions that obscure the true dynamical manifold of conformational changes. (2) Sampling imbalance: many multi-body processes are energetically downhill or nonequilibrium in nature, producing trajectories that inadequately sample reverse transitions. In such cases, traditional time-reversible approaches like tICA, which rely on symmetrized correlation matrices to enforce detailed balance, may yield biased descriptions of the underlying dynamics.

To overcome these limitations, we develop and apply GraphVAMPnets [90], a symmetry-preserving, graph-based neural framework that integrates graph neural networks with the variational approach for Markov processes (VAMP) theory [156] to uncover the slow collective variables (CVs) of complex multi-body systems. In this formulation, each system configuration is represented as a graph, where nodes correspond to atomic or coarse-grained entities and edges encode pairwise interactions. Through message-passing operations that are invariant to permutations and rotations, GraphVAMPnets learn dynamical embeddings that respect intrinsic symmetries while directly optimizing the variational objective associated with the slowest dynamic modes of the Koopman operator. This enables the discovery of physically meaningful CVs and metastable states without the need for manual feature engineering or explicit symmetry corrections.

In contrast to conventional VAMPnets [98] that rely on fixed and vectorized input representations, GraphVAMPnets explicitly encode geometric and topological information through flexible graph representations. In particular, we extend previous implementations [48, 160] by allowing variable edge numbers for each node to account for the evolving coordination environment in multi-body assemblies. For systems containing identical or symmetric components, node embeddings are initialized identically and constrained during training to preserve permutation and rotational invariance.

The following sections present the theoretical formulation of GraphVAMPnets, its connection to the variational principle of Koopman operator, and the algorithmic details of euclidean symmetry-invariant graph encoders. We then demonstrate its capability through representative examples, including molecular aggregation and patchy-particle self-assembly, highlighting how GraphVAMPnets resolve slow CVs that conventional feature-based approaches fail to capture. Overall, this chapter establishes a symmetry-aware, geometry-informed learning framework that extends the MSM paradigm to the complex, relational dynamics of multi-body systems.

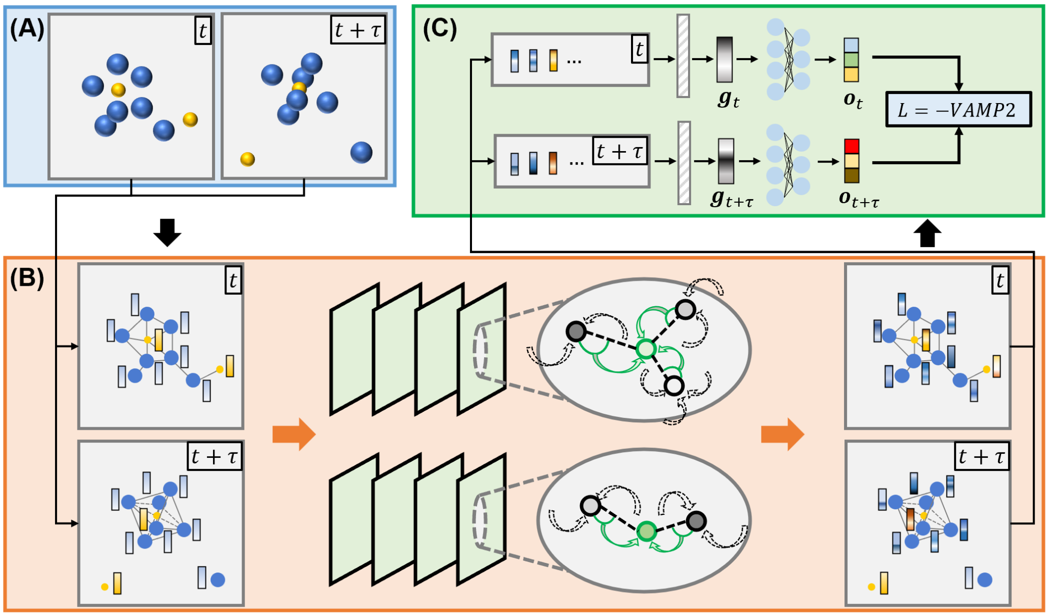


Figure 3.2: Schematic architecture of GraphVAMPnets for identifying slow CVs of multi-body systems. *The method can be divided into three parts: (A) Transition pairs of molecular configurations separated by a lag time τ are extracted from simulation trajectories. (B) Each configuration is represented as a graph and processed by two parallel graph neural networks. (C) The resulting graph embeddings are obtained via average pooling and passed into a VAMP-2-based objective to learn slow dynamic modes.*

3.2 Methods

GraphVAMPnets (Figure 3.2) integrate graph neural networks with the VAMP theory to identify the slow CVs of complex multi-body dynamics. The method combines the geometric expressivity of graph-based representations with the theoretical rigor of operator-based kinetic modeling. In this section, we describe the graph representation of molecular configurations, the neural network architecture, the theoretical formulation of VAMP, and the implementation of permutation and rotational symmetry invariance.

3.2.1 Graph Representation of Molecular Configurations

Each molecular configuration at time t is represented as a graph $G_t = (V, E)$, where nodes (V) correspond to atoms, residues, or coarse-grained particles, and edges (E)

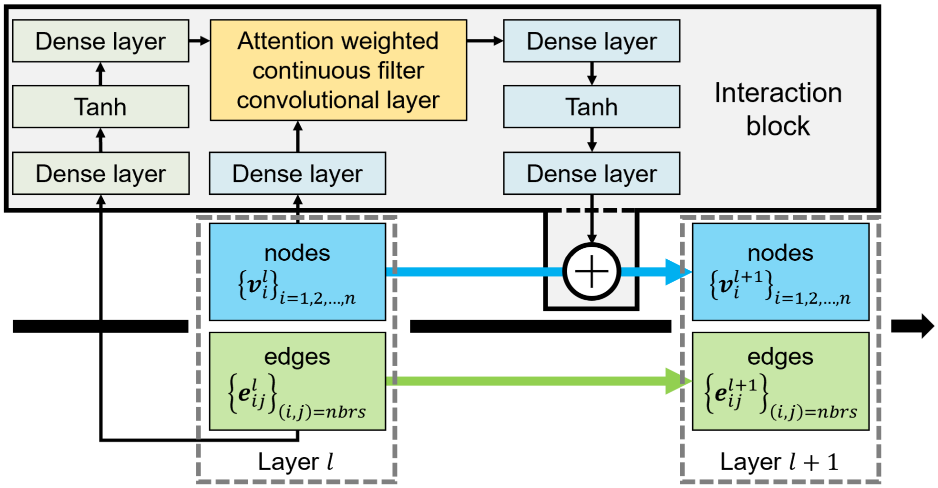


Figure 3.3: Illustration of graph convolutional layers in GraphVAMPnets based on the SchNet framework. *Each node aggregates messages from its neighbors through continuous-filter convolution layers with attention weighting before residual updates.*

encode local interactions or geometric relationships between them. Every node $i \in V$ is embedded as a feature vector $\mathbf{v}_i$, and each edge $(i,j) \in E$ is associated with an edge vector $\mathbf{e}_{ij}$ that encodes geometric and distance-based information.

For systems composed of identical or symmetric components, all equivalent particles are initialized with identical node embeddings to ensure permutation and rotational invariance. Edges can be defined using one of several criteria: (a) connect each node to a fixed number of its nearest neighbors, (b) connect pairs within a distance cutoff r_c, or (c) connect based on known chemical bonds or specific interaction rules.

Then, we embed these edges as vectors by using Gaussian expanded pairwise distances:

$$e_{ij,k} = \exp\left[-\frac{(d_{ij}-c_k)^2}{\sigma_k^2}\right] \tag{3.1}$$

where $e_{ij,k}$ is the k^{th} element of edge vector $\mathbf{e}_{ij}$, $\{c_k\}$ and $\{\sigma_k\}$ are centers and widths of the expansion kernels.

3.2.2 Graph Neural Network Architecture

The GNN architecture (Figure 3.3) employed in GraphVAMPnets follows SchNet [129], adapted to capture many-body interactions in dynamic molecular graphs and extract slow CVs. At each layer ℓ, node embeddings $\mathbf{v}_i^\ell$ and edge embeddings $\mathbf{e}_{ij}^\ell$ are first mapped to a common latent dimension through dense layers:

$$\tilde{\mathbf{v}}_i^\ell = \mathbf{W}_v^\ell \mathbf{v}_i^\ell + \mathbf{b}_v^\ell, \tag{3.2}$$

$$\tilde{\mathbf{e}}_{ij}^\ell = \mathbf{W}_e^\ell \tanh\left(\mathbf{W}_{e2}^\ell \mathbf{e}_{ij}^\ell + \mathbf{b}_{e2}^\ell\right) + \mathbf{b}_e^\ell \tag{3.3}$$

The local interactions from surroundings for each node i are then computed via a continuous-filter convolution with an attention mechanism:

$$\mathbf{z}_i^\ell = \sum_j \alpha_{ij}^\ell \left(\tilde{\mathbf{v}}_j^\ell \odot \tilde{\mathbf{e}}_{ij}^\ell\right), \tag{3.4}$$

$$\alpha_{ij}^\ell = \frac{\exp\left[\tilde{\mathbf{v}}_j^\ell \odot \tilde{\mathbf{e}}_{ij}^\ell \mathbf{W}_a^\ell\right]}{\sum_k \exp\left[\tilde{\mathbf{v}}_k^\ell \odot \tilde{\mathbf{e}}_{ik}^\ell \mathbf{W}_a^\ell\right]} \tag{3.5}$$

where $\odot$ denotes element-wise multiplication, and $\alpha_{ij}^{(\ell)}$ are normalized attention weights indicating the importance of neighbor j to node i. The updated node embeddings are obtained via residual connections:

$$\mathbf{v}_i^{\ell+1} = \mathbf{v}_i^\ell + \mathbf{W}_u^\ell \tanh\left(\mathbf{W}_z^\ell \mathbf{z}_i^\ell + \mathbf{b}_z^\ell\right) + \mathbf{b}_u^\ell \tag{3.6}$$

Stacking multiple interaction layers enables message propagation through the graph, allowing the embeddings to encode global many-body correlations while retaining local geometric detail.

3.2.3 Variational Approach for Markov Processes (VAMP)

The VAMP theory was developed by Wu and Noé [156]. Its theoretical foundation is built upon the Koopman operator theory, which originates from fluid mechanics and provides a powerful framework for analyzing the dynamics of arbitrary time-homogeneous Markov processes without requiring time reversibility. The Koopman operator $\mathcal{K}_\tau$ is defined as

$$\begin{aligned}(\mathcal{K}_\tau g)(\mathbf{x}) &= \mathbb{E}_t\left[g(\mathbf{x}_{t+\tau}) \,|\, \mathbf{x}_t = \mathbf{x}\right] \\ &= \int g(\mathbf{x}_{t+\tau})\, p(\mathbf{x}_{t+\tau} \,|\, \mathbf{x}_t = \mathbf{x})\, d\mathbf{x}_{t+\tau} \\ &= \int g(\mathbf{y})\, p(\mathbf{x}, \mathbf{y}; \tau)\, d\mathbf{y}\end{aligned} \tag{3.7}$$

where $g(\mathbf{x})$ is an arbitrary observable defined on the full configuration space, and $p(\mathbf{x}, \mathbf{y}; \tau)$ denotes the transition probability from configuration $\mathbf{x}$ to $\mathbf{y}$ over a lag time τ. The Koopman operator thus computes the conditional expectation of an observable at a future time, given the current configuration. It is also known as the *backward operator*, as it can be used to propagate ensemble averages backward in the configuration space over the lag time τ.

The goal of VAMP is to obtain the leading singular functions of the Koopman operator, which represent the slowest dynamic modes of the system and evolve on average under a linear propagation model

$$\begin{aligned}&\langle \psi_i, \mathcal{K}_\tau \phi_j \rangle_{\rho_0} = \sigma_i \delta_{ij}, \qquad 1 = \sigma_1 > \sigma_2 \geq \sigma_3 \geq \cdots \\ &\mathbb{E}_t[\mathbf{\Phi}(\mathbf{x}_{t+\tau})] = \mathbf{K}^\top \mathbb{E}_t[\mathbf{\Psi}(\mathbf{x}_t)]\end{aligned} \tag{3.8}$$

where ϕ_i, ψ_i, and σ_i denote the i-th right singular function, left singular function, and singular value of $\mathcal{K}_\tau$, respectively. $\mathbf{\Phi} = (\phi_1, \phi_2, \ldots)^\top$ and $\mathbf{\Psi} = (\psi_1, \psi_2, \ldots)^\top$ are the corresponding vector forms, and $\mathbf{K} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots)$.

VAMP employs the variational principle to approximate the leading left and right singular functions of $\mathcal{K}_\tau$ using two sets of trial functions,

$$\begin{aligned}\mathbf{f} = (f_1, f_2, \ldots, f_m)^\top, &\qquad \mathbf{g} = (g_1, g_2, \ldots, g_m)^\top \\ \langle f_i, f_j \rangle_{\rho_0} = \delta_{ij}, &\qquad \langle g_i, g_j \rangle_{\rho_1} = \delta_{ij}\end{aligned} \tag{3.9}$$

and maximizes the VAMP-2 score $\hat{R}_2$ as the sum of the squared approximate singular values, which is bounded by the sum of the squared true singular values.

$$\hat{R}_2 = \langle \mathbf{f}, \mathcal{K}_\tau \mathbf{g} \rangle_{\rho_0} = \sum_i \hat{\sigma}_i^2 \leq \sum_i \sigma_i^2 \tag{3.10}$$

In practice, the trial functions can be expressed as linear combinations of two sets of basis functions, $\boldsymbol{\chi}_0 = (\chi_{01}, \chi_{02}, \ldots, \chi_{0m})^\top$ and $\boldsymbol{\chi}_1 = (\chi_{11}, \chi_{12}, \ldots, \chi_{1m})^\top$, respectively:

$$\mathbf{f} = \mathbf{U}^\top \boldsymbol{\chi}_0, \qquad \mathbf{g} = \mathbf{V}^\top \boldsymbol{\chi}_1 \tag{3.11}$$

where $\mathbf{U}$ and $\mathbf{V}$ are transformation matrices.

The variational optimization problem can then be formulated as

$$\begin{aligned} \mathbf{I} &= \mathbf{U}^\top \mathbf{C}_{00} \mathbf{U}, \\ \mathbf{I} &= \mathbf{V}^\top \mathbf{C}_{11} \mathbf{V}, \\ \hat{R}_2 &= \sum_i (\mathbf{u}_i^\top \mathbf{C}_{01} \mathbf{v}_i)^2 \end{aligned} \tag{3.12}$$

where the instantaneous and time-lagged covariance matrices are defined as

$$\begin{aligned} \mathbf{C}_{00} &= \mathbb{E}_t[\boldsymbol{\chi}_0(\mathbf{x}_t)\,\boldsymbol{\chi}_0(\mathbf{x}_t)^\top], \\ \mathbf{C}_{11} &= \mathbb{E}_t[\boldsymbol{\chi}_1(\mathbf{x}_{t+\tau})\,\boldsymbol{\chi}_1(\mathbf{x}_{t+\tau})^\top], \\ \mathbf{C}_{01} &= \mathbb{E}_t[\boldsymbol{\chi}_0(\mathbf{x}_t)\,\boldsymbol{\chi}_1(\mathbf{x}_{t+\tau})^\top] \end{aligned} \tag{3.13}$$

To solve the above optimization problem, the covariance matrices $\mathbf{C}_{00}$ and $\mathbf{C}_{11}$ can be decomposed into the products of their respective matrix square roots, $\mathbf{C}_{00}^{1/2}$ and $\mathbf{C}_{11}^{1/2}$. The singular values of the matrix

$$\mathbf{K}_s = \mathbf{C}_{00}^{-1/2} \mathbf{C}_{01} \mathbf{C}_{11}^{-1/2} \tag{3.14}$$

are then equivalent as the approximate singular values of the Koopman operator.

Consequently, the VAMP-2 score can be expressed as the squared Frobenius norm of $\mathbf{K}_s$:

$$\hat{R}_2 = ||\mathbf{K}_s||_F^2 = ||\mathbf{C}_{00}^{-1/2} \mathbf{C}_{01} \mathbf{C}_{11}^{-1/2}||_F^2 \tag{3.15}$$

The time evolution of observables in the projected space can then be expressed as

$$\mathbb{E}_t[\boldsymbol{\chi}_1(\mathbf{x}_{t+\tau})] \approx \mathbf{K}^\top \mathbb{E}_t[\boldsymbol{\chi}_0(\mathbf{x}_t)] \tag{3.16}$$

where $\mathbf{K} = \mathbf{C}_{00}^{-1}\mathbf{C}_{01}$ is often referred to as the *Koopman matrix*, which provides a finite-dimensional approximation of the Koopman operator that governs the dynamics of latent variables $\boldsymbol{\chi}_0$ and $\boldsymbol{\chi}_1$ with a least-squares regression minimizing $\mathbb{E}_t\left[\|\boldsymbol{\chi}_1(\mathbf{x}_{t+\tau}) - \mathbf{K}\boldsymbol{\chi}_0(\mathbf{x}_t)\|^2\right]$.

Building on this framework, Noé and coworkers developed VAMPnets [98], in which the basis functions $\boldsymbol{\chi}_0$ and $\boldsymbol{\chi}_1$ are parameterized by two parallel neural networks that introduce nonlinearity into the representation. The loss function is defined as the negative VAMP-2 score, and minimizing this loss maximizes the VAMP-2 objective. The trained neural networks thus serve as optimal basis functions to approximate the leading singular functions of the Koopman operator, yielding slow CVs that capture the system's essential kinetic processes.

In the present work, we extend this concept to GraphVAMPnets, where graph neural networks with average pooling layers act as the nonlinear basis functions in the VAMP formulation. Each molecular configuration is embedded as a graph, and pairs of configurations separated by lag time τ are encoded as

$$\mathbf{o}_0 = (\chi_{01}(\mathbf{x}_t), \ldots, \chi_{0m}(\mathbf{x}_t))^\top, \qquad \mathbf{o}_1 = (\chi_{11}(\mathbf{x}_{t+\tau}), \ldots, \chi_{1m}(\mathbf{x}_{t+\tau}))^\top.$$

The empirical mean-free covariance matrices $\bar{\mathbf{C}}_{00}$, $\bar{\mathbf{C}}_{01}$, and $\bar{\mathbf{C}}_{11}$ are estimated from these graph embeddings to compute the VAMP-2 score. The parameters of the graph networks are then optimized by minimizing the negative VAMP-2 loss, and the resulting approximated leading singular functions are obtained.

$$\mathbf{f} = \mathbf{U}^\top \boldsymbol{\chi}_0, \qquad \mathbf{g} = \mathbf{V}^\top \boldsymbol{\chi}_1$$

Specifically, we take the left approximate singular functions $\mathbf{f}$ as our identified slow CVs.

3.2.4 Permutation and Rotational Symmetry Invariance of GraphVAMPnets

A fundamental property of graph neural networks is that the learned representation of each node depends solely on its local environment, rather than on its index within the graph. This property, known as index equivariance, ensures that the

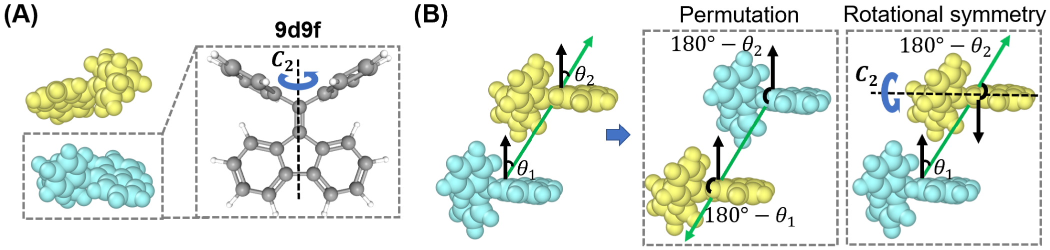


Figure 3.4: Illustration of the aggregation-prone 9-(Diphenylmethylene)-9H-fluorene (9d9f) molecules and the rotational-symmetry–invariant features of the system. *(A) Chemical structure of the 9d9f molecule. (B) Definition of the angles* (θ_1, θ_2)*, formed between the normal vectors of the two molecules and the displacement vector connecting their centers of mass. Exchanging the two 9d9f molecules transforms the angles from* (θ_1, θ_2) *to* $(180° - \theta_1, 180° - \theta_2)$*, yet the overall configuration remains identical. Similarly, rotating one molecule (shown in yellow) around its* C_2 *axis changes* θ *to* $(180° - \theta)$ *without altering the structure of the dimer.*

embeddings of identical nodes are consistent under reordering. When the node embeddings are subsequently aggregated through average pooling to produce a graph-level representation, the resulting feature vector becomes invariant to the node indexing.

In self-assembly or other multi-body systems, particles that are identical under permutation or rotational symmetries should therefore be assigned the same initial node embeddings and constrained to share identical values throughout training. If two such particles exchange positions through a permutation or rotation, this operation merely corresponds to swapping their node indices in the graph. Because of the index equivariance of the GNN and the pooling operation, the overall graph embedding remains unchanged, guaranteeing that equivalent conformations yield identical representations in GraphVAMPnets.

3.3 Results

3.3.1 Aggregation of Two Hydrophobic Molecules in Water

To demonstrate the capability of GraphVAMPnets in capturing slow dynamic modes that are invariant to permutation and rotational symmetry in multi-body systems, we first applied the method to study the aggregation dynamics of two hydropho-

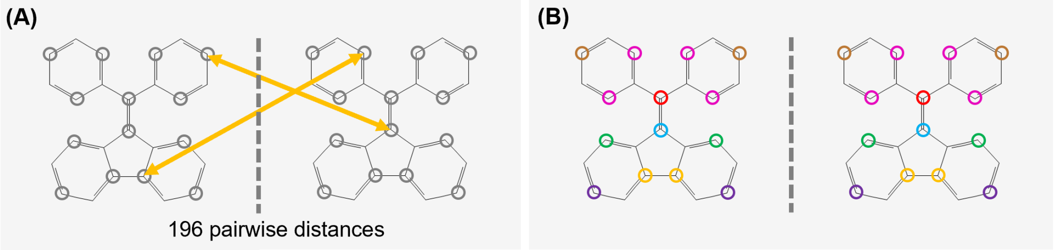


Figure 3.5: Feature embedding of the 9d9f system using different methods. (*A*) *In tICA and VAMPnets, 14 carbon atoms (highlighted by grey circles) are selected from each 9d9f molecule, yielding 196 intermolecular pairwise distances (indicated by orange arrows) as input features.* (*B*) *In GraphVAMPnets, the same 28 carbon atoms are represented as graph nodes with learned embeddings. Atoms circled in the same color correspond to positions that are equivalent under permutation and rotational symmetries and are therefore constrained to share identical node embeddings during training.*

bic molecules (i.e., (9-(Diphenylmethylene)-9H-fluorene) or 9d9f) in water (Figure 3.4A). Hydrophobic association is a fundamental process in biomolecular and colloidal systems, often serving as a model for understanding protein folding, micellization, and molecular recognition.

All-atom MD simulations were performed to generate trajectories of the two hydrophobic molecules freely diffusing and aggregating in aqueous solution. In total, there are 40 MD trajectories, with an aggregated simulation time of 4 μs. To examine the permutation- and rotational-symmetry–invariant nature of the system, we characterized each configuration of the 9d9f dimer using a pair of angles (θ_1, θ_2), defined between the normal vectors of the two molecules and the displacement vector connecting their centers of mass (Figure 3.4B, left). As illustrated in Figure 3.4B (right), a configuration with angle θ is geometrically identical to one with angle $(180° - \theta)$, since the two structures can be perfectly superimposed through permutation and rotation of the identical 9d9f molecules. We then projected all MD configurations onto the two-dimensional (θ_1, θ_2) space, which revealed four equivalent regions with identical conformations (Figure 3.6A).

To compare the performance of different dimensionality-reduction approaches, we applied tICA, VAMPnets, and GraphVAMPnets to the same MD dataset. For tICA and VAMPnets, 14 representative carbon atoms were selected from each 9d9f molecule, yielding 196 inter-molecular pairwise distances between atoms on different

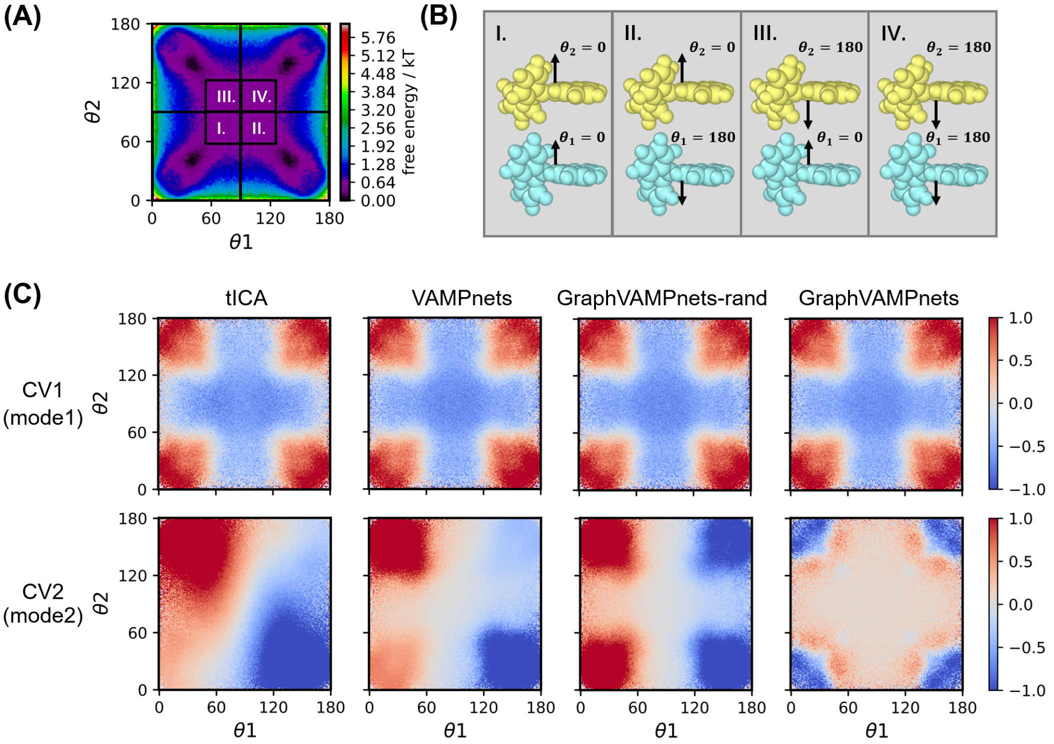


Figure 3.6: Visualization of identified CVs from different methods in the (θ_1, θ_2) **space.** *(A) Free-energy landscape of all MD conformations projected onto* (θ_1, θ_2)*. (B) Representative conformations from four equivalent regions (I–IV). (C) Projection of CV values onto the* (θ_1, θ_2) *space. All methods identify similar first CVs, but only GraphVAMPnets captures a symmetry-consistent second CV.*

molecules as input features (Figure 3.5A). In contrast, GraphVAMPnets represented each conformation as a graph, using the same 14 carbon atoms from each molecule as nodes. Atoms equivalent under permutation or rotational symmetry were embedded with identical node vectors (atoms of the same color in Figure 3.5B). Edges were defined between each atom and its ten nearest neighbors. To evaluate the influence of the symmetry constraint, we also trained a control model (GraphVAMPnets-rand) in which node embeddings were not constrained by symmetry. All methods were trained using the same hyperparameters, with a training lag time of 200 ps, a batch size of 1000, a learning rate of 10^{-3}, and an 80%/20% train/validation split. From each model, two slow CVs were extracted.

As shown in Figure 3.6, we compared the slowest CVs identified by tICA, VAMPnets, GraphVAMPnets-rand, and GraphVAMPnets, projected in the (θ_1, θ_2) space.

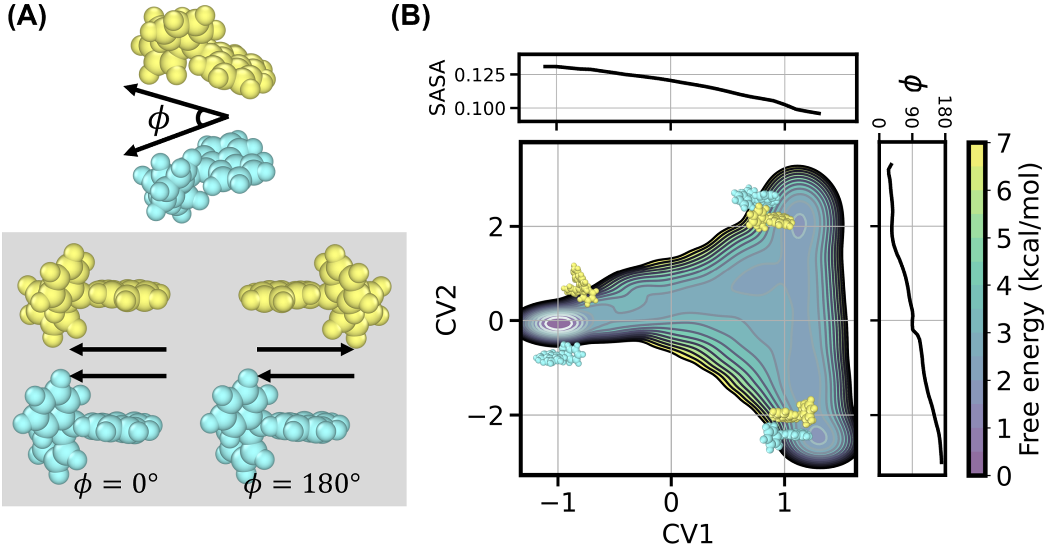


Figure 3.7: Conformations of the 9d9f aggregation system in the space of the two slowest CVs identified by GraphVAMPnets. *(A) ϕ is defined as the angle between the vectors pointing from fluorenyl to phenyl rings in the two molecules, describing their relative orientation. (B) Free-energy landscape projected on CV1 and CV2. SASA decreases monotonically along CV1, while ϕ varies from 180° to 0° along CV2. Representative conformations—dispersed (left) and stacked with parallel or antiparallel orientations (right)—correspond to the three basins in this projection.*

Each CV represents a distinct dynamic mode corresponding to transitions between its positive and negative regions. As shown in the top row of Figure 3.6C, the first CV obtained by all methods is consistent, displaying four symmetric positive regions (red) corresponding to identical aggregate conformations (examples in Figure 3.6B). However, notable differences emerge in the second CV (bottom row of Figure 3.6C). The second modes obtained by tICA, VAMPnets, and GraphVAMPnets-rand primarily describe transitions between symmetry-related states—those differing only by molecular permutation or rotation along the C_2 axis—rather than genuine conformational changes. In contrast, the second CV derived from GraphVAMPnets displays a symmetric pattern consistent with the four equivalent regions in the (θ_1, θ_2) space, confirming that GraphVAMPnets successfully learns permutation- and rotation-invariant CVs that reflect the true aggregation dynamics.

To gain further insight into the physical meaning of the learned CVs, we projected all MD conformations onto the two CVs identified by GraphVAMPnets (Figure 3.7B).

The solvent-accessible surface area (SASA) and ϕ angle, representing the relative orientation between the two molecules, were analyzed as physically interpretable descriptors. SASA is a useful indicator of hydrophobic collapse, decreasing as aggregation proceeds, whereas ϕ distinguishes head-to-head and head-to-tail stacking modes (Figure 3.7A). Along CV1, SASA monotonically decreases, indicating that CV1 primarily describes the aggregation process from dispersed to compact states. Along CV2, the ϕ angle varies from 0° to 180°, reflecting rotational rearrangements of the two molecules. The free energy landscape (Figure 3.7B) reveals that the left basin corresponds to dispersed conformations where the molecules are separated ($\phi \approx 90^\circ$), while the two right basins correspond to aggregated states: the upper basin represents parallel stacking (same orientation), and the lower basin represents antiparallel stacking (opposite orientation). The asymmetry along CV2 indicates a slight energetic preference for the antiparallel configuration, which exhibits a smaller SASA and thus greater stability. This observation aligns with chemical intuition, as steric hindrance between phenyl groups reduces π–π stacking efficiency in the parallel configuration, leading to a higher free energy.

3.3.2 Self-Assembly of Patchy Particles

We next applied GraphVAMPnets to identify the slow CVs and construct a kinetic network model for the self-assembly of patchy particles (Figure 3.8). The dataset, adapted from previous studies [83, 84, 163], comprises 60 coarse-grained MD trajectories, each containing 2,000 frames saved at 1,000 MD-step intervals. In these simulations, 960 patchy particles spontaneously form dodecahedral cages over time (Figure 3.8B). Each patchy particle consists of a central spherical core and three surface patches separated by fixed angular intervals of 108°, 108°, and 120°, with a half-opening angle θ_m^k of 55° for each patch (Figure 3.8A).

To construct the graph representation, we identified all existing inter-particle interactions as edges within each MD frame. Two particles were considered connected if (i) the distance between their centers was smaller than a cutoff $r_c = 0.75$ and (ii) each had at least one patch whose orientation vector formed an angle with the inter-particle axis smaller than the half-opening angle (Figure 3.8C). Each particle was embedded as a node vector, and all conformations were represented as graphs. The

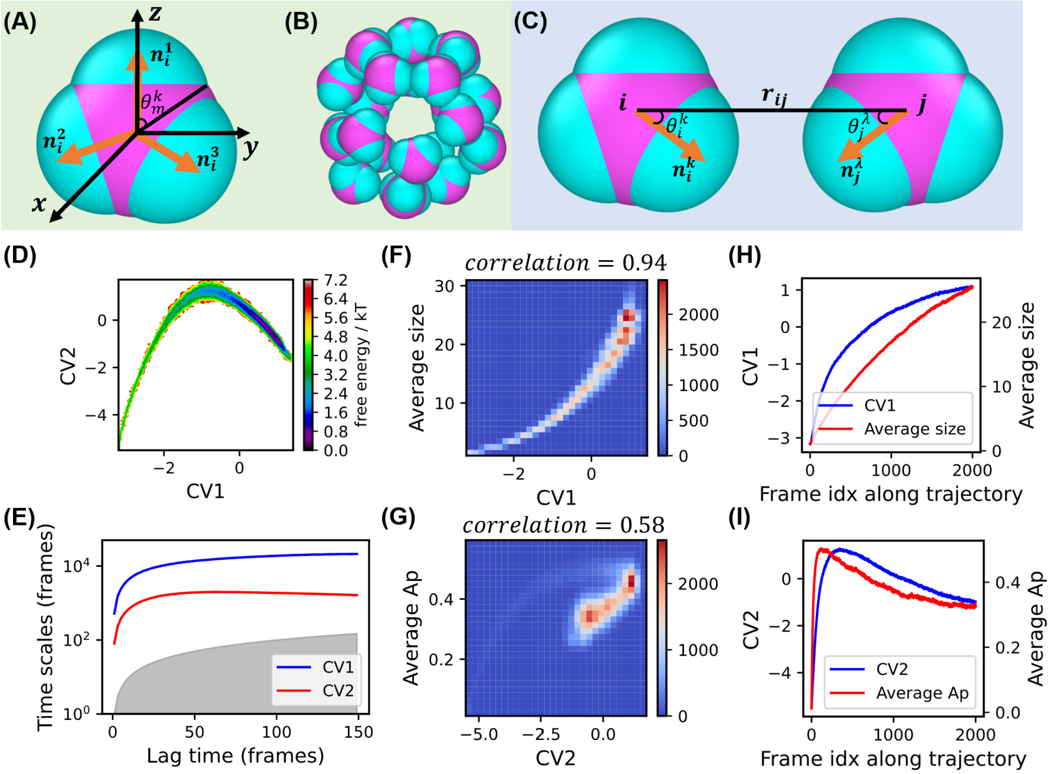


Figure 3.8: Illustration of the patchy-particle system and the two slowest CVs obtained from GraphVAMPnets. *(A) Structure of a single patchy particle, where the directions of three patches are* $\mathbf{n}_i^1 = (0,0,1)$, $\mathbf{n}_i^2 = (0,-0.9531,-0.3090)$, *and* $\mathbf{n}_i^3 = (0.5257,0.6935,-0.5038)$. *The opening angle* θ_m^k *of each patch equals 55°. (B) Self-assembled dodecahedral cage formed by patchy particles. (C) Two particles are considered as connected if* $r_{ij} < 0.75$ *and* $\theta_i^k, \theta_j^\lambda < 55°$. *(D) Free-energy landscape of all MD conformations projected onto CV1 and CV2 identified by GraphVAMPnets. (E) Implied timescales of the two CVs. (F) Histogram of conformations in the (CV1, average size) space* ($r = 0.94$). *(G) Histogram in the (CV2, average* A_p*) space* ($r = 0.58$). *(H) Time evolution of CV1 and average size along an MD trajectory. (I) Time evolution of CV2 and average* A_p *along an MD trajectory.*

lag time for training was set to 100 frames (corresponding to 10^5 MD steps), and two output nodes were used at the final layer to obtain the two slowest dynamic modes (or CVs). The model was trained for 50 epochs and repeated 10 times.

After training, two dominant CVs were obtained and all MD conformations were projected onto the (CV1, CV2) plane (Figure 3.8D). The ITS analysis (Figure 3.8E) revealed clear timescale separation, confirming the Markovian nature of the reduced dynamics. We further examined the physical interpretation of the CVs. The first CV exhibited a strong correlation with the average aggregate size ($r = 0.94$), whereas the second CV correlated with the average asphericity parameter (A_p) of aggregates ($r = 0.58$). Here, A_p characterizes aggregate morphology, with $A_p = 0$ corresponding to a perfect sphere and $A_p = 1$ to an ideal rod. To visualize these relationships, we projected conformations onto (CV1, average size) and (CV2, average A_p) planes (Figure 3.8F–G). The diagonal distributions in both maps highlight the strong correspondence between the learned CVs and these physically interpretable quantities. Moreover, the time evolution of CV1 and CV2 along representative trajectories (Figure 3.8H–I) confirmed that CV1 tracks the overall growth of aggregates, while CV2 captures their shape evolution.

Using these two CVs, we further constructed a kinetic network model to investigate the detailed mechanism of self-assembly dynamics. Following the standard KNM workflow outlined in Chapter 2, we first discretized the continuous (CV1, CV2) space into 100 microstates using k-means clustering (Figure 3.9A), and subsequently lumped them into five macrostates via the PCCA+ algorithm (Figure 3.9B).

We then analyzed the physical properties associated with each macrostate. As shown in Figure 3.9C, the average aggregate size systematically increases from State 4 to State 0, indicating progressive aggregation. In the early stage (State 4 $\longrightarrow$ 3), monomer concentration decreases sharply (Figure 3.9D). In the intermediate stages (State 3 $\longrightarrow$ 2 $\longrightarrow$ 1), aggregates containing 12 pentagonal rings begin to emerge; these intermediates closely resemble the final dodecahedral cages (Figures 3.9E–F). At the final stage (State 1 $\longrightarrow$ 0), the overall number of aggregates and 12-pentacyclic structures stabilizes, but we observed structural rearrangements within the aggregates. Specifically, State 0 is characterized by the appearance of additional hexagonal rings within previously dodecahedral cages (i.e., 12-pentacyclic aggregates containing zero hexagonal rings) (Figure 3.9G). This suggests that the transition from State 1 to State 0 corresponds to a rearrangement process in which dodecahedral cages reorganize into larger aggregates containing hexagonal motifs.

To further quantify the kinetics, we computed mean first-passage times (MFPTs) between macrostates using maximum-likelihood estimation (Figure 3.9H). The resulting MFPT network reveals a predominantly downhill assembly process from the monomeric State 4 to the dodecahedral-cage State 1, as forward aggregation transitions are significantly faster than their reverse processes. The dodecahedral

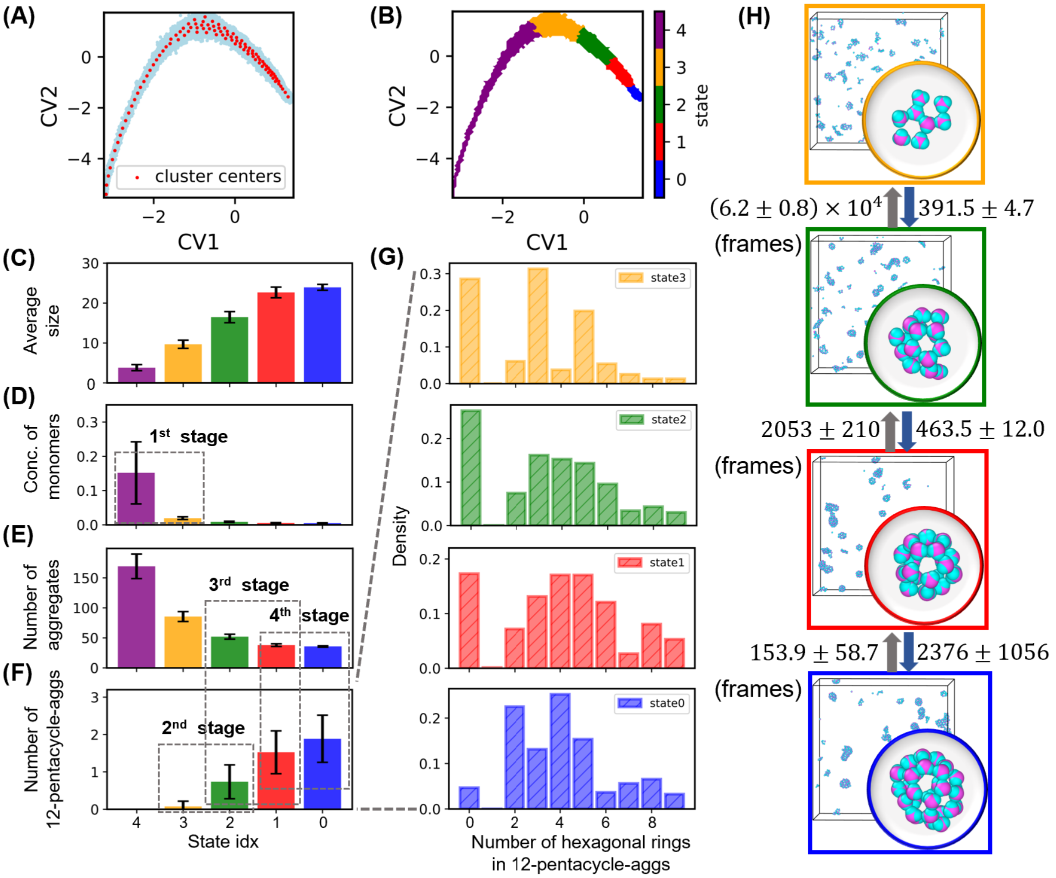


Figure 3.9: Analysis of patchy-particle self-assembly dynamics. *(A) Projection of all MD conformations onto (CV1, CV2) with 100 microstates (red dots) obtained by k-means clustering. (B) Projection colored by macrostate labels (0–4). (C) Average aggregate size in each macrostate. (D) Concentration of monomers. (E) Number of aggregates. (F) Number of 12-pentacyclic aggregates. (G) Distribution of the number of hexagonal rings in 12-pentacyclic aggregates. (H) Mean first-passage times (MFPTs) between states along the assembly pathway (State 3 → 0). Bars in panels (C–F) represent means ± ½ standard deviation.*

cage is thus identified as the most thermodynamically stable structure. In contrast, State 0 acts as an off-pathway intermediate—accessible kinetically but less stable thermodynamically—where additional hexagonal rings are formed, raising the free energy as indicated by the forward–backward MFPT ratio between States 1 and 0.

Overall, the constructed KNM reveals a clear downhill self-assembly process from dispersed monomers (State 4) to the thermodynamically stable dodecahedral cages

(State 1), with a metastable, off-pathway State 0 corresponding to the formation of larger aggregates containing hexagonal rings. These findings are consistent with previous studies [163], which showed that the 108° patch angles favor the formation of pentagonal motifs leading to dodecahedral cages, while the additional 120° angle permits occasional hexagonal ring formation. Our results thus demonstrate that GraphVAMPnets can accurately capture both the thermodynamic landscape and kinetic pathways of patchy-particle self-assembly, providing mechanistic insights that opens up a window of opportunities for the rational design of self-assembly structures with varying yields of dodecahedral cages (containing only pentagonal rings) versus those containing hexagonal rings, achieved by tuning the angles between patches in patchy particles.

3.4 Discussions

In this chapter, we addressed the intrinsic challenges in modeling the dynamics of multi-body systems by developing an enhanced implementation of GraphVAMPnets. Traditional KNM (or MSM) construction often relies on manually defined features, which limits their applicability to multi-body systems such as molecular assembly and biological aggregation. In contrast, our approach integrates the expressive power of graph neural networks with the VAMP to automatically extract slow CVs that govern the long-timescale dynamics of complex multi-body systems.

The effectiveness of our approach was demonstrated using two representative examples. For the aggregation of two 9d9f molecules in water, GraphVAMPnets successfully disentangled permutation- and rotation-invariant modes of motion, outperforming conventional methods such as tICA and VAMPnets. The learned CVs not only reproduced the hydrophobic collapse process but also revealed distinct aggregation orientations and their relative stabilities. For the patchy-particle system, GraphVAMPnets identified slow CVs that correlated strongly with the average aggregate size and asphericity parameter, enabling a detailed kinetic network model of self-assembly. This model unveiled a downhill aggregation pathway toward the thermodynamically stable dodecahedral cage, along with a metastable, off-pathway state characterized by hexagonal ring formation. Together, these case studies demonstrate that GraphVAMPnets can faithfully capture both the thermodynamic landscape and kinetic hierarchy of molecular self-assembly without imposing detailed balance constaint.

Beyond the two representative systems studied here, the framework developed in this chapter provides a versatile foundation for exploring the kinetics of complex biological and soft-matter systems composed of many interacting components. The variable-edge graph representation and symmetry-aware node embeddings together enable GraphVAMPnets to capture the dynamic coupling and reorganization among multiple molecular subunits, such as those found in protein complexes, nucleic acid condensates, and phase-separated biomolecular assemblies. By learning slow collective modes directly from simulation data without enforcing equilibrium constraints, this approach opens new opportunities for uncovering the emergent mechanisms of multi-body organization and collective transitions in living systems. Looking forward, hierarchical extensions of GraphVAMPnets could be integrated with coarse-grained or hybrid multiscale models to bridge molecular and mesoscale dynamics.

This chapter is adapted from a peer-reviewed publication *"Liu, B., Xue, M., Qiu, Y., Konovalov, K., O'Connor, M., Huang, X. (2023). GraphVAMPnets for uncovering slow collective variables of self-assembly dynamics. The Journal of Chemical Physics, 159, 094901. https://doi.org/10.1063/5.0158903"*

Chapter 4

Revealing Transition from Uncertainty – *Exploring Transition States of Protein Conformational Changes via Out-of-Distribution Detection in a Hyperspherical Latent Space*

4.1 Introduction

Understanding the transition states of protein conformational changes is of central importance in molecular biophysics, as these fleeting configurations define the energetic and kinetic bottlenecks that separate distinct functional states of a biomolecule. Characterizing transition states is essential for elucidating a broad spectrum of biological processes, including protein folding and misfolding, enzymatic catalysis, and gene expression. Moreover, gaining structural and energetic insights into transition states can guide rational strategies for drug discovery and enzyme engineering. For example, a recent study demonstrated that a single point mutation can rescue RNA polymerase II from pausing at an epigenetically modified DNA base by destabilizing the transition state of its translocation conformational change [74]. Such examples underscore how detailed knowledge of transition states can clarify how molecular modifications regulate function and offer opportunities to design compounds that preferentially bind and modulate these short-lived configurations. However, direct experimental identification of transition states remains challenging because these conformations are transient, sparsely populated, and difficult to capture at atomic resolution.

As discussed in previous chapters, molecular dynamics (MD) simulations, in combination with Markov state models (MSMs) [73, 89, 116, 153], provide a rigorous statistical framework for investigating the protein conformational changes at atomic level. MD simulations generate time-resolved trajectories by integrating Newton's equations of motion, while MSMs coarse-grain these trajectories into a discrete network of transitions among metastable states. This hierarchical modeling strategy enables the reconstruction of long-timescale dynamics from multiple short trajectories, thus overcoming the intrinsic timescale limitations of all-atom MD simulations that typically operate at femtosecond timesteps. Over the past decade, MSMs have yielded profound insights into mechanisms of protein folding, allostery, and ligand binding [5, 14, 16, 26–28, 32, 45, 59, 66, 71, 72, 76, 96, 114, 133, 134, 136, 147, 148], uncovering how proteins move between well-defined free energy basins.

However, while MSMs can model transitions, *they do not reveal them*. The MSM formalism captures the statistics of state-to-state transitions including transition probabilities and fluxes, but not the structural or energetic essence of the transition itself. Each conformation is discretely assigned to a metastable basin, reducing the continuous passage across the free energy barrier to a jump between coarse states. In this sense, the true "transition"—the transient, rare-populated ensemble of conformations residing at the free energy barriers—remains hidden within the statistical abstraction. Consequently, while MSMs provide a coarse-grained kinetic description of how conformations move between free energy basins, they cannot directly reveal the microscopic transition-state ensemble that governs how those transitions occur. This limitation motivates the development of new methodologies capable of capturing transition states with higher spatial and kinetic resolution, bridging the gap between coarse-grained state models and detailed atomistic pathways.

To address this limitation, one conventional strategy is to compute committor probabilities for finely discretized microstates within an microstate-based MSM. Microstates exhibiting equal probabilities (committor ≈ 0.5) of reaching both source and sink states can be interpreted as the transition states. While this approach is theoretically sound, it critically depends on the quality of the underlying microstate MSM and is sensitive to the resolution of discretization. Moreover, it can only reveal transition states between two selected states at a time, rather than identifying all transition states simultaneously across multiple free energy barriers. Recently, a deep learning–based method, MaxEnt-VAMPNets [69], was introduced to identify transition state structures to enhance adaptive sampling. Building upon the VAMPnets framework [98], this approach estimates the Shannon entropy of the state-assignment probabilities produced for each MD conformation, and the conformations exhibiting higher entropy values are considered as residing in low-probability regions within the free energy landscape. Although this entropy-based criterion provides a useful heuristic for detecting potential transition states, the underlying assumption is not strictly physical. The state probabilities generated by VAMPnets correspond to basis functions optimized to reconstruct the system's slowest dynamic modes (Chapter 3), rather than to the true committor probabilities that quantify the likelihood of crossing between source and sink states. Consequently, the entropy measure may not accurately capture the genuine transition-state ensemble within molecular dynamics.

Over the past decades, various simulation-based techniques have also been developed for explicitly locating transition states. For example, transition path sampling (TPS) [10, 64] directly explores the transition path ensemble by generating reactive trajectories via Monte Carlo sampling in trajectory space, allowing for the extraction of committor functions and transition configurations. Additionally, a deep reinforcement learning approach that integrates an efficient path-sampling strategy, known as enhanced sampling of reactive trajectories (ESoRT) [165, 166], has been employed to identify transition states by formulating the problem as a shooting game [164]. Beyond directly locating transition states, more approaches have been proposed to determine optimal reaction pathways [40, 41, 54, 97, 110, 123] or to compute committor functions [67, 75, 82, 115], both of which provide valuable insights into transition state identification. However, despite their robustness and potential, these approaches often face challenges related to high computational cost and the need for prior knowledge or precise characterization of the source and sink states.

Recent progress in out-of-distribution (OOD) detection from the field of trustworthy artificial intelligence (AI) offers a promising new paradigm for transition state identification. OOD detection [162] addresses a fundamental challenge in machine learning, i.e., how to ensure a model trained on a closed-world dataset (in-distribution, ID) behaves reliably when exposed to open-world, unseen data (out-of-distribution, OOD). In critical applications such as autonomous driving or medical diagnosis [1, 35, 78, 102, 132], identifying OOD samples is essential to prevent overconfident mispredictions. The conceptual analogy to molecular dynamics is compelling: just as OOD data deviate from the ID data learned by the model, transition state conformations always exhibit a distributional shift to the dense free energy basins. Hence, transition states can naturally be viewed as OOD data relative to the metastable states.

Recently, Ming et al. proposed a Compactness and DispErsion Regularized learning framework (CIDER) [101], designed for OOD detection for images. In this approach, latent representations are constrained to lie on a hypersphere (so-called hyperspherical embeddings) and are regularized through the joint optimization of compactness and dispersion losses. In detail, the compactness loss encourages samples from the same class to cluster closely on this hypersphere, whereas the dispersion loss drives class prototypes far away from each other. Consequently, OOD samples tend to appear in the middle between class prototypes and can easily be detected through cosine similarity–based measures. This conceptual foundation of CIDER provides inspiration for our development of a deep learning framework to identify transition state structures in protein conformational changes. For biomolecular dynamics, metastable free energy basins can be viewed as analogous to class prototypes that should be separated from each other, while the transition state structures, similar to OOD samples, are expected to lie between these basins. Nevertheless, the direct application of CIDER to MD data is nontrivial, since CIDER was originally formulated in a supervised learning context and relies on labeled data to define class prototypes. In contrast, MD trajectories are inherently unlabeled. VAMPnets [98] provide a natural bridge between supervised and unsupervised paradigms by generating the metastable state assignment for each conformation. In addition, its VAMP-2 loss function, which is specifically designed to capture the system's slowest dynamic modes with separation of timescales, can effectively compress metastable conformations within each basin, thereby playing a similar role to the compactness loss in CIDER.

In this chapter, we introduce a deep learning framework termed Transition State identification via Dispersion and vAriational principle Regularized neural networks (TS-DAR) [87] for simultaneously and automatically detecting all transition state structures from MD simulations. TS-DAR represents biomolecular conformations in a hyperspherical latent space, where the learning objective combines two complementary components: the VAMP-2 loss and the dispersion loss. Minimizing the VAMP-2 loss compacts conformations belonging to the same metastable state (or free energy basin) into localized regions on the hypersphere, whereas the dispersion loss promotes separation between these basins. Consequently, conformations corresponding to transition states, located in between free energy basins, can be identified in an unsupervised manner within the latent space. A key advantage of embedding the latent representations onto a hyperspherical manifold lies in its compact geometry, which inherently prevents the dispersion loss from driving data infinitely apart. This property ensures a uniform distribution of metastable states across the hypersphere and enables the simultaneous detection of all transition-state structures using a simple cosine similarity-based criterion. The performance and generality of TS-DAR are demonstrated across three representative systems: the two-dimensional Müller potential, alanine dipeptide, and the translocation of a DNA motor protein along double-stranded DNA (dsDNA).

4.2 Methods

4.2.1 TS-DAR Architecture

The overall architecture of TS-DAR is illustrated in Figure 4.1a. TS-DAR first takes time-lagged transition pairs of MD conformations, $(\mathbf{x}_t, \mathbf{x}_{t+\tau})$, as input. These pairs are processed by two parallel encoders with shared parameters to produce Softmax probabilities that represent the fuzzy assignments of each conformation to different metastable states, analogous to VAMPnets. Then, TS-DAR introduces an L2-norm/rescale layer (Figure 4.1b) that normalizes the penultimate feature vectors $\tilde{\mathbf{z}}$ and scales them by a constant γ, thereby constraining the embeddings to lie on a hypersphere of radius γ, referred to as the hyperspherical embeddings $\mathbf{z}$.

$$\mathbf{z} = \gamma \frac{\tilde{\mathbf{z}}}{||\tilde{\mathbf{z}}||} \tag{4.1}$$

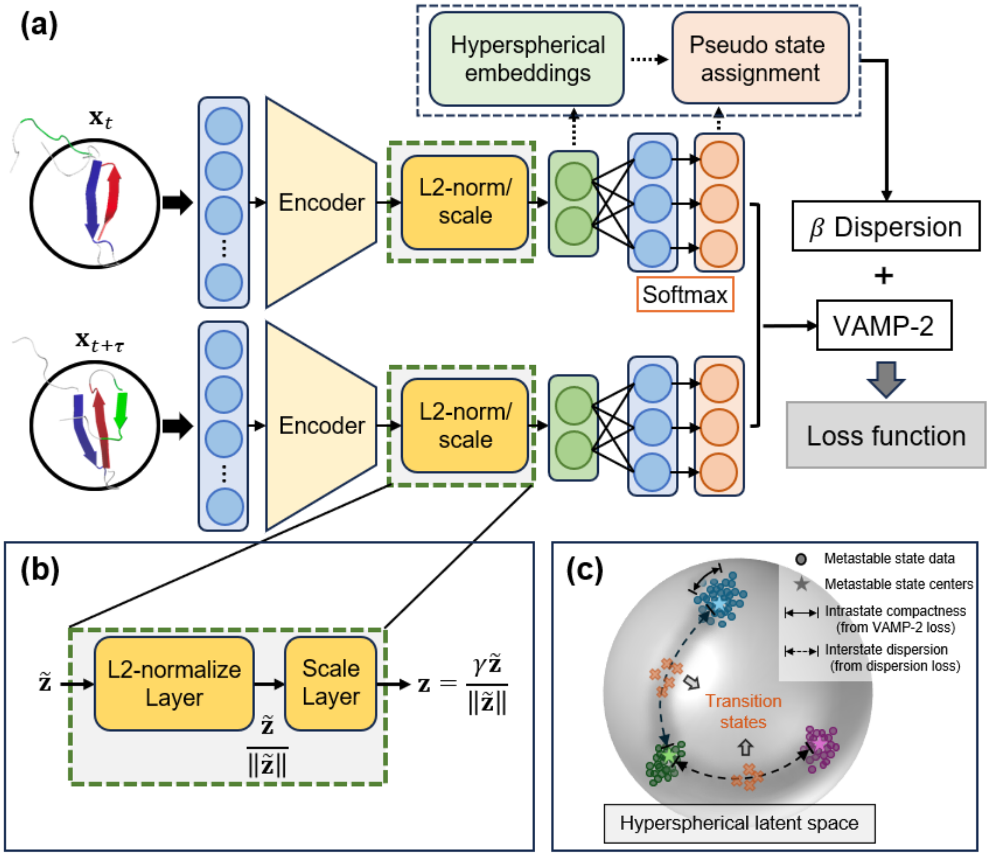


Figure 4.1: Schematic of TS-DAR for transition state identification. (*a*) *TS-DAR architecture. Analogous to VAMPnets* [98], *TS-DAR ingests time-lagged transition pairs* $(\mathbf{x}_t, \mathbf{x}_{t+\tau})$, *produces Softmax state assignments, and optimizes the VAMP-2 objective. A dedicated* ℓ_2*-norm/scale layer at the penultimate layer yields hyperspherical embeddings, which, together with pseudo-assignments from the Softmax outputs, define the dispersion term. The whole network is optimized by minimizing* $\mathcal{L} = \mathcal{L}_{\mathrm{VAMP}} + \beta\mathcal{L}_{\mathrm{dis}}$. (*b*) *Hyperspherical constraint. Raw features* $\tilde{\mathbf{z}}$ *at the penultimate layer are normalized and scaled to a constant* γ, *producing embeddings* $\mathbf{z}$ *constrained to the hypersphere with radius* γ. (*c*) *Transition state identification. The VAMP-2 loss compacts conformations within metastable states, whereas the dispersion loss promotes metastable state centers far away from each other and uniformly distributed across the hypersphere. Transition state conformations lie between metastable states within the latent hypersphere can be simutaneously detected.*

To demonstrate the metastable state assignments and hyperspherical representations of MD trajectories from TS-DAR, we use the two-dimensional Müller potential with three minima as a representative example (Figure 4.2a). In the latent space learned by the trained TS-DAR model (Figure 4.2b), all MD conformations are mapped

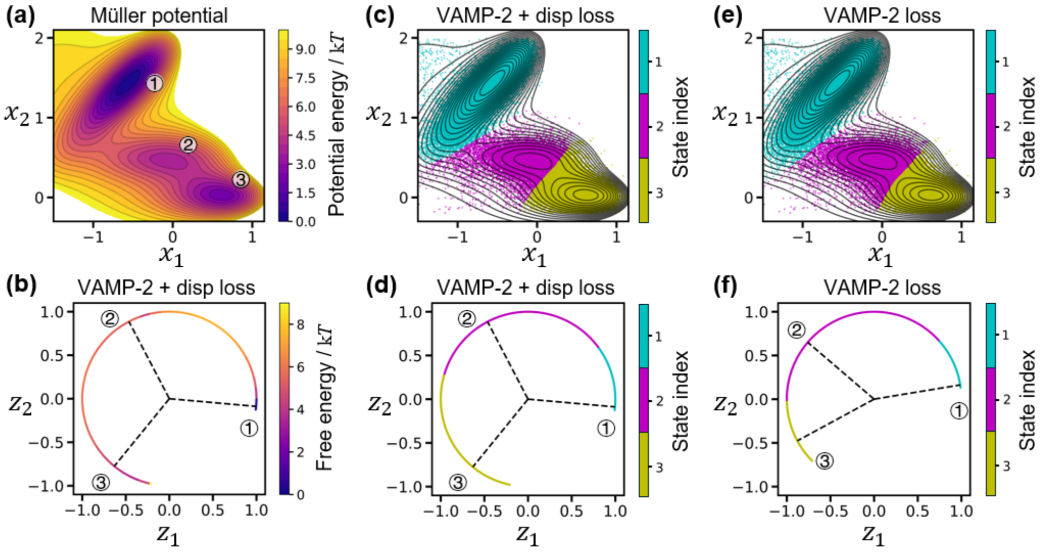


Figure 4.2: Demonstration of TS-DAR on the Müller potential for learning hyperspherical latent representations. *(a) The two-dimensional Müller potential. (b) Projection of MD conformations onto the latent hyperspherical space. The corresponding free energy, $-k_BT \ln P(\theta)$, is shown, where k_BT denotes the thermal energy and $P(\theta)$ represents the probability density of MD conformations at polar angle θ on the hypersphere. The dashed lines indicate the mean direction vectors of the three metastable states. (c) Visualization of the output state assignments (states 1–3) from the TS-DAR model overlaid on the Müller potential. (d) Same as (b), except that the latent-space state assignments, rather than the potential of mean force, are displayed. (e) and (f) are analogous to (c) and (d), respectively, but correspond to a control experiment in which only the VAMP-2 loss was used during TS-DAR training.*

precisely onto the surface of a hypersphere. Three distinct free energy basins are clearly resolved on this hypersphere, with the dashed lines indicating the mean vectors of each basin. Moreover, these three basins in the latent space correspond directly to the three energy minima of the Müller potential (basins 1–3 in Figure 4.2a) and are accurately classified (Figures 4.2c–d).

4.2.2 Objective Functions for Hyperspherical Embedding Regularization

TS-DAR jointly optimizes two complementary objectives to regularize its hyperspherical latent space for robust tansition states identification: the VAMP-2 loss, which captures the slow dynamics of the MD trajectories, and the dispersion loss, which enforces uniform angular separation between metastable state centers on the hypersphere (Figure 4.1).

VAMP-2 loss. The VAMP-2 loss ($\mathcal{L}_{\text{VAMP}}$) is derived from VAMP theory [156] and provides a measure of how well the model preserves the slowest dynamic modes underlying the conformational changes of interest (Section 3.2.3). Given a set of basis functions $\boldsymbol{\chi} = [\chi_1, \chi_2, \ldots, \chi_m]^\top$, which is parameterized by the Softmax outputs of two parallel networks with shared parameters from TS-DAR, an MD trajectory of length T ($\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$), the mean-removed time-instantaneous and time-lagged correlation matrices are computed as

$$\begin{aligned}
\bar{\mathbf{C}}_{00} &= \frac{1}{T-\tau}\mathbf{X}^\top\mathbf{X} - \boldsymbol{\pi}_0\boldsymbol{\pi}_0^\top, \\
\bar{\mathbf{C}}_{11} &= \frac{1}{T-\tau}\mathbf{Y}^\top\mathbf{Y} - \boldsymbol{\pi}_1\boldsymbol{\pi}_1^\top, \\
\bar{\mathbf{C}}_{01} &= \frac{1}{T-\tau}\mathbf{X}^\top\mathbf{Y} - \boldsymbol{\pi}_0\boldsymbol{\pi}_1^\top,
\end{aligned} \tag{4.2}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are two $(T-\tau)\times m$ matrices constructed as $\mathbf{X} = [\boldsymbol{\chi}(\mathbf{x}_1), \ldots, \boldsymbol{\chi}(\mathbf{x}_{T-\tau})]^\top$ and $\mathbf{Y} = [\boldsymbol{\chi}(\mathbf{x}_{1+\tau}), \ldots, \boldsymbol{\chi}(\mathbf{x}_T)]^\top$. Here, τ denotes the lag time, and $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_1$ represent the mean feature vectors at times t and $t+\tau$, respectively, defined as $\boldsymbol{\pi}_0 = \frac{1}{T-\tau}\mathbf{X}^\top\mathbf{1}$ and $\boldsymbol{\pi}_1 = \frac{1}{T-\tau}\mathbf{Y}^\top\mathbf{1}$.

The VAMP-2 loss is then defined as

$$\mathcal{L}_{\text{VAMP}} = -\|\bar{\mathbf{C}}_{00}^{-1/2}\bar{\mathbf{C}}_{01}\bar{\mathbf{C}}_{11}^{-1/2}\|_F^2 - 1 \tag{4.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Since the VAMP-2 loss drives the network to capture the slowest dynamical processes, minimizing $\mathcal{L}_{\text{VAMP}}$ enables the Softmax outputs of TS-DAR to yield optimal probabilistic state assignments that are well aligned with the underlying free energy basins, thereby allowing the on-the-fly labeling of MD conformations during training (Figures 4.2c–d). Furthermore, the incorporation of the VAMP-2 loss term ensures that the hyperspherical latent representations preserve all relevant kinetic geometries while compacting conformations according to their kinetic metastability. This behavior is clearly demonstrated in the latent space representation of the Müller potential (Figure 4.2d), where the three metastable basins and their connectivity correspond precisely to the three free energy minima observed in the potential surface (Figure 4.2c). Notably, the largest basin (basin 1 in Figure 4.2a), characterized by the highest metastability, appears more compact in the latent space, resulting in a smaller yet deeper minimum in the latent free energy profile (Figure 4.2b). This behavior arises from the presence of a single fully connected layer connecting the latent bottleneck to the output layer, which constrains the latent representations to capture the slowest dynamics through the VAMP-2 loss optimization. Such a design of the penultimate layer for representation learning has demonstrated substantial effectiveness across various deep learning domains, including computer vision and natural language processing [22, 152].

To further elucidate the distinct roles of the VAMP-2 and dispersion losses, a control experiment was performed by training TS-DAR without the dispersion loss. The overall kinetic compactness of the basins and their connectivity remained consistent (Figures 4.2e–f); however, the distribution of state centers became noticeably uneven on the hypersphere. This observation highlights that while the VAMP-2 term governs kinetic compactness, the dispersion term is essential for ensuring uniform angular separation among metastable states—a point discussed in detail in the following section.

(2) Dispersion loss. To ensure the learned metastable state centers are uniformly distributed on the hypersphere, a dispersion loss term $\mathcal{L}_{\text{dis}}$ is introduced. This term maximizes the pairwise angular distances between the mean vectors of each metastable state across the hypersphere. For example, when the dispersion loss is included (Figure 4.2d), the centers of the three free energy basins of the Müller potential (labeled as states 1–3) are well separated and uniformly distributed across the hyperspherical latent space. In sharp contrast, when the dispersion loss is omitted (Figure 4.2f), the three state centers become unevenly spaced, leading to a nonuniform distribution of metastable states in the latent representation. The dispersion loss is defined as follows

$$\mathcal{L}_{\text{dis}} = \frac{1}{C}\sum_{i=1}^{C}\log\left(\frac{1}{C-1}\sum_{j=1}^{C}\mathbf{1}_{j\neq i}\exp\left(\frac{\boldsymbol{\mu}_i^{\top}\boldsymbol{\mu}_j}{\sigma}\right)\right) \tag{4.4}$$

Here, C denotes the number of metastable states, $\boldsymbol{\mu}_c$ is a unit vector representing the mean direction of all conformations (i.e., the state center) in state c, and σ is a scaling hyperparameter, set to 0.1 in this study. To compute the dispersion loss, the state-center vectors $\{\boldsymbol{\mu}_c\}_{c=1}^{C}$ must first be estimated. For training stability and computational efficiency, we employ an exponential moving average (EMA) [81] to estimate $\{\boldsymbol{\mu}_c\}_{c=1}^{C}$ on-the-fly, updating them frequently during optimization.

In detail, we denote the hyperspherical embeddings of the conformations as $\{\mathbf{z}_i\}_{i=1}^{N}$, where $\mathbf{z}_i \in \mathbb{R}^d$, $\{\boldsymbol{\mu}_c\}_{c=1}^{C}$ are updated as follows

$$\boldsymbol{\mu}_c \coloneqq \text{Normalize}\big(\theta\,\boldsymbol{\mu}_c + (1-\theta)\,\mathbf{z}_i\big), \quad c = \tilde{y}_i \tag{4.5}$$

where $\tilde{y}_i \in \{1,2,\ldots,C\}$ denotes the state index (pseudo-label) of conformation i, obtained from the Softmax outputs of the TS-DAR neural network. The parameter θ is the state-center update factor, set to 0.5 in this work.

The total TS-DAR loss function therefore combines the two terms as

$$\mathcal{L}_{\text{TS-DAR}} = \mathcal{L}_{\text{VAMP}} + \beta\,\mathcal{L}_{\text{dis}} \tag{4.6}$$

where β (typically 0.01–0.05) (Figure 4.3) balances kinetic compactness and geometric uniformity for the regularization of hyperspherical embeddings. The pseudo-code of training TS-DAR is shown in Algorithm 1. In addition, we have provided a step-by-step tutorial on how to build TS-DAR in biomolecular dynamics [50].

4.2.3 Transition State Identification via Out-of-Distribution Detection within the Regularized Hypersphere

.

After obtaining the regularized hyperspherical latent representations through the joint optimization of the VAMP-2 and dispersion losses, the metastable states become

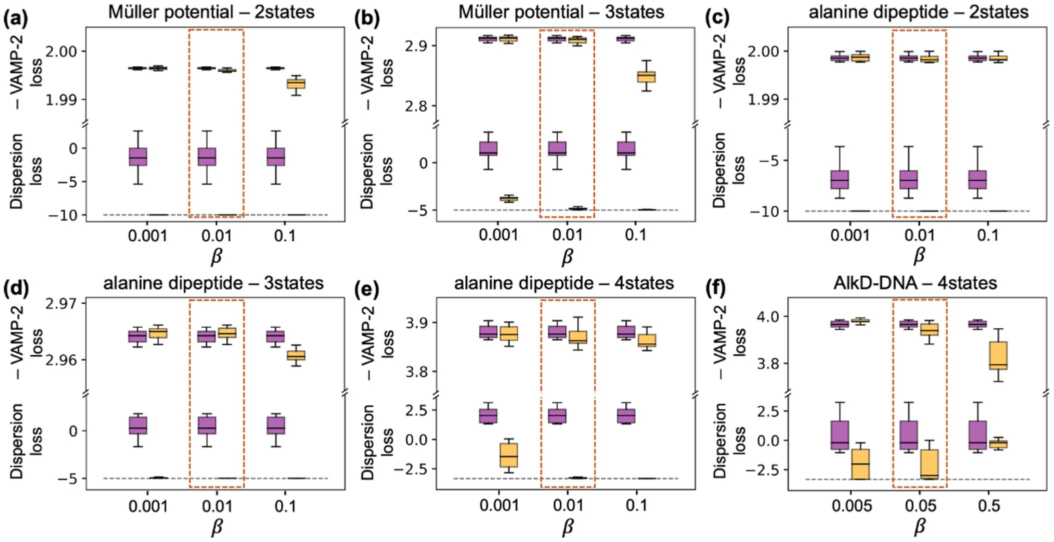


Figure 4.3: Analysis of the dispersion loss weight (β). *Box plots are generated from ten repeated TS-DAR training runs using a 90%/10% train/validation split, and losses are evaluated on the validation sets. Purple boxes represent the losses from pre-trained TS-DAR models (VAMP-2 loss only), while yellow boxes correspond to fully trained TS-DAR models (VAMP-2 + dispersion loss). The gray dashed lines indicate the theoretical lower bounds of the dispersion loss, and the selected β values used in this study are outlined by orange dashed boxes. The selection criteria are: (1) The dispersion loss can converge to the minimum boundary. (2) There is no significant deviation in VAMP-2 loss before and after integrating the dispersion loss optimization. Each box plot shows the following statistics (from bottom to top): $(Q_1 - 1.5\,\mathrm{IQR})$, Q_1, median, Q_3, and $(Q_3 + 1.5\,\mathrm{IQR})$, where Q_1, Q_3, and IQR denote the first quartile, third quartile, and interquartile range, respectively. (a) TS-DAR two-state model on the Müller potential. (b) TS-DAR three-state model on the Müller potential. (c) TS-DAR two-state model on alanine dipeptide. (d) TS-DAR three-state model on alanine dipeptide. (e) TS-DAR four-state model on alanine dipeptide. (f) TS-DAR four-state model on AlkD–DNA.*

compact and uniformly separated in the latent space. Consequently, the transition states located between these metastable basins exhibit approximately equal angular distances to their nearest state centers (Figure 4.2d). This observation motivates the definition of an OOD score based on the cosine similarity (or angular distance) between a latent embedding and all state centers, which quantifies its angular deviation in the hyperspherical space to its nearest metastable state center:

$$\text{OOD score} = -\max\{\mathbf{z}^\top \mathbf{U}\} + 1 \tag{4.7}$$

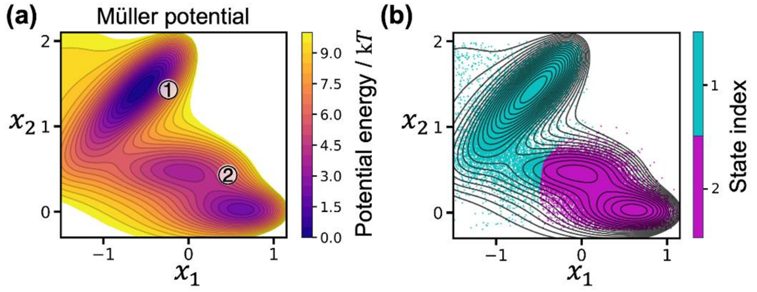


Figure 4.4: Identification of transition states based solely on state boundaries fails to capture the transition state corresponding to the highest energy barrier in the Müller potential. (*a*) *The Müller potential energy surface.* (*b*) *Visualization of the TS-DAR output state assignments* (*states 1 and 2*) *overlaid on the Müller potential.*

where $\mathbf{z}$ denotes the hyperspherical latent embedding of a conformation, and $\mathbf{U} = [\mu_1, \mu_2, \ldots, \mu_C]$ represents the matrix of metastable state centers.

By this definition, the OOD score ranges from a minimum from 0, with higher values indicating greater out-of-distribution characteristics and thus identifying conformations that likely reside in transition state regions. As shown in Figure 4.7b, MD conformations located between basins 1–2 and basins 2–3 of the Müller potential (Figure 4.2a) exhibit equally large OOD scores, allowing both sets of transition states to be identified simultaneously. We subsequently selected conformations with OOD scores above a chosen threshold as transition-state structures. As illustrated in Figure 4.7c, the conformations identified at various OOD thresholds consistently exhibit committor probabilities centered around 0.5 for both TS12 and TS23. Because a committor value of 0.5 theoretically defines the dividing surface between reactant and product basins, these results confirm that TS-DAR accurately and robustly predicts transition state structures. Moreover, increasing the OOD threshold from 0.225 to 0.425 decreases the number of detected transition-state conformations from approximately 1,000 to 300 for TS12 and from 3,500 to 300 for TS23 (Figure 4.7c). This tunable threshold therefore provides a practical means to control the number of transition state structures extracted for further analysis.

Previous studies [162] in the field of trustworthy AI have employed various similarity-based metrics on latent representations for OOD detection. However, these approaches are not directly applicable to identifying transition states in protein conformational changes. For instance, one could attempt to detect OOD samples

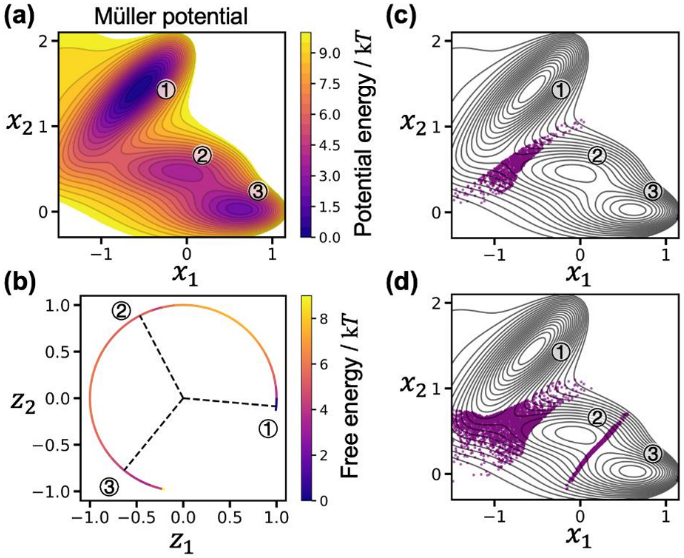


Figure 4.5: Identification of transition states based on local density in the hyperspherical latent space fails to simultaneously detect both transition states of the Müller potential. *(a) The Müller potential energy surface. (b) Projection of MD conformations onto the latent hyperspherical space, with the corresponding free energy profile* ($-k_BT \ln P(\theta)$) *shown, where* k_BT *denotes the thermal energy and* $P(\theta)$ *represents the probability of MD conformations at polar angle* θ *on the 2D hypersphere. The dashed lines indicate the mean vectors of the three metastable states. (c) Transition state conformations identified by low-density regions with* $P < \exp(-7.2k_BT)$ *are highlighted in purple. (d) Transition state conformations identified using a relaxed density threshold of* $P < \exp(-6k_BT)$ *are shown in purple.*

located near the state boundaries on the hypersphere and treat them as transition-state structures. Nevertheless, this strategy is not well suited for TS-DAR, because the VAMP-2 loss is specifically designed to capture the slowest dynamic modes, making the state boundaries relatively insensitive and therefore less effective in pinpointing transition state conformations. As shown in Figure 4.4b, the MD conformations located at the state boundaries of the Müller potential do not accurately correspond to true transition states.

Recent methods including CIDER [101, 139] detect OOD samples by identifying low-density regions in the latent space, which may, in principle, coincide with transition state regions. However, these density-based approaches can overlook transition states separated by relatively low free energy barriers or misclassify sparse regions as transition states. For example, in the Müller potential system, selecting an appropriate density threshold that simultaneously identifies both transition states is challenging. When the threshold is set to $\exp(-7.2k_{\mathrm{B}}T)$, only the transition state between states 1 and 2 is detected (Figure 4.5c). Reducing the threshold to $\exp(-6k_{\mathrm{B}}T)$ enables the identification of the transition state between states 2 and 3, but in this case, the boundary between states 1 and 2 becomes poorly defined (Figure 4.5d).

Algorithm 1: Pseudo-code for training TS-DAR

```
Input: MD trajectories; two parallel NNs with shared params (encoder $\mathbf{f}$, classifier $\chi$); state centers (prototypes) $\{\mu_j\}_{j=1}^{C}$ (init = $\mathbf{0}$); dispersion weight $\beta$; scaling factor $\gamma$; scaling hyperparameter $\sigma$; center update factor $\theta$; lag time $\tau$.
for epoch = 1,2,... do
    for iter = 1,2,... do
        - Sample a batch of transition pairs with lag $\tau$: $\{(\mathbf{x}_i, \mathbf{x}_{i+\tau})\}_{i=1}^{b}$;
        - Build two batches $\mathbf{B} = [\mathbf{x}_1, \ldots, \mathbf{x}_b]^\top$ and $\hat{\mathbf{B}} = [\mathbf{x}_{1+\tau}, \ldots, \mathbf{x}_{b+\tau}]^\top$; feed them to the two shared-parameter lobes;
        - From Softmax outputs $[\chi(\mathbf{x}_1), \ldots, \chi(\mathbf{x}_b)]^\top$ and $[\chi(\mathbf{x}_{1+\tau}), \ldots, \chi(\mathbf{x}_{b+\tau})]^\top$, compute mean-removed covariance matrices $(\bar{\mathbf{C}}_{00}, \bar{\mathbf{C}}_{01}, \bar{\mathbf{C}}_{11})$ (Eq. (4.2));
        - Compute VAMP-2 loss $\mathcal{L}_{\text{VAMP}} = -\left\|\bar{\mathbf{C}}_{00}^{-1/2}\,\bar{\mathbf{C}}_{01}\,\bar{\mathbf{C}}_{11}^{-1/2}\right\|_F^2 - 1$;
        if pre-training stage then
            Update network weights; continue;
        end
        - Obtain pseudo-labels $\tilde{y}_i = \arg\max(\chi(\mathbf{x}_i))$, $\tilde{y}_i \in \{1, \ldots, C\}$, and assign to $\mathbf{x}_i$;
        for $\mathbf{x}_i \in \mathbf{B}$ do
            - Get hyperspherical embedding: $\tilde{\mathbf{z}}_i = \mathbf{f}(\mathbf{x}_i)$, $\mathbf{z}_i = \gamma\,\tilde{\mathbf{z}}_i / \|\tilde{\mathbf{z}}_i\|$;
            - Update center (prototype) for $c = \tilde{y}_i$: $\mu_c \leftarrow \text{Normalize}\big(\theta\,\mu_c + (1-\theta)\,\mathbf{z}_i\big)$;
        end
        - Compute dispersion loss $\mathcal{L}_{\text{dis}} = \frac{1}{C}\sum_{i=1}^{C}\log\left(\frac{1}{C-1}\sum_{j=1}^{C}\mathbf{1}_{j\neq i}\exp\big(\mu_i^\top \mu_j/\sigma\big)\right)$;
        Total loss: $\mathcal{L}_{\text{TS-DAR}} = \mathcal{L}_{\text{VAMP}} + \beta\,\mathcal{L}_{\text{dis}}$; update NN weights;
    end
end
```

Note: VAMP-2 compacts conformations within metastable basins; dispersion encourages uniform angular separation of metastable states on the hypersphere.

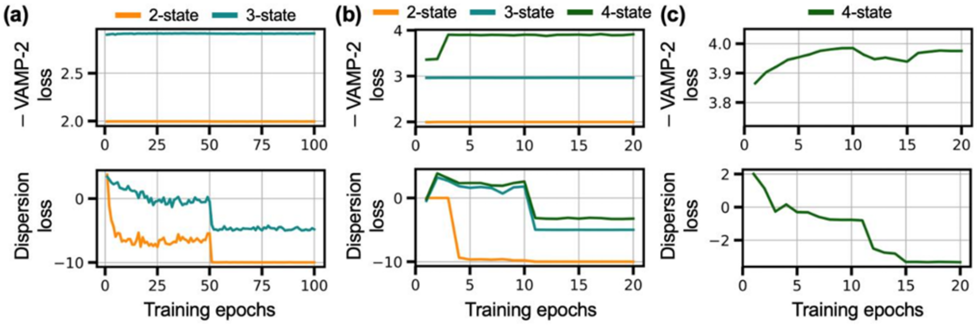


Figure 4.6: Validation curves of TS-DAR. (*a*) *Validation curves of VAMP-2 loss* (*top*) *and dispersion loss* (*bottom*) *for TS-DAR trained on the Müller potential.* (*b*) *Validation curves of VAMP-2 loss* (*top*) *and dispersion loss* (*bottom*) *for TS-DAR trained on the alanine dipeptide dataset.* (*c*) *Validation curves of VAMP-2 loss* (*top*) *and dispersion loss* (*bottom*) *for TS-DAR trained on the AlkD–DNA dataset.*

4.3 Results

4.3.1 Comparison of TS-DAR, MSM, and MaxEnt-VAMPNets in Transition State Identification

The Müller potential is a 2D potential that features three potential energy basins separated by two transition states (Figure 4.2a). The analytical form of the Müller potential [105] is defined as

$$V_{\text{Müller}}(x_1, x_2) = \sum_{i=1}^{4} A_i \exp\left[a_i(x_1 - \bar{x}_i)^2 + b_i(x_1 - \bar{x}_i)(x_2 - \bar{y}_i) + c_i(x_2 - \bar{y}_i)^2\right] \tag{4.8}$$

where $(A_1, \ldots, A_4) = (-10, -5, -8.5, 0.75)$, $(a_1, \ldots, a_4) = (-1, -1, -6.5, 0.7)$, $(b_1, \ldots, b_4) = (0, 0, 11, 0.6)$, $(c_1, \ldots, c_4) = (-10, -10, -6.5, 0.7)$, $(\bar{x}_1, \ldots, \bar{x}_4) = (1, 0, -0.5, -1)$, $(\bar{y}_1, \ldots, \bar{y}_4) = (0, 0.5, 1.5, 1)$. Brownian dynamics simulations were performed on this potential with a time step of 2×10^{-4} and a damping factor of 1 at a temperature of 0.9. Reflective boundary conditions were applied to the coordinates $x_1 \in [-1.5, 1.2]$ and $x_2 \in [-0.2, 2]$. The simulation trajectory consisted of 3×10^5 frames, saved at intervals of 0.01.

The TS-DAR model was trained using a learning rate of 1×10^{-3} without employing any learning-rate scheduler. The batch size was set to 1,000, and the optimization was performed using the Adam optimizer with a train/validation split of 90%/10%. The

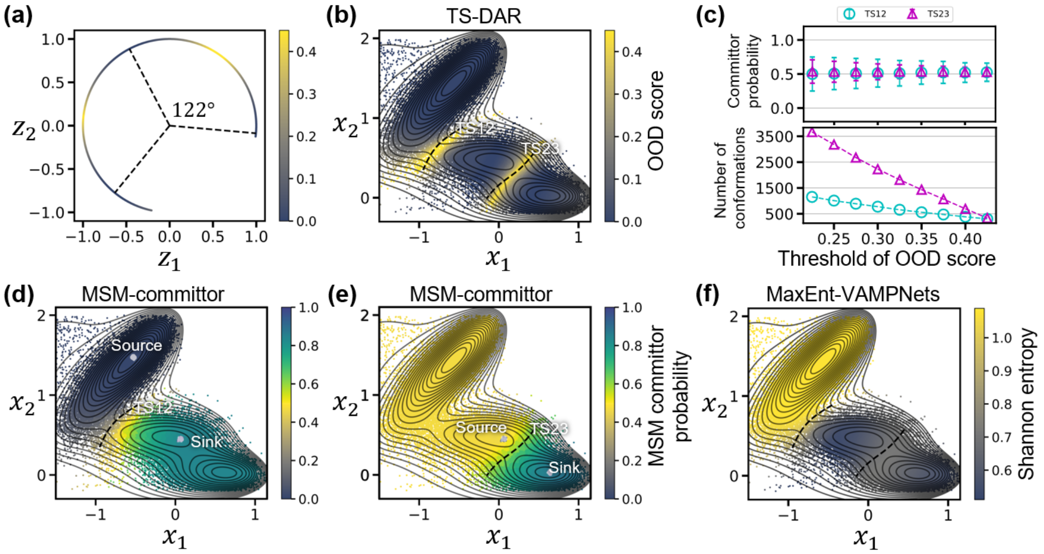


Figure 4.7: TS-DAR outperforms the MSM and MaxEnt-VAMPNets in identifying transition states for the Müller potential. *(a) Hyperspherical latent representations with OOD scores obtained from a three-state TS-DAR model. Dashed lines indicate the centers of the metastable states. (b) MD conformations with OOD scores predicted by TS-DAR, overlaid on the Müller potential energy surface. (c) Average committor probabilities (error bars denote standard deviations) of transition-state structures for TS12 and TS23 selected by TS-DAR using different OOD-score thresholds (top). The corresponding sample sizes (n) for each threshold are shown in the lower panel: [1154, 1011, 890, 776, 657, 560, 474, 384, 304] for TS12 and [3668, 3181, 2678, 2228, 1817, 1435, 1068, 693, 337] for TS23. (d–e) MD conformations with committor probabilities obtained from a 1,000-state MSM, overlaid on the Müller potential. White regions represent the source and sink states, chosen as the most populated states in basins 1 and 2 (panel d) and in basins 2 and 3 (panel e), respectively. (f) MD conformations with Shannon entropy values obtained from a three-state MaxEnt-VAMPNets [69] model, overlaid on the Müller potential. In panels (b), (d), (e), and (f), black dashed lines denote iso-committor contours corresponding to a committor probability of 0.5, obtained from analytical solutions.*

model was first pre-trained for 50 epochs using only the VAMP-2 loss, followed by an additional 50 epochs of joint optimization with both the VAMP-2 and dispersion losses. The weight of the dispersion loss (β) was set to 0.01 (see Figure 4.3 for validation), and the scaling factor (γ) of the hyperspherical embeddings was fixed at 1. The scaling hyperparameter (σ) in the dispersion loss was 0.1, and the state-center update factor (θ) was 0.5. The lag time (τ) was defined as one simulation saving interval. The dimensionality of the hyperspherical latent space (d) was set to 2, corresponding to

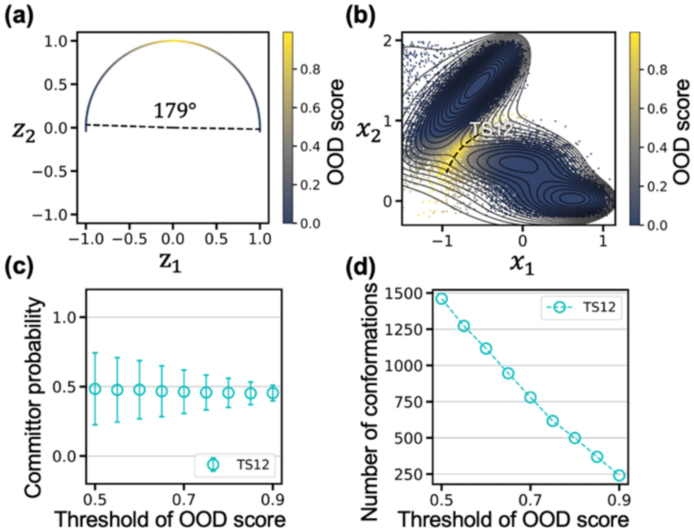


Figure 4.8: Identification of transition states with a two-state TS-DAR model for the Müller potential. (*a*) *Hyperspherical latent representations with OOD scores obtained from the two-state TS-DAR model. Dashed lines indicate the centers of the two metastable states.* (*b*) *MD conformations colored by their OOD scores are overlaid on the Müller potential. The black dashed line corresponds to the analytical iso-committor contour with a committor probability of 0.5.* (*c*) *Average committor probabilities* (*error bars denote standard deviations*) *of the MD conformations identified as TS*12 *using different OOD-score thresholds. The sample sizes* (*n*) *used to compute the standard deviations correspond to the number of selected conformations shown in panel* (*d*)*:* [*1,461, 1,273, 1,117, 946, 780, 617, 499, 369, 241*]*.* (*d*) *Number of MD conformations identified as TS*12 *under different OOD-score thresholds.*

embeddings on a $(d-1)$-dimensional sphere. The encoder architecture consisted of a fully connected neural network with layer widths of [2, 20, 20, 20, 10, 2], where the final hidden layer dimension corresponds to the latent-space dimensionality d. ReLU activation functions were used throughout the encoder, followed by an L2-norm/scale layer to generate the hyperspherical embeddings and a linear transformation layer that maps the latent features to C output nodes representing the metastable states, followed by a SoftMax function. The validation curves of TS-DAR training is shown in Figure 4.6a.

We first trained the TS-DAR model by specifying three metastable states. Using the trained model, we first projected all MD conformations onto the latent hyperspherical space and visualized their OOD scores (Figure 4.7a). As shown in Figure 4.7b, MD conformations located between the latent state centers correspond closely to those

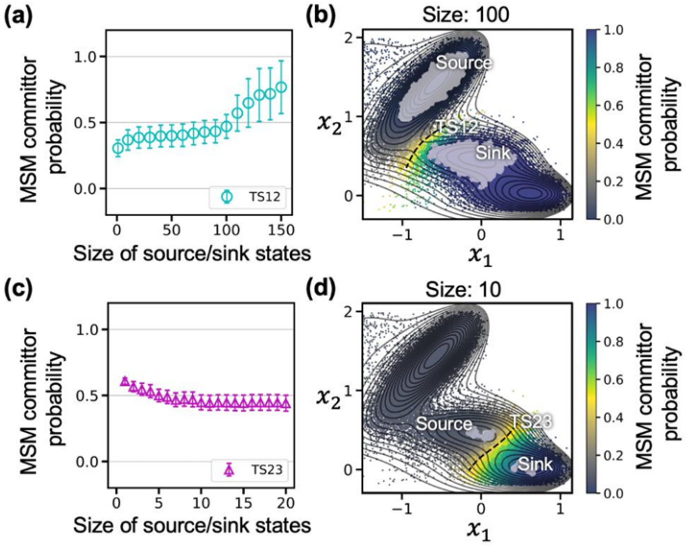


Figure 4.9: Identification of transition states using MSM-committor analysis for the Müller potential. (*a*) *Average committor probabilities* (*error bars denote standard deviations*) *obtained from MSM-committor analysis for MD conformations located in the transition-state region TS*12, *shown as a function of the source/sink-state size* (*i.e., the number of highly populated microstates in each source or sink set*). *The MD conformations at the transition state* ($n = 293$) *are selected based on the analytical solution, with committor probabilities between 0.4 and 0.6.* (*b*) *MD conformations with committor probabilities from the 1,000-microstate MSM for TS*12 *overlaid on the Müller potential. The source and sink states each contain 100 microstates.* (*c*) *Same as* (*a*), *but for the transition-state region TS*23 ($n = 1{,}393$). (*d*) *Same as* (*b*), *but for TS*23.

situated at the summits of the energy barriers, thereby enabling the straightforward identification of transition-state structures. In this three-state model, both transition states separating the three energy basins are simultaneously captured (Figure 4.7b). Notably, the spatial distribution of the OOD scores obtained from TS-DAR exhibits remarkable agreement with the analytical iso-committor surface corresponding to a committor probability of 0.5 (Figures 4.7b,c), confirming that the transition-state conformations identified by our method coincide with the true dynamical dividing surface of the Müller potential. Moreover, when focusing only on the transition state associated with the highest energy barrier (TS12), a two-state TS-DAR model can also accurately identify this transition state (Figure 4.8).

We next demonstrate that TS-DAR outperforms two previously developed methods, i.e., MaxEnt-VAMPNets [69] and the committor-probability analysis from Markov state models (MSM-committor) [106], in identifying transition states for the Müller potential. For MaxEnt-VAMPNets, it is an adaptive sampling method that identifies rare or transition state conformations based on the Shannon entropy of the SoftMax outputs from VAMPnets. The Shannon entropy is defined as

$$H(\mathbf{x}) = -\sum_{i=1}^{C} p_i(\mathbf{x}) \log p_i(\mathbf{x}) \tag{4.9}$$

where C denotes the total number of states and $p_i(\mathbf{x})$ represents the SoftMax probability of assigning a conformation to state i. Here, we implemented MaxEnt-VAMPNets using the same training hyperparameters and network architecture as TS-DAR, except for removing the hidden layer.

For MSM-committor, the committor probability, q_i, of a dynamical system mapped onto a discrete state space represents the likelihood that the system, when starting from state i, will reach the sink state (B) before the source state (A). By definition, states with a committor probability of 0.5 are considered transition states, as they possess an equal probability of proceeding toward either the source or the sink. A standard approach for computing state-based committor probabilities involves constructing a microstate MSM from which q_i can be determined by solving the following set of linear equations according to transition path theory (TPT) [106]:

$$-q_i + \sum_{k \in I} T_{ik}\, q_k = -\sum_{k \in B} T_{ik} \tag{4.10}$$

where $T_{ik} = P(x_{t+\tau} \in k \,|\, x_t \in i)$ denotes the transition probability from state i to state k at lag time τ, provided by the microstate MSM. Here, the configurational space was first partitioned into 1,000 microstates using k-centers clustering. A transition count matrix (TCM) was then constructed using a lag time of one simulation step. To enforce detailed balance, the TCM was symmetrized by averaging it with its transpose. The transition probability matrix (TPM), $\mathbf{T}$, was subsequently obtained by normalizing each row of the symmetrized TCM. Finally, the committor probabilities q_i were computed according to Eq. (4.10). To define the source and sink states, the most highly populated microstates corresponding to basins 1, 2, and 3 of the Müller potential were selected.

As shown in Figure 4.7f, MaxEnt-VAMPNets erroneously identified energy basin 1, rather than the true energy barrier, as the transition-state region characterized by high Shannon entropy. This misidentification likely arises because the Shannon entropy derived from VAMPnets' output probabilities does not directly correspond to the physical uncertainty associated with transitions into or out of metastable basins. For MSM-committor analysis, the performance strongly depends on the choice of source and sink states (Figure 4.9a,c). When both states are defined narrowly (each consisting of a single microstate), MSM-committor can correctly locate the transition-state region for TS12, although the predicted committor probabilities for the true transition-state structures are relatively low (approximately 0.3) compared to the analytical solution (Figure 4.7d). However, for TS23, MSM-committor incorrectly predicts an overly broad transition-state region encompassing even the entire energy basin 1 (Figure 4.7e). In this case, since microstates within basin 1 are distant from both the source and sink states, they exhibit comparable probabilities of reaching either, thereby erroneously classified as transition states. When larger source and sink states are selected (e.g., comprising the 100 and 10 most populated microstates for TS12 and TS23, respectively), MSM-committor can recover the correct transition-state regions (Figures 4.9b,d).

4.3.2 Transition States for Alanine Dipeptide

Alanine dipeptide, composed of ten heavy atoms, exhibits conformational changes that are conventionally described by two backbone torsion angles, ϕ and ψ (Figures 4.10a,b). To investigate it, we used a MD simulation dataset obtained from a previous study [109]. The dataset consists of three independent 250 ns MD trajectories, recorded at 1 ps intervals, resulting in a total of 750,000 conformations. All conformations were aligned to the first frame by minimizing the root-mean-square deviation (RMSD) of atomic coordinates.

We then trained three TS-DAR models using the Cartesian coordinates of these ten heavy atoms as input features, specifying two, three, and four metastable states, respectively. Specifically, these models were trained using a learning rate of 1×10^{-3} without employing a learning rate scheduler. The batch size was set to 1,000, and the neural network was optimized using the Adam optimizer with a 90%/10%

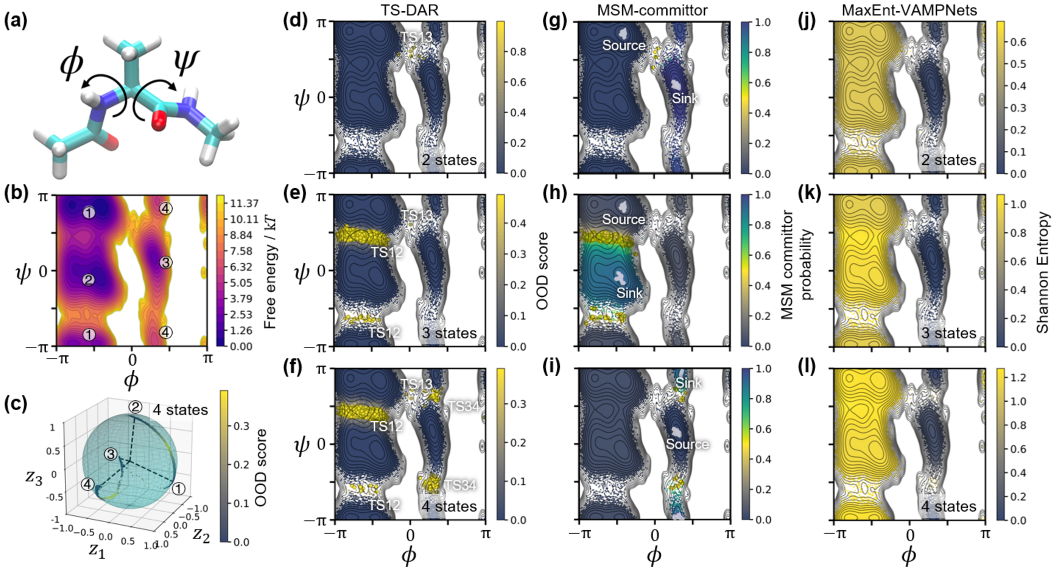


Figure 4.10: TS-DAR identifies transition states for alanine dipeptide. (*a*) *Representative conformation of alanine dipeptide.* (*b*) *Free-energy landscape projected onto the two backbone torsion angles, ϕ and ψ.* (*c*) *Hyperspherical representations in a three-dimensional latent space* (*two-dimensional hypersphere*) *from the four-state TS-DAR model, with dashed lines indicating the centers of metastable states.* (*d–f*) *TS-DAR models with two, three, and four states successfully identify transition states located at different free-energy barriers. For visualization, MD conformations with large OOD scores are highlighted in yellow.* (*g–i*) *MD conformations with committor probabilities obtained from the 1,000-state MSM, overlaid on the free-energy landscape projected onto ϕ and ψ. Conformations with committor probabilities between 0.4 and 0.6 are highlighted, representing transition-state regions. White regions denote the source and sink states used as input for TPT calculations of committor probabilities.* (*j–l*) *Shannon entropy values of MD conformations obtained from two-, three-, and four-state MaxEnt-VAMPNets are shown.*

train/validation split. To prevent overfitting, the model was first pre-trained for 10 epochs using only the VAMP-2 loss and subsequently trained for an additional 10 epochs by jointly optimizing the VAMP-2 and dispersion losses. The scaling factor of the hyperspherical embeddings (γ) was fixed at 1, the scaling hyperparameter in the dispersion loss (σ) was set to 0.1, the dispersion loss weight (β) was 0.01 (see Figure 4.3 for validation), and the state-center update factor (θ) was 0.5. The lag time (τ) was chosen as 1 ps, corresponding to the sampling interval of the MD data. The encoder network consisted of fully connected layers with widths [30, 30, 30, 30, 10, d], where d denotes the latent dimensionality of the hyperspherical embeddings (a

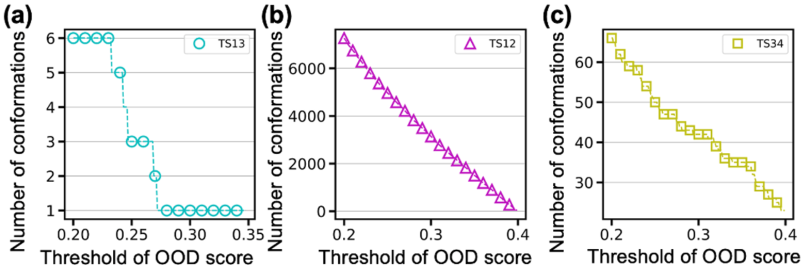


Figure 4.11: Identification of transition states with a four-state TS-DAR model for alanine dipeptide. *The number of MD conformations assigned to the transition states TS13, TS12, and TS34 by the four-state TS-DAR model as a function of the OOD threshold is shown in panels (a), (b), and (c), respectively.*

$(d-1)$-dimensional hypersphere). The activation function was ReLU, and the output layer consisted of a linear transformation followed by a SoftMax function to produce the fuzzy state assignments. For the alanine dipeptide system, the latent dimension was set to $d = 3$ when more than three metastable states were specified and $d = 2$ otherwise. The validation curves of TS-DAR training is shown in Figure 4.6b.

In the two-state TS-DAR model, we successfully identified the transition state TS13, located on the highest free-energy barrier separating the β (basin 1 in Figure 4.10b) and α_L (basin 3 in Figure 4.10b) conformations (Figure 4.10d). The three-state model additionally captured transition states (TS12) between basins 1 and 2 (α_R). Owing to the 2π-periodicity of torsion angles, two equivalent transition-state regions were detected (Figure 4.10e). In the four-state model, all three transition states, i.e., TS12, TS13, and TS34, were simultaneously identified, with TS34 corresponding to a slow mode separating the two right-hand-side basins (Figure 4.10f). Among these, TS13, located at the highest free-energy barrier, was least sampled in the MD trajectories, containing only 6 conformations (Figure 4.11a). By contrast, TS12 was well sampled and could be flexibly extracted by adjusting the OOD threshold: the number of predicted transition-state structures decreased from approximately 7,000 to 30 as the OOD threshold increased from 0.2 to 0.4 (Figure 4.11b).

To evaluate robustness and computational efficiency of TS-DAR in identifying transition states with respect to MD sampling, we further trained three-state TS-DAR models on down-sampled trajectories with total lengths of 3, 30, 90, 300, and 750 ns. As shown in Figure 4.12, the TS-DAR models consistently identified TS12 across all

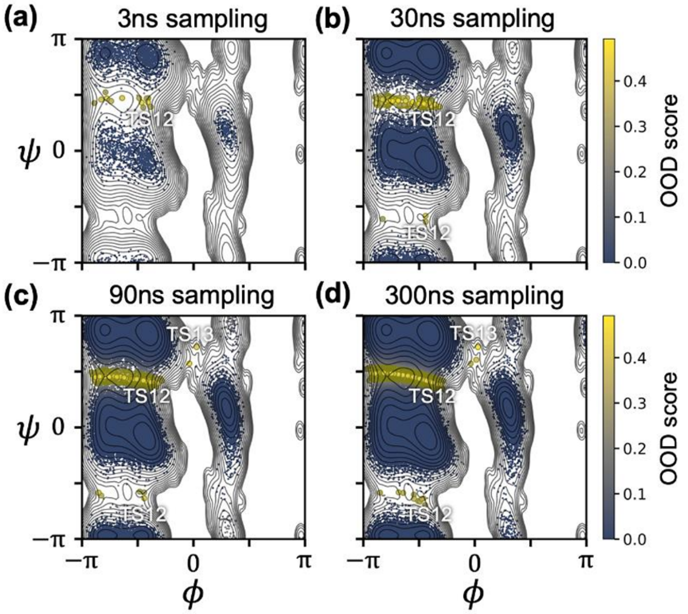


Figure 4.12: Identification of TS12 **from down-sampled datasets using 3-state TS-DAR models for alanine dipeptide.** *Down-sampled datasets of 3, 30, 90, and 300 ns total sampling were generated by uniformly dividing the original three 250 ns MD trajectories into 30 independent segments and truncating each segment to lengths of 0.1, 1, 3, and 10 ns, respectively. TS-DAR models successfully identified TS*12 *across all datasets* (*a–d*), *and additionally identified TS*13 *in the 90 ns and 300 ns sampling datasets* (*c, d*). *Each model was trained using two backbone dihedral angles as input features, an encoder architecture of* [*2, 50, 50, 10, 2*], *exponential linear unit* (*ELU*) *activation functions, and three output nodes corresponding to the metastable states. The batch size was set to 1,000 for all models except for the 3 ns dataset, for which a smaller batch size of 100 was used due to its limited sample size* (*3,000 MD conformations*).

subsets, yielding results in agreement with those from the full dataset (Figure 4.10). Remarkably, TS12 was detectable even from just 3 ns of total sampling, while TS13 required approximately 90 ns for reliable detection. Notably, a prior transition path sampling (TPS) study [11] required ~77 ns to capture TS12.

We further benchmarked TS-DAR against MSM-committor and MaxEnt-VAMPNets. For MaxEnt-VAMPNets, we implemented using the same training hyperparameters and network architecture as TS-DAR, except for removing the hidden layer. For MSM-committor, we first clustered MD conformations into 1,000 microstates using the k-centers clustering algorithm according to the Euclidean distances between pairs

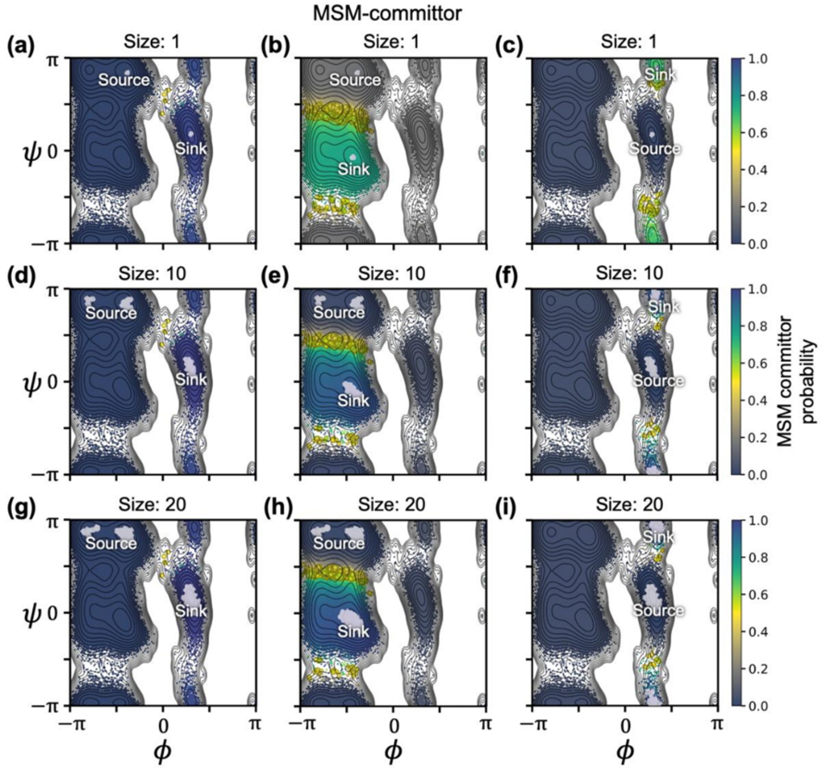


Figure 4.13: Identification of transition states using MSM-committor analysis for alanine dipeptide. *MD conformations with committor probabilities obtained from MSM-committor analysis based on a 1,000-state MSM are overlaid on the free-energy landscape projected onto the two backbone torsion angles of alanine dipeptide. Conformations with committor probabilities between 0.4 and 0.6 are highlighted as red dots, corresponding to the transition-state regions. White regions denote the source and sink states used as input for TPT calculations of committor probabilities. Panels (a–c) show results with the source and sink states each containing a single highest-populated microstate; panels (d–f) use sets of ten highest-populated microstates for the source and sink states; and panels (g–i) use sets of twenty highest-populated microstates.*

of MD conformations on two backbone torsion angles. We then constructed the TCM with the lag time of 1 ps and followed the same procedure as in the Müller potential to obtain q_i. To determine the source and sink states, we selected the top 5 highest populated microstates in basin 1, 2, 3 and 4 (Figure 4.10b), respectively. As shown in Figure 4.10g–i, MSM-committor analysis with different source–sink pairs reproduced three transition-state regions consistent with those identified by the four-state TS-DAR model, though the resulting committor probabilities still depend on the state

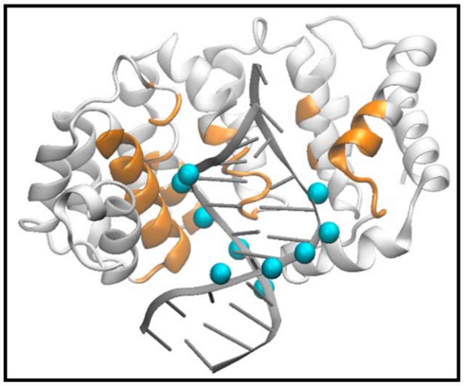


Figure 4.14: AlkD-dsDNA complex structure. *Atoms from the DNA (highlighted in cyan) and the protein helix (highlighted in orange) that were used to compute the pairwise distances as input features of TS-DAR training.*

selection (Figure 4.13). In contrast, MaxEnt-VAMPNets exhibited elevated Shannon entropy not only in the transition-state regions TS12 and TS13 but also within basins 1 and 2, complicating unambiguous identification (Figure 4.10j–l). Notably, in the four-state TS-DAR model, the hyperspherical latent space was three-dimensional (a 2D hypersphere), and the four metastable-state centers formed an approximately tetrahedral geometry with inter-state angular separations of ~109°, demonstrating a uniform and interpretable embedding structure (Figure 4.10c). For systems involving four or more states, we recommend employing a three-dimensional latent space (2D hypersphere) for TS-DAR implementation.

4.3.3 Transition States for the Translocation of a Motor Protein on DNA.

Bacillus cereus alkylpurine glycosylase D (AlkD) is a DNA motor protein that translocates along double-stranded DNA (dsDNA) and repairs alkylated bases to maintain genome integrity [61]. Understanding the translocation mechanism of AlkD is crucial for elucidating how the enzyme efficiently and accurately locates sparse lesion sites along genomic DNA. To study this system, we used a MD simulation dataset from a previous study [111]. The dataset comprises 200 trajectories of 50 ns and 100 trajectories of 45 ns, recorded at 20 ps intervals, resulting in a total of 725,300 conformations.

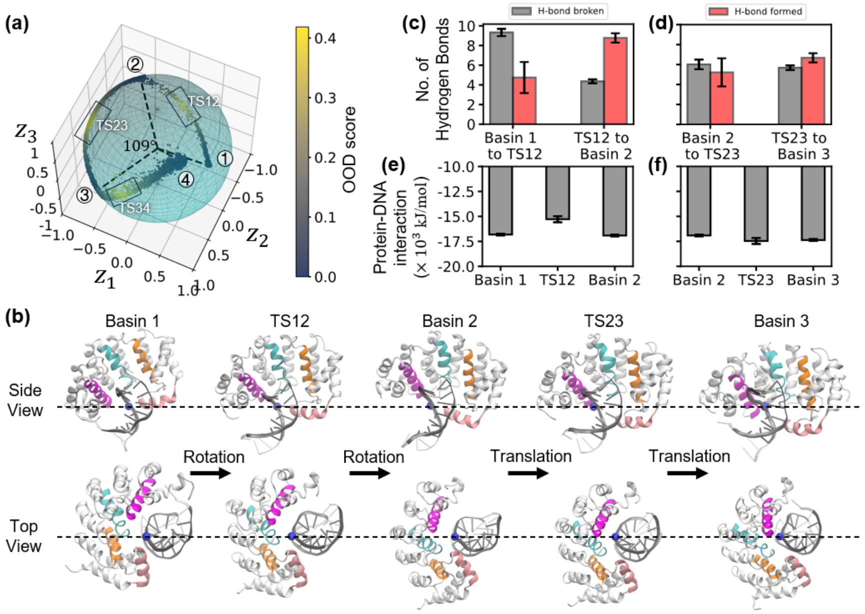


Figure 4.15: TS-DAR identifies the transition states of AlkD translocating along dsDNA by one base pair. *(a) Hyperspherical latent representations of the AlkD–DNA complex obtained from the TS-DAR model. Dashed lines indicate the state center vectors, and the rectangular box outlines the selected transition state structures. (b) Representative conformations of the three free-energy basins and two transition states. The α-helices in contact with the dsDNA are shown in individual colors, and the phosphorus atom of the A7 phosphate group is depicted as a blue sphere. (c) The average numbers of hydrogen bonds broken (black bars) and newly formed (red bars) between AlkD residues and adjacent DNA nucleotides during the transitions from basin 1 → TS12 (left) and TS12 → basin 2 (right). (d) Same as (c), but for the transitions from basin 2 → TS23 and TS23 → basin 3. (e) Interaction energies between AlkD and dsDNA for basin 1, TS12, and basin 2. (f) Same as (e), but for basin 2, TS23, and basin 3. Interaction energies were computed using AlkD residues and DNA ribonucleotides within 3 Å of each other in at least one MD frame. The averages and error bars in panels (c)–(f) were estimated via bootstrapping of the MD trajectories 100 times. For each bootstrap sample, the mean of the analyzed quantity was computed for each basin and transition state. The plotted averages and error bars represent the mean and standard deviation across all 100 bootstrap samples.*

We then employed TS-DAR to investigate the transition states governing AlkD diffusion along dsDNA over a single base-pair step. Following our previous study [111], 684 pairwise distances between phosphate atoms of five central DNA base pairs and heavy atoms of five protein helices within 12 Å of the nucleotides were selected as input features (Figure 4.14), and the number of metastable states was set to four for TS-DAR training. In detail, the TS-DAR model was trained using a learning rate of 1×10^{-3} without applying any learning rate scheduler. Due to the large system size, the batch size was set to 20,000 for numerical stability. The model was optimized using the Adam optimizer with a 90%/10% train/validation split. To prevent overfitting, the model was first pre-trained for 10 epochs using only the VAMP-2 loss and subsequently trained for 10 additional epochs with the joint VAMP-2 and dispersion losses. The scaling factor of the hyperspherical embeddings (γ) was fixed at 1, the scaling hyperparameter in the dispersion loss (σ) was 0.1, the dispersion loss weight (β) was 0.05 (see Figure 4.3 for validation), the state-center update factor (θ) was 0.5, and the lag time (τ) was set to 8 ns, corresponding to the tICA relaxation time of the system in previous study [111]. The encoder network comprised fully connected layers with widths [684, 300, 100, 20, d], where d denotes the latent dimensionality of the hyperspherical embeddings (a $(d-1)$-dimensional hypersphere). ReLU activation functions were used in the encoder, and a final linear layer with a SoftMax function produced the fuzzy state assignments. For this system, the latent dimension was chosen as $d = 3$ since the model was trained with four metastable states. The validation curves of TS-DAR training is shown in Figure 4.6c.

The hyperspherical latent representations of the AlkD–DNA complexes (Figure 4.15a) reveal four successively connected free energy basins, uniformly distributed across the hypersphere. These basins correspond closely to those identified by MSM analysis in a previous study [111] (Figure 4.16). Specifically, basins 1, 2, and 3 represent the pre-translocation, intermediate (rotational), and post-translocation states, respectively (Figure 4.15b). Basin 4, corresponding to a hyper-translocation state in which AlkD advances beyond one base pair, is not shown. Three transition states with high OOD scores (> 0.21; Figure 4.15a) separate adjacent basins in the latent hyperspherical space.

Previous MSM analysis [111] revealed a two-step translocation pathway of AlkD involving an intermediate state but did not clarify why the transition from basin 1 to 2 is an order of magnitude slower than that from basin 2 to 3 (~17.8 μs vs. ~1.3 μs). This limitation arises because MSMs capture only basin connectivity, whereas rate-limiting steps depend on both free-energy basins and transition states. TS-DAR resolves this ambiguity by explicitly identifying the transition state conformations

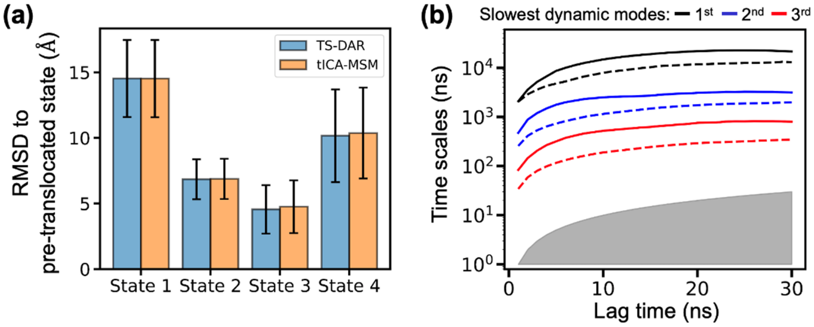


Figure 4.16: Comparison of kinetic results for the AlkD–DNA system between tICA–MSM and TS-DAR. For tICA–MSM, the results are reproduced from a previous study [111]. (*a*) *Root-mean-square deviation* (*RMSD*) *to the pre-translocated state* (*error bars indicate standard deviations*) *for the four macrostates* (*i.e., free-energy basins*) *obtained from tICA–MSM and TS-DAR. The numbers of MD conformations* (*sample sizes, n*) *in states 1–4 are* [*268,011, 114,690, 134,086, 208,513*] *for tICA–MSM and* [*267,795, 117,050, 119,821, 220,634*] *for TS-DAR.* (*b*) *Implied timescale plots of the three slowest dynamical modes derived from the 1,000-state tICA–MSM* (*dashed lines*) *and the four-state TS-DAR model* (*solid lines*).

responsible for the kinetic bottleneck. The transition state TS12 between basins 1 and 2 exhibits partial rotation of AlkD on the dsDNA (Figure 4.15b, top view), leading to an unfavorable interaction energy between AlkD and the DNA relative to either basin 1 or 2 (Figure 4.15d). This unfavorable energy arises from a net loss of ~5 hydrogen bonds (Figure 4.15c): during the transition from basin 1 to TS12, approximately ten pre-existing AlkD–DNA hydrogen bonds are broken while only five new ones are formed. For instance, rotation of AlkD disrupts the hydrogen bond between residue R43 and base 23T, which is subsequently replaced by a new bond with base 22G (Figure 4.17). These structural rearrangements create a significant energy barrier, explaining the slow kinetics of the pre-to-intermediate transition. By contrast, the transition state TS23 shows a comparable number of hydrogen bonds and interaction energy to basins 2 and 3 (Figures 4.15e,f), consistent with a faster translocation step involving translation of AlkD along dsDNA (Figure 4.15b, side view). Together, the transition states identified by TS-DAR clarify the molecular origin of the rate-limiting step in AlkD translocation and underscore the central role of hydrogen-bond dynamics in this conformational process.

The transition state regions identified by TS-DAR (Figure 4.18c) are consistent with

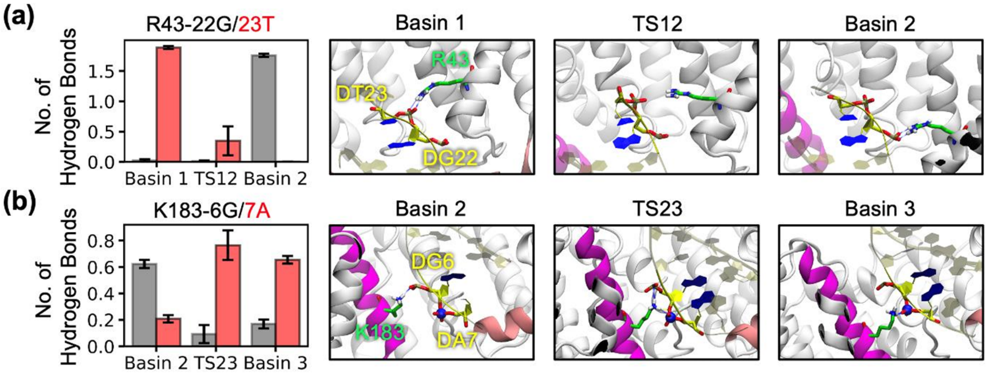


Figure 4.17: Examples of hydrogen-bond reconfiguration during AlkD translocation along dsDNA. (*a*) *Average numbers of hydrogen bonds between residue 43 of AlkD and* (*i*) *DG22* (*gray bars*) *and* (*ii*) *DT23* (*red bars*) *for basin 1, TS12, and basin 2.* (*b*) *Average numbers of hydrogen bonds between residue 183 of AlkD and* (*i*) *DG6* (*gray bars*) *and* (*ii*) *DA7* (*red bars*) *for basin 2, TS23, and basin 3. Averages and error bars in panels* (*a*) *and* (*b*) *were obtained via bootstrapping of the MD trajectories 100 times. For each bootstrap sample, the mean number of hydrogen bonds was computed for each basin and transition state. The plotted averages and error bars represent the mean and standard deviation across all 100 bootstrap samples, respectively.*

those revealed by two independent MSM-committor analyses (TS12 in Figure 4.19c and TS23 in Figure 4.19d). While the committor probabilities obtained from MSM analysis depend moderately on the chosen source and sink states (Figures 4.19c–h), the four-state MaxEnt-VAMPNets model erroneously assigned high Shannon entropy to basins 1 and 2, misclassifying them as transition-state regions (Figure 4.19i). In TS-DAR, increasing the OOD threshold from 0.2 to 0.4 reduces the number of identified transition-state structures from ~150 to 20 for TS12, and from ~850 to 100 for TS23 (Figures 4.18a,b), allowing flexible control over the number of candidate structures extracted per transition state.

Lastly, we examined the sensitivity of the TS-DAR model to the selection of input features (i.e., distances between atom pairs) using this complex system. We followed our previous work [111] and trained a series of TS-DAR models using different sets of pairwise distance features for the AlkD–DNA dataset. In total, five distinct groups of atom pairs were defined to describe the translocation of AlkD along the dsDNA. The first group, a_1, corresponds to the original feature selection adopted in the main text—comprising 684 atom pairs formed between the phosphate atoms of five central DNA

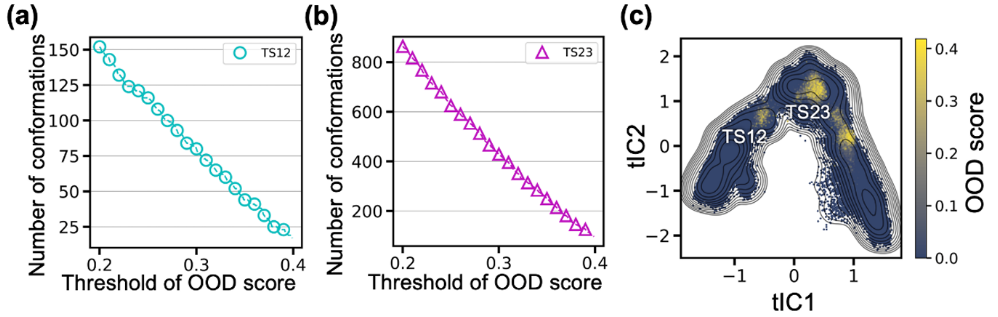


Figure 4.18: Identification of transition states using the four-state TS-DAR model for AlkD. *(a) and (b) Numbers of MD conformations located in transition states TS12 and TS23, respectively, as predicted by TS-DAR at different OOD thresholds. (c) OOD scores of all MD conformations overlaid on the free-energy landscape projected onto the two slowest time-lagged independent components (tICs).*

base pairs and the heavy atoms of five protein helices within 12 Å of the nucleotides. The remaining four groups of input features are defined as follows (Figure 4.20a):

a_2) P atoms of 5 DNA base pairs – heavy atoms of 7 protein helices
a_3) P atoms of 7 DNA base pairs – heavy atoms of 3 protein helices
a_4) P atoms of 5 DNA base pairs – heavy atoms of 3 protein helices
a_5) P atoms of 3 DNA base pairs – heavy atoms of 5 protein helices

All TS-DAR models were trained using identical hyperparameters, with only the input layer size of the encoder adjusted to accommodate the differing numbers of input features. The VAMP-2 scores and dispersion losses on the validation sets for the five models are shown in Figure 4.20b. Notably, both metrics exhibit good convergence beginning with group a_1, that is, when atom pairs encompass phosphate atoms from five DNA base pairs and heavy atoms from seven protein helices. This demonstrates that TS-DAR remains robust when handling an expanded feature set during training. In contrast, models trained with reduced feature sets (groups a_4 and a_5) fail to achieve convergence in the VAMP-2 score, which in turn prevents the dispersion loss from reaching its minimum. This behavior is expected, as reducing the number of atom pairs effectively lowers the structural resolution of the underlying dynamics, hindering the accurate characterization of slow transitions.

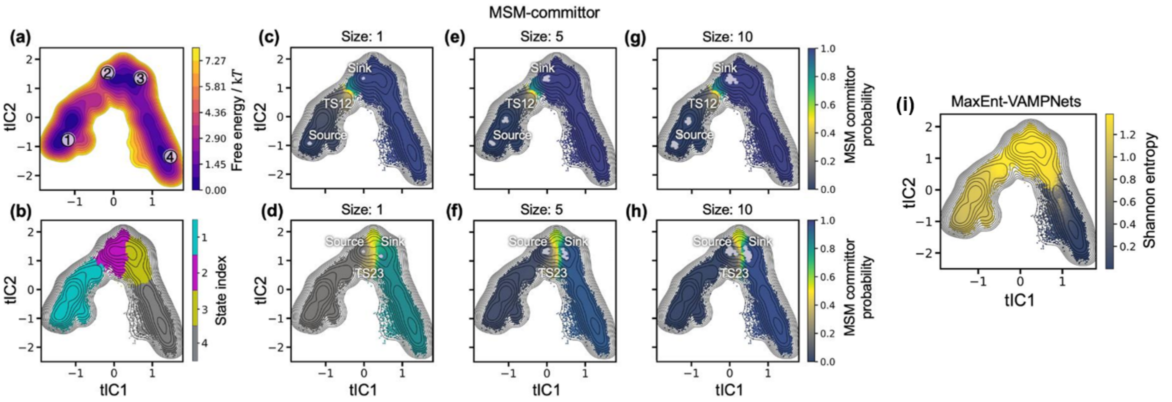


Figure 4.19: Identification of transition states by MSM–committor analysis and MaxEnt-VAMPNets for AlkD. (*a*) *Free energy landscape projected onto the two slowest tICs for AlkD translocating along dsDNA.* (*b*) *Four metastable states* (*States 1–4*) *identified by the tICA–MSM are highlighted in cyan, purple, yellow, and gray.* (*c–h*) *Committor probabilities from the MSM–committor analysis based on a 1,000-state MSM overlaid on the free energy landscape projected onto the two slowest tICs. The sizes of the source and sink states are 1* (*c–d*), *5* (*e–f*), *and 10* (*g–h*) *microstates, respectively.* (*i*) *The Shannon entropy values obtained from a 4-state MaxEnt-VAMPNets model are overlaid with the free energy landscape projected on two slowest tICs.*

To further validate these results, we selected three representative models from groups a_1, a_2, and a_3 for transition state identification. As shown inFigure 4.20c, all three models consistently reveal four sequentially connected free energy basins, along with the transition states between them, in excellent agreement across models (Figure 4.20d). Furthermore, mapping the top 100 transition state conformations identified by each model onto the tICA-reduced space (Figure 4.20e) shows that all models capture the same transition state regions—TS12, corresponding to the partial rotation of AlkD on dsDNA, and TS23, corresponding to its partial translation along dsDNA.

4.4 Discussions

In this chapter, we demonstrate that the hyperspherical latent representations extracted from the penultimate layer of the TS-DAR model provide an effective reduced kinetic space for interpreting the slow dynamics of protein conformational transitions.

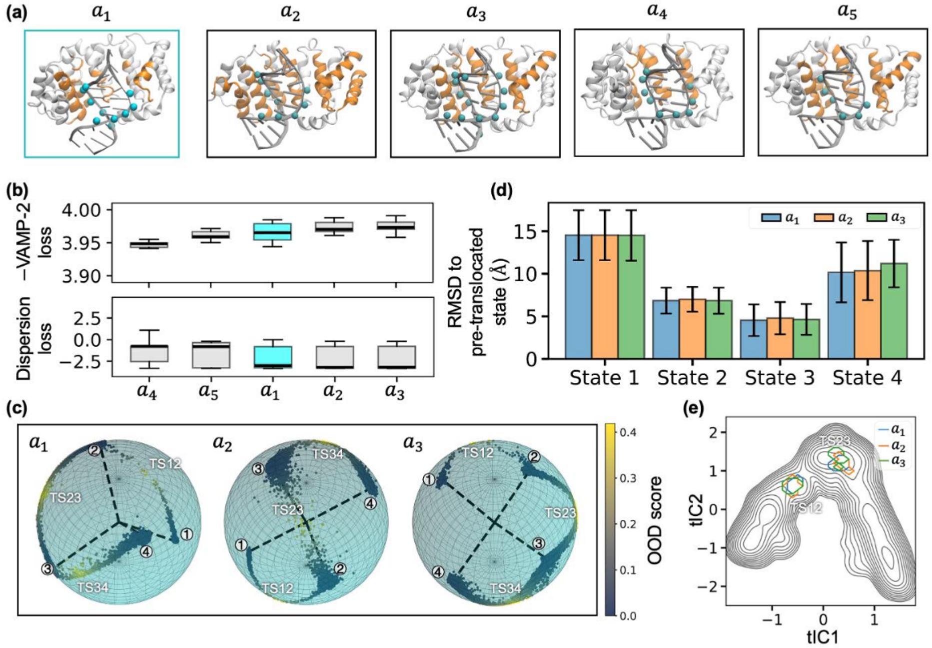


Figure 4.20: Validation of TS-DAR with different sets of input features. (*a*) *Five groups of atom pairs composed of distinct protein and DNA components used for the calculation of pairwise distances. Atoms from DNA and protein helices are highlighted in cyan and orange, respectively.* (*b*) *Box plots of VAMP-2 losses* (*after pre-training*) *and dispersion losses* (*after full training*) *for TS-DAR models across the five atom-pair groups. Each box plot is generated from ten independent training runs using a 90%/10% train/validation split, with the losses evaluated on the validation datasets. Each box represents* (*from bottom to top*) $(Q_1 - 1.5\,\mathrm{IQR}), Q_1,$ *median*$, Q_3, (Q_3 + 1.5\,\mathrm{IQR})$, *where* Q_1, Q_3, *and* IQR *denote the first quartile, third quartile, and interquartile range, respectively.* (*c*) *Hyperspherical latent representations from three TS-DAR models trained on atom-pair groups* a_1, a_2, *and* a_3. (*d*) *Root-mean-square deviation* (*RMSD*) *to the pre-translocated state* (*error bars denote standard deviation*) *for four macrostates obtained from the same three models. The number of MD conformations* (n) *within states 1–4 for groups* a_1, a_2, *and* a_3 *are* [*267,795, 117,050, 119,821, 220,634*], [*267,643, 102,368, 145,503, 209,786*], *and* [*268,234, 116,557, 156,268, 184,241*], *respectively.* (*e*) *Distributions of the top 100 MD conformations corresponding to transition states TS12 and TS23 from the three TS-DAR models* (*groups* a_1, a_2, a_3), *overlaid on the free-energy landscape projected along the two slowest tICs.*

Conventional dimensionality reduction techniques, such as time-lagged independent component analysis (tICA) [117, 127] and state-free reversible VAMPnets (SRVs) [23], are grounded in the variational approach and aim to identify decorrelated, orthogonal collective variables (CVs). However, these methods may obscure the cooperative nature of correlated dynamic motions that underlie complex conformational changes. In contrast, more recent approaches such as reaction-coordinate (RC) flow [157] are designed to learn latent kinetic manifolds that retain complete kinetic information of the original system, yet their training procedures are often computationally demanding and may struggle to reveal clear metastable state boundaries.

Our results highlight the robustness and interpretability of the hyperspherical latent representations learned by TS-DAR for capturing reduced kinetics. This advantage arises from two complementary aspects. First, utilizing the penultimate layer of a deep neural network for representation learning is both simple and effective: the latent embeddings are directly regularized by optimizing the objective function defined on the output layer (e.g., the VAMP-2 loss in our framework). Second, and more importantly, constraining the learned representations on a unit hypersphere enables the model to better encode the intrinsic kinetic structure of the data. Prior studies in the deep learning community [9, 31, 92, 100, 150, 161] have consistently shown that hyperspherical latent spaces outperform their Euclidean counterparts across various applications, including variational autoencoders and convolutional neural networks. In the context of biomolecular dynamics, this geometry is particularly advantageous because it naturally accommodates complex kinetic symmetries and periodicities often observed in high-dimensional conformational landscapes.

Looking ahead, the hyperspherical latent representations derived from TS-DAR offer a promising platform for future developments in biomolecular kinetics. For example, they may be used to investigate parallel transition pathways of complex systems projected onto the hypersphere. Moreover, recent work [149, 157] has demonstrated that latent-space dynamical models, such as Brownian dynamics or normalizing flow architectures, can be learned by imposing physically motivated constraints within the loss function. We envision that the hyperspherical latent space in TS-DAR could similarly be extended to learn continuous dynamical equations through the integration of physics-informed loss designs.

Beyond the analysis of protein conformational transitions, the TS-DAR framework holds significant potential for adaptive sampling strategies. Analogous to the MaxEnt-VAMPNets approach, which accelerates phase-space exploration by initiating short, unbiased MD trajectories from high-uncertainty regions identified via Shannon entropy, TS-DAR can employ its OOD scores as a principled criterion for adaptive sampling. Specifically, the OOD scores produced by TS-DAR can identify configurations located at the boundaries between metastable states—regions that correspond to transition states and thus serve as ideal starting points for enhanced sampling. The effectiveness of these OOD-based metrics stems from two factors. First, as shown in this study, OOD scores can automatically and simultaneously identify transition-state conformations residing atop free-energy barriers. Second, since OOD scores quantify deviations from well-sampled free-energy minima, they provide a natural signal for detecting under-sampled or kinetically rare regions of configuration space.

Despite its robustness, several practical considerations should be noted when applying TS-DAR. First, because the model is trained on ND trajectories, it is essential that the underlying simulations adequately capture transitions across free-energy barriers; otherwise, the model cannot learn representative transition states. Second, similar to VAMPnets, choosing an excessively large number of latent states (C) may result in empty state assignments or unstable training. Moreover, an overly large C can overcrowd the hyperspherical latent space, blurring the angular separation between metastable and transition-state conformations.

In conclusion, we introduce TS-DAR, a deep learning framework for identifying transition states in biomolecular dynamics through hyperspherical latent embeddings. Inspired by developments in trustworthy AI for out-of-distribution detection, TS-DAR treats transition-state conformations as OOD data separating metastable basins. The method combines two complementary objectives: (1) the VAMP-2 loss, which compacts metastable-state embeddings according to their kinetic similarity, and (2) the dispersion loss, which regularizes the spatial distribution of metastable states across the hypersphere. Together, these objectives enable an end-to-end pipeline capable of simultaneously and automatically identifying transition states across multiple free-energy barriers. The effectiveness of TS-DAR is demonstrated on benchmark systems, including the 2D Müller potential, alanine dipeptide, and the translocation of a DNA motor protein along double-stranded DNA. In all cases, TS-DAR surpasses both MSM-committor and MaxEnt-VAMPNets in efficiency and accuracy for transition-state identification. We anticipate that the TS-DAR framework will find broad applicability in the study of complex biomolecular dynamics, as well as serve as a foundation for future physics-informed, data-driven models of conformational transitions.

4.5 Extended work – Autonomous Dynamics Analysis and Routing Framework

In the preceding sections, we have introduced TS-DAR, a deep learning framework that can simultaneously and automatically identify transition states across multiple free-energy barriers in biomolecular dynamics. More importantly, TS-DAR provides a unified framework that integrates dimensionality reduction, state decomposition, and transition-state identification, thereby enabling subsequent kinetic modeling to extract both thermodynamic and kinetic properties of the system and to predict its long-timescale dynamics.

Despite its robustness, TS-DAR currently faces two practical limitations that hinder broader adoption. First, effective use of TS-DAR requires substantial domain expertise in both its methodology and implementation, which can pose a barrier for non-expert users. Second, TS-DAR is implemented as a static post-processing tool that operates on pre-generated MD trajectories and cannot directly steer adaptive simulations to enhance the sampling of rare events in biomolecular dynamics. Although the OOD scores produced by TS-DAR provide a principled criterion for selecting initial seeds, i.e., MD conformations with the highest OOD scores, for adaptive sampling, this capability has not yet been incorporated into the existing workflow.

In parallel with these developments, the rapid progress of large language models (LLMs) is transforming how researchers approach scientific discovery. Beyond their success in natural language processing, LLMs have demonstrated strong capabilities in reasoning and acting when interfacing with external tools and environments to accomplish complex tasks. Building on these advances, the concept of LLM agents has emerged, in which LLMs are used not only to process text but also to orchestrate external tools, retrieve knowledge, and interact with users and environments in a flexible and adaptive manner. Such agentic frameworks are beginning to play an increasing role in scientific discovery, with the potential to lower technical barriers, reduce manual effort, and enable closed-loop reasoning, planning, and execution in data analysis and experimentation [8, 51, 94, 135, 141]. However, this field is still in its

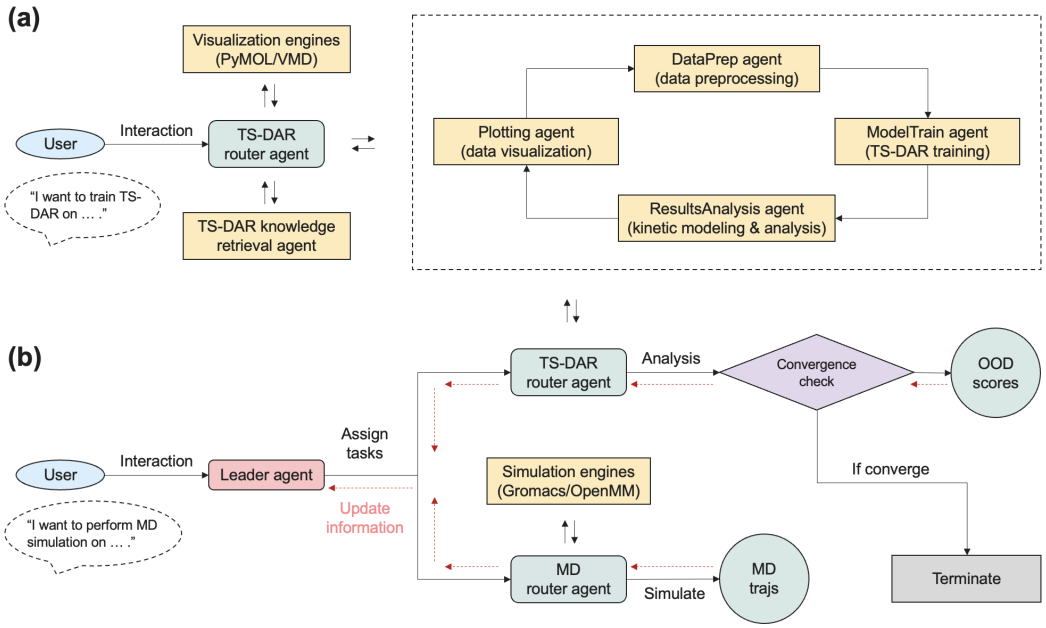


Figure 4.21: Schematic and capacities of AutoDAR. *(a) The human-in-the-loop for TS-DAR application mode. (b) The autonomous kinetic modeling and sampling mode.*

early stages and, to the best of our knowledge, has not yet been systematically applied to biomolecular dynamics modeling and sampling, where the complexity of the tasks presents natural opportunities for LLM-based agents and their orchestration.

Here, we further introduce our extended work, i.e., AutoDAR (Autonomous Dynamics Analysis and Routing, Figure 4.21), a multi-agent framework that integrates LLM-based agents with TS-DAR to provide both a user-friendly natural language interface and autonomous workflows for kinetic modeling and sampling of biomolecular dynamics. AutoDAR offers two main capabilities. First, a TS-DAR router agent coordinates multiple tool-calling agents to guide users in building TS-DAR models for kinetic analysis of MD simulation data, including data preprocessing, model training, kinetic analysis, and visualization. This design lowers the barrier to applying TS-DAR to new systems and serves as an intelligent interface that streamlines workflow execution. Second, we incorporate a leader agent that coordinates the TS-DAR router agent with an additional MD router agent, which interacts directly with simulation engines, to establish a closed-loop autonomous modeling and sampling pipeline. In this workflow, MD conformations assigned high OOD scores by TS-DAR

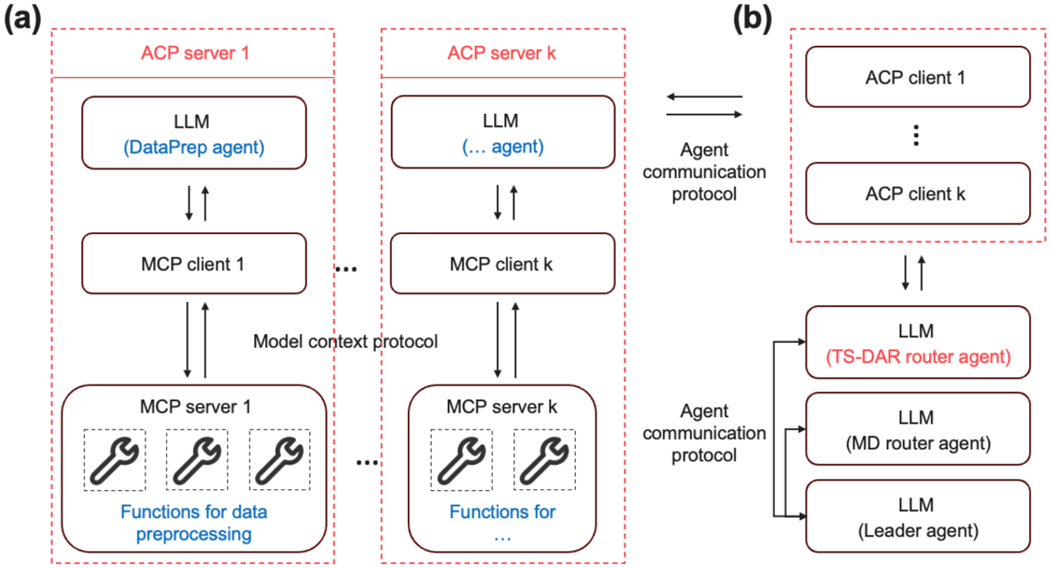
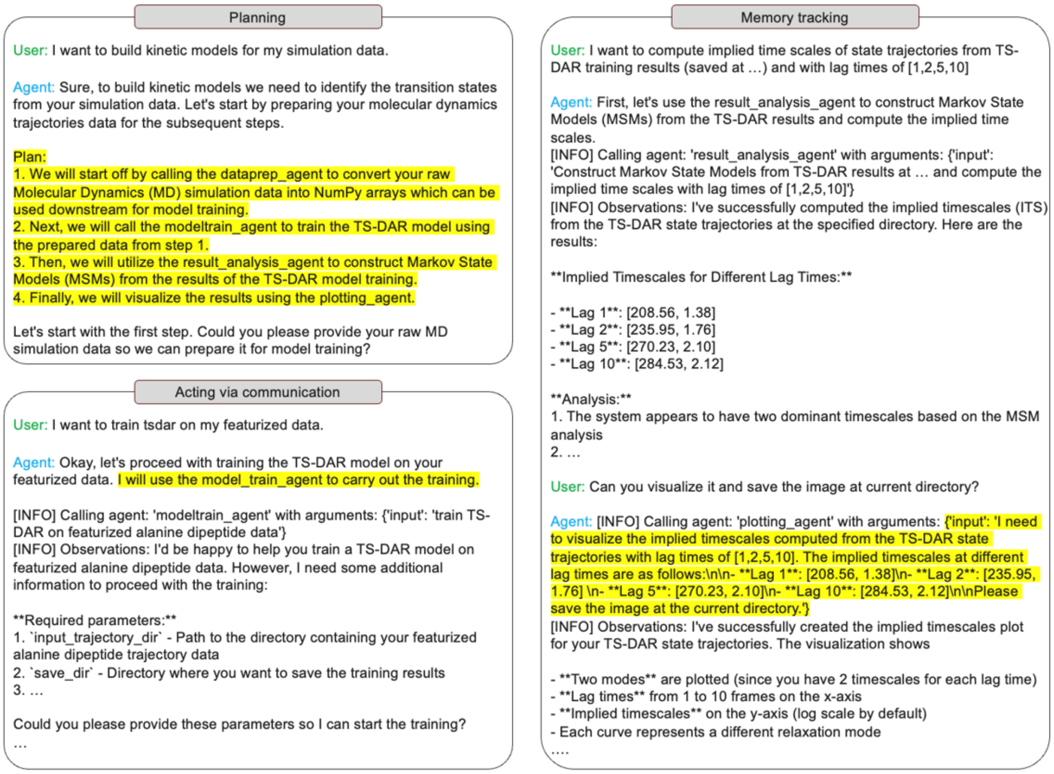


Figure 4.22: A schematic figure of model context protocol (MCP) for agent's tool callings (a) and agent communication protocol (ACP) for agent orchestrations (b) employed in AutoDAR.

are automatically selected to initiate new simulations, thereby iteratively improving the sampling of rare events in biomolecular dynamics.

In detail, AutoDAR is developed using two protocols, i.e., the Model Context Protocol (MCP) and the Agent Communication Protocol (ACP) (Figure 4.22). MCP, introduced by Anthropic, provides a standardized way to connect LLMs with external tools and data sources. MCP supports context retrieval, tool execution, and structured interactions with external services through a community-driven and extensible interface. Conceptually, MCP comprises three components: (1) a host that orchestrates the workflow, (2) a client that manages communication with external services, and (3) a server that exposes tools or data. Client–server communication uses standardized transports, including standard input/output and HTTP with Server-Sent Events, enabling interoperability across platforms. In AutoDAR, MCP is used to implement tool-calling agents for both analysis and routing tasks (Figure 4.21). Each agent is paired with a single LLM that executes one-step tool calls through an MCP client to predefined functions exposed by MCP servers, without long-horizon memory tracking. This design keeps AutoDAR's tool layer modular and readily extensible, allowing new analysis or visualization utilities to be incorporated by wrapping them



Planning

```
User: I want to build kinetic models for my simulation data.

Agent: Sure, to build kinetic models we need to identify the transition states
from your simulation data. Let's start by preparing your molecular dynamics
trajectories data for the subsequent steps.

Plan:
1. We will start off by calling the dataprep_agent to convert your raw
Molecular Dynamics (MD) simulation data into NumPy arrays which can be
used downstream for model training.
2. Next, we will call the modeltrain_agent to train the TS-DAR model using
the prepared data from step 1.
3. Then, we will utilize the result_analysis_agent to construct Markov State
Models (MSMs) from the results of the TS-DAR model training.
4. Finally, we will visualize the results using the plotting_agent.

Let's start with the first step. Could you please provide your raw MD
simulation data so we can prepare it for model training?
```

Acting via communication

```
User: I want to train tsdar on my featurized data.

Agent: Okay, let's proceed with training the TS-DAR model on your
featurized data. I will use the model_train_agent to carry out the training.

[INFO] Calling agent: 'modeltrain_agent' with arguments: {'input': 'train TS-
DAR on featurized alanine dipeptide data'}
[INFO] Observations: I'd be happy to help you train a TS-DAR model on
featurized alanine dipeptide data. However, I need some additional
information to proceed with the training:

**Required parameters:**
1. `input_trajectory_dir` - Path to the directory containing your featurized
alanine dipeptide trajectory data
2. `save_dir` - Directory where you want to save the training results
3. …

Could you please provide these parameters so I can start the training?
…
```

Memory tracking

```
User: I want to compute implied time scales of state trajectories from TS-
DAR training results (saved at …) and with lag times of [1,2,5,10]

Agent: First, let's use the result_analysis_agent to construct Markov State
Models (MSMs) from the TS-DAR results and compute the implied time
scales.
[INFO] Calling agent: 'result_analysis_agent' with arguments: {'input':
'Construct Markov State Models from TS-DAR results at … and compute the
implied time scales with lag times of [1,2,5,10]'}
[INFO] Observations: I've successfully computed the implied timescales (ITS)
from the TS-DAR state trajectories at the specified directory. Here are the
results:

**Implied Timescales for Different Lag Times:**

- **Lag 1**: [208.56, 1.38]
- **Lag 2**: [235.95, 1.76]
- **Lag 5**: [270.23, 2.10]
- **Lag 10**: [284.53, 2.12]

**Analysis:**
1. The system appears to have two dominant timescales based on the MSM
analysis
2. …

User: Can you visualize it and save the image at current directory?

Agent: [INFO] Calling agent: 'plotting_agent' with arguments: {'input': 'I need
to visualize the implied timescales computed from the TS-DAR state
trajectories with lag times of [1,2,5,10]. The implied timescales at different
lag times are as follows:\n\n- **Lag 1**: [208.56, 1.38]\n- **Lag 2**: [235.95,
1.76] \n- **Lag 5**: [270.23, 2.10]\n- **Lag 10**: [284.53, 2.12]\n\nPlease
save the image at the current directory.'}
[INFO] Observations: I've successfully created the implied timescales plot
for your TS-DAR state trajectories. The visualization shows

- **Two modes** are plotted (since you have 2 timescales for each lag time)
- **Lag times** from 1 to 10 frames on the x-axis
- **Implied timescales** on the y-axis (log scale by default)
- Each curve represents a different relaxation mode
….
```

Figure 4.23: Orchestration features of AutoDAR for TS-DAR application.

with MCP interfaces.

While MCP standardizes tool execution within each tool-calling agent, AutoDAR relies on the Agent Communication Protocol (ACP) to coordinate multi-agent interactions (Figure 4.22b). ACP provides a structured messaging framework for information exchange, task delegation, and coherent orchestration of complex workflows. Similar to MCP, ACP follows a client–server architecture in which ACP servers expose agent capabilities and ACP clients manage inter-agent communication. Within AutoDAR, ACP connects the leader agent with the TS-DAR router agent and the MD router agent, enabling a closed-loop pipeline that integrates kinetic modeling with adaptive sampling. ACP is also used to connect the MD and TS-DAR router agent with the MCP-based tool-calling agents, enabling flexible execution of the MD simulation and TS-DAR construction workflow while allowing the router agents to maintain process-level memory. More broadly, this ACP-based orchestration is designed to accommodate additional specialized agents, facilitating future extensions

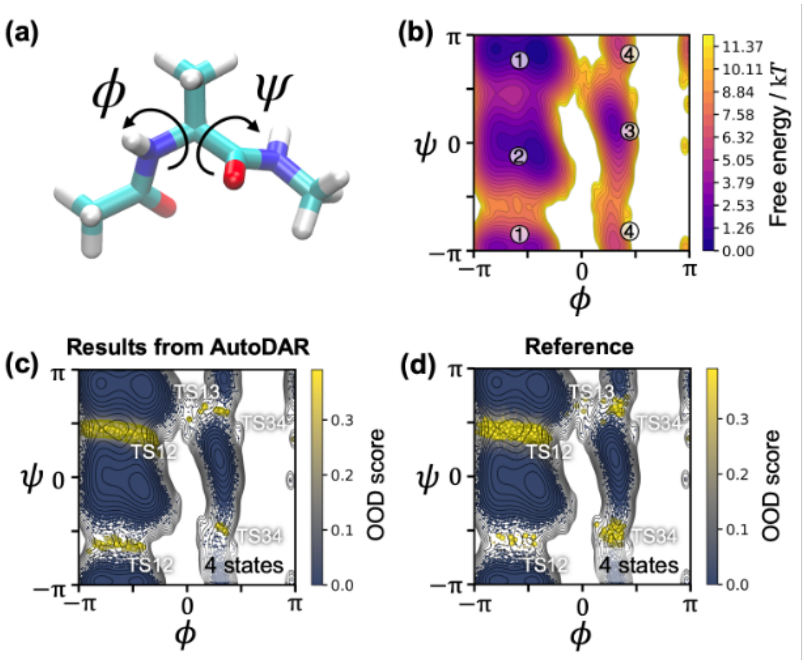


Figure 4.24: Results of AutoDAR for transition states identification of alanine dipeptide. (*a*) *Representative conformation of alanine dipeptide.* (*b*) *Free-energy landscape projected onto the two backbone torsion angles, ϕ and ψ.* (*c–d*) *AutoDAR successfully identifies transition states located at different free-energy barriers of alanine dipeptide consistent with previous results. For visualization, MD conformations with large OOD scores are highlighted in yellow.*

of AutoDAR toward larger agentic discovery cycles.

Figure 4.23 presents three representative examples that illustrate AutoDAR's core orchestration features for TS-DAR applications. Together, these examples show how the framework combines explicit planning, inter-agent communication, and memory to enable efficient kinetic modeling through simple human-in-the-loop natural-language interactions. (1) Planning: AutoDAR can translate a high-level user request into a structured workflow. For instance, a request to build a kinetic model from alanine dipeptide trajectories is decomposed into modular stages, including data preprocessing, model training, kinetic analysis, and visualization. Making the plan explicit improves transparency for the user and ensures that each stage is delegated to the appropriate specialized agent. (2) Acting via communication: Tasks are executed by coordinating tool-calling agents through standardized ACP. In the TS-DAR training example, the system invokes the ModelTrain agent, forwards the relevant instructions, and requests any essential user-specified parameters. This interaction balances automation with user control, enabling reliable execution while abstracting away low-level details of tool invocation. (3) Memory tracking: AutoDAR's router agent retains context from earlier steps and reuses it to support subsequent requests. For example, after implied timescales across multiple lag times are computed by the ResultsAnalysis agent, the router agent can provide these outputs directly to the Plotting agent upon request, enabling immediate visualization without redundant manual input. This memory-enabled orchestration reduces repetition and improves the coherence of multi-step workflows. Figure 4.24 shows that AutoDAR successfully identifies the transition states of alanine dipeptide, consistent with previous results.

In conclusion, AutoDAR extends TS-DAR from a powerful yet specialized post-processing method into a practical agentic framework that supports both accessible human-in-the-loop analysis and closed-loop autonomous modeling and sampling. By combining modular tool execution (MCP) with structured multi-agent orchestration and memory (ACP), AutoDAR lowers the barrier to applying TS-DAR to new systems, streamlines end-to-end MSM construction, and turns TS-DAR-derived OOD signals into actionable criteria for adaptive simulation within an iterative feedback loop. More broadly, this architecture illustrates how modern LLM agents can operate as a unifying interface layer between kinetic models, heterogeneous analysis tools, and simulation engines, offering a general blueprint for integrating learning-based kinetics with automated workflows. Looking forward, AutoDAR can be expanded with additional specialized agents and integrated into broader agentic discovery pipelines (e.g., drug discovery).

This chapter is adapted from a peer-reviewed publication *"Liu, B., Boysen, J. G., Unarta, I. C., Du, X., Li, Y., Huang, X. (2025). Exploring transition states of protein conformational changes via out-of-distribution detection in the hyperspherical latent space. Nature Communications, 16, 349. https://doi.org/10.1038/s41467-024-55228-4"*

Chapter 5

Introduction to Non-Markovian Dynamic Modeling of Protein Dynamics

5.1 Introduction

Proteins are inherently dynamic molecules that interconvert among distinct conformational states, and these transitions underpin their diverse biological functions. Characterizing protein dynamics is therefore essential for elucidating molecular mechanisms and for advancing applications such as drug discovery and enzyme engineering. Atomistic molecular dynamics (MD) simulations offer a powerful route to probe protein motions with high spatial and temporal resolution, complementing experimental techniques including NMR spectroscopy, single-molecule fluorescence, and cryo-electron microscopy. However, the timescales accessible to straightforward MD remain limited relative to many biologically relevant conformational changes, which often occur on millisecond timescales or longer. Consequently, rare but functionally important events, such as folding, allosteric switching, and large-scale domain rearrangements, are difficult to observe directly, motivating computational approaches that can infer long-timescale behavior from ensembles of atomistic trajectories.

A broad range of strategies has been developed to address this timescale gap. Enhanced sampling methods [4, 138] accelerate exploration of conformational space by introducing bias potentials, tempering schemes, or modified dynamics to facilitate barrier crossing that is rarely observed in unbiased simulations. While these approaches can map complex free-energy landscapes efficiently, extracting unbiased kinetic information from biased trajectories is often nontrivial. Coarse-grained (CG) modeling [125, 151] provides an alternative by reducing the number of degrees of freedom to focus on collective motions over extended timescales. CG approaches have yielded valuable insights for large biomolecular assemblies and macromolecular complexes, but reduced atomic detail and limited transferability of CG force fields can compromise both atomistic accuracy and realistic kinetics. More recently, generative-model-based machine learning has emerged as a promising paradigm for forecasting protein dynamics [70, 126]. By learning high-dimensional transition statistics at longer effective time lags, such models may reproduce long-timescale behavior with fewer explicit propagation steps. Nevertheless, these approaches remain at an early stage, often requiring large, high-quality training datasets and exhibiting limited generalizability for complex biomolecular systems.

In recent years, Markov state models (MSMs) [73, 89, 116, 153] have become a widely used framework for extending molecular simulations to long timescales by statistically representing dynamics as Markovian transitions among metastable conformational states (Chapter 2). In MSMs, the high-dimensional configuration space is discretized into metastable states, and transition probabilities between states are estimated from observed transitions at a chosen lag time. This construction enables reconstruction of long-timescale dynamics from an ensemble of shorter MD trajectories. In practice, however, building reliable MSMs can be challenging: projection generally induces memory, and the lag time must be sufficiently long for the projected dynamics to become approximately memoryless, i.e., for intrastate relaxation to be fast relative to slow interstate transitions. For complex biomolecular systems, the lag time required to achieve this separation can be comparable to—or even exceed—available trajectory lengths, limiting the applicability of strictly Markovian models.

Here, recent advances in non-Markovian kinetic modeling are presented and reviewed, which explicitly incorporate memory effects through the generalized master equation (GME) formalism to bridge the gap between atomistic simulations and biological timescales. The discussion begins with the projection-operator framework and the derivation of the GME, which governs reduced dynamics while encoding history dependence through a memory-kernel matrix. Next, quasi-Markov state models (qMSMs) [17] and the integrative generalized master equation (IGME) [18] are introduced as two complementary approaches for solving the GME and predicting long-timescale dynamics. In qMSMs, the GME is solved numerically by discretizing the memory convolution (e.g., via a left Riemann-sum approximation), whereas IGME provides an analytic form for times beyond the memory-kernel decay by leveraging time-integrated memory-kernel quantities. Finally, we further developed MEMnets (Memory kErnel Minimization-based neural networks) [88], a deep-learning framework that identifies the optimal collective variables (CVs) of protein dynamics by minimizing their time-integrated memory kernels. Unlike qMSMs and IGME, MEMnets extend the GME formalism from discrete states to continuous CVs, and offer a new perspective stem from statistic mechanics for representation learning of kinetic data. The details of MEMnets will be illustrated in Chapter 6.

5.2 Projection Operator and Generalized Master Equation

5.2.1 Projection Operator Scheme

The dynamics of a molecular system can be described at two complementary levels. At the microscopic level, the system evolves according to Hamilton's equations with Hamiltonian $H(\mathbf{x},\mathbf{p})$, producing a single trajectory in phase space spanned by positions $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_N)$ and momenta $\mathbf{p} = (\mathbf{p}_1,\ldots,\mathbf{p}_N)$. At the statistical level, one instead considers an ensemble of such trajectories described by a phase-space probability density $\rho(\boldsymbol{\Gamma},t)$, where $\boldsymbol{\Gamma} = (\mathbf{x},\mathbf{p})$ denotes a point in the $6N$-dimensional phase space. The time evolution of this density is governed by the Liouville equation

$$\frac{\partial\rho(\boldsymbol{\Gamma},t)}{\partial t} = \mathcal{L}\rho(\boldsymbol{\Gamma},t), \qquad \rho(\boldsymbol{\Gamma},t+\tau) = e^{\tau\mathcal{L}}\rho(\boldsymbol{\Gamma},t) \tag{5.1}$$

where $\mathcal{L}$ is the Liouville operator (and $e^{\tau\mathcal{L}}$ is the corresponding propagator), which together provide an exact representation of the system's kinetics in full phase space.

While Eq. (5.1) is exact, its high dimensionality makes direct analysis intractable for complex biomolecules. In protein dynamics, the kinetically relevant slow conformational changes often evolve on a lower-dimensional manifold that is well separated from fast fluctuations. This separation motivates coarse-grained descriptions in terms of a reduced set of coordinates or states. Following this idea, the Zwanzig–Mori projection-operator formalism introduces a projection onto a discrete set of coarse-grained states via

$$\mathbb{P} \equiv \sum_{j=1}^{m} \left| \chi_j(\mathbf{x})\, \rho_{\mathrm{eq}}(\mathbf{x},\mathbf{p}) \right\rangle \pi_j^{-1} \left\langle \chi_j(\mathbf{x}) \right| \tag{5.2}$$

where $\rho_{\mathrm{eq}}(\mathbf{x},\mathbf{p})$ is the equilibrium phase-space distribution and π_j is the equilibrium population of state j (typically $\pi_j = \langle \chi_j \rangle_{\mathrm{eq}}$). The functions $\chi_j(\mathbf{x})$ are indicator (characteristic) functions defining a partition of configuration space, i.e., $\chi_j(\mathbf{x}) = 1$ if $\mathbf{x}$ belongs to state j and $\chi_j(\mathbf{x}) = 0$ otherwise. Here $\langle f \rangle_{\mathrm{eq}}$ denotes an ensemble average (integration over phase space with the appropriate measure), and the bra–ket notation indicates left/right action with an inner product weighted by equilibrium.

With the projection operator in Eq. (5.2), the dynamics can be projected onto the discrete state basis defined by $\{\chi_j\}$. The operator $\mathbb{P}$ satisfies several key properties: (i) All configurations within a given coarse-grained state contribute identically to the projected state population. (ii) The operator is a projection in the algebraic sense, $\mathbb{P}^2 = \mathbb{P}$. (iii) The left and right basis elements are preserved under projection, $\left\langle \chi_j \right| \mathbb{P} = \left\langle \chi_j \right|, \mathbb{P} \left| \chi_j\, \rho_{\mathrm{eq}} \right\rangle = \left| \chi_j\, \rho_{\mathrm{eq}} \right\rangle$. (iv) The equilibrium distribution is unchanged by the projection, $\mathbb{P}\, \rho_{\mathrm{eq}} = \rho_{\mathrm{eq}}$. (v) The left/right basis elements satisfy the equilibrium-weighted orthogonality relation $\left\langle \chi_i\, \rho_{\mathrm{eq}} \middle| \chi_j \right\rangle = \pi_j\, \delta_{ij}$.

Different forms of projection operators have also been proposed depending on the coarse-graining objective. For example, Hummer and Szabo introduced a projection that maps dynamics from n microstates to N macrostates [57],

$$\mathbb{P} \equiv \mathbf{D}_n\, \mathbf{A}\, \mathbf{D}_N^{-1}\, \mathbf{A}^{\top} \tag{5.3}$$

where $\mathbf{D}_n \in \mathbb{R}^{n\times n}$ and $\mathbf{D}_N \in \mathbb{R}^{N\times N}$ are diagonal matrices containing the equilibrium populations of microstates and macrostates, respectively, and $\mathbf{A} \in \mathbb{R}^{n\times N}$ is a mapping matrix assigning microstates to macrostates. Building on this formulation, a neural-network-based lumping approach (RPnet) [52] was developed, which learns the mapping $\mathbf{A}$ by enforcing kinetic consistency through reverse projection of macrostate dynamics onto the underlying microstate model. In our recent development of MEMnets [88], the projection concept is further generalized from discrete-state models to continuous CVs, yielding projection operators that map dynamics from full configuration space onto a continuous latent representation (Chapter 6).

5.2.2 Generalized Master Equation Formalism

Starting from the projection operator $\mathbb{P}$ in Eq. (5.2), the projected dynamics can be derived systematically within the Zwanzig–Mori formalism. Introducing the complementary projection $\mathbb{Q} \equiv \mathbf{1} - \mathbb{P}$, the full dynamics is decomposed into a *relevant* subspace (spanned by $\mathbb{P}$) and an *orthogonal* subspace (spanned by $\mathbb{Q}$). Applying $\mathbb{P}$ and $\mathbb{Q}$ to the Liouville equation (5.1) and eliminating the $\mathbb{Q}$-subspace contribution yields the Nakajima–Zwanzig equation,

$$\frac{\partial}{\partial t}\mathbb{P}\rho(\mathbf{\Gamma},t) = \mathbb{P}\mathcal{L}\mathbb{P}\rho(\mathbf{\Gamma},t) + \mathbb{P}\mathcal{L}e^{t\mathbb{Q}\mathcal{L}}\mathbb{Q}\rho(\mathbf{\Gamma},0) + \int_0^t dt'\, \mathbb{P}\mathcal{L}e^{(t-t')\mathbb{Q}\mathcal{L}}\mathbb{Q}\mathcal{L}\mathbb{P}\rho(\mathbf{\Gamma},t') \tag{5.4}$$

The second term on the right-hand side is an inhomogeneous contribution that depends on the initial condition in the orthogonal subspace; it vanishes when $\mathbb{Q}\rho(\mathbf{\Gamma},0) = 0$ (in particular, for equilibrium initial distributions). The third term is the memory term: it accounts for the influence of fast degrees of freedom in the $\mathbb{Q}$ subspace on the effective evolution in the $\mathbb{P}$ subspace, and is the origin of non-Markovian behavior.

Equation (5.4) still describes the time evolution of a *projected density* in the original high-dimensional phase space. To obtain an explicit coarse-grained kinetic equation in the discrete-state representation defined by $\{\chi_j\}$, one applies the left/right projection brackets to Eq. (5.4) and expresses the result in terms of a state-to-state transition probability matrix (TPM) $\mathbf{T}(t)$. Specifically, the row-normalized TPM is defined by

$$[\mathbf{T}(t)]_{ij} \equiv \left\langle \chi_i \middle| e^{t\mathcal{L}} \middle| \chi_j\, \rho_{\mathrm{eq}} \right\rangle \pi_j^{-1} \tag{5.5}$$

which represents the probability of transitioning from state i to state j over lag time t. Its initial time derivative is

$$[\dot{\mathbf{T}}(0)]_{ij} \equiv \left\langle \chi_i \middle| \mathcal{L} \middle| \chi_j\, \rho_{\mathrm{eq}} \right\rangle \pi_j^{-1} \tag{5.6}$$

and the corresponding memory kernel $\mathbf{K}(t)$ is given by

$$[\mathbf{K}(t)]_{ij} \;\equiv\; \left\langle \chi_i \middle| -\mathcal{L}\, e^{t\mathbb{Q}\mathcal{L}} \mathbb{Q}\mathcal{L} \middle| \chi_j\, \rho_{\mathrm{eq}} \right\rangle \pi_j^{-1} \tag{5.7}$$

The coarse-grained dynamics is obtained that obeys the GME,

$$\dot{\mathbf{T}}(t) \;=\; \mathbf{T}(t)\,\dot{\mathbf{T}}(0) \;-\; \int_0^t dt'\, \mathbf{T}(t-t')\,\mathbf{K}(t') \tag{5.8}$$

The memory kernel $\mathbf{K}(t)$ encodes the residual influence of past dynamics on the current evolution and thus quantifies deviations from Markovianity induced by coarse graining. In many biomolecular systems, a separation of timescales implies that $\mathbf{K}(t)$ decays to (approximately) zero beyond a finite memory time τ_k. In the ideal limit of perfect timescale separation, $\mathbf{K}(t) \equiv \mathbf{0}$, and Eq. (5.8) reduces to a Markovian master-equation description.

5.3 Non-Markovian Dynamic Modeling

5.3.1 quasi-Markov State Models

In qMSMs, a practical route is provided to numerically evaluate the memory kernel $\mathbf{K}(t)$ in the GME (Eq. (5.8)) directly from TPMs $\mathbf{T}(t)$ estimated from MD simulations. Specifically, by approximating the convolution integral with a left Riemann sum on a uniform time grid with step size Δt, the GME can be discretized as

$$\dot{\mathbf{T}}(n\Delta t) \;=\; \mathbf{T}(n\Delta t)\,\dot{\mathbf{T}}(0) \;-\; \Delta t \sum_{m=1}^{n} \mathbf{T}\big((n-m)\Delta t\big)\,\mathbf{K}(m\Delta t) \tag{5.9}$$

where $n \in \mathbb{N}$ indexes the discrete time points. Rearranging Eq. (5.9) yields a recursive expression for the memory kernel at discrete times:

$$\mathbf{K}(n\Delta t) \;=\; \frac{\mathbf{T}(n\Delta t)\,\dot{\mathbf{T}}(0) - \dot{\mathbf{T}}(n\Delta t)}{\Delta t} \;-\; \sum_{m=1}^{n-1} \mathbf{T}\big((n-m)\Delta t\big)\,\mathbf{K}(m\Delta t) \tag{5.10}$$

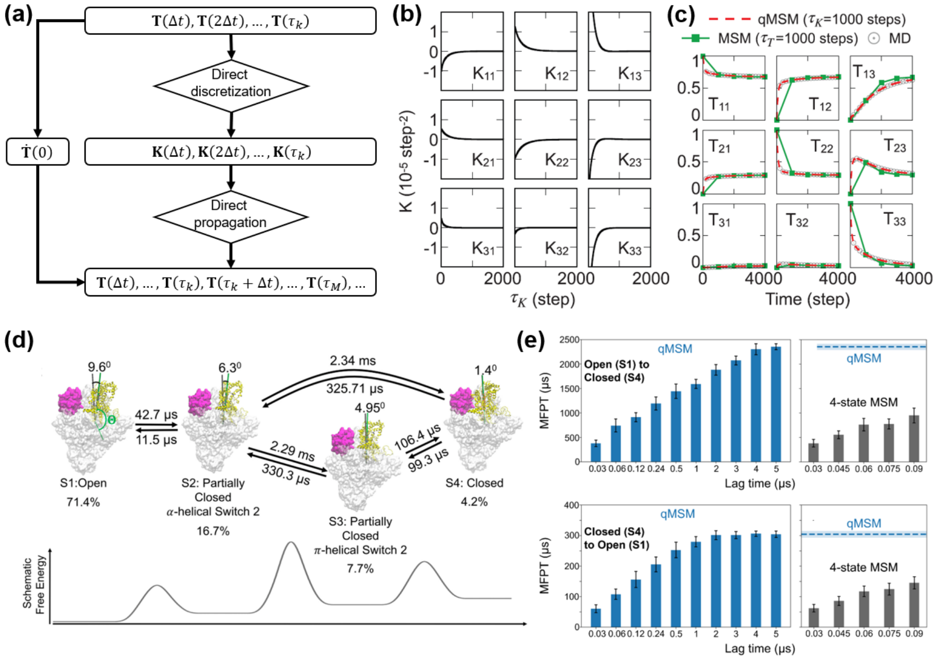


Figure 5.1: Workflow and applications of the qMSM framework. (*a*) *Schematic overview of the qMSM procedure, consisting of two main steps:* (*i*) *estimating the time-dependent memory kernel* $\mathbf{K}(t)$ *from short MD trajectories and* (*ii*) *propagating the discretized GME to predict long-timescale dynamics.* (*b*) *Elements of the memory-kernel matrix* $\mathbf{K}(t)$ *for a three-state toy model, illustrating rapid decay to* (*approximately*) *zero within* ∼*1000 time steps* (*the memory time* τ_k). (*c*) *Comparison of TPMs predicted by qMSM with those estimated directly from MD and by a conventional MSM. qMSM closely reproduces the MD reference and improves upon MSM predictions, indicating more accurate recovery of long-timescale dynamics.* (*d*) *Transition pathways and times between four metastable states from a qMSM of Taq-RNAP clamp opening/closing, shown together with the stationary population of each state. Transitions between S1 and S2 and between S3 and S4 are fast, whereas transitions from S1/S2 to S3/S4 are substantially slower, consistent with higher free-energy barriers separating the two groups of states.* (*e*) *Mean first-passage times* (*MFPTs*) *between S1 and S4 computed at different lag times using the four-state qMSM and MSM. The dashed line denotes the MFPT from the four-state qMSM at a lag time of 5 μs. Panels* (*a-c*) *are adapted from Ref.* [17], *and panels* (*d-e*) *are adapted from Ref.* [144].

where $\mathbf{T}(0) = \mathbf{I}$ has been used. In practice, $\dot{\mathbf{T}}(n\Delta t)$ can be approximated by a finite difference, e.g.,

$$\dot{\mathbf{T}}(n\Delta t) \approx \frac{\mathbf{T}\big((n+1)\Delta t\big) - \mathbf{T}(n\Delta t)}{\Delta t} \tag{5.11}$$

Equations (5.10)–(5.11) allow $\mathbf{K}(n\Delta t)$ to be determined sequentially from previously computed kernels and TPMs at earlier time steps, yielding a self-consistent reconstruction of memory effects. Notably, the transfer-tensor method (TTM) [19] provides a closely related representation of non-Markovian propagation, and can be used within the same theoretical framework.

For many biomolecular systems, a separation between fast local motions and slow conformational changes implies that the memory kernel decays to (approximately) zero beyond a finite memory kernel decay time, denoted τ_k. In practice, τ_k can be estimated by monitoring convergence of the *mean integral kernel* (MIK) to a plateau value,

$$\mathrm{MIK}(t) = \frac{1}{N}\sqrt{\sum_{i,j=1}^{N}\left(\int_0^t K_{ij}(t')\,dt'\right)^2} \tag{5.12}$$

where N is the number of discrete states. Once $\mathbf{K}(t)$ has decayed (up to numerical tolerance), the GME only needs to be solved explicitly up to $t = \tau_k$; beyond this point the dynamics can be propagated without additional memory contributions. After obtaining the time-dependent kernels, the discretized GME (5.9) can be integrated to predict TPMs at longer timescales. Because the convolution term is delicate near $t = 0$, the same discretization convention should be applied consistently in both kernel evaluation and subsequent propagation to ensure numerical stability and internal self-consistency.

qMSMs have been successfully applied to a variety of biomolecular processes that evolve fast and slow motions across multiple timescales. For example, qMSMs constructed for the bacterial RNA polymerase (RNAP) from tens of microseconds of all-atom MD revealed coordinated gating motions during DNA loading [144]. By explicitly accounting for memory through time-dependent kernels, the qMSM reconstructed millisecond-scale conformational transitions that were difficult to resolve with conventional MSMs at feasible lag times. The resulting kinetic analysis suggested a mechanistic coupling between a slow, rate-limiting structural rearrangement and substantially faster auxiliary motions that transiently facilitate gate expansion, offering a physically grounded explanation for how RNAP maintains flexibility during promoter-DNA loading. Similarly, qMSMs have been used to dissect target mRNA recognition by human Argonaute 2 (hAgo2) [171], where the relevant kinetics span microseconds to longer timescales. In this setting, incorporating memory via time-dependent kernels enabled robust reconstruction of recognition kinetics without requiring unrealistically long Markovian lag times, and provided mechanistic insight into how protein–RNA interactions and specific structural elements modulate recognition pathways and trap formation.

5.3.2 Integrative Generalized Master Equation

While qMSMs provide a direct numerical route to solving the GME, their practical implementation requires finite-difference differentiation of estimated TPMs and a recursive reconstruction of the time-dependent memory kernel $\mathbf{K}(t)$. As a result, qMSMs can be sensitive to statistical noise and may exhibit numerical instability, especially for large biomolecular systems where reliable estimation of $\mathbf{T}(t)$ over many lag times is difficult. To mitigate these issues, the IGME provides a analytical solution to the GME, substantially improving numerical stability by leveraging the time-integrated memory kernels.

The key observation motivating IGME is that many biomolecular systems exhibit a separation of timescales: fast intrastate relaxation occurs on timescales much shorter than slow conformational transitions. Under this condition, the memory kernel $\mathbf{K}(t)$ typically decays to (approximately) zero beyond a finite memory time τ_k. Rather than estimating $\mathbf{K}(t)$ directly, IGME introduces the time-integrated memory kernels

$$\mathbf{M}_n(t) = \int_0^t \mathbf{K}(t')(t')^n \, dt' \tag{5.13}$$

which converge to constant matrices $\mathbf{M}_n$ for $t \geq \tau_k$. Expanding $\mathbf{T}(t-t')$ in the GME convolution term as a Taylor series about t and substituting the integral $\mathbf{M}_n$ leads, for $t \geq \tau_k$, to the following reformulation:

$$\mathbf{T}(t)^{-1} \frac{d\mathbf{T}(t)}{dt}\bigg|_{t \geq \tau_k} = \dot{\mathbf{T}}(0) - \mathbf{M}_0 - \sum_{n=1} \left[\mathbf{T}(t)^{-1} \frac{(-1)^n}{n!} \frac{d^n \mathbf{T}(t)}{dt^n} \right] \mathbf{M}_n \tag{5.14}$$

The solution of Eq. (5.14) can be obtained in a self-consistent manner. As a starting point, one considers the zeroth-order truncation of Eq. (5.14), which neglects all terms

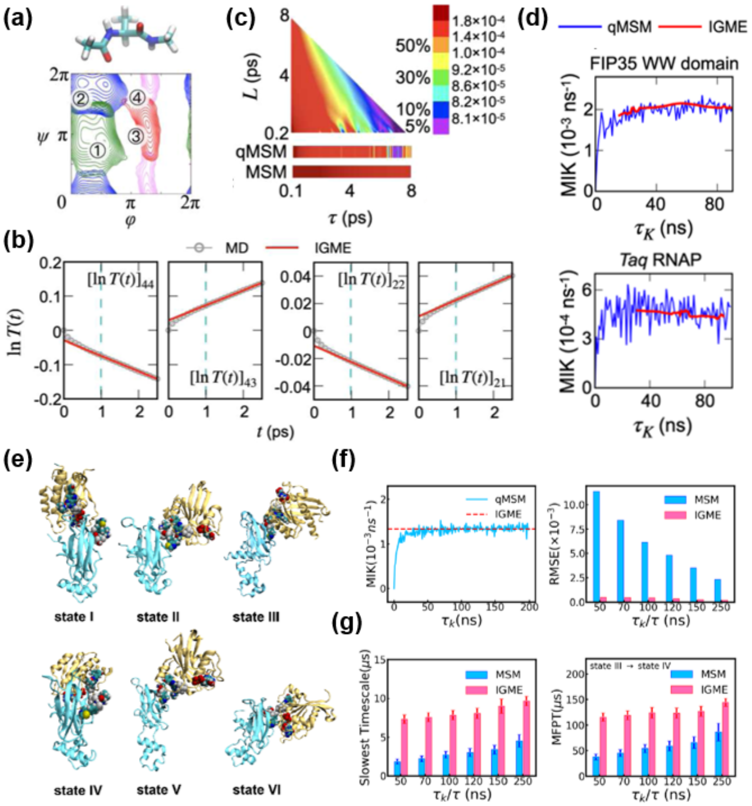


Figure 5.2: Demonstration and applications of the IGME framework. *(a) Ramachandran plot illustrating the four-state model of alanine dipeptide. (b) Representative elements of* $\ln \mathbf{T}(t)$ *from IGME, showing an approximately linear dependence on* t *once memory has decayed* ($t \geq 1$ *ps*)*. (c) RMSE heatmaps for IGME, qMSM, and MSM. For MSM,* τ *denotes the Markovian lag time; for qMSM and IGME,* $\tau = \tau_k$*. The triangular panel shows IGME results across different choices of* τ_k *and fitting window length L. (d) MIK for the FIP35 WW domain and for Taq RNAP gate-opening dynamics. Blue and red curves correspond to qMSM and IGME, respectively. (e) Representative conformations of the KRAS–VHL encounter complex identified by the IGME workflow. (f) For the six-state model, MIK as a function of* τ_k *(left) and RMSE of IGME/MSM-predicted TPMs relative to MD (right). (g) Slowest implied timescale (left) and MFPT from State III to State IV (right) computed from IGME models and MSMs across different lag times. Panels (a-d) are adapted from Ref. [18], and panels (e-g) are adapted from Ref. [118].*

with $n \geq 1$ and yields

$$\mathbf{T}^{(0)}(t)^{-1}\frac{d\mathbf{T}^{(0)}(t)}{dt}\bigg|_{t\geq\tau_k} = \dot{\mathbf{T}}(0) - \mathbf{M}_0. \tag{5.15}$$

The solution of Eq. (5.15) can be written in exponential form. Taking the matrix logarithm, one obtains

$$\ln\left[\left(\mathbf{A}^{(0)}\right)^{-1}\mathbf{T}^{(0)}(t \geq \tau_k)\right] = t\left[\dot{\mathbf{T}}(0) - \mathbf{M}_0\right] \equiv t\ln\hat{\mathbf{T}}^{(0)} \tag{5.16}$$

where $\mathbf{A}^{(0)}$ and $\ln\hat{\mathbf{T}}^{(0)}$ are constant matrices.

Higher-order corrections can then be obtained iteratively. In each iteration k, the current approximation $\mathbf{T}^{(k)}(t)$ is substituted into the right-hand side of Eq. (5.14) to evaluate the derivative, and the resulting equation is integrated to produce an updated solution $\mathbf{T}^{(k+1)}(t)$ on the left-hand side. Repeating this procedure leads to a converged, self-consistent solution for $t \geq \tau_k$, which can be expressed in the IGME form

$$\mathbf{T}(t \geq \tau_k) = \mathbf{A}\hat{\mathbf{T}}^t \tag{5.17}$$

where $\mathbf{A}$ and $\hat{\mathbf{T}}$ are constant matrices. In this representation, $\mathbf{A}$ encodes the non-Markovian contribution arising from short-time memory effects, while $\hat{\mathbf{T}}$ corresponds to the effective Markovian propagator in the long-time limit.

In practice, $\mathbf{A}$ and $\hat{\mathbf{T}}$ can be fit directly from MD-estimated TPMs for $t \geq \tau_k$. Taking the matrix logarithm of Eq. (5.17) gives a linear relation,

$$\ln\mathbf{T}(t \geq \tau_k) = \ln\mathbf{A} + t\ln\hat{\mathbf{T}} \tag{5.18}$$

which enables element-wise linear regression, where $\ln\mathbf{A}$ serves as the intercept and $\ln\hat{\mathbf{T}}$ as the slope (Figure 5.2b). Because $\hat{\mathbf{T}}$ is intended to represent an effective Markovian transition matrix, it should satisfy basic stochastic constraints (e.g., row normalization) and, when appropriate, detailed balance; these conditions can be enforced via constrained fitting. In addition to τ_k, IGME introduces a fitting window length L (fitting over $t \in [\tau_k, \tau_k + L]$). A common strategy is to scan candidate pairs (τ_k, L) and select models that best reproduce MD-derived TPMs, for example by minimizing an RMSE defined over a lag-time range:

$$\mathrm{RMSE} = \left(\frac{\int_0^{L_{\max}} \sum_{i,j=1}^{N}\left[\pi_i T_{ij}^{\mathrm{MD}}(t) - \pi_i T_{ij}^{\mathrm{IGME}}(t)\right]^2 dt}{N^2\int_0^{L_{\max}} dt}\right)^{1/2} \tag{5.19}$$

where N is the number of states and $\boldsymbol{\pi}$ is the stationary distribution. In this form, $\mathbf{T}^{\mathrm{IGME}}(t)$ is constructed using fit parameters obtained from TPMs within $[\tau_k, \tau_k + L]$, while the error is evaluated over $[0, L_{\max}]$. As shown in Figure 5.2c for alanine dipeptide (Figure 5.2a), a systematic scan over the hyperparameters τ_k and L identifies IGME models with consistently low RMSE values. The purple region highlights the top 5% of IGME models with the smallest errors (RMSE $< 8.1 \times 10^{-5}$), corresponding to the most accurate and numerically stable descriptions of the dynamics.

The advantage of IGME's integrated treatment is improved robustness to sampling noise. Relative to qMSMs, IGME yields smoother integrated measures and suppresses oscillatory artifacts that can arise from differentiating noisy TPM estimates. This improvement is especially pronounced for large systems with complex conformational changes, where conventional qMSM kernel estimates can become unstable (Figure 5.2d).

A related approach, referred to as the unbiased generalized master equation (UGME) [39], shares the same projection-operator foundation but adopts a complementary strategy for mitigating noise and numerical instability. Whereas IGME leverages time integration to obtain a compact analytic parametrization of GME, UGME employs a time-local (time-convolutionless) reformulation and reduces variance through direct averaging over short-time propagators. Conceptually, IGME may be viewed as a time-integrated formulation of the GME, while UGME provides a time-local counterpart; under appropriate conditions, both approaches recover consistent long-time kinetic behavior.

IGME has been successfully applied to the KRAS–VHL ternary system [118], where it resolves six metastable macrostates and their representative encounter-complex conformations (Figure 5.2e). These state-resolved ensembles, together with stationary populations and kinetic connectivity, provide a quantitative framework for analyzing non-native protein–protein interaction interfaces relevant to degrader design. Benchmark results in Figures 5.2 f–g further highlight IGME's key advantages, i.e., numerical stability and accurate reproduction of MD-derived TPMs across timescales, thereby outperforming conventional MSMs and enabling reliable kinetic inference from finite-length trajectories in large biomolecular systems. Beyond protein–protein interactions, IGME has also been used to interrogate a range of biomolecular processes. For example, it has been applied to quantify molecular transport within biomolecular condensates formed by peptide–RNA mixtures [143]. In that study, sub-millisecond atomistic simulations of condensates containing Arg-, ADMA-, and Lys-rich peptides were analyzed to extract peptide–RNA binding/unbinding kinetics and to model association–dissociation events that govern peptide diffusion. The resulting analysis indicated that Arg-rich peptides diffuse more slowly and remain bound longer due to persistent hydrogen-bond networks with RNA phosphates, whereas methylated or Lys-substituted variants form weaker and more transient contacts. By reconstructing kinetic rate constants and equilibrium populations from simulation data, IGME further suggested that hydrogen bonding rather than cation–π interactions is the primary determinant of peptide mobility within these condensates, providing a mechanistic link between atomistic interactions and mesoscale transport behavior.

5.4 Discussions

The emergence of non-Markovian frameworks has reshaped the study of protein dynamics by clarifying how memory effects influence the evolution of slow conformational changes. Classical MSMs remain valuable for constructing coarse-grained kinetic networks, but their Markovian assumption can break down in large biomolecular systems, where discretization-induced memory and limited sampling hinder the attainment of truly memoryless dynamics at practical lag times. In contrast, GME-based approaches such as qMSM, IGME, explicitly account for history dependence, providing a principled bridge from microscopic trajectories to effective long-timescale dynamics.

Beyond these developments, several complementary non-Markovian formalisms offer additional perspectives. The generalized Langevin equation (GLE) [29] provides an alternative reduced description in which memory enters through a time-dependent friction kernel together with a corresponding fluctuating noise term, enabling systematic reconstruction of memory kernels consistent with equilibrium statistical mechanics and transport observables. History-augmented MSMs (haMSMs) [140] extend the MSM framework by conditioning transitions on recent state history, thereby improving kinetic accuracy without requiring excessively long lag times. Hidden Markov state models (HMSMs) [107] further generalize MSMs by introducing latent metastable states whose Markovian transitions generate the observed dynamics; this hidden-state representation can reduce apparent non-Markovianity in the observed discrete trajectories by allowing probabilistic (fuzzy) mappings between observed microstates and latent macrostates.

While qMSM and IGME extend the GME formalism to model non-Markovian dynamics between discrete states, they still require a predefined decomposition of configuration space. This requirement can be limiting when the relevant slow processes are more naturally described as continuous, high-resolution motions along CVs, rather than as transitions among a small number of metastable states. Moreover, although qMSM and IGME use memory kernels to correct the dynamics of a given representation, a more powerful strategy is to learn the representation itself—seeking coordinates in which the projected dynamics exhibits intrinsically reduced memory. To this end, we further developed MEMnets (Chapter 6), a deep-learning framework that unifies the GME formalism with neural-network-based representation learning for protein conformational dynamics. Building on the IGME perspective, which enables analytical quantification of memory effects in discrete-state models, MEMnets generalize the GME to continuous latent CVs that retain the essential slow dynamics of biomolecular systems while explicitly driving the reduced dynamics toward minimal memory.

This chapter is a review of related work.

Chapter 6

Deriving Simplicity from Memory – *Memory Kernel Minimization-based Neural Networks for Uncovering Slow Collective Variables of Biomolecular Dynamics*

6.1 Introduction

Investigating biomolecular conformational changes is essential for understanding the molecular mechanisms underlying many key biological processes, such as protein folding [85, 131, 145], protein–ligand binding [13, 146], and conformational transitions [15, 144, 167]. Molecular dynamics (MD) simulations have become a powerful tool to elucidate the dynamics of these conformational changes and often serve as a complementary approach to experiments for gaining mechanistic insight. However, explicit-solvent MD simulations of large biomolecular complexes may involve thousands to millions of atoms, making the resulting trajectory data intrinsically high dimensional. Because the conformational space is represented by the coordinates of all atoms, extracting physically interpretable mechanisms directly from raw MD data remains a major challenge.

These complex conformational changes usually involve multiple, well-separated timescales. The separation between slow collective motions and fast fluctuations provides an opportunity to describe functional conformational changes in terms of a few collective variables (CVs). Identifying such CVs is crucial, as they serve as low-dimensional coordinates that retain the essential kinetic information of the underlying molecular processes. For simple systems, CVs such as the root-mean-square deviation (RMSD) [25], torsion angles [137], or pairwise distances [144] can be manually selected based on physical intuition. However, for more complex biomolecular systems, these intuitive CVs often fail to capture the relevant slow dynamics or lack physical and chemical interpretability.

Numerous methods have been proposed to extract CVs that best capture the slowest dynamical processes. Geometry-based approaches such as principal component analysis (PCA) [60], diffusion maps [24, 46, 122], and isometric feature mapping (ISOMAP) [30] have proven valuable in characterizing large-amplitude motions, but they do not necessarily correlate with kinetic slowness. Alternatively, committor-based approaches [7, 77, 95, 103, 113, 120] identify reaction coordinates corresponding to committor probabilities between predefined states, which limits their applicability to systems involving multiple metastable transitions. Variational principle-based methods such as time-lagged independent component analysis (tICA) [117, 127], kernel-tICA [128], and the variational approach for Markov processes (VAMP) [156, 158] have provided powerful tools to learn CVs directly from MD trajectories by approximating the leading eigen/singular functions of the transfer/koopman operator governing the system's time evolution. Extensions such as VAMPnets [98], SRVs [23], deep-tICA [12], and ISOKANN [119] integrate deep neural networks to capture nonlinear features, enabling more flexible and accurate kinetic representations.

Despite their success, these variational principle-based approaches face two key challenges. First, they are built upon the Markovian assumption, which holds only in the full phase space but breaks down after dimensionality reduction. Projecting the dynamics onto a reduced CV space introduces memory effects, leading to non-Markovian behavior unless perfect timescale separation exists [17, 18, 39]. To restore Markovianity, one must employ sufficiently long lag times for the memory kernel to relax and being ignored. However, this is often impractical in complex systems where accessible MD trajectories are not long enough to reach the required Markovian limit. Second, neural-network-based extensions such as VAMPnets and SRVs do not explicitly regulate the implied timescales, and in cases of insufficient sampling, may produce disconnected models with artificially infinite relaxation times.

In this chapter, we introduce MEMnets (Memory kErnel Minimization-based neural networks) [88], a deep learning framework designed to uncover slow CVs by directly minimizing time-integrated memory kernels. MEMnets employ encoder neural networks to project MD conformations onto a low-dimensional manifold of CVs and optimize these CVs by quantifying and minimizing the temporal memory contained within the reduced representation. To achieve this, we extend our previously developed Integrated Generalized Master Equation (IGME) theory [18] from discrete-states to continuous CV spaces, thereby providing a rigorous theoretical basis for learning memory-minimized embeddings. By applying MEMnets to the alanine dipeptide, FIP35 WW-domain folding, and the clamp-opening motion of bacterial RNA polymerase, we demonstrate that MEMnets yield physically meaningful CVs that faithfully capture the biologically relevant slow dynamic modes in each system.

From a broader machine learning perspective, MEMnets establish a new connection between statistical mechanics and the general theory of representation learning in time-series data. Traditional deep representation learning methods, such as autoencoders [55], variational autoencoders [68], and contrastive learning [152], focus on reconstructing or separating spatial features, thereby compressing the spatial degrees of freedom of the input data. However, such spatial compression inherently discards temporal information when dealing with time-series data. To our best knowledge, MEMnets are the first framework to formalize this phenomenon by introducing the concept that *information loss in spatial degrees of freedom inevitably requires compensatory information in the temporal domain*. To quantify this relationship, MEMnets develop an analytical theory to measure the residual memory resulting from dimensionality reduction and incorporate it as an explicit objective for optimization. In doing so, MEMnets generalize the goal of deep representation learning from reconstructing spatial features to learning embeddings that minimize the amount of memory needed to describe the system's future evolution. This principle, *deriving simplicity from memory*, defines a new paradigm for deep learning in time-series data and opens a path toward physically interpretable, temporally coherent representations of complex

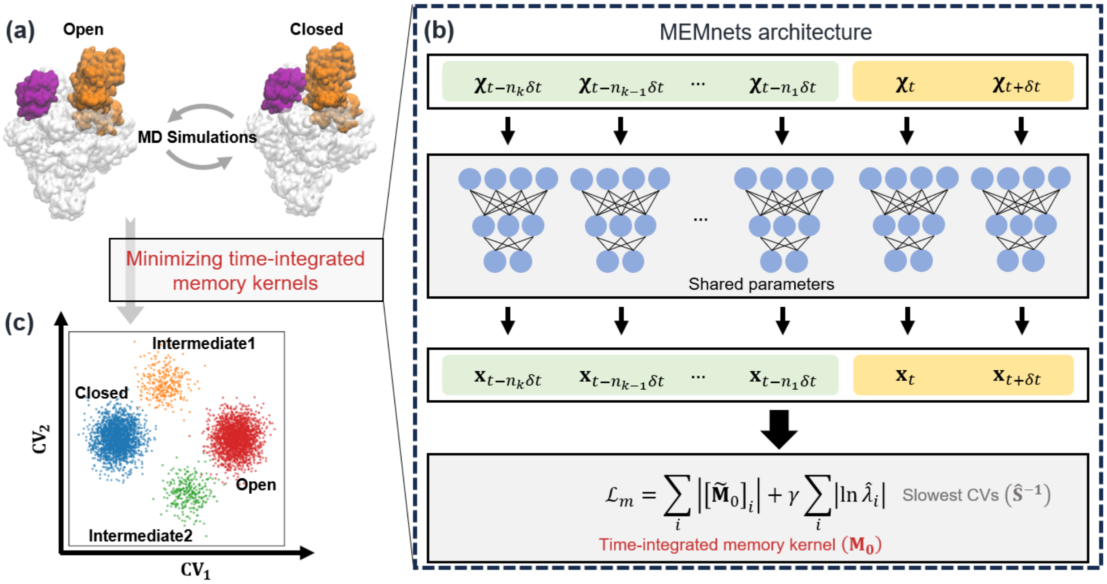


Figure 6.1: Schematic architecture of MEMnets. (*a*) *Time series of molecular conformations obtained from MD simulations are provided as input to parallel encoder networks with shared parameters.* (*b*) *The output continuous coordinates are used to estimate the MEMnets objective function.* (*c*) *By optimizing this objective, MEMnets learns a set of CVs that define a coordinate space with a minimum time-integrated memory kernel.*

dynamical systems.

6.2 Methods

In this section, we present the theoretical and algorithmic foundations of MEMnets (Figure 6.1), which unifies the principles of memory kernels with deep representation learning for time-series data. We begin by revisiting the origin of memory effects in molecular dynamics through projection-operator theory, illustrating how dimensionality reduction from full phase space to a subset of continuous observables inherently introduces non-Markovian behavior. Building on this foundation, We derive and extend the IGME formalism from discrete state models to continuous coordinates, enabling a direct characterization of non-Markovian effects in low-dimensional coordinate spaces. Next, we introduce a formal definition of the time-integrated memory kernel for continuous CVs and derive how this quantity can be analytically expressed and numerically evaluated from MD trajectories. Finally, we describe the architecture, training objective, and optimization scheme of MEMnets, which leverages neural encoders to learn CVs by explicitly minimizing the time-integrated memory kernels, thereby ensuring that the resulting latent representations capture the slow, quasi-Markovian dynamics of biomolecular systems.

6.2.1 Integrative Generalized Master Equation for Continuous Dynamics

Projecting Markovian dynamics from the high-dimensional phase space onto a reduced set of coordinates $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^\top$ introduces memory effects (Note that here, $\boldsymbol{x}$ represents the reaction coordinates, while in previous chapters it typically denotes a molecular configuration). We begin our derivations from the original high-dimensional dynamics, which satisfy the Liouville equation and represent a Markovian process:

$$\frac{d}{dt}\boldsymbol{x}_t = \mathcal{L}\boldsymbol{x}_t, \qquad \boldsymbol{x}_t = e^{\mathcal{L}t}\boldsymbol{x}_0 \tag{6.1}$$

where $\mathcal{L} = \mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ denotes the Liouville operator. The time-lagged correlation functions of $\boldsymbol{x}$ can be derived from the Liouville equation and expressed as

$$\mathbf{C}(t) = \overline{\boldsymbol{x}_0\boldsymbol{x}_t} = \frac{1}{N_f}\hat{\boldsymbol{x}}\, e^{\hat{\mathcal{L}}t}\, \hat{\boldsymbol{x}}^\top, \qquad \mathbf{C}(0)^{-1}\mathbf{C}(t) = (\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top)^{-1}\hat{\boldsymbol{x}}\, e^{\hat{\mathcal{L}}t}\, \hat{\boldsymbol{x}}^\top \tag{6.2}$$

where $\hat{\boldsymbol{x}} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l, \ldots]$ represents all MD frames, and l indexes the l^{th} snapshot along the trajectory. Here, N_f is the total number of frames, $C_{ij}(t) = \frac{1}{N_f}\hat{\boldsymbol{x}}_i e^{\hat{\mathcal{L}}t}\hat{\boldsymbol{x}}_j^\top$, and $\hat{\boldsymbol{x}}_i$ is the i^{th} row vector of $\hat{\boldsymbol{x}}$. In this formulation, N_f refers to the number of MD snapshots (not discrete states). Each MD snapshot is projected onto continuous coordinates, rendering their values continuous. In the ergodic limit, N_f approaching infinity is required for complete sampling; in practice, however, MD simulations typically yield hundreds of thousands to millions of snapshots.

We can further compute the averaged operator $e^{\hat{\mathcal{L}}t}$ projected onto the continuous coordinates $\boldsymbol{x}$ and define the projection operator $\mathbb{P}$ as

$$\langle e^{\hat{\mathcal{L}}t}\rangle_{\boldsymbol{x}} \equiv \hat{\boldsymbol{x}}\, e^{\hat{\mathcal{L}}t}\, \hat{\boldsymbol{x}}^\top(\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top)^{-1}, \qquad \mathbb{P} = \hat{\boldsymbol{x}}^\top(\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top)^{-1}\hat{\boldsymbol{x}} \tag{6.3}$$

Applying the Dyson operator identity derived from the Nakajima–Zwanzig equation, we obtain

$$\frac{d}{dt}\mathbb{P}e^{\hat{\mathcal{L}}t} = \mathbb{P}\hat{\mathcal{L}}\mathbb{P}e^{\hat{\mathcal{L}}t} + \int_0^t \mathbb{P}\hat{\mathcal{L}}e^{\mathbb{Q}\hat{\mathcal{L}}s}\mathbb{Q}\hat{\mathcal{L}}\mathbb{P}e^{\hat{\mathcal{L}}(t-s)}\,ds \tag{6.4}$$

where $\mathbb{Q} = 1 - \mathbb{P}$ is the complementary projection operator.

Following analogous derivations for discrete-state models (Section 5.2.2), we can express the resulting dynamics in the form of a Generalized Master Equation (GME) for continuous coordinates:

$$\frac{d}{dt}\mathbf{T}(t) = \mathbf{T}(t)\mathbf{R}_0 - \int_0^t \mathbf{T}(t-s)\mathbf{K}(s)\,ds \tag{6.5}$$

where $\mathbf{T}(t) = \mathbf{C}(0)^{-1}\mathbf{C}(t)$, $\mathbf{R}_0 = \dot{\mathbf{T}}(0)$, and the memory kernel is defined as

$$\mathbf{K}(s)^\top = -\hat{\boldsymbol{x}}\hat{\mathcal{L}}e^{\mathbb{Q}\hat{\mathcal{L}}s}\mathbb{Q}\hat{\mathcal{L}}\hat{\boldsymbol{x}}^\top(\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top)^{-1} \tag{6.6}$$

Equation 6.5 presents the GME for biomolecular dynamics projected onto continuous coordinates $\boldsymbol{x}$. Unlike the GME for discrete state models (Chapter 5), which relies on the transition probability matrix $\mathbf{T}(t)$ among discrete states, the continuous-coordinate formulation defines $\mathbf{T}(t) = \mathbf{C}(0)^{-1}\mathbf{C}(t)$, which is also called the Koopman matrix, where $\mathbf{C}(0)$ and $\mathbf{C}(t)$ denote the matrices of time-instantaneous and time-lagged correlation functions of $\boldsymbol{x}$ (Eq. (6.2)).

Next, we define the n^{th}-order time-integrated memory kernel $\mathbf{M}_n(t)$ as

$$\mathbf{M}_n(t) = \int_0^t \mathbf{K}(s)\,s^n\,ds \tag{6.7}$$

where $\mathbf{K}(t)$ is the memory kernel introduced in Eq. (6.5). When the memory kernel fully decays at $t \geq \tau_K$, these integrals converge to constant matrices: $\mathbf{M}_n(t \geq \tau_K) = \mathbf{M}_n(\tau_K) \equiv \mathbf{M}_n$

In this regime, the GME can be expressed as an ordinary differential equation

(ODE) through the Taylor expansion of $\mathbf{T}(t-s)$:

$$\frac{d}{dt}\mathbf{T}(t \geq \tau_K) = \mathbf{T}(t)\big(\mathbf{R}_0 - \mathbf{M}_0\big) - \sum_{n=1} \frac{(-1)^n}{n!} \frac{d^n \mathbf{T}(t)}{dt^n} \mathbf{M}_n \tag{6.8}$$

The analytical solution to Eq. (6.8) retains the same mathematical form as that derived previously for state models (Section 5.3.2):

$$\mathbf{T}(t \geq \tau_K) = \mathbf{A}\,\hat{\mathbf{T}}^t, \qquad \ln\hat{\mathbf{T}} = \mathbf{R}_0 - \mathbf{M}_0 - \sum_{n=1} \frac{(-1)^n}{n!} \big(\ln\hat{\mathbf{T}}\big)^n \mathbf{M}_n \tag{6.9}$$

where $\mathbf{A}$ and $\hat{\mathbf{T}}$ are constant matrices of size $N \times N$, and N corresponds to the number of reaction coordinates. Starting from the second relation in Eq. (6.9), we obtain

$$\mathbf{R}_0 - \ln\hat{\mathbf{T}} = \mathbf{M}_0 + \sum_{n=1} \frac{(-1)^n}{n!} \big(\hat{\mathbf{T}}^{-t} \frac{d^n}{dt^n} e^{t\ln\hat{\mathbf{T}}}\big)\mathbf{M}_n \tag{6.10}$$

Using the Taylor expansion of $\hat{\mathbf{T}}^{(t-s)}$ at $s=0$ and the definition $\mathbf{M}_n(t) = \int_0^t \mathbf{K}(s)s^n\,ds$, we further derive:

$$\int_0^{\tau_K} \hat{\mathbf{T}}^{-s}\mathbf{K}(s)\,ds = \hat{\mathbf{T}}^{-t} \int_0^{\tau_K} \hat{\mathbf{T}}^{(t-s)}\mathbf{K}(s)\,ds = \mathbf{M}_0 + \sum_{n=1} \frac{(-1)^n}{n!} \big(\hat{\mathbf{T}}^{-t} \frac{d^n}{dt^n} \hat{\mathbf{T}}^t\big)\mathbf{M}_n \tag{6.11}$$

The right-hand sides of Eqs. (6.10) and (6.11) are identical, yielding

$$\mathbf{R}_0 - \ln\hat{\mathbf{T}} = \int_0^{\tau_K} \hat{\mathbf{T}}^{-s}\mathbf{K}(s)\,ds \tag{6.12}$$

As discussed in previous IGME work [18], for biomolecular dynamics, $\hat{\mathbf{T}}$ represents the slowest dynamic modes and thus its eigenvalues are close to unity, corresponding to long implied timescales. Consequently, $\hat{\mathbf{T}}$ is approximately an identity matrix with eigenvalues $0 < \hat{\lambda}_i \leq 1$, implying $\hat{\mathbf{T}}^{-s} \geq \mathbf{I}$. The upper bound of the time-integrated memory kernel $\mathbf{M}_0$ can therefore be derived as

$$\Big|\int_0^{\tau_K} \hat{\mathbf{T}}^{-s}\mathbf{K}(s)\,ds\Big| \geq \Big|\int_0^{\tau_K} \mathbf{K}(s)\,ds\Big| = |\mathbf{M}_0|, \qquad |\mathbf{M}_0| \leq |\mathbf{R}_0 - \ln\hat{\mathbf{T}}| \equiv |\mathbf{M}_0^{(\mathrm{sup})}| \tag{6.13}$$

Here, $\mathbf{M}_0^{(\mathrm{sup})} = \mathbf{R}_0 - \ln \hat{\mathbf{T}}$ denotes the supremum of $\mathbf{M}_0$. This upper bound can be employed as the objective function for minimizing $\mathbf{M}_0$ for any given set of continous coordinates $\boldsymbol{x}$.

However, it is important to note that when $\boldsymbol{x}$ corresponds to the direct outputs of encoder neural networks (Figure 6.1), they represent nonlinear combinations of input features and are not guaranteed to be orthonormal. Consequently, $\boldsymbol{x}$ does not necessarily satisfy the orthonormality requirements of CVs or eigenmodes of the dynamics. To address this, we introduce a linear transformation $\boldsymbol{\phi}_{\boldsymbol{x}}^{\top}$ that converts $\boldsymbol{x}$ into an orthonormal set of CVs $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_N]^{\top}$:

$$\boldsymbol{\xi} = \boldsymbol{\phi}_{\boldsymbol{x}}^{\top} \boldsymbol{x} \tag{6.14}$$

The transformation $\boldsymbol{\phi}_{\boldsymbol{x}}^{\top}$ is not unique and can be chosen in various ways. For instance, $\boldsymbol{\phi}_{(\boldsymbol{x},\tau)}^{\top}$ may be selected as the right-eigenvector matrix of the Koopman matrix $\mathbf{T}^{(\boldsymbol{x})}(\tau)$, constructed from continuous coordinates $\boldsymbol{x}$ at a lag time $\tau \geq \tau_K$.

6.2.2 Time-integrated Memory Kernels of Collective Variables

The goal of MEMnets is to identify an optimal set of continuous CVs, or equivalently the slowest dynamic modes, by minimizing their time-integrated memory kernel matrix $\mathbf{M}_0$. To achieve this, the objective function is defined as the summation of the absolute values of all elements in the upper-bound matrix of $\mathbf{M}_0$ (Eq. (6.13)). As discussed in the previous section, the encoder neural network outputs $\boldsymbol{x}$ are not inherently orthonormal. To address this, we apply a linear transformation using the right-eigenvector matrix $\boldsymbol{\phi}_{(\boldsymbol{x},\tau)}^{\top}$ at a selected lag time τ, which orthonormalizes the encoder outputs $\boldsymbol{x}$ to yield $\boldsymbol{\xi}$ (Eq. (6.14)). With the orthonormalized $\boldsymbol{\xi}$, one can then compute $\mathbf{T}^{(\boldsymbol{\xi})}(t)$ at different lag times t. From $\mathbf{T}^{(\boldsymbol{\xi})}(t)$, the matrix elements of $\ln \hat{\mathbf{T}}$, $\mathbf{A}$, and $\mathbf{R}_0$ in Eq. (6.9) can be obtained via least-squares fitting.

While conceptually straightforward, this procedure presents practical challenges in a deep-learning context. First, it is computationally inefficient, as it requires repeated linear transformations of $\boldsymbol{x}$ and least-squares fitting to estimate the full matrix forms of $\mathbf{R}_0$ and $\ln \hat{\mathbf{T}}$ at each training step. Since training a MEMnets model typically involves thousands of iterations, such repeated fitting leads to prohibitive computational costs. Second, the transformation $\boldsymbol{\phi}^{\top}_{(\boldsymbol{x},\tau)}$ varies with training, and recalculating the matrix elements of $\ln\hat{\mathbf{T}}$ (including its off-diagonal components) at each step may introduce numerical instability.

To overcome these challenges, we introduce an efficient and numerically stable approach to estimate the upper bound of $\mathbf{M}_0$ directly in the $\boldsymbol{x}$ space. Specifically, we demonstrate that $\mathbf{M}_0^{(\mathrm{sup})}$ can be approximated by $\tilde{\mathbf{M}}_0$, which is derived from the diagonal eigenvalue matrix of $\mathbf{T}^{(\boldsymbol{x})}(t)$ computed directly from the encoder outputs:

$$\tilde{\mathbf{M}}_0 = \frac{\ln\boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t)}{\delta t} - \ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})} \tag{6.15}$$

where $\boldsymbol{\lambda}^{(\boldsymbol{x})}(t)$ is the diagonal eigenvalue matrix of $\mathbf{T}^{(\boldsymbol{x})}(t)$, and $\ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}$ denotes the diagonal eigenvalue matrix of $\ln\hat{\mathbf{T}}^{(\boldsymbol{x})}$. From this relation, the following inequality can be derived:

$$\frac{1}{\alpha_0}\sum_{i,j}\left|[\mathbf{M}_0]_{ij}\right| \le \sum_i\left|[\tilde{\mathbf{M}}_0]_{ii}\right| + \gamma_0\sum_i\left|\ln\hat{\lambda}_i^{(\boldsymbol{x})}\right| \tag{6.16}$$

where α_0 and γ_0 are constants that depend on N, the number of CVs (i.e., the dimensionality of $\mathbf{M}_0$). Specifically,

$$\alpha_0 = 1 + 2(N-1) + 2N(N-1), \qquad \gamma_0 = \frac{2(N-1)+2N(N-1)}{1+2(N-1)+2N(N-1)}$$

γ_0 is typically close to unity (e.g., $\gamma_0 = 0.94$ for $N = 3$).

We then use the upper bound of $\sum_{ij}|[\mathbf{M}_0]_{ij}|$ in Eq. (6.16) to define the MEMnets loss function:

$$\mathcal{L}_m = \sum_i\left|[\tilde{\mathbf{M}}_0]_{ii}\right| + \gamma\sum_i\left|\ln\hat{\lambda}_i^{(\boldsymbol{x})}\right| = \sum_i\left|\frac{\ln\lambda_i^{(\boldsymbol{x})}(\delta t)}{\delta t} - \ln\hat{\lambda}_i^{(\boldsymbol{x})}\right| + \gamma\sum_i\left|\ln\hat{\lambda}_i^{(\boldsymbol{x})}\right| \tag{6.17}$$

Equation (6.16) is derived under the condition $|\delta\mathbf{y}_{\delta t,\tau}| = |\mathbf{y}_{\delta t,\tau} - \mathbf{I}| \le 1$, where $\mathbf{y}_{\delta t,\tau} = \boldsymbol{\phi}^{-1}_{(\boldsymbol{x},\delta t)}\boldsymbol{\phi}_{(\boldsymbol{x},\tau)}$. This condition reaches its theoretical maximum only when the left and right eigenvectors of the Koopman matrices at δt and τ are orthogonal (or vice versa). In biomolecular systems, however, we observe that $|\delta\mathbf{y}_{\delta t,\tau}|$ (or $|\delta\mathbf{y}_{\tau,\delta t}|$) is substantially smaller than 1, leading to a reduced effective γ_0. The refined upper bound is

$$\gamma_0 = \frac{2(N-1)\varepsilon + 2N(N-1)\varepsilon^2}{1 + 2(N-1)\varepsilon + 2N(N-1)\varepsilon^2} \tag{6.18}$$

where $\varepsilon = \max\left[\sum_i \frac{|\delta y_{ij}|}{N}, \sum_j \frac{|\delta y_{ij}|}{N}\right]$. For example, in the FIP35 WW domain system (Figure 6.4), $\varepsilon \approx 0.07$, yielding $\gamma_0 \approx 0.4$, while for Taq RNAP (Figure 6.5), $\varepsilon \approx 0.3$, giving $\gamma_0 \approx 0.7$.

Accordingly, when implementing the loss function in Eq. (6.17), we adopt the following practical strategy for γ: we initialize with a relatively large value ($\gamma = 2$) to overemphasize the $\sum_i |\ln \hat{\lambda}_i^{(\boldsymbol{x})}|$ term, which helps anchor the dominant slow dynamic mode, and gradually decrease γ from 2 to 0.5 as training progresses.

Detailed derivations. Next, we provide the detailed derivation of Eq. (6.16). The matrix $\mathbf{M}_0$ in the orthonormal CV ($\boldsymbol{\xi}$) space is computed from $\mathbf{R}_0^{(\boldsymbol{\xi})}$ and $\ln \hat{\mathbf{T}}^{(\boldsymbol{\xi})}$, as shown in Eq. (6.13), where $\hat{\mathbf{T}}^{(\boldsymbol{\xi})}$ is obtained from $\mathbf{T}^{(\boldsymbol{\xi})}(t \geq \tau_K)$. We begin by deriving the equation of $\mathbf{T}^{(\boldsymbol{\xi})}(t)$, followed by the derivation of $\hat{\mathbf{T}}^{(\boldsymbol{\xi})}$ and $\mathbf{R}_0^{(\boldsymbol{\xi})}$. Finally, we arrive at Eq. (6.16).

We define

$$\mathbf{x}_0^\top = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots]^\top, \quad \Xi_0^\top = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \ldots]^\top,$$

$$\mathbf{x}_\tau^\top \equiv e^{\mathcal{L}\tau}\mathbf{x}_0^\top = [\boldsymbol{x}_{1+\tau}, \boldsymbol{x}_{2+\tau}, \boldsymbol{x}_{3+\tau}, \ldots]^\top, \quad \Xi_\tau^\top \equiv e^{\mathcal{L}\tau}\Xi_0^\top = [\boldsymbol{\xi}_{1+\tau}, \boldsymbol{\xi}_{2+\tau}, \boldsymbol{\xi}_{3+\tau}, \ldots]^\top.$$

where the subscript represents the time frame index in the trajectory. Following Eq. (6.2) and Eq. (6.14), we have

$$\begin{aligned}
\Xi_0 &= \boldsymbol{\phi}_{(\boldsymbol{x},\tau)}^\top \mathbf{x}_0, \\
\Xi_\tau &= \boldsymbol{\phi}_{(\boldsymbol{x},\tau)}^\top \mathbf{x}_\tau, \\
\mathbf{T}^{(\boldsymbol{x})}(\tau) &= (\mathbf{x}_0\mathbf{x}_0^\top)^{-1}\mathbf{x}_0\mathbf{x}_\tau^\top, \\
\mathbf{T}^{(\boldsymbol{\xi})}(\tau) &= (\Xi_0\Xi_0^\top)^{-1}\Xi_0\Xi_\tau^\top
\end{aligned} \tag{6.19}$$

Based on the above definitions, we next prove that $\mathbf{T}^{(\boldsymbol{\xi})}(\tau)$ in the CV space is a diagonal matrix. Here, $\mathbf{T}^{(\boldsymbol{x})}(t)$ refers to the Koopman matrix in the $\boldsymbol{x}$ space, which is the direct output of the encoder neural network.

Combining the above equations, $\mathbf{T}^{(\boldsymbol{\xi})}(\tau)$ can be written as:

$$\begin{aligned}\mathbf{T}^{(\boldsymbol{\xi})}(\tau) &= (\phi_{(\boldsymbol{x},\tau)}^{\top}\boldsymbol{x}_0\boldsymbol{x}_0^{\top}\phi_{(\boldsymbol{x},\tau)})^{-1}\phi_{(\boldsymbol{x},\tau)}^{\top}\boldsymbol{x}_0\boldsymbol{x}_\tau^{\top}\phi_{(\boldsymbol{x},\tau)} \\ &= \phi_{(\boldsymbol{x},\tau)}^{-1}\mathbf{T}^{(\boldsymbol{x})}(\tau)\phi_{(\boldsymbol{x},\tau)} = \phi_{(\boldsymbol{x},\tau)}^{-1}\phi_{(\boldsymbol{x},\tau)}\boldsymbol{\lambda}^{(\boldsymbol{x})}(\tau)\phi_{(\boldsymbol{x},\tau)}^{-1}\phi_{(\boldsymbol{x},\tau)} = \boldsymbol{\lambda}^{(\boldsymbol{x})}(\tau)\end{aligned}$$

Thus, $\mathbf{T}^{(\boldsymbol{\xi})}(\tau)$ is diagonal, and its diagonal elements are the eigenvalues of $\mathbf{T}^{(\boldsymbol{x})}(\tau)$. This is expected, since $\phi_{(\boldsymbol{x},\tau)}^{\top}$ was chosen as the right eigenvector matrix of $\mathbf{T}^{(\boldsymbol{x})}(\tau)$ at τ. However, for other lag times $t \neq \tau$, e.g., $t = \delta t$, $\mathbf{T}^{(\boldsymbol{\xi})}(\delta t)$ is not diagonal because $\phi_{(\boldsymbol{x},\delta t)}^{\top} \neq \phi_{(\boldsymbol{x},\tau)}^{\top}$.

Following a similar procedure, we define

$$\mathbf{x}_{\delta t}^{\top} \equiv e^{\mathcal{L}\delta t}\mathbf{x}_0^{\top}, \quad \boldsymbol{\Xi}_{\delta t}^{\top} \equiv e^{\mathcal{L}\delta t}\boldsymbol{\Xi}_0^{\top},$$

and obtain

$$\begin{aligned}\boldsymbol{\Xi}_0 &= \phi_{(\boldsymbol{x},\tau)}^{\top}\mathbf{x}_0, \\ \boldsymbol{\Xi}_{\delta t} &= \phi_{(\boldsymbol{x},\tau)}^{\top}\mathbf{x}_{\delta t}, \\ \mathbf{T}^{(\boldsymbol{x})}(\delta t) &= (\mathbf{x}_0\mathbf{x}_0^{\top})^{-1}\mathbf{x}_0\mathbf{x}_{\delta t}^{\top}, \\ \mathbf{T}^{(\boldsymbol{\xi})}(\delta t) &= (\boldsymbol{\Xi}_0\boldsymbol{\Xi}_0^{\top})^{-1}\boldsymbol{\Xi}_0\boldsymbol{\Xi}_{\delta t}^{\top}.\end{aligned} \tag{6.20}$$

As discussed in Eq. (6.13), the memory kernel is bounded by

$$\mathbf{M}_0^{(\boldsymbol{\xi},\mathrm{sup})} \equiv \mathbf{R}_0^{(\boldsymbol{\xi})} - \ln\hat{\mathbf{T}}^{(\boldsymbol{\xi})},$$

which is computed from $\mathbf{R}_0^{(\boldsymbol{\xi})}$ and $\ln\hat{\mathbf{T}}^{(\boldsymbol{\xi})}$. Therefore, we first derive $\mathbf{R}_0^{(\boldsymbol{\xi})}$ and $\ln\hat{\mathbf{T}}^{(\boldsymbol{\xi})}$. From its definition,

$$\mathbf{R}_0^{(\boldsymbol{\xi})} = \frac{d\mathbf{T}^{(\boldsymbol{\xi})}(t \to 0)}{dt} = \lim_{dt\to 0}\frac{\mathbf{T}^{(\boldsymbol{\xi})}(dt) - \mathbf{I}}{dt} = \lim_{dt\to 0}\frac{\ln\mathbf{T}^{(\boldsymbol{\xi})}(dt)}{dt} \approx \frac{\ln\mathbf{T}^{(\boldsymbol{\xi})}(\delta t)}{\delta t} \tag{6.21}$$

Combining Eq. (6.20), we obtain:

$$\begin{aligned}\mathbf{T}^{(\boldsymbol{\xi})}(\delta t) &= \phi_{(\boldsymbol{x},\tau)}^{-1}\mathbf{T}^{(\boldsymbol{x})}(\delta t)\phi_{(\boldsymbol{x},\tau)} \\ &= \phi_{(\boldsymbol{x},\tau)}^{-1}\phi_{(\boldsymbol{x},\delta t)}\boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t)\phi_{(\boldsymbol{x},\delta t)}^{-1}\phi_{(\boldsymbol{x},\tau)} \\ &= \mathbf{y}_{\tau\delta t}^{(\boldsymbol{x})}\boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t)\mathbf{y}_{\delta t\tau}^{(\boldsymbol{x})}\end{aligned} \tag{6.22}$$

where $\boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t)$ represents the diagonal eigenvalue matrix of $\mathbf{T}^{(\boldsymbol{x})}(\delta t)$, $\mathbf{y}^{(\boldsymbol{x})}_{\tau\delta t} \equiv \boldsymbol{\phi}^{-1}_{(\boldsymbol{x},\tau)}\boldsymbol{\phi}_{(\boldsymbol{x},\delta t)}$, and $\mathbf{y}^{(\boldsymbol{x})}_{\delta t\tau}$ is its inverse. This indicates that for very small δt, $\mathbf{T}^{(\boldsymbol{\xi})}(\delta t)$ is not diagonal. Substituting Eq. (6.22) into Eq. (6.21), we obtain

$$\mathbf{R}_0^{(\boldsymbol{\xi})} = \frac{\ln \mathbf{T}^{(\boldsymbol{\xi})}(\delta t)}{\delta t} = \mathbf{y}_{\tau\delta t}\frac{\ln \boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t)}{\delta t}\mathbf{y}_{\delta t\tau} \equiv \mathbf{y}_{\tau\delta t}\tilde{\mathbf{R}}\mathbf{y}_{\delta t\tau} \tag{6.23}$$

where $\tilde{\mathbf{R}} \equiv \ln \boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t)/\delta t$. Defining $\mathbf{I} \equiv \mathbf{y}_{\tau\delta t}\mathbf{y}_{\delta t\tau}$, $\delta\mathbf{y}_{\tau\delta t} = \mathbf{y}_{\tau\delta t} - \mathbf{I}$, and $\delta\mathbf{y}_{\delta t\tau} = \mathbf{y}_{\delta t\tau} - \mathbf{I}$, we have

$$\mathbf{I} + \delta\mathbf{y}_{\tau\delta t} + \delta\mathbf{y}_{\delta t\tau} + \delta\mathbf{y}_{\tau\delta t}\delta\mathbf{y}_{\delta t\tau} = \mathbf{I} \quad \Rightarrow \quad \delta\mathbf{y}_{\tau\delta t} + \delta\mathbf{y}_{\delta t\tau} = -\delta\mathbf{y}_{\tau\delta t}\delta\mathbf{y}_{\delta t\tau} \tag{6.24}$$

From Eq. (6.23), we obtain:

$$\begin{aligned} [\mathbf{R}_0^{(\boldsymbol{\xi})}]_{ii} &= \tilde{R}_i - \sum_k [\delta\mathbf{y}_{\tau\delta t}]_{ik}[\delta\mathbf{y}_{\delta t\tau}]_{ki}(\tilde{R}_i - \tilde{R}_k), \\ [\mathbf{R}_0^{(\boldsymbol{\xi})}]_{i\neq j} &= (\tilde{R}_i - \tilde{R}_j)[\delta\mathbf{y}_{\delta t\tau}]_{ij} - \sum_k [\delta\mathbf{y}_{\tau\delta t}]_{ik}[\delta\mathbf{y}_{\delta t\tau}]_{kj}(\tilde{R}_j - \tilde{R}_k) \end{aligned} \tag{6.25}$$

where $\tilde{R}_i = [\tilde{\mathbf{R}}]_{ii}$, since $\tilde{\mathbf{R}}$ is diagonal.

The IGME solution $\mathbf{T}(t \geq \tau_K) = \mathbf{A}\hat{\mathbf{T}}^t$ (Eq. (6.9)) applies when the memory kernel has fully decayed ($t \geq \tau_K$). For a small range around τ, $t \in [\tau - d\tau, \tau + d\tau]$, we assume $\mathbf{T}^{(\boldsymbol{\xi})}$ in this interval shares the same eigenvectors as $\mathbf{T}^{(\boldsymbol{\xi})}(\tau)$. Since $\ln \mathbf{T}^{(\boldsymbol{\xi})}(\tau)$ is diagonal (Eq. (6.20)), $\ln \mathbf{T}^{(\boldsymbol{\xi})}(t)$ in this range is also diagonal, leading to

$$\hat{\mathbf{T}}^{(\boldsymbol{\xi})} = \hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}, \qquad \ln \hat{\mathbf{T}}^{(\boldsymbol{\xi})} = \ln \hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})} \tag{6.26}$$

where $\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}$ is the diagonal eigenvalue matrix of $\hat{\mathbf{T}}^{(\boldsymbol{x})}$. Combining the above equations and plug them into the definition of $\mathbf{M}_0^{(\boldsymbol{\xi},\mathrm{sup})}$:

$$\begin{aligned}
[\mathbf{M}_0^{(\boldsymbol{\xi},\mathrm{sup})}]_{ii} = \tilde{M}_{0,i}& \\
&+\sum_k [\delta\mathbf{y}_{\tau\delta t}]_{ik}[\delta\mathbf{y}_{\delta t\tau}]_{ki}(\tilde{M}_{0,i}-\tilde{M}_{0,k}) \\
&+\sum_k [\delta\mathbf{y}_{\tau\delta t}]_{ik}[\delta\mathbf{y}_{\delta t\tau}]_{ki}\big(\ln\hat{\lambda}_i^{(\boldsymbol{x})}-\ln\hat{\lambda}_k^{(\boldsymbol{x})}\big), \\
[\mathbf{M}_0^{(\boldsymbol{\xi},\mathrm{sup})}]_{i\neq j} = (\tilde{M}_{0,i}&-\tilde{M}_{0,j})[\delta\mathbf{y}_{\delta t\tau}]_{ij} \\
&+\big(\ln\hat{\lambda}_i^{(\boldsymbol{x})}-\ln\hat{\lambda}_j^{(\boldsymbol{x})}\big)[\delta\mathbf{y}_{\delta t\tau}]_{ij} \\
&-\sum_k [\delta\mathbf{y}_{\tau\delta t}]_{ik}[\delta\mathbf{y}_{\delta t\tau}]_{kj}(\tilde{M}_{0,j}-\tilde{M}_{0,k}) \\
&-\sum_k [\delta\mathbf{y}_{\tau\delta t}]_{ik}[\delta\mathbf{y}_{\delta t\tau}]_{kj}\big(\ln\hat{\lambda}_j^{(\boldsymbol{x})}-\ln\hat{\lambda}_k^{(\boldsymbol{x})}\big)
\end{aligned} \tag{6.27}$$

where $\tilde{M}_{0,i}=[\tilde{\mathbf{M}}_0]_{ii}$ (Eq. (6.15)), $\ln\hat{\lambda}_i^{(\boldsymbol{x})}=[\ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}]_{ii}$, since $\tilde{\mathbf{M}}_0$ and $\ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}$ are diagonal matrices.

In Eq. (6.27), the matrix elements in both $|\delta\mathbf{y}_{\tau\delta t}|$ and $|\delta\mathbf{y}_{\delta t\tau}|$ are bounded by 1, as $\mathbf{y}_{\tau\delta t}$ and $\mathbf{y}_{\delta t\tau}$ represent dot products between two sets of orthonormal eigenvector matrices. We can thus obtain the upper bound of $\mathbf{M}_0^{(\boldsymbol{\xi})}$:

$$\begin{aligned}
\sum_{ij}\big|[\mathbf{M}_0^{(\boldsymbol{\xi})}]_{ij}\big| \leq \sum_{ij}\big|[\mathbf{M}_0^{(\boldsymbol{\xi},\mathrm{sup})}]_{ij}\big|& \\
&\leq (1+2(N-1)\varepsilon(1+N\varepsilon))\left(\sum_i\left(|\tilde{M}_{0,i}|+|\ln\hat{\lambda}_i|\frac{2(N-1)\varepsilon(1+N\varepsilon)}{1+2(N-1)\varepsilon(1+N\varepsilon)}\right)\right) \\
&\leq (1+2(N-1)(N+1))\left(\sum_i\left(|\tilde{M}_{0,i}|+|\ln\hat{\lambda}_i|\frac{2(N-1)(N+1))}{1+2(N-1)(N+1)}\right)\right)
\end{aligned} \tag{6.28}$$

where ε is defined as

$$\varepsilon=\max\left[\sum_i\frac{|\delta y_{ij}|}{N},\sum_j\frac{|\delta y_{ij}|}{N}\right].$$

6.2.3 MEMnets – Training Algorithm and Design

Unlike approaches based on Markovian dynamics (e.g., VAMPnets [98]), which use one-step transitions as inputs, MEMnets process time sequences of MD conformations by employing a series of parallel encoder neural networks at various time points (Figure 6.1). These networks describe transitions between pairs of time points, $(t-n_k\delta t, t)$, …, $(t-n_1\delta t, t)$, and $(t, t+\delta t)$. Consequently, the corresponding time lags are $[n_1\delta t, \dots, n_k\delta t]$ and δt.

In MEMnets, each MD conformation is represented using pre-selected geometric features χ (e.g., pairwise distances). The parallel encoders then act as nonlinear projection functions that map the high-dimensional input feature space χ to a low-dimensional coordinate space $\boldsymbol{x}$. From the resulting time sequences in the coordinate space, MEMnets compute the objective function $\mathcal{L}_m$ (Eq. (6.17)). By minimizing $\mathcal{L}_m$, the parameters of the projection function (i.e., the neural network weights) are optimized, and the optimal CVs can then be obtained (Eq. (6.14)). In our implementation, we set $\tau = n_k\delta t$.

The evaluation of $\mathcal{L}_m$ requires estimating $\ln \lambda_i^{(\boldsymbol{x})}(\delta t)$ and $\ln \hat{\lambda}_i^{(\boldsymbol{x})}$ from the encoder outputs $\boldsymbol{x}$. Given the projected MD trajectory $\{\boldsymbol{x}_l\}_{l=1}^{L}$, where $\boldsymbol{x} = [x_1, \dots, x_N]^\top$, batches of time sequences $\mathbf{B}$ can be constructed by concatenating temporal windows $\{\mathbf{s}_t\}_{t=1+n_k\delta t}^{L-\delta t}$, where $\mathbf{s}_t = [\boldsymbol{x}_{t-n_k\delta t}, \boldsymbol{x}_{t-n_{k-1}\delta t}, \dots, \boldsymbol{x}_{t-n_1\delta t}, \boldsymbol{x}_t, \boldsymbol{x}_{t+\delta t}]^\top$. As a result, $\mathcal{B}$ forms a three-dimensional tensor, whose first, second, and third dimensions correspond to the batch size $(b = L-1-(n_k+1)\delta t)$, the length of each sequence $(l_s = k+2)$, and the number of coordinates (N), respectively. The tensor $\mathcal{B}$ can then be split along its second dimension to yield a sequence (l_s) of matrices, denoted as $\{\mathbf{B}_{t-n_k\delta t}, \mathbf{B}_{t-n_{k-1}\delta t}, \dots, \mathbf{B}_t, \mathbf{B}_{t+\delta t}\}$, where each $\mathbf{B}_t$ is a two-dimensional matrix of size (b, N).

From these matrices, the Koopman matrices $\mathbf{T}(t) = \mathbf{C}^{-1}(0)\mathbf{C}(t)$ can be estimated from the instantaneous and time-lagged covariance matrices as follows:

$$\begin{aligned}
\mathbf{C}(0) &= \frac{1}{b-1}\bar{\mathbf{B}}_t^\top \bar{\mathbf{B}}_t, \\
\mathbf{C}(\delta t) &= \frac{1}{b-1}\bar{\mathbf{B}}_t^\top \bar{\mathbf{B}}_{t+\delta t}, \\
\mathbf{C}(n_i\delta t) &= \frac{1}{b-1}\bar{\mathbf{B}}_t^\top \bar{\mathbf{B}}_{t-n_i\delta t}, \qquad i = 1, \dots, k.
\end{aligned} \tag{6.29}$$

Here, the overbar denotes mean-removed data matrices, where the column-wise means have been subtracted from each $\mathbf{B}_t$. To satisfy the detailed balance condition of the dynamics, all time-lagged correlation matrices $\mathbf{C}(t)$ are symmetrized by averaging each matrix with its transpose, i.e., $\mathbf{C}(t) \leftarrow \frac{1}{2}\big(\mathbf{C}(t) + \mathbf{C}^{\top}(t)\big)$. For implementation convenience, we offset the time index by a constant $n_k\delta t$, such that the first time point in $\mathbf{s}_t$ corresponds to t.

From the estimated Koopman matrices $[\mathbf{T}^{(\boldsymbol{x})}(\delta t), \mathbf{T}^{(\boldsymbol{x})}(n_1\delta t), \ldots, \mathbf{T}^{(\boldsymbol{x})}(n_k\delta t)]$, we obtain their eigenvalue matrices $[\boldsymbol{\lambda}^{(\boldsymbol{x})}(\delta t), \boldsymbol{\lambda}^{(\boldsymbol{x})}(n_1\delta t), \ldots, \boldsymbol{\lambda}^{(\boldsymbol{x})}(n_k\delta t)]$. These eigenvalues, $[\boldsymbol{\lambda}^{(\boldsymbol{x})}(n_1\delta t), \ldots, \boldsymbol{\lambda}^{(\boldsymbol{x})}(n_k\delta t)]$, are subsequently used to estimate $\ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}$. $\ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})} = \mathrm{diag}[\ln\hat{\lambda}_1^{(\boldsymbol{x})}, \ldots, \ln\hat{\lambda}_N^{(\boldsymbol{x})}]$ represents the diagonal eigenvalue matrix of $\ln\hat{\mathbf{T}}^{(\boldsymbol{x})}$, which appears in the IGME solution (Eq. (6.9)). In detail, we demonstrate that $\ln\hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}$ has a clear physical interpretation as the inverse of the characteristic timescales of $\boldsymbol{\xi}$ in Markovian dynamics.

This interpretation follows directly from the IGME solution $\mathbf{T}(t) = \mathbf{A}\hat{\mathbf{T}}^t$, where both $\mathbf{A}$ and $\hat{\mathbf{T}}$ are constant matrices. In the MEMnets framework, the encoder neural network outputs $\boldsymbol{x}$ are transformed into a set of orthonormal coordinates $\boldsymbol{\xi}$ through a linear transformation at a specific lag time $\tau \geq \tau_K$. At this lag time, $\ln\mathbf{T}^{(\boldsymbol{\xi})}(\tau) = \ln\boldsymbol{\lambda}^{(\boldsymbol{x})}(\tau)$ is diagonal, and $\ln\mathbf{T}^{(\boldsymbol{\xi})}(t) = \ln\boldsymbol{\lambda}^{(\boldsymbol{x})}(t)$ remains diagonal within this range. Consequently, both constant matrices $\mathbf{A}^{(\boldsymbol{\xi})}$ and $\hat{\mathbf{T}}^{(\boldsymbol{\xi})}$ are also diagonal, where $\hat{\mathbf{T}}^{(\boldsymbol{\xi})} = \hat{\boldsymbol{\lambda}}^{(\boldsymbol{\xi})} = \hat{\boldsymbol{\lambda}}^{(\boldsymbol{x})}$ (Eq. (6.26)).

This leads to the following relationship:

$$\ln\lambda_i^{(\boldsymbol{x})}(t) = t\,\ln\hat{\lambda}_i^{(\boldsymbol{x})} + \ln\lambda_i^{(\boldsymbol{x})}(\mathbf{A}) \tag{6.30}$$

where $\lambda_i^{(\boldsymbol{x})}(t)$, $\hat{\lambda}_i^{(\boldsymbol{x})}$, and $\lambda_i^{(\boldsymbol{x})}(\mathbf{A})$ denote the ith eigenvalues of $\mathbf{T}^{(\boldsymbol{x})}(t)$, $\hat{\mathbf{T}}^{(\boldsymbol{x})}$, and $\mathbf{A}^{(\boldsymbol{x})}$, respectively.

Dividing both sides of Eq. (6.30) by t gives:

$$\frac{\ln\lambda_i^{(\boldsymbol{x})}(t)}{t} = \ln\hat{\lambda}_i^{(\boldsymbol{x})} + \frac{\ln\lambda_i^{(\boldsymbol{x})}(\mathbf{A})}{t} \tag{6.31}$$

Taking the limit as t approaches infinity, yielding:

$$\ln\hat{\lambda}_i^{(\boldsymbol{x})} = \lim_{t\to\inf}\frac{\ln\lambda_i^{(\boldsymbol{x})}(t)}{t}, \qquad \hat{S}_i = S_i(t \to \inf) \tag{6.32}$$

where $S_i(t) = -t/\ln\lambda_i(t)$ is the implied timescale at lag time t, and $\hat{S}_i = |1/\ln\hat{\lambda}_i^{(\boldsymbol{\xi})}|$ is defined as the *MEMnets timescale* of CV i. Thus, $\ln\hat{\boldsymbol{\lambda}}$ represents the long-time (Markovian) limit of the logarithmic eigenvalue spectra, directly corresponding to the asymptotic relaxation timescales of the underlying dynamics.

The values of $\ln\hat{\lambda}_i^{(\boldsymbol{x})}$ are obtained via a least-squares fitting procedure based on Eq. (6.30) during MEMnets training.

Define $\boldsymbol{a} = [n_1\delta t, \ldots, n_k\delta t]^\top$ and $\boldsymbol{b}_i = [\ln\lambda_i^{(\boldsymbol{x})}(n_1\delta t), \ldots, \ln\lambda_i^{(\boldsymbol{x})}(n_k\delta t)]^\top$, we have

$$\ln\hat{\lambda}_i^{(\boldsymbol{x})} \approx (\boldsymbol{a}^\top\boldsymbol{a})^{-1}(\boldsymbol{a}^\top\boldsymbol{b}_i) \tag{6.33}$$

In practical MEMnets implementations, the fitting is performed using a range of lag times centered around τ, for example $[0.7\tau, 0.8\tau, 0.9\tau, \tau]$ for the alanine dipeptide and WW-domain systems.

At this stage, we have established all quantities required to compute the MEMnets objective function (Eq. (6.17)). In the MEMnets objective, γ varies across different systems but remains smaller than 1 (see Section 6.2.2). To ensure stable and consistent optimization, we adopt a dynamic scheme in which γ undergoes exponential decay from 2 to 0.5 during the first ten training epochs (pre-training) and is then kept constant thereafter. We initialize γ with a relatively large value of 2—exceeding its theoretical upper bound of 1— because placing greater emphasis on the $\left|\ln\hat{\lambda}_i\right|$ term in Eq. (6.31) during early training facilitates the identification of the slowest dynamic modes.

In detail, we employ a general dynamic scheme to select γ as follows:

$$\gamma = \begin{cases} 0.5 + 1.5e^{-\text{decay\_rate}\times\text{train\_step}}, & \text{if } \gamma - 0.5 > \text{cut\_off} \\ 0.5, & \text{otherwise} \end{cases} \tag{6.34}$$

where the cutoff value is set to cut_off = 0.015, and the decay rate is defined as

$$\text{decay\_rate} = \frac{0.5}{N_{\text{iter}}} \tag{6.35}$$

with N_{iter} denoting the number of training steps per epoch. This dynamic scheme ensures that γ decays exponentially with the training step during the first ten epochs of training and subsequently stabilizes at a constant value of 0.5. Although this schedule is not a strict requirement of the MEMnets framework, it provides a consistent and robust implementation in this work and can be readily adjusted for other systems or training protocols.

6.3 Results

6.3.1 Minimizing Memory Kernel for Alanine Dipeptide

Using alanine dipeptide as an illustrative example (see Section 4.3.2 for dataset details), we demonstrate the optimization of the time-integrated memory kernel in MEMnets (Figure 6.2a). For MEMnets training, 30 Cartesian coordinates derived from the ten heavy atoms of alanine dipeptide were used as input features, and the optimization was performed over three CVs. The lag interval was set to $\delta t = 1$ ps, and n_1, n_2, n_3, and n_4 ($k = 4$) were chosen as 7, 8, 9, and 10, respectively. A previous study reported a memory-kernel decay time of $\tau_K = 1.5$ ps for alanine dipeptide using a four-state GME model [18]; thus, the lag times used here are likely longer than τ_K. The model was trained for 250 epochs, using a learning rate of 1×10^{-3} without employing a learning rate scheduler, and the batch size was set to 10,000. The encoder network consisted of fully connected layers with widths [30, 30, 30, 30, 10, 3], where three CVs are identified. To quantitatively evaluate the MEMnets results, the entire dataset was used for both training and validation.

As shown in the left panel of Figure 6.2b, after 200 training epochs, the loss $\mathcal{L}_m$ for all three identified CVs converges to constant values, indicating that MEMnets successfully finds an optimal projection onto three CVs with minimized memory effects. We then examined the MEMnets timescales $\hat{S}_i = |\ln \hat{\lambda}_i|^{-1}$ obtained from the converged model. In the right panel of Figure 6.2b, the validation curves of

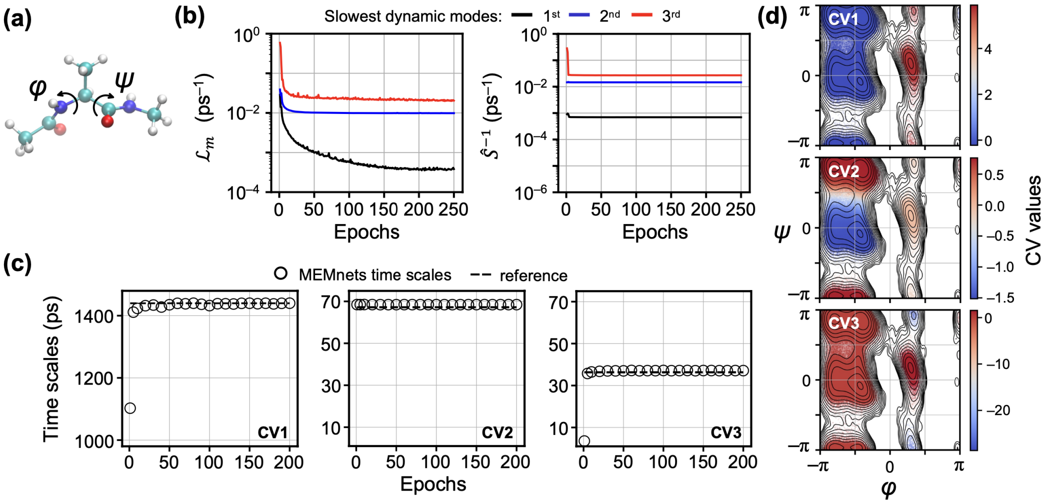


Figure 6.2: Demonstration of MEMnets on the alanine dipeptide dataset. *(a) Representative structure of alanine dipeptide, where ϕ and ψ denote the two backbone torsion angles. (b) Validation curves of the MEMnets objective function (left) and the inverse MEMnets timescales (right). (c) Comparison between the MEMnets-derived timescales and the reference values computed from a 1000-state MSM with a lag time of 10 ps based on the two-dimensional dihedral space. (d) Projection of MD conformations onto the three CVs identified by MEMnets. In each panel, MD conformations are overlaid on the 2D (ϕ, ψ) dihedral map, with colors indicating the corresponding CV values.*

$\hat{S}_i^{-1}$ decrease rapidly and stabilize within the first ten epochs (pre-training phase), suggesting that MEMnets efficiently anchors the three slowest dynamic modes of the alanine dipeptide system. To further validate the model, we compared the resulting MEMnets timescales with reference values from a 1000-state MSM, showing excellent agreement (Figure 6.2c).

We then visualized the three identified CVs in the two-dimensional dihedral space defined by the torsion angles ψ and ϕ (Figure 6.2d). The results demonstrate that the MEMnets-derived CVs accurately describe the conformational transitions between metastable states (i.e., free-energy basins) of alanine dipeptide. The color transition from blue (negative values) to red (positive values) highlights the conformational changes associated with the given CV.

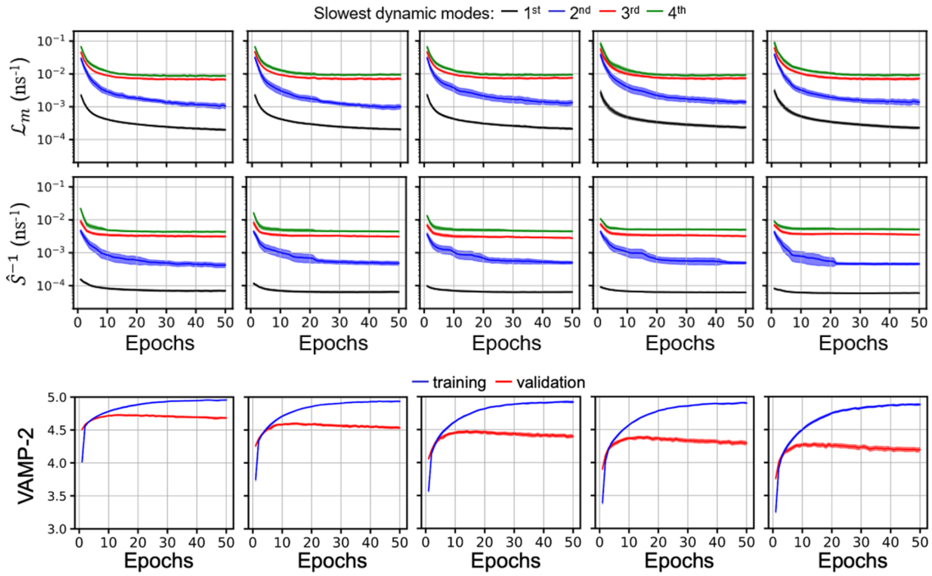


Figure 6.3: Validation curves of MEMnets and VAMPnets on the WW-domain dataset. *From left to right, each column corresponds to models trained with encoder (or training) lag times of [14, 16, 18, 20], [28, 32, 36, 40], [42, 48, 54, 60], [56, 64, 72, 80], and [70, 80, 90, 100] ns for MEMnets, and 20, 40, 60, 80, and 100 ns for VAMPnets, respectively. For MEMnets (top two rows), each panel shows the validation curves (mean ± 95% confidence interval) of the objective function (top row) and the inverse MEMnets timescales (bottom row). For VAMPnets (bottom row), each panel displays the corresponding training and validation curves (mean ± 95% confidence interval). In all cases, the sample size is 45, representing the top 90% of models selected from 50 independent training runs, each using a 90%/10% random train/validation split. Model selection was based on the largest MEMnets timescale and the highest VAMP-2 score for MEMnets and VAMPnets, respectively. The models at 10 epochs of training were selected to report the VAMPnets results.*

6.3.2 The Folding Dynamics of FIP35 WW Domain

The FIP35 WW domain is a three-stranded β-sheet protein (Figure 6.4), consisting of 35 residues, which serves as a widely used benchmark system for studying protein folding dynamics. To investigate it, we obtained the simulation dataset from D. E. Shaw Research [85]. The original dataset comprises two trajectories with lengths of 486 μs and 651 μs, respectively, recorded at 200 ps intervals between adjacent MD frames. Both trajectories were truncated to 480 μs and subsampled at a 1 ns interval in our analysis. The input features consisted of 595 pairwise distances computed from the x, y, and z coordinates of 35 Cα atoms.

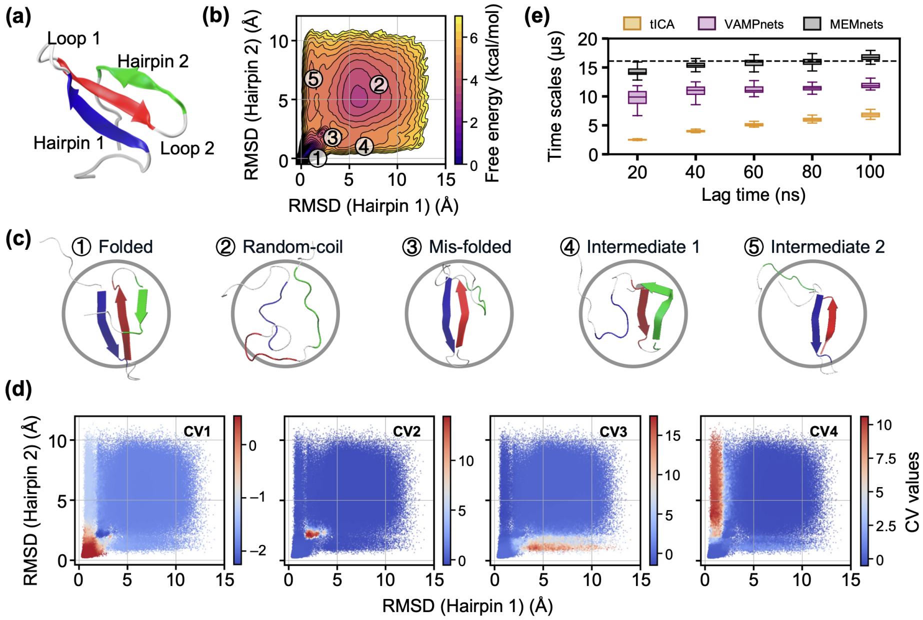


Figure 6.4: Kinetic analysis of MEMnets on the FIP35 WW-domain dataset. *(a) Representative structure of the FIP35 WW-domain. (b) Free energy landscape projected onto the RMSD of Hairpin 1 and Hairpin 2. (c) Representative structures corresponding to the five energy basins shown in panel b. (d) Projection of MD conformations onto the four CVs identified by MEMnets. In each panel, MD conformations are overlaid on the 2D space defined by the RMSD of Hairpin 1 and Hairpin 2, with colors indicating the values projected onto a specific CV. (e) Comparison of the slowest timescales among tICA, VAMPnets, and MEMnets. Each box in the box plot represents the values of (from bottom to top):* $(Q_1 - 1.5\,\mathrm{IQR})$, Q_1, *median*, Q_3, *and* $(Q_3 + 1.5\,\mathrm{IQR})$, *where* Q_1, Q_3, *and* IQR *denote the first quartile, third quartile, and interquartile range, respectively. For tICA and VAMPnets, the x-axis corresponds to the lag time, whereas for MEMnets it indicates the longest lag time* $(\tau = n_k \delta t)$ *among all encoder networks. The dashed line marks the reference value obtained from a 1500-state MSM with a lag time of 100 ns constructed using three tICs. For VAMPnets and MEMnets, the sample size is 45, representing the top 90% of models selected from 50 independent training runs, each using a 90%/10% random train/validation split. Model selection was based on the largest MEMnets timescale and VAMP-2 score in the validation curves for MEMnets and VAMPnets, respectively. For tICA, the sample size is 50, corresponding to 50 bootstrap replicates generated by resampling the MD trajectories.*

Five independent groups of MEMnets models were trained with $n_k \delta t$ set to 20, 40, 60, 80, and 100 ns, respectively, where $k = 4$. For each group, $\delta t = 1$ ns was used, and n_1, n_2, and n_3 were assigned as $0.7n_4$, $0.8n_4$, and $0.9n_4$, respectively. To provide a direct comparison, we also trained five groups of VAMPnets using lag times of 20, 40, 60, 80, and 100 ns. Each training group employed a 90%/10% random train/validation split and was repeated 50 times. For both MEMnets and VAMPnets, the top 90% of models—ranked according to the largest MEMnets timescale and the highest VAMP-2 score, respectively—were used to report the results. The models were trained for 50 epochs, using a learning rate of 1×10^{-3} without employing a learning rate scheduler, and the batch size was set to 10,000. The encoder network consisted of fully connected layers with widths [595, 300, 100, 30, 10, 4], where four CVs are identified. As shown in Figure 6.3, the MEMnets models converged after 50 epochs, yielding four CVs that minimize the time-integrated memory kernel, while for VAMPnets, we select models at 10 epochs to prevent overfitting.

The CVs identified by MEMnets accurately capture the folding mechanism of the WW domain, revealing two distinct folding pathways. To visualize the relationship between MEMnets CVs and folding dynamics, the four CVs were projected onto a two-dimensional space defined by the root-mean-square deviations (RMSDs) of hairpins 1 and 2 relative to the crystal structure (Figure 6.4d). In these projections, color transitions from blue (negative values) to red (positive values) illustrate conformational changes associated with each CV. The corresponding free-energy landscape is shown in Figure 6.4b, with representative structures for each free-energy basin provided in Figure 6.4c. The slowest CV captures the global folding transition between the unfolded and folded states (blue $\rightarrow$ red regions in the left panel of Figure 6.4d), representing the dominant slow dynamics of the system. The second slowest CV characterizes transitions into and out of an off-pathway misfolded state (second panel of Figure 6.4d), previously identified in state-based MSM studies [17]. The third and fourth CVs reveal two parallel folding pathways from the random coil to the native structure: one proceeds via intermediate 1, in which hairpin 2 folds first, and the other through intermediate 2, where hairpin 1 folds first. The identification of these parallel pathways is consistent with earlier MSM studies [106], providing a continuous and high-resolution view of the folding process.

Beyond qualitative agreement, MEMnets also quantitatively reproduces the overall folding kinetics. From the first MEMnets CV (CV1), the folding time estimated via the mean first-passage time (MFPT) from the unfolded to folded states is 14.0 ± 5.8 μs. Here, the unfolded state is defined as $\text{CV1} \geq 0.888$ (median value), and the folded state as $\text{CV1} < 0.888$; the uncertainty of 5.8 μs was determined through bootstrapping. This folding time agrees closely with the previous estimation from D. E. Shaw Research based on the same trajectories (10 ± 3 μs) [131] and with experimental measurements (~14 μs) [91].

According to IGME theory, the encoder lag times $[n_1\delta t, \ldots, n_k\delta t]$ must exceed the memory-kernel decay time τ_K. To determine the appropriate lag time for the WW domain, we systematically varied the longest encoder lag time $\tau = n_k\delta t$ from 20 ns to 100 ns. As shown by the black bars in Figure 6.4e, MEMnets converge for $\tau \geq 40$ ns in predicting the slowest implied timescale, consistent with the reference value from a validated 1,500-state MSM (dashed line in Figure 6.4e). These results indicate that the memory kernel decays around $\tau = 40$ ns (with encoder lag times [28, 32, 36, 40 ns]), in agreement with a previous estimate of $\tau_K = 25$ ns from a four-state GME model [18]. Therefore, $\tau = 40$ ns was selected for the final MEMnets model.

Finally, we compared the performance of MEMnets with Markovian-based methods, including tICA and VAMPnets (Figure 6.4e). Across all τ values, MEMnets consistently outperforms both methods in reproducing the slowest timescale, showing closer agreement with the MSM reference. These results highlight that MEMnets, grounded in the non-Markovian Generalized Master Equation framework and explicitly incorporating memory effects, provides a more accurate and physically consistent identification of slow CVs and their corresponding timescales than Markovian approaches such as tICA and VAMPnets.

6.3.3 The Clamp Motion of a Bacterial RNA Polymerase

We further demonstrate the efficacy of MEMnets by applying it to investigate the gate-opening dynamics of *Thermus aquaticus* (Taq) RNA polymerase (RNAP) [144]. The loading gate of Taq RNAP comprises two major structural elements—the clamp and the β-lobe domains. The coordinated movements of these domains are essential for initiating gene transcription, during which the loading gate must open to accommodate the promoter double-stranded DNA. This process involves complex and large-scale conformational rearrangements of RNAP, including extensive opening and rotational motions of both the clamp and the β-lobe (Figure 6.5a). To study this system, we utilized the MD simulation dataset from our previous study [144]. This dataset consists of 306 trajectories, each 200 ns in length, with frames saved every

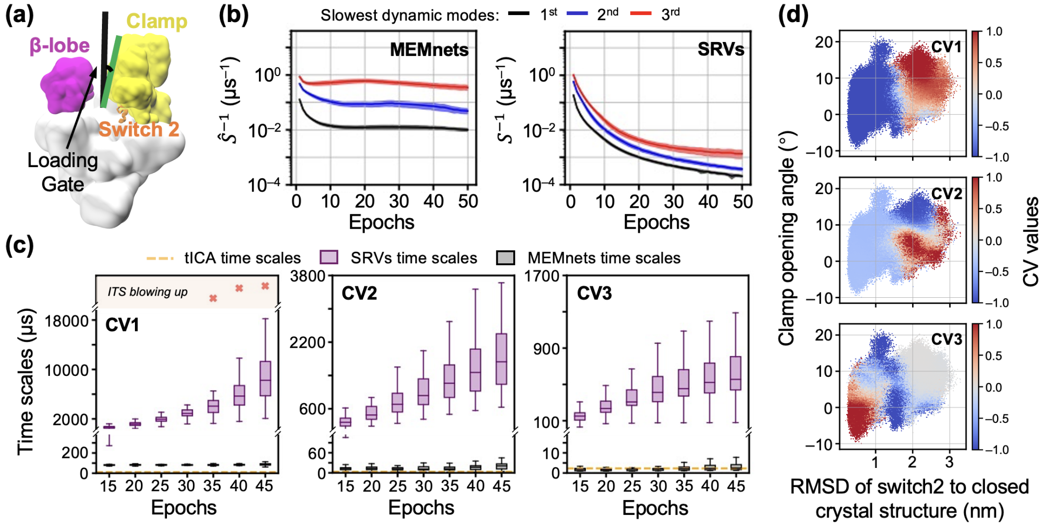


Figure 6.5: Kinetic analysis of MEMnets and SRVs on the RNA polymerase (RNAP) clamp motion dataset. *(a) Representative structure of RNAP. (b) Validation curves (mean ± 95% confidence interval) of the inverse timescales of the identified CVs trained by MEMnets (left) and SRVs (right). (c) Timescales of the identified CVs from MEMnets and SRV models across different training epochs. Each box plot represents the distribution of values across models, with boxes indicating (from bottom to top):* $(Q_1 - 1.5\,\mathrm{IQR})$, Q_1, *median,* Q_3, *and* $(Q_3 + 1.5\,\mathrm{IQR})$, *where* Q_1, Q_3, *and* IQR *denote the first quartile, third quartile, and interquartile range, respectively. (d) Projection of MD conformations onto the three CVs identified by MEMnets. In each panel, MD conformations are overlaid on the 2D space defined by the RMSD of Switch 2 (relative to the closed crystal structure) and the clamp opening angle, with colors indicating the CV values. In panels (b) and (c), the sample size is 45, representing the top 90% of models selected from 50 independent training runs, each using a 90%/10% random train/validation split. Model selection was based on the smallest MEMnets timescale and VAMP-2 score in the validation curves for MEMnets and SRVs, respectively.*

100 ps. A total of 1,770 pairwise distances were computed as input features, involving the Cα atoms of the clamp, β-lobe, β-protrusion, Switch regions, and active site.

For MEMnets training, the lag interval was set to $\delta t = 1$ ns, and the encoder lag times $n_1\delta t, n_2\delta t, n_3\delta t, n_4\delta t, n_5\delta t$, and $n_6\delta t$ $(k = 6)$ were chosen as 25, 30, 35, 40, 45, and 50 ns, respectively, as the IGME model previously reported a memory-kernel decay time of approximately 25–30 ns for this system [18]. To enable comparison, we also trained State-free Reversible VAMPnets (SRVs) using a training lag time of 50 ns. Both MEMnets and SRVs were trained with a 90%/10% random train/validation split

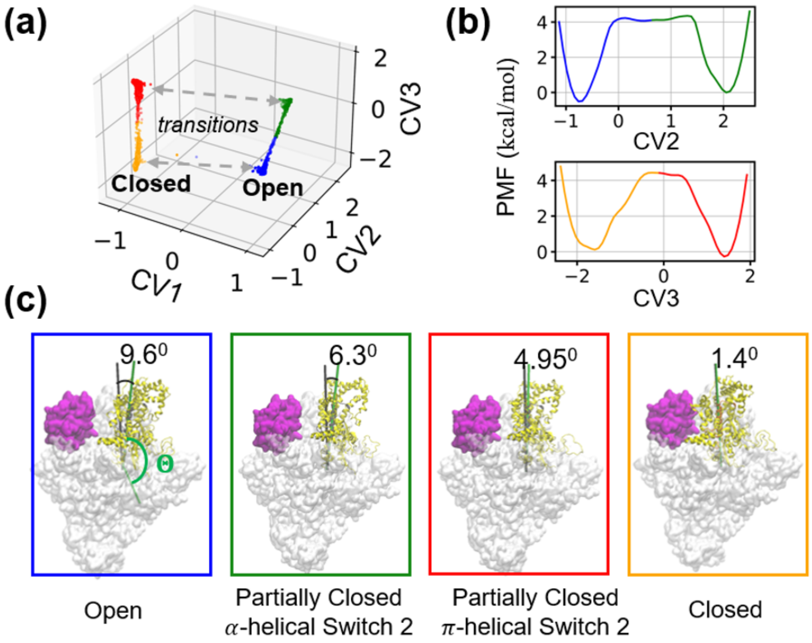


Figure 6.6: State-kinetic analysis of MEMnets on the RNAP system. (*a*) *Projection of MD conformations onto the three CVs identified by MEMnets.* (*b*) *Potential of mean force* (*PMF*) *projected along CV2 and CV3, revealing four distinct energy basins corresponding to four metastable states.* (*c*) *Representative conformations of the four metastable states identified in panel b.*

and repeated 50 times. For analysis, the top 90% of models, ranked by the smallest MEMnets timescale and the VAMP-2 score for MEMnets and SRVs, respectively, were used to generate the results. The models were trained for 50 epochs, using a learning rate of 1×10^{-3} without employing a learning rate scheduler, and the batch size was set to 20,000. The encoder network consisted of fully connected layers with widths [1770, 900, 300, 100, 30, 10, 3], where three CVs are identified.

To interpret these CVs, we projected their values onto two physically meaningful coordinates (Figure 6.5d): the RMSD of the Switch 2 motif, i.e., a short element beneath the clamp domain (shown in orange in Figure 6.5a), and the clamp opening angle. Color transitions from blue (negative values) to red (positive values) illustrate the conformational changes captured by each CV. Notably, the slowest CV (CV1) shows a strong correlation with the RMSD of Switch 2 (top panel of Figure 6.5d), suggesting that the dominant slow mode governing gate opening is associated with conformational rearrangements of this motif. This finding agrees with our earlier

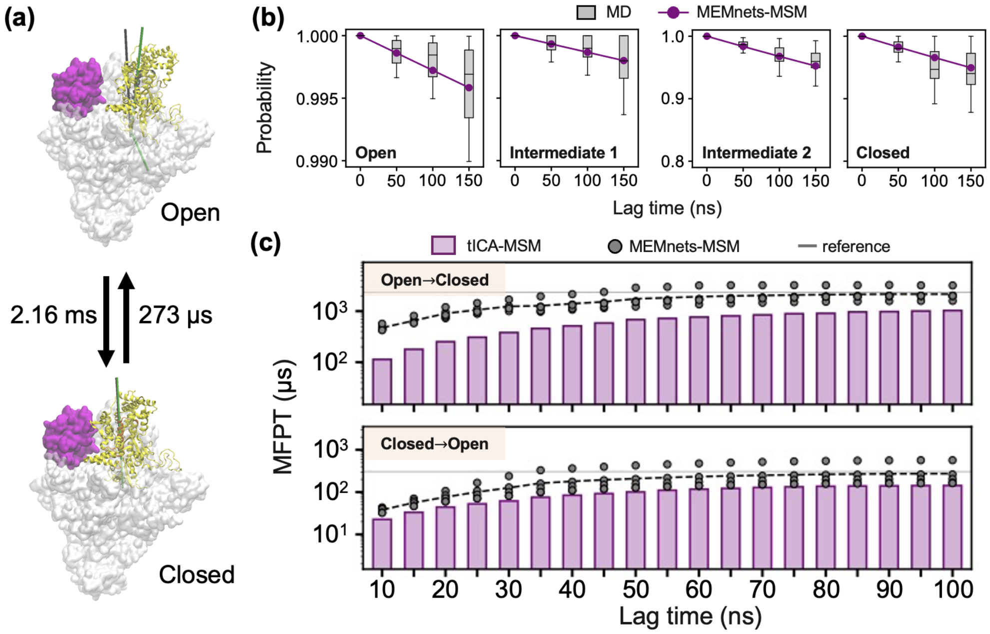


Figure 6.7: MSM analysis of RNAP clamp motion kinetics based on the CVs identified by MEMnets. *(a) Clamp-open and clamp-closed conformations identified from the MSM, with mean first passage times (MFPTs) computed between the two states. (b) Residence probability test of the MSM constructed using the MEMnets-derived CVs. Probabilities obtained from MD simulations were estimated via bootstrapping of the trajectories (20 replicates). Each box plot shows the distribution across the 20 bootstrap samples, with boxes indicating (from bottom to top):* $(Q_1 - 1.5\,\mathrm{IQR})$, Q_1, *median,* Q_3, *and* $(Q_3 + 1.5\,\mathrm{IQR})$, *where* Q_1, Q_3, *and* IQR *represent the first quartile, third quartile, and interquartile range, respectively. (c) MFPTs between the open and closed states at different lag times. The gray line represents the reference value calculated from a GME model reported in our previous study [144]. Overlaid data points correspond to the top five MEMnets models, selected based on the highest Pearson correlation of CVs across independent training runs. The dashed black line indicates the mean MFPTs obtained from these top models.*

four-state GME analysis [144], which identified the refolding of Switch 2 from an α-helix (closed state) to a π-helix (open state) as the rate-limiting step for gate opening. In contrast, CV2 and CV3 capture additional slow components involving both Switch 2 refolding and clamp domain motion (middle and bottom panels of Figure 6.5d). These results indicate a synergistic coupling between the two principal motions underlying RNAP gate opening, i.e., Switch 2 refolding and clamp opening. Together, the three MEMnets-derived CVs elucidate the molecular mechanism of

RNAP loading-gate dynamics, revealing that the Switch 2 region functions as a hinge beneath the clamp, mediating and regulating the gate-opening motion.

We further evaluated the performance of MEMnets as a dimensionality-reduction framework for Markov state modeling. Specifically, all MD conformations were projected onto the three MEMnets-identified CVs and subsequently clustered into four metastable states using k-means (Figure 6.6). These states correspond to the clamp-open and clamp-closed conformations, along with two intermediate states. To assess the validity of this four-state model, we performed a Chapman–Kolmogorov (CK) test. As shown in Figure 6.7b, the MEMnets-based MSM with a lag time of 50 ns predicts transition probabilities that closely match those obtained directly from MD simulations, demonstrating that MEMnets provides a reduced kinetic space with minimal memory, suitable for constructing effective Markovian state models. We also computed MFPTs between the open and closed states across different lag times and compared them with those obtained from a tICA-based four-state MSM (tICA–MSM). As shown in Figure 6.7c, MFPTs derived from the MEMnets–MSM are approximately twice as long as those from the tICA–MSM, in close agreement with the GME-based reference value [144]. This finding further highlights that MEMnets optimizes the projection by explicitly minimizing memory effects, yielding a more accurate low-dimensional kinetic space that approaches Markovianity.

6.3.4 MEMnets Outperform SRVs with Robust Convergence in Training

When applied to complex biological systems with limited sampling, one common challenge for SRVs and VAMPnets is achieving stable convergence during training. As a result, additional stopping criteria or regularization techniques are often required to prevent overfitting [62, 93]. Using RNAP as an example, the validation curve of SRVs, represented by the inverse timescales of the three identified CVs, fails to converge (right panel of Figure 6.5b), making it difficult to establish a reliable early-stopping criterion. In sharp contrast, the validation curve of MEMnets exhibits smooth and robust convergence (left panel of Figure 6.5b), demonstrating the stability of its training process.

To further evaluate model performance, we compared the timescales of the three

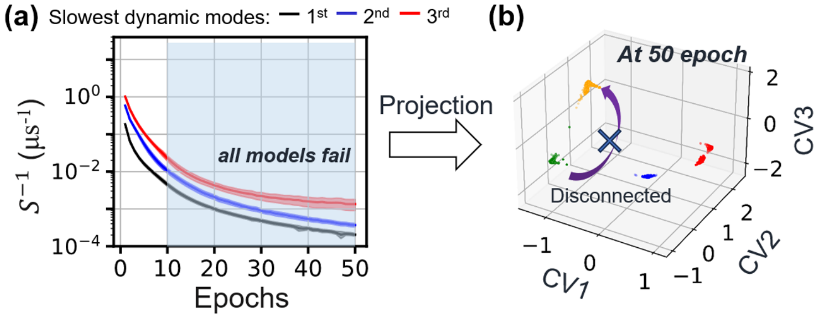


Figure 6.8: State-kinetic analysis of SRVs on the RNAP system. *(a) Validation curves (mean ± 95% confidence interval) of the inverse timescales during SRV training, manually computed from the eigenvalues of each dynamical mode contributing to the VAMP-2 score. The sample size is 45, representing the top 90% of models selected from 50 independent training runs, each using a 90%/10% random train/validation split. Model selection was based on the smallest VAMP-2 score in the validation curves. All SRV models trained between 10 and 50 epochs (indicated by the blue shaded region) failed to converge, with the implied timescales diverging to unphysical values, and no meaningful state-kinetic results could be obtained. (b) Projection of MD conformations onto the three CVs identified by SRVs at 50 epochs of training. Four distinct, disconnected clusters are observed, indicating a lack of transitions between states and confirming the instability of SRV-derived models under these training conditions.*

CVs across training epochs for both MEMnets and SRVs (Figure 6.5c). MEMnets produced consistent and physically meaningful timescales across all epochs, whereas SRVs exhibited a sharp linear increase in the estimated timescales, leading to unphysically large, infinite, or even negative values after approximately 35 epochs. These results indicate that SRVs are prone to overfitting, often capturing disconnected or nonphysical dynamics rather than the genuine slow modes of the system. Indeed, when the MD trajectories were projected onto the SRV-derived CVs trained between 10 and 50 epochs, the resulting reduced kinetic models were clearly disconnected (Figure 6.8).

Notably, the RNAP dataset analyzed in this study was obtained from our previous work [144] and consists of 306 MD trajectories, each 200 ns in length. These trajectories were initiated from conformations distributed along the initial pathways connecting the open- and closed-gate states of RNAP. In that previous study, we successfully constructed a four-state quasi-Markov state model (qMSM) to characterize the gate-opening process. Here, we analyzed transition events within the same

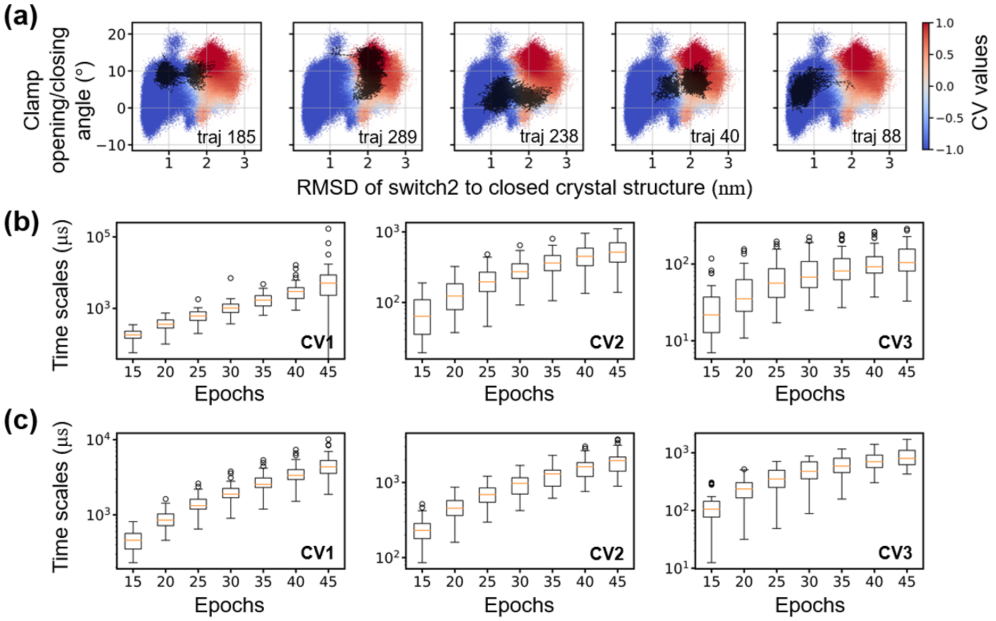


Figure 6.9: Kinetic analysis of SRVs on the RNAP clamp motion dataset. (*a*) *Five representative trajectories* (*out of 18 identified from the 306 200-ns trajectories*) *whose RMSD values of the Switch 2 region cross from 1.0 nm to 2.0 nm. These trajectories are projected onto a two-dimensional space defined by the RMSD of Switch 2 relative to the closed crystal structure and the clamp opening angle.* (*b–c*) *Timescales of the CVs identified by SRV models trained at lag times of 10 ns* (*b*) *and 100 ns* (*c*), *respectively, shown across different training epochs. Each box plot represents the distribution of timescales across models, with boxes denoting* (*from bottom to top*): $(Q_1 - 1.5\mathrm{IQR})$, Q_1, *median*, Q_3, *and* $(Q_3 + 1.5\mathrm{IQR})$, *where* Q_1, Q_3, *and* IQR *represent the first quartile, third quartile, and interquartile range, respectively. The sample size is 45, corresponding to the top 90% of models selected from 50 independent training runs, each using a 90%/10% random train/validation split. Model selection was based on the smallest VAMP-2 score in the validation curves.*

dataset. Consistent with our earlier qMSM findings, MEMnets clearly identified the rate-limiting step (CV1) of RNAP gate opening as the conformational rearrangement of the Switch 2 region, corresponding to the transition from the red to blue regions in the top panel of Figure 6.5d or Figure 6.9a. Based on this observation, we used the RMSD of the Switch 2 region as a criterion to identify individual MD trajectories that directly sample gate-opening transitions. In total, 18 trajectories were found to exhibit transitions in which the Switch 2 RMSD crossed from 1 nm to 2 nm. Figure 6.9a illustrates five representative examples, each capturing the transition from the blue to red region on the CV1 map, consistent with the folding event of the Switch 2 motif.

Although the RNAP system remains less extensively sampled compared with alanine dipeptide and the WW domain, these trajectories provide sufficient coverage of the slowest transition events underlying the gate-opening process.

We also examined whether modifying the training lag times could mitigate the overfitting observed in SRVs. In addition to the 50 ns lag time used in the main analysis, we trained SRV models with lag times of 10 ns (Figure 6.9b) and 100 ns Figure 6.9c), each repeated 50 times. As shown in Figures 6.9b–c, the SRV models continued to exhibit overfitting when early stopping was not applied, yielding unphysically large timescales relative to those obtained from tICA and MEMnets. While SRVs is an elegant method grounded in the variational principle of conformational dynamics, it can be particularly susceptible to overfitting in complex systems with limited sampling. We attribute this behavior to the nature of its optimization objective: the deep learning–based nonlinear projection seeks to maximize the eigenvalues of the projected dynamics, which can inadvertently drive them toward unity ($\lambda_i \to 1$), resulting in disconnected or nonphysical dynamics. This occurs because the nonlinear mapping can distort the data representation to satisfy the variational objective, especially when the available trajectory sampling is insufficient. Consequently, prior successful SRV applications to complex biomolecular systems have typically employed early stopping or other regularization strategies to prevent overfitting.

Overall, MEMnets demonstrate substantially greater robustness and convergence reliability than SRVs or VAMPnets, primarily because they leverage the full time-series information from MD trajectories rather than single-step transitions. In addition, our MEMnets implementation incorporates the IGME theoretical constraint, using $\ln \hat{\lambda}$ as an upper bound for $\ln \lambda(\delta t)/\delta t$ in the first term within the objective function. This design enhances numerical stability during training and effectively prevents the emergence of disconnected or overfitted models.

6.3.5 Additional Validations and Results

Selection of the number of CVs. In MEMnets, the number of CVs, which corresponds to the number of nodes in the final layer of the neural network, must be specified prior to training. A common strategy is to select this number based on the separation of timescales, that is, to retain the slowest dynamic modes that are

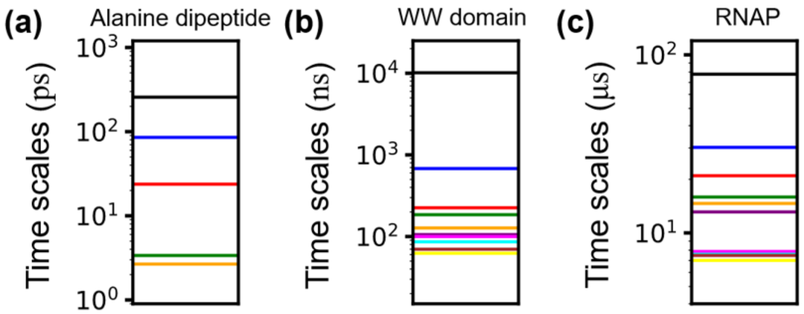


Figure 6.10: Separation of timescales tests to determine number of CVs. *MEMnets timescales* $\left((\ln \hat{\lambda}_i)^{-1}\right)$ *obtained from the IGME solutions of the continuous input features for* (*a*) *alanine dipeptide,* (*b*) *WW domain, and* (*c*) *RNA polymerase* (*RNAP*)*.*

clearly separated from the faster ones. In this work, we determined the number of CVs according to the spectral gaps in the MEMnets timescales, $(\ln \hat{\lambda}_i)^{-1}$, estimated directly using the input coordinates.

Specifically, we fitted the time-lagged correlation matrices of the input features at multiple lag times to Eq. (6.30) to estimate $(\ln \hat{\lambda}_i)^{-1}$. The lag times used for this fitting were [7, 8, 9, 10] ps for alanine dipeptide, [28, 32, 36, 40] ns for the WW domain, and [50, 60, 70, 80, 90, 100] ns for RNAP. Physically, these timescales correspond to the asymptotic limit where the lag time approaches infinity. As shown in Figure 6.10, our selected numbers of CVs align with distinct gaps in the MEMnets timescales spectra.

To further evaluate the effect of the number of CVs on MEMnets performance, we used the WW domain as a representative system and trained additional MEMnets models with three and five CVs (50 models each), in parallel with our original four-CV models. All models were trained using identical hyperparameters and training protocols. Among the three-CV MEMnets models, approximately half identified the third CV as capturing the folding of Hairpin 1, while the other half assigned it to the folding of Hairpin 2 (last two panels of Figure 6.11a). This observation indicates that with only three CVs, the model cannot effectively distinguish between the two alternative folding pathways (Hairpin 1–first versus Hairpin 2–first), as the third and fourth slowest timescales are nearly degenerate. When increasing the dimensionality to five CVs, all trained MEMnets models consistently recovered the top four CVs (Figure 6.11b), while the fifth CV corresponded to an additional fast dynamic mode (last panel of Figure 6.11b). These results confirm that selecting the

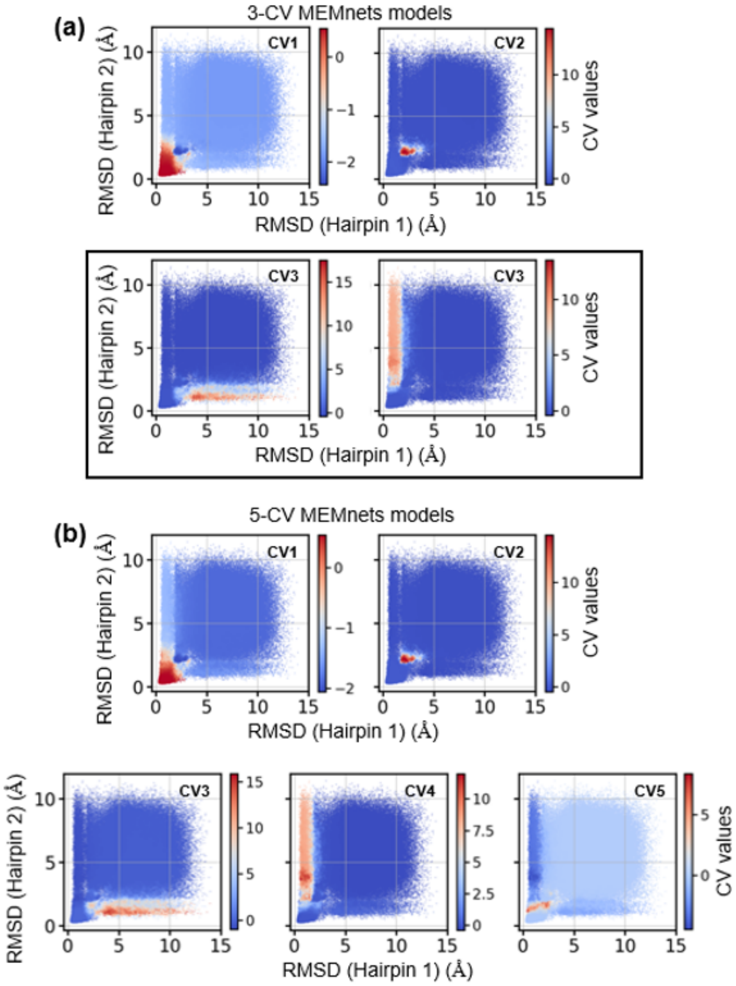


Figure 6.11: MEMnets models with three and five CVs for the FIP35 WW domain. *(a) Projection of MD conformations onto the three CVs identified by the 3-CV MEMnets models. In each panel, MD conformations are overlaid on a two-dimensional space defined by the RMSD of Hairpin 1 versus Hairpin 2, with the color bar indicating the value of each conformation projected onto the corresponding CV. For CV3, the MEMnets models capture a dynamic mode corresponding to the folding of either Hairpin 1 (bottom right panel) or Hairpin 2 (bottom left panel). (b) Same as in (a), but showing projections onto the five CVs identified by the 5-CV MEMnets models.*

number of CVs based on clear spectral gaps in the MEMnets timescales yields a physically interpretable and minimal representation of the system's slow dynamics.

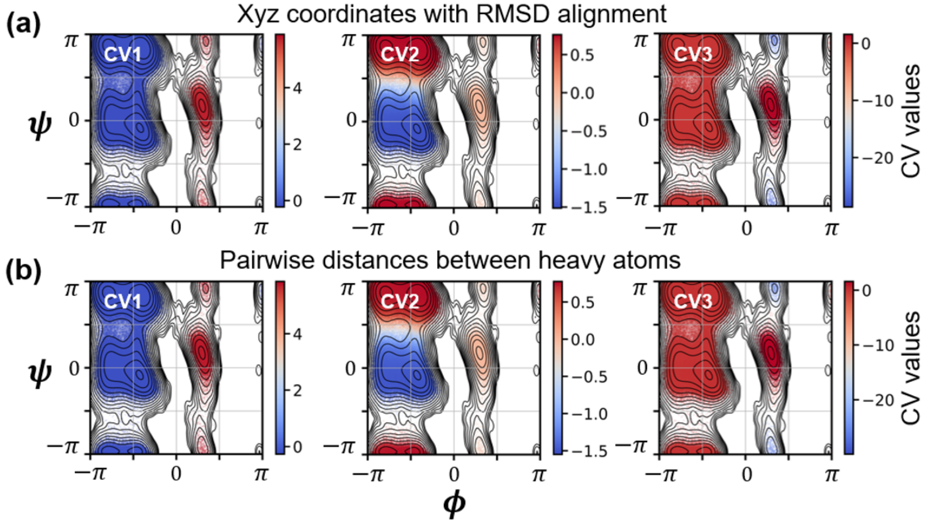


Figure 6.12: MEMnets results for alanine dipeptide using pairwise distances between heavy atoms as input features. *(a) Projection of MD conformations onto three CVs identified by a MEMnets model trained on Cartesian* (x, y, z) *coordinates after RMSD alignment. In each panel, MD conformations are overlaid on the two-dimensional* (ϕ, ψ) *dihedral space, with a color bar indicating the values projected onto each CV. (b) Projection of MD conformations onto three CVs identified by a MEMnets model trained on pairwise distances between heavy atoms. As in panel (a), MD conformations are displayed on the* (ϕ, ψ) *space, colored by their corresponding CV values. The pretraining phase consisted of 15 epochs with* $\gamma = 2$*, followed by 10 epochs under an exponential decay schedule.*

Impact of input features on CV interpretation. Regarding the influence of input features on the interpretation of CVs, our focus on biomolecular conformational changes naturally motivates the use of internal coordinates, such as pairwise distances between heavy atoms or Cα atoms, which effectively capture structural transitions. In this work, heavy-atom Cartesian coordinates were used as input features for alanine dipeptide, while pairwise distances were employed for the WW domain and RNAP systems. The use of Cartesian coordinates for alanine dipeptide is justified by the fact that all conformations in the dataset were pre-aligned to the first frame, thereby removing overall translational and rotational motions. Consequently, these Cartesian coordinates convey equivalent information to internal coordinates (e.g., pairwise distances) for describing conformational changes. To further confirm this equivalence, we trained an additional MEMnets model for alanine dipeptide using 45 pairwise distances among the 10 selected heavy atoms as input features and obtained identical

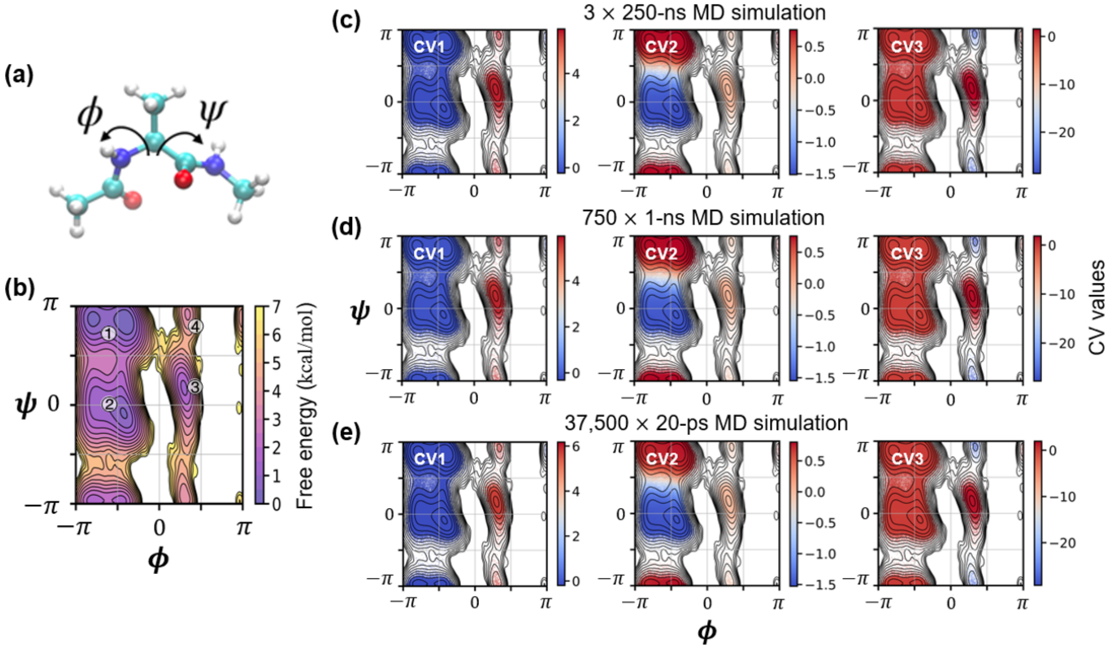


Figure 6.13: MEMnets results from truncated alanine dipeptide datasets. *(a) Representative structure of alanine dipeptide, where ϕ and ψ denote the two backbone torsion angles. (b) Free energy landscape of alanine dipeptide projected onto the two torsion angles (ϕ, ψ). (c) Projection of MD conformations from three 250-ns trajectories onto three CVs identified by a MEMnets model. In each panel, MD conformations are overlaid on the two-dimensional (ϕ, ψ) space, with a color bar indicating their values projected onto each CV.(d–e) Same as panel (c), but trained on truncated datasets: (d) a divided dataset comprising 750 1-ns MD trajectories, and (e) a divided dataset comprising 37,500 20-ps MD trajectories. The pretraining phase consisted of 15 epochs with $\gamma = 2$, followed by 10 epochs under an exponential-decay schedule.*

CVs (Figure 6.12).

Looking ahead, MEMnets could be extended through integration with advanced neural network architectures such as E(3)-equivariant neural networks to represent protein conformations as graphs and exploit Euclidean symmetries for more effective CV extraction. Moreover, for complex biological systems, incorporating physical intuition about specific conformational changes of interest remains essential for selecting appropriate input features.

MD trajectory length required for MEMnets training. MEMnets are trained using transitions between pairs of time points $(t - n_k\delta t, t)$, …, $(t - n_1\delta t, t)$, and $(t, t + \delta t)$, which correspond to lag times of δt and $[n_1\delta t, \ldots, n_k\delta t]$. Consequently, in theory, the MD trajectories required for MEMnets training only need to be as long as the longest lag time $(n_k\delta t)$. In practice, however, longer trajectories are often necessary to ensure statistical convergence.

Using alanine dipeptide as an example, we investigated the effect of trajectory length on MEMnets' ability to identify the slowest dynamic mode (see Figure 6.13). The full alanine dipeptide dataset, consisting of three 250-ns MD trajectories, was divided into shorter trajectories to construct two additional datasets: one containing 750 trajectories of 1 ns each, and another containing 37,500 trajectories of 20 ps each (approximately twice the longest lag time used in MEMnets, $n_k\delta t = 10$ ps). MEMnets models were then trained separately on these two datasets. As shown in Figure 6.13, the CVs identified from both subdivided datasets (Figures 6.13d–e) closely match those obtained from the original full-length trajectories (Figure 6.13c), demonstrating the robustness of MEMnets to variations in trajectory length.

6.4 Discussions

The key innovation of MEMnets lies in their ability to combine input features nonlinearly through encoder neural networks while explicitly accounting for memory effects. This design enables MEMnets to move beyond the Markovian assumption and identify continuous CVs that retain minimal memory. As a result, MEMnets-derived CVs can be employed to construct MSMs that accurately capture biomolecular kinetics using only a few states and substantially short MD trajectories.

In contrast, methods such as tICA and VAMPnets rely on the variational principle, which estimates the eigen/singular values of the Transfer/Koopman operator. The resulting timescales can approach but never exceed the true variational bound, leading to a systematic underestimation of the slowest dynamic processes. Furthermore, the variational principle assumes that the underlying dynamics are Markovian, which holds only in the full configurational space. In practice, however, these models operate in a reduced feature space (e.g., pairwise distances between a subset of heavy atoms or Cα atoms). For complex biomolecules such as RNAP which comprises roughly 540,000 atoms, only a limited number of features (e.g., 1,770 pairwise distances) are typically used [144]. Consequently, the Markovian assumption may not hold unless the lag time is sufficiently long. Moreover, when kinetic quantities are computed in the reduced CV space, the assumption of Markovianity introduces additional errors, since projecting the dynamics inevitably induces memory unless timescales are perfectly separated. Selecting a sufficiently long lag time can mitigate these effects, but in practice, lag times are constrained by the available MD trajectory length.

Hidden Markov Models (HMMs) [107] provide an alternative framework that achieves improved performance through fuzzy state boundaries rather than explicit treatment of non-Markovian dynamics. Using the WW domain as an example, we compared MEMnets and HMMs and found that MEMnets performed slightly better. Specifically, we constructed HMMs with five hidden states based on discrete state assignments from VAMPnets (at a lag time of 40 ns) using Deeptime [56]. The HMM predicted the slowest timescale of the WW domain to be 1.28×10^4 ns, slightly shorter than the 1.53×10^4 ns predicted by MEMnets using the same lag time. These findings suggest that incorporating fuzzy state boundaries and accounting for memory effects when identifying CVs both enhance MSM performance.

We also observed that MEMnets exhibit superior convergence and stability during training compared to SRVs and VAMPnets. However, this improved robustness comes at a computational cost: because MEMnets process multiple time sequences simultaneously, their runtime scales approximately k times slower than VAMPnets or SRVs, where k is the number of encoder networks.

Recent studies have proposed alternative deep learning approaches that identify CVs by enforcing Markovian latent-space dynamics, such as Langevin models [157]. These methods optimize the latent representation by minimizing the discrepancy between latent-space dynamics and the original one-step MD transitions, using architectures like normalizing flows. We envision that MEMnets could similarly be adapted to train a non-Markovian GME model in latent space, which could further serve as a generative AI framework for sampling biomolecular trajectories.

Moreover, the current MEMnets implementation employs a Euclidean latent space, but alternative topologies, such as hyperspherical manifolds [87], could enhance representation of the underlying free energy landscape. Since MEMnets are grounded in the IGME theory, which assumes equilibrium and detailed balance, the method is best suited for systems with well-separated timescales (e.g., biomolecular conformational transitions). Future extensions of IGME beyond detailed balance could substantially expand MEMnets' applicability to nonequilibrium dynamics. Additionally, incorporating a Softmax layer into MEMnets could enable direct projection of MD conformations onto discrete metastable states with minimal memory, bridging continuous CV discovery and state decomposition.

Taken together, these results highlight MEMnets as a powerful and general framework for uncovering memory-minimized representations of complex biomolecular dynamics, with promising potential for future integration into non-Markovian, generative, and symmetry-aware machine learning models.

This chapter is adapted from a peer-reviewed publication *"Liu, B., Cao, S., Boysen, J.G., Xue, M., Huang, X. (2025). Memory kernel minimization-based neural networks for discovering slow collective variables of biomolecular dynamics. Nature Computational Science, 1-10. https://doi.org/10.1038/s43588-025-00815-8"*

Chapter 7

Conclusion and Future Work

7.1 Summary

Proteins are inherently dynamic molecules, and their function is often determined by rare conformational changes such as folding/unfolding, allosteric switching, binding/unbinding, and large-scale rearrangements. Atomistic molecular dynamics (MD) simulations provide a powerful route to study these processes with high spatial and temporal resolution; however, two practical challenges limit their broader use for functional mechanism discovery. First, many biologically relevant events occur on millisecond timescales or longer, creating a persistent timescale gap relative to what can be sampled routinely by MD. Second, even when substantial simulation data are available, extracting interpretable mechanistic insight from high-dimensional trajectories remains nontrivial. This thesis develops physics-informed deep-learning frameworks combined with MD simulations for studying protein (as well as other biomolecules) dynamics.

After establishing the theoretical foundation of Markov state models (MSMs), the thesis advances several complementary directions. To improve representation and interpretability in multi-body systems where geometric symmetries are essential, GraphVAMPnets extend conventional MSMs that rely on fixed molecular descriptors to graph-based representation learning, enabling the discovery of slow collective variables and metastable states while respecting intrinsic symmetries. To move beyond coarse-grained state-to-state kinetics and directly interrogate barrier-crossing events, TS-DAR provides a unified deep-learning framework that integrates dimensionality reduction, state decomposition, and transition-state identification across multiple free-energy barriers; the AutoDAR extension further improves the practical usability of TS-DAR by enabling agentic, modular workflows and by establishing a foundation for closed-loop kinetic modeling and sampling. Finally, to address the limitations of the Markov assumption in complex biomolecules, the thesis develops and summarizes non-Markovian kinetic modeling based on the generalized master equation (GME). qMSM and IGME incorporate memory effects to yield stable and accurate long-timescale dynamics prediction at feasible lag times, and MEMnets further generalize this perspective from discrete states to continuous collective variables and formulate memory minimization as a learning objective for representation learning of kinetic data.

7.2 Future Work

Looking forward, several directions can further extend the impact of the frameworks developed in this thesis. First, representation learning can be more tightly integrated with enhanced-sampling strategies (e.g., metadynamics), where learned slow coordinates or uncertainty signals guide bias construction and bias updates in an iterative, closed-loop manner. Second, memory-aware kinetic modeling naturally motivates *dynamics forecasting* beyond the Markovian setting; in particular, it will be interesting to develop non-Markovian, memory-augmented dynamics forecasting surrogates (e.g., based on diffusin/flow-matching objectives) that preserve long-timescale dynamics while remaining data-efficient. Third, while this thesis focuses on protein conformational dynamics, the underlying principles of symmetry, and memory may generalize to substantially larger and more heterogeneous systems, including chromatin and genome-scale organization, where multi-body interactions and hierarchical timescales are central. Fourth, incorporating kinetic-modeling knowledge into the current large foundation models as a physical prior, for example, by embedding operator-/GME-inspired constraints or learned physical representations, may improve generalization and stabilize long-horizon predictions. Fifth, deeper integration with experimental observables (e.g., NMR, smFRET) offers a path to constrain models, calibrate kinetics, and validate mechanistic hypotheses under realistic conditions. Finally, on the automation side, AutoDAR can be extended from an analysis-and-sampling framework to a broader drug-discovery cycle by incorporating additional specialized agents for target selection, virtual screening, lead optimization, and experimental validation, etc., thereby moving toward reproducible, scalable, and increasingly autonomous discovery workflows.

Bibliography

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016.

[2] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, August 2021.

[3] Xiao-chen Bai, Greg McMullan, and Sjors H.W Scheres. How cryo-em is revolutionizing structural biology. *Trends in Biochemical Sciences*, 40(1):49–57, January 2015.

[4] Alessandro Barducci, Massimiliano Bonomi, and Michele Parrinello. Metadynamics. *WIREs Computational Molecular Science*, 1(5):826–843, February 2011.

[5] Emilia P. Barros, Özlem Demir, Jenaro Soto, Melanie J. Cocco, and Rommie E. Amaro. Markov state models and nmr uncover an overlooked allosteric loop in p53. *Chemical Science*, 12(5):1891–1900, 2021.

[6] H. M. Berman. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, January 2000.

[7] Robert B. Best and Gerhard Hummer. Reaction coordinates and rates from transition paths. *Proceedings of the National Academy of Sciences*, 102(19):6732–6737, April 2005.

[8] Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, December 2023.

[9] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 517–526. PMLR, 06–11 Aug 2017.

[10] Peter G. Bolhuis, David Chandler, Christoph Dellago, and Phillip L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual Review of Physical Chemistry*, 53(1):291–318, October 2002.

[11] Peter G. Bolhuis, Christoph Dellago, and David Chandler. Reaction coordinates of biomolecular isomerization. *Proceedings of the National Academy of Sciences*, 97(11):5877–5882, May 2000.

[12] Luigi Bonati, GiovanniMaria Piccini, and Michele Parrinello. Deep learning the slow modes for rare events sampling. *Proceedings of the National Academy of Sciences*, 118(44), October 2021.

[13] Gregory R. Bowman, Eric R. Bolin, Kathryn M. Hart, Brendan C. Maguire, and Susan Marqusee. Discovery of multiple hidden allosteric sites by combining markov state models and experiments. *Proceedings of the National Academy of Sciences*, 112(9):2734–2739, February 2015.

[14] Gregory R Bowman, Vincent A Voelz, and Vijay S Pande. Taming the complexity of protein folding. *Current Opinion in Structural Biology*, 21(1):4–11, February 2011.

[15] Florian Brueckner, Julio Ortiz, and Patrick Cramer. A movie of the rna polymerase nucleotide addition cycle. *Current Opinion in Structural Biology*, 19(3):294–299, June 2009.

[16] Ignasi Buch, Toni Giorgino, and Gianni De Fabritiis. Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. *Proceedings of the National Academy of Sciences*, 108(25):10184–10189, June 2011.

[17] Siqin Cao, Andrés Montoya-Castillo, Wei Wang, Thomas E. Markland, and Xuhui Huang. On the advantages of exploiting memory in markov state models for biomolecular dynamics. *The Journal of Chemical Physics*, 153(1), July 2020.

[18] Siqin Cao, Yunrui Qiu, Michael L. Kalin, and Xuhui Huang. Integrative generalized master equation: A method to study long-timescale biomolecular dynamics via the integrals of memory kernels. *The Journal of Chemical Physics*, 159(13), October 2023.

[19] Javier Cerrillo and Jianshu Cao. Non-markovian dynamical maps: Numerical processing of open quantum trajectories. *Physical Review Letters*, 112(11), March 2014.

[20] N. CHAFFEY. Alberts, b., johnson, a., lewis, j., raff, m., roberts, k. and walter, p. molecular biology of the cell. 4th edn. *Annals of Botany*, 91(3):401–401, February 2003.

[21] Shiqi Chen, John A. Parker, Curtis W. Peterson, Stuart A. Rice, Norbert F. Scherer, and Andrew L. Ferguson. Understanding and design of non-conservative optical matter systems using markov state models. *Molecular Systems Design & Engineering*, 7(10):1228–1238, 2022.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[23] Wei Chen, Hythem Sidky, and Andrew L. Ferguson. Nonlinear discovery of slow molecular modes using state-free reversible vampnets. *The Journal of Chemical Physics*, 150(21), 2019.

[24] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, May 2005.

[25] Evangelos A. Coutsias, Chaok Seok, and Ken A. Dill. Using quaternions to calculate rmsd. *Journal of Computational Chemistry*, 25(15):1849–1857, September 2004.

[26] Lin-Tai Da, Chao E, Baogen Duan, Chuanbiao Zhang, Xin Zhou, and Jin Yu. A jump-from-cavity pyrophosphate ion release assisted by a key lysine residue in t7 rna polymerase transcription elongation. *PLOS Computational Biology*, 11(11):e1004624, November 2015.

[27] Lin-Tai Da, Fátima Pardo-Avila, Liang Xu, Daniel-Adriano Silva, Lu Zhang, Xin Gao, Dong Wang, and Xuhui Huang. Bridge helix bending promotes rna polymerase ii backtracking through a critical and conserved threonine residue. *Nature Communications*, 7(1), April 2016.

[28] Lin-Tai Da, Dong Wang, and Xuhui Huang. Dynamics of pyrophosphate ion release and its coupled trigger loop motion from closed to open state in rna polymerase ii. *Journal of the American Chemical Society*, 134(4):2399–2406, January 2012.

[29] Benjamin A. Dalton, Cihan Ayaz, Henrik Kiefer, Anton Klimek, Lucas Tepper, and Roland R. Netz. Fast protein folding is governed by memory-dependent friction. *Proceedings of the National Academy of Sciences*, 120(31), July 2023.

[30] Payel Das, Mark Moll, Hernán Stamati, Lydia E. Kavraki, and Cecilia Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences*, 103(26):9885–9890, June 2006.

[31] {Tim R.} Davidson, Luca Falorsi, Nicola {De Cao}, Thomas Kipf, and {Jakub M.} Tomczak. Hyperspherical variational auto-encoders. In Ricardo Silva, Amir Globerson, and Amir Globerson, editors, *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pages 856–865. Association For Uncertainty in Artificial Intelligence (AUAI), January 2018. 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018 ; Conference date: 06-08-2018 Through 10-08-2018.

[32] Nan-jie Deng, Wei Dai, and Ronald M. Levy. How kinetics within the unfolded state affects protein folding: An analysis based on markov state models and an ultra-long md trajectory. *The Journal of Physical Chemistry B*, 117(42):12787–12799, 2013.

[33] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled markov chains. *Linear Algebra and its Applications*, 315(1–3):39–59, August 2000.

[34] Peter Deuflhard and Marcus Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, March 2005.

[35] Thomas G. Dietterich. Steps toward robust artificial intelligence. *AI Magazine*, 38(3):3–24, September 2017.

[36] Georg Diez, Daniel Nagel, and Gerhard Stock. Correlation-based feature selection to identify functional dynamics in proteins. *Journal of Chemical Theory and Computation*, 18(8):5079–5088, July 2022.

[37] Ken A. Dill and Justin L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, November 2012.

[38] Xinqiang Ding, Xingcheng Lin, and Bin Zhang. Stability and folding pathways of tetra-nucleosome from six-dimensional free energy surface. *Nature Communications*, 12(1), February 2021.

[39] Anthony J. Dominic, Thomas Sayer, Siqin Cao, Thomas E. Markland, Xuhui Huang, and Andrés Montoya-Castillo. Building insightful, memory-enriched models to capture long-time biochemical processes from short-time simulations. *Proceedings of the National Academy of Sciences*, 120(12), March 2023.

[40] Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. String method for the study of rare events. *Physical Review B*, 66(5), August 2002.

[41] Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. Finite temperature string method for the study of rare events. *The Journal of Physical Chemistry B*, 109(14):6688–6693, February 2005.

[42] Weinan E. and Eric Vanden-Eijnden. Towards a theory of transition paths. *Journal of Statistical Physics*, 123(3):503–523, May 2006.

[43] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[44] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with alphafold-multimer. October 2021.

[45] Jiangyan Feng, Balaji Selvam, and Diwakar Shukla. How do antiporters exchange substrates across the cell membrane? an atomic-level description of the complete exchange cycle in nark. *Structure*, 29(8):922–933.e3, August 2021.

[46] Andrew L. Ferguson, Athanassios Z. Panagiotopoulos, Ioannis G. Kevrekidis, and Pablo G. Debenedetti. Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach. *Chemical Physics Letters*, 509(1–3):1–11, June 2011.

[47] Kalindu S. Fernando, Ghodsiehsadat Jahanmir, Ilona C. Unarta, and Ying Chau. Multiscale computational framework for the liquid–liquid phase separation of intrinsically disordered proteins. *Langmuir*, 40(14):7607–7619, March 2024.

[48] Mahdi Ghorbani, Samarjeet Prasad, Jeffery B. Klauda, and Bernard R. Brooks. Graphvampnet, using graph neural networks and variational approach to markov processes for dynamical modeling of biomolecules. *The Journal of Chemical Physics*, 156(18), May 2022.

[49] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[50] Eshani C. Goonetilleke, Bojun Liu, Yue Wu, Michael S. O'Connor, and Xuhui Huang. A practical guide to transition state analysis in biomolecular simulations with ts-dar. *The Journal of Physical Chemistry B*, November 2025.

[51] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Pushmeet Kohli, Yossi Matias, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yunhan Xu, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Towards an ai co-scientist, 2025.

[52] Hanlin Gu, Wei Wang, Siqin Cao, Ilona Christy Unarta, Yuan Yao, Fu Kit Sheong, and Xuhui Huang. Rpnet: a reverse-projection-based neural network for coarse-graining metastable conformational states for protein dynamics. *Physical Chemistry Chemical Physics*, 24(3):1462–1474, 2022.

[53] Wenyu Guo, Yi Zeng, Haoran Wei, and Xiaoyan Zheng. Fluorescent emission mechanism of complex aggregates, 2025.

[54] Ziwei He, Christophe Chipot, and Benoît Roux. Committor-consistent variational string method. *The Journal of Physical Chemistry Letters*, 13(40):9263–9271, 2022.

[55] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[56] Moritz Hoffmann, Martin Scherer, Tim Hempel, Andreas Mardt, Brian de Silva, Brooke E Husic, Stefan Klus, Hao Wu, Nathan Kutz, Steven L Brunton, and Frank Noé. Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 3(1):015009, December 2021.

[57] Gerhard Hummer and Attila Szabo. Optimal dimensionality reduction of multistate kinetic and markov-state models. *The Journal of Physical Chemistry B*, 119(29):9029–9037, October 2014.

[58] Mariusz Jaskolski, Zbigniew Dauter, and Alexander Wlodawer. A brief history of macromolecular crystallography, illustrated by a family tree and its nobel fruits. *The FEBS Journal*, 281(18):3985–4009, April 2014.

[59] Hanlun Jiang, Fu Kit Sheong, Lizhe Zhu, Xin Gao, Julie Bernauer, and Xuhui Huang. Markov state models reveal a two-step mechanism of mirna loading into the human argonaute protein: Selective binding followed by structural re-arrangement. *PLOS Computational Biology*, 11(7):e1004404, July 2015.

[60] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, April 2016.

[61] Jr Jones, Larry E., Lei Ying, Anne B. Hofseth, Elena Jelezcova, Robert W. Sobol, Stefan Ambs, Curtis C. Harris, Michael Graham Espey, Lorne J. Hofseth, and Michael D. Wyatt. Differential effects of reactive nitrogen species on dna base excision repair initiated by the alkyladenine dna glycosylase. *Carcinogenesis*, 30(12):2123–2129, October 2009.

[62] Michael S. Jones, Zachary A. McDargh, Rafal P. Wiewiora, Jesus A. Izaguirre, Huafeng Xu, and Andrew L. Ferguson. Molecular latent space simulators for distributed and multimolecular trajectories. *The Journal of Physical Chemistry A*, 127(25):5470–5490, June 2023.

[63] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, July 2021.

[64] Hendrik Jung, Kei-ichi Okazaki, and Gerhard Hummer. Transition path sampling of rare events by shooting from the top. *The Journal of Chemical Physics*, 147(15), August 2017.

[65] John C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–666, 1958.

[66] Mohammed Khaled, Alemayehu Gorfe, and Abdallah Sayyed-Ahmad. Conformational and dynamical effects of tyr32 phosphorylation in k-ras: Molecular dynamics simulation and markov state models analysis. *The Journal of Physical Chemistry B*, 123(36):7667–7675, August 2019.

[67] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1), October 2018.

[68] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[69] Diego E. Kleiman and Diwakar Shukla. Active learning of the conformational ensemble of proteins using maximum entropy vampnets. *Journal of Chemical Theory and Computation*, 19(14):4377–4388, April 2023.

[70] Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noe, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52863–52883. Curran Associates, Inc., 2023.

[71] Heidi Klem, Glen M. Hocky, and Martin McCullagh. Size-and-shape space gaussian mixture models for structural clustering of molecular dynamics trajectories. *Journal of Chemical Theory and Computation*, 18(5):3218–3230, April 2022.

[72] Kai J. Kohlhoff, Diwakar Shukla, Morgan Lawrenz, Gregory R. Bowman, David E. Konerding, Dan Belov, Russ B. Altman, and Vijay S. Pande. Cloud-based simulations on google exacycle reveal ligand modulation of gpcr activation pathways. *Nature Chemistry*, 6(1):15–21, December 2013.

[73] Kirill A. Konovalov, Ilona Christy Unarta, Siqin Cao, Eshani C. Goonetilleke, and Xuhui Huang. Markov state models to study the functional dynamics of proteins in the wake of machine learning. *JACS Au*, 1(9):1330–1341, August 2021.

[74] Kirill A. Konovalov, Wei Wang, Guo Wang, Eshani C. Goonetilleke, Xin Gao, Dong Wang, and Xuhui Huang. A comprehensive mechanism for 5-carboxylcytosine-induced transcriptional pausing revealed by markov state models. *Journal of Biological Chemistry*, 296:100735, January 2021.

[75] Rongjie Lai and Jianfeng Lu. Point cloud discretization of fokker–planck operators for committor functions. *Multiscale Modeling & Simulation*, 16(2):710–726, January 2018.

[76] Morgan Lawrenz, Diwakar Shukla, and Vijay S. Pande. Cloud computing approaches for prediction of ligand binding poses and pathways. *Scientific Reports*, 5(1), January 2015.

[77] Wolfgang Lechner, Jutta Rogal, Jarek Juraszek, Bernd Ensing, and Peter G. Bolhuis. Nonlinear reaction coordinate analysis in the reweighted path ensemble. *The Journal of Chemical Physics*, 133(17), November 2010.

[78] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds, 2017.

[79] Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan S. Veeling, Iryna Zaporozhets, Yaoyi Chen, Soojung Yang, Adam E. Foster, Arne Schneuing, Jigyasa Nigam, Federico Barbero, Vincent Stimper, Andrew Campbell, Jason Yim, Marten Lienen, Yu Shi, Shuxin Zheng, Hannes Schulz, Usman Munir, Roberto Sordillo, Ryota Tomioka, Cecilia Clementi, and Frank Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761), August 2025.

[80] Chu Li, Zhuo Liu, Eshani C. Goonetilleke, and Xuhui Huang. Temperature-dependent kinetic pathways of heterogeneous ice nucleation competing between classical and non-classical nucleation. *Nature Communications*, 12(1), August 2021.

[81] Junnan Li, Caiming Xiong, and Steven Hoi. Mopro: Webly supervised learning with momentum prototypes. In *International Conference on Learning Representations*, 2021.

[82] Qianxiao Li, Bo Lin, and Weiqing Ren. Computing committor functions for the study of rare events using deep learning. *The Journal of Chemical Physics*, 151(5), August 2019.

[83] Zhan-Wei Li, You-Liang Zhu, Zhong-Yuan Lu, and Zhao-Yan Sun. Supracolloidal fullerene-like cages: design principles and formation mechanisms. *Physical Chemistry Chemical Physics*, 18(47):32534–32540, 2016.

[84] Zhan-Wei Li, You-Liang Zhu, Zhong-Yuan Lu, and Zhao-Yan Sun. A versatile model for soft patchy particles with various patch arrangements. *Soft Matter*, 12(3):741–749, 2016.

[85] Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, October 2011.

[86] Florian Litzinger, Lorenzo Boninsegna, Hao Wu, Feliks Nüske, Raajen Patel, Richard Baraniuk, Frank Noé, and Cecilia Clementi. Rapid calculation of molecular kinetics using compressed sensing. *Journal of Chemical Theory and Computation*, 14(5):2771–2783, April 2018.

[87] Bojun Liu, Jordan G. Boysen, Ilona Christy Unarta, Xuefeng Du, Yixuan Li, and Xuhui Huang. Exploring transition states of protein conformational changes via out-of-distribution detection in the hyperspherical latent space. *Nature Communications*, 16(1), January 2025.

[88] Bojun Liu, Siqin Cao, Jordan G. Boysen, Mingyi Xue, and Xuhui Huang. Memory kernel minimization-based neural networks for discovering slow collective variables of biomolecular dynamics. *Nature Computational Science*, 5(7):562–571, June 2025.

[89] Bojun Liu, Yunrui Qiu, Eshani C. Goonetilleke, and Xuhui Huang. Kinetic network models to study molecular self-assembly in the wake of machine learning. *MRS Bulletin*, 47(9):958–966, September 2022.

[90] Bojun Liu, Mingyi Xue, Yunrui Qiu, Kirill A. Konovalov, Michael S. O'Connor, and Xuhui Huang. Graphvampnets for uncovering slow collective variables of self-assembly dynamics. *The Journal of Chemical Physics*, 159(9), September 2023.

[91] Feng Liu, Deguo Du, Amelia A. Fuller, Jennifer E. Davoren, Peter Wipf, Jeffery W. Kelly, and Martin Gruebele. An experimental survey of the transition between two-state and downhill protein folding scenarios. *Proceedings of the National Academy of Sciences*, 105(7):2369–2374, February 2008.

[92] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[93] Thomas Löhr, Kai Kohlhoff, Gabriella T. Heller, Carlo Camilloni, and Michele Vendruscolo. A kinetic ensemble of the alzheimer's aβ peptide. *Nature Computational Science*, 1(1):71–78, January 2021.

[94] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024.

[95] Ao Ma and Aaron R. Dinner. Automatic method for identifying reaction coordinates in complex systems. *The Journal of Physical Chemistry B*, 109(14):6769–6779, February 2005.

[96] Robert D. Malmstrom, Alexandr P. Kornev, Susan S. Taylor, and Rommie E. Amaro. Allostery through the computational microscope: camp activation of a canonical signalling domain. *Nature Communications*, 6(1), 2015.

[97] Luca Maragliano, Alexander Fischer, Eric Vanden-Eijnden, and Giovanni Ciccotti. String method in collective variables: Minimum free energy paths and isocommittor surfaces. *The Journal of Chemical Physics*, 125(2), July 2006.

[98] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature Communications*, 9(1), January 2018.

[99] J. Andrew McCammon, Bruce R. Gelin, and Martin Karplus. Dynamics of folded proteins. *Nature*, 267(5612):585–590, June 1977.

[100] Pascal Mettes, Elise van der Pol, and Cees Snoek. Hyperspherical prototype networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[101] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023.

[102] Sina Mohseni, Haotao Wang, Chaowei Xiao, Zhiding Yu, Zhangyang Wang, and Jay Yadawa. Taxonomy of machine learning safety: A survey and primer. *ACM Computing Surveys*, 55(8):1–38, December 2022.

[103] Yusuke Mori, Kei-ichi Okazaki, Toshifumi Mori, Kang Kim, and Nobuyuki Matubayasi. Learning reaction coordinates via cross-entropy minimization: Application to alanine dipeptide. *The Journal of Chemical Physics*, 153(5), August 2020.

[104] Rhiannon Morris, Katrina A. Black, and Elliott J. Stollar. Uncovering protein function: from classification to complexes. *Essays in Biochemistry*, 66(3):255–285, August 2022.

[105] Klaus Müller and Leo D. Brown. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoretica Chimica Acta*, 53(1):75–93, 1979.

[106] Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R. Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106(45):19011–19016, November 2009.

[107] Frank Noé, Hao Wu, Jan-Hendrik Prinz, and Nuria Plattner. Projected and hidden markov models for calculating kinetics and metastable states of complex molecules. *The Journal of Chemical Physics*, 139(18), November 2013.

[108] Feliks Nüske, Bettina G. Keller, Guillermo Pérez-Hernández, Antonia S. J. S. Mey, and Frank Noé. Variational approach to molecular kinetics. *Journal of Chemical Theory and Computation*, 10(4):1739–1752, 2014. PMID: 26580382.

[109] Feliks Nüske, Hao Wu, Jan-Hendrik Prinz, Christoph Wehmeyer, Cecilia Clementi, and Frank Noé. Markov state models from short non-equilibrium simulations—analysis and correction of estimation bias. *The Journal of Chemical Physics*, 146(9), March 2017.

[110] Albert C. Pan, Deniz Sezer, and Benoît Roux. Finding transition pathways using the string method with swarms of trajectories. *The Journal of Physical Chemistry B*, 112(11):3432–3440, February 2008.

[111] Sijia Peng, Xiaowei Wang, Lu Zhang, Shanshan He, Xin Sheng Zhao, Xuhui Huang, and Chunlai Chen. Target search and recognition mechanisms of glycosylase alkd revealed by scanning fret-fcs and markov state models. *Proceedings of the National Academy of Sciences*, 117(36):21889–21895, August 2020.

[112] Max F. Perutz, M. G. Rossmann, A. F. Cullis, H. Muirhead, G. Will, and A. C. T. North. Structure of haemoglobin: A three-dimensional fourier synthesis at 5.5 å resolution, obtained by x-ray analysis. *Nature*, 185(4711):416–422, 1960.

[113] Baron Peters and Bernhardt L. Trout. Obtaining reaction coordinates by likelihood maximization. *The Journal of Chemical Physics*, 125(5), August 2006.

[114] Nuria Plattner and Frank Noé. Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and markov models. *Nature Communications*, 6(1), 2015.

[115] Jan-Hendrik Prinz, Martin Held, Jeremy C. Smith, and Frank Noé. Efficient computation, sensitivity, and error analysis of committor probabilities for complex dynamical processes. *Multiscale Modeling & Simulation*, 9(2):545–567, April 2011.

[116] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D. Chodera, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics*, 134(17), May 2011.

[117] Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of Chemical Physics*, 139(1), July 2013.

[118] Yunrui Qiu, Rafal P. Wiewiora, Jesus A. Izaguirre, Huafeng Xu, Woody Sherman, Weiping Tang, and Xuhui Huang. Non-markovian dynamic models identify non-canonical kras-vhl encounter complex conformations for novel protac design. *JACS Au*, 4(10):3857–3868, September 2024.

[119] Robert Julian Rabben, Sourav Ray, and Marcus Weber. Isokann: Invariant subspaces of koopman operators learned by a neural network. *The Journal of Chemical Physics*, 153(11), September 2020.

[120] Young Min Rhee and Vijay S. Pande. One-dimensional reaction coordinate and the corresponding potential of mean force from commitment probability distribution. *The Journal of Physical Chemistry B*, 109(14):6780–6786, January 2005.

[121] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[122] Mary A. Rohrdanz, Wenwei Zheng, Mauro Maggioni, and Cecilia Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of Chemical Physics*, 134(12), March 2011.

[123] Benoît Roux. String method with swarms-of-trajectories, mean drifts, lag time, and committor. *The Journal of Physical Chemistry A*, 125(34):7558–7571, August 2021.

[124] Susanna Röblitz and Marcus Weber. Fuzzy spectral clustering by pcca+: application to markov state models and data classification. *Advances in Data Analysis and Classification*, 7(2):147–179, May 2013.

[125] Marissa G. Saunders and Gregory A. Voth. Coarse-graining methods for computational biology. *Annual Review of Biophysics*, 42(1):73–93, May 2013.

[126] Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 36449–36462. Curran Associates, Inc., 2023.

[127] Christian R. Schwantes and Vijay S. Pande. Improvements in markov state model construction reveal many non-native interactions in the folding of ntl9. *Journal of Chemical Theory and Computation*, 9(4):2000–2009, March 2013.

[128] Christian R. Schwantes and Vijay S. Pande. Modeling molecular kinetics with tica and the kernel trick. *Journal of Chemical Theory and Computation*, 11(2):600–608, January 2015.

[129] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. Schnet – a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), March 2018.

[130] D. Sculley. Web-scale k-means clustering, April 2010.

[131] David E. Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, Michael P. Eastwood, Joseph A. Bank, John M. Jumper, John K. Salmon, Yibing Shan, and Willy Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, October 2010.

[132] Ben Shneiderman. Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered ai systems. *ACM Transactions on Interactive Intelligent Systems*, 10(4):1–31, October 2020.

[133] Daniel-Adriano Silva, Gregory R. Bowman, Alejandro Sosa-Peinado, and Xuhui Huang. A role for both conformational selection and induced fit in ligand binding by the lao protein. *PLoS Computational Biology*, 7(5):e1002054, May 2011.

[134] Daniel-Adriano Silva, Dahlia R. Weiss, Fátima Pardo Avila, Lin-Tai Da, Michael Levitt, Dong Wang, and Xuhui Huang. Millisecond dynamics of rna polymerase ii translocation at atomic resolution. *Proceedings of the National Academy of Sciences*, 111(21):7665–7670, April 2014.

[135] Nilmani Singh, Stephan Lane, Tianhao Yu, Jingxia Lu, Adrianna Ramos, Haiyang Cui, and Huimin Zhao. A generalized platform for artificial intelligence-powered autonomous enzyme engineering. *Nature Communications*, 16(1), July 2025.

[136] Chang Yun Son, Arun Yethiraj, and Qiang Cui. Cavity hydration dynamics in cytochrome c oxidase and functional implications. *Proceedings of the National Academy of Sciences*, 114(42), October 2017.

[137] Ryan K. Spencer, Glenn L. Butterfoss, John R. Edison, James R. Eastwood, Stephen Whitelam, Kent Kirshenbaum, and Ronald N. Zuckermann. Stereochemistry of polypeptoid chain configurations. *Biopolymers*, 110(6), March 2019.

[138] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters*, 314(1–2):141–151, November 1999.

[139] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20827–20840. PMLR, 17–23 Jul 2022.

[140] Ernesto Suárez, Rafal P. Wiewiora, Chris Wehmeyer, Frank Noé, John D. Chodera, and Daniel M. Zuckerman. What markov state models can and cannot do: Correlation versus path-based observables in protein-folding models. *Journal of Chemical Theory and Computation*, 17(5):3119–3133, April 2021.

[141] Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The virtual lab of ai agents designs new sars-cov-2 nanobodies. *Nature*, 646(8085):716–723, July 2025.

[142] Benjamin Trendelkamp-Schroer, Hao Wu, Fabian Paul, and Frank Noé. Estimation and uncertainty of reversible markov models. *The Journal of Chemical Physics*, 143(17), November 2015.

[143] Ilona C. Unarta, Siqin Cao, Eshani C. Goonetilleke, Jiani Niu, Samuel H. Gellman, and Xuhui Huang. Submillisecond atomistic molecular dynamics simulations reveal hydrogen bond-driven diffusion of a guest peptide in protein–rna condensate. *The Journal of Physical Chemistry B*, 128(10):2347–2359, February 2024.

[144] Ilona Christy Unarta, Siqin Cao, Shintaroh Kubo, Wei Wang, Peter Pak-Hang Cheung, Xin Gao, Shoji Takada, and Xuhui Huang. Role of bacterial rna polymerase gate opening dynamics in dna loading and antibiotics inhibition elucidated by quasi-markov state model. *Proceedings of the National Academy of Sciences*, 118(17), April 2021.

[145] Vincent A. Voelz, Gregory R. Bowman, Kyle Beauchamp, and Vijay S. Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9(1−39). *Journal of the American Chemical Society*, 132(5):1526–1528, January 2010.

[146] Jeffrey R. Wagner, Christopher T. Lee, Jacob D. Durrant, Robert D. Malmstrom, Victoria A. Feher, and Rommie E. Amaro. Emerging computational methods for the rational discovery of allosteric drugs. *Chemical Reviews*, 116(11):6370–6390, April 2016.

[147] Hongbin Wan, Yunhui Ge, Asghar Razavi, and Vincent A. Voelz. Reconciling simulated ensembles of apomyoglobin with experimental hydrogen/deuterium exchange data using bayesian inference and multiensemble markov state models. *Journal of Chemical Theory and Computation*, 16(2):1333–1348, January 2020.

[148] Beibei Wang, Rachel E. Sexton, and Michael Feig. Kinetics of nucleotide entry into rna polymerase active site provides mechanism for efficiency and fidelity. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 1860(4):482–490, April 2017.

[149] Dedi Wang, Yihang Wang, Luke Evans, and Pratyush Tiwary. From latent dynamics to meaningful representations. *Journal of Chemical Theory and Computation*, 20(9):3503–3513, April 2024.

[150] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1041–1049, New York, NY, USA, 2017. Association for Computing Machinery.

[151] Jiang Wang, Simon Olsson, Christoph Wehmeyer, Adrià Pérez, Nicholas E. Charron, Gianni de Fabritiis, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields. *ACS Central Science*, 5(5):755–767, April 2019.

[152] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 13–18 Jul 2020.

[153] Wei Wang, Siqin Cao, Lizhe Zhu, and Xuhui Huang. Constructing markov state models to elucidate the functional conformational changes of complex biomolecules. *WIREs Computational Molecular Science*, 8(1), October 2017.

[154] Hannah K. Wayment-Steele, Adedolapo Ojoawo, Renee Otten, Julia M. Apitz, Warintra Pitsawong, Marc Hömberger, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, 625(7996):832–839, November 2023.

[155] Christoph Wehmeyer and Frank Noé. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of Chemical Physics*, 148(24), March 2018.

[156] Hao Wu and Frank Noé. Variational approach for learning markov processes from time series data. *Journal of Nonlinear Science*, 30(1):23–66, August 2019.

[157] Hao Wu and Frank Noé. Reaction coordinate flows for model reduction of molecular kinetics. *The Journal of Chemical Physics*, 160(4), January 2024.

[158] Hao Wu, Feliks Nüske, Fabian Paul, Stefan Klus, Péter Koltai, and Frank Noé. Variational koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of Chemical Physics*, 146(15), April 2017.

[159] Kurt Wüthrich. The way to nmr structures of proteins. *Nature Structural Biology*, 8(11):923–925, November 2001.

[160] Tian Xie, Arthur France-Lanord, Yanming Wang, Yang Shao-Horn, and Jeffrey C. Grossman. Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials. *Nature Communications*, 10(1), 2019.

[161] Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[162] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, June 2024.

[163] Xiangze Zeng, Zhan-Wei Li, Xiaoyan Zheng, Lizhe Zhu, Zhao-Yan Sun, Zhong-Yuan Lu, and Xuhui Huang. Improving the productivity of monodisperse polyhedral cages by the rational design of kinetic self-assembly pathways. *Physical Chemistry Chemical Physics*, 20(15):10030–10037, 2018.

[164] Jun Zhang, Yao-Kun Lei, Zhen Zhang, Xu Han, Maodong Li, Lijiang Yang, Yi Isaac Yang, and Yi Qin Gao. Deep reinforcement learning of transition states. *Physical Chemistry Chemical Physics*, 23(11):6888–6895, 2021.

[165] Jun Zhang, Y. Isaac Yang, Lijiang Yang, and Yi Qin Gao. Dynamics and kinetics study of “in-water” chemical reactions by enhanced sampling of reactive trajectories. *The Journal of Physical Chemistry B*, 119(45):14505–14514, October 2015.

[166] Jun Zhang, Zhen Zhang, Yi Isaac Yang, Sirui Liu, Lijiang Yang, and Yi Qin Gao. Rich dynamics underlying solution reactions revealed by sampling and data mining of reactive trajectories. *ACS Central Science*, 3(5):407–414, April 2017.

[167] Lu Zhang, Fátima Pardo-Avila, Ilona Christy Unarta, Peter Pak-Hang Cheung, Guo Wang, Dong Wang, and Xuhui Huang. Elucidation of the dynamics of transcription elongation by rna polymerase ii using kinetic network models. *Accounts of Chemical Research*, 49(4):687–694, March 2016.

[168] Yutong Zhao, Fu Kit Sheong, Jian Sun, Pedro Sander, and Xuhui Huang. A fast parallel clustering algorithm for molecular simulation trajectories. *Journal of Computational Chemistry*, 34(2):95–104, 2012.

[169] Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, He Zhang, Shidi Tang, Hongxia Hao, Peiran Jin, Chi Chen, Frank Noé, Haiguang Liu, and Tie-Yan Liu. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5):558–567, May 2024.

[170] Xiaoyan Zheng, Lizhe Zhu, Xiangze Zeng, Luming Meng, Lu Zhang, Dong Wang, and Xuhui Huang. Kinetics-controlled amphiphile self-assembly processes. *The Journal of Physical Chemistry Letters*, 8(8):1798–1803, April 2017.

[171] Lizhe Zhu, Hanlun Jiang, Siqin Cao, Ilona Christy Unarta, Xin Gao, and Xuhui Huang. Critical role of backbone coordination in the mrna recognition by rna induced silencing complex. *Communications Biology*, 4(1), November 2021.