ABSTRACT

Title of Dissertation: BUILDING KINETIC MODELS
FOR COMPLEX SYSTEMS
WITH ARBITRARY MEMORIES

Sun-Ting Tsai
Doctor of Philosophy, 2022

Dissertation Directed by: Professor Pratyush Tiwary
Department of Chemistry and Biochemistry

Analyzing time series from complex dynamical systems in nature is a common yet challenging task in scientific computation since these time series are usually high-dimensional. To apply our physics intuitions to these dynamical systems often requires projecting these time series to certain low-dimensional degrees of freedom, which often introduces complicated memory effect. A simplest and classic example can be a 2-dimensional coupled differential equation. When one only looks at one of the Cartesian coordinates, one loses the predictability to predict what will happen next given the current 1-dimensional coordinate. The well-known solution is to describe the solution using the eigenvector, and the coupled equation is decoupled into a constant and a 1-dimensional memoryless equation. However, it can be imagined in a more complicated system we may have to look back to more time steps in the past, and it can be impossible to obtain a simple 1-dimensional eigenvector. In this work, we examine such memory effect within

time series generated from Langevin dynamics, Molecular Dynamics (MD) simulations, and some experimental time series. We also develop computational methods to minimize and model such memory effects using statistical mechanics and machine learning.

In recent years, MD simulation has become a powerful tool to model complex molecular dynamics in physics, chemistry, material science, biology, and many other fields. However, rare events such as droplet formation, nucleation, and protein conformational changes are hard to sample using MD simulations since they happen on the timescales far away from what all-atom MD simulation can reach. This makes MD simulation less useful for studying the mechanism of rare event kinetics. Therefore, it is a common practice to perform enhanced sampling techniques to help sample rare events, which requires performing dimensionality reduction from atomic coordinates to a low-dimensional representation that has a minimal memory effect.

In the first part of this study, we focus on reducing the memory effect by capturing slow degrees of freedom using a set of low-dimensional reaction coordinates (RCs). The RCs are a low-dimensional surrogate of the eigenvector in the example of coupled equations. When describing the system using RCs, other dimensions become constant except fast randomly fluctuating noise. These RCs can then be used to help reproducing correct kinetic connectivity between metastable states using enhanced sampling methods such as metadynamics. We demonstrate the utility of our method by applying them to the droplet formation from the gaseous phase of Lennard-Jones particles and the conformational changes of a small peptide Ace-Ala_3-Nme.

The second part of the study aims at modeling another type of memory coming from intrinsic long-term dependency induced by ignored fast degrees of freedom wherein

we utilize one of the fundamental machine learning techniques called the recurrent neural network to model non-Markovianity within time-series generated from MD simulations. This method has been shown to work not only on the molecular model of alanine dipeptide but also on experimental time series taken from single-molecule force spectroscopy. At the end of this second part, we also improve this method to extrapolate physics that the neural network had never seen in the training dataset by incorporating static or dynamical constraints on the path ensemble it generates.

BUILDING KINETIC MODELS FOR COMPLEX SYSTEMS WITH ARBITRARY MEMORIES

by

Sun-Ting Tsai

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor Pratyush Tiwary, Chair/Advisor
Professor Christopher Jarzynski, Co-Advisor
Professor John D. Weeks
Professor Michelle Girvan
Professor Jeffery Klauda

© Copyright by
Sun-Ting Tsai
2022

Acknowledgments

If I have ever accomplished any at the University of Maryland, it can't happen without any of the following people:

- My advisor Pratyush Tiwary. I am the first PhD student he takes. He often said as a new PI, he gave us some projects in wrong directions. That said, we also got an invaluable experience with intense research and maximal freedom at the same time. He is creative, energetic, and supportive to us. He is so good that I believe he is among the top advisors you can find at UMD. I feel sorry for those who didn't accept his offer of joining the group.

- My friends Yihang Wang, Yixu Wang, En-Jui Kuo, Yijia Xu, and Yalun Yu. Yihang and Yalun are the two best people I've met at UMD. I have received countless helps from them. Yixu is my roommate since my second year. He is perhaps the wonderful roommate a person can have. I also can't ignore his useful suggestions and comments in my research even though he does not coauthor with me in any of my papers. En-Jui and Yijia are the two greatest minds I know in UMD physics. They are both self-motivated and energetic. They bring me back with the true value of science. Our collaborations are of craziness and legend, and it has become one of my precious treasures of my PhD.

- My group members Zachary Smith, Connor Zou, Dedi Wang, Luke Evans, Shams Mehdi, and Eric Fields. Zack's program of SGOOP helps me a lot in my research. His multi-SGOOP paper motivates part of my research. I've also learned a lot from his presentation in group meetings. I would also like to thank Zack for proofreading this dissertation. Connor helps me code the angle order parameter and interfacial water. I very much enjoy discussing and doing research with him. He is very hardworking and talented. I hope I will have the chance to work with him again at the University of Michigan. Dedi helps me clarify difficult concepts in language processing and also gives me a lot of ideas in path sampling LSTM. I always enjoy discussing with him, as he always gives me the most critical comments and he is always correct. I feel honored to work with such a smart person like him. I still remember one of my discussions with Luke really motivates SGOOP-d. His profound presentation on the diffusion map inspired me with the idea of relating commute map and SGOOP. Shams is the person who sets up the Aib9 simulation. I am surprised by his amazing visualization plot of Aib9 and he also helps me make one for my ps-LSTM paper. Eric is the only undergraduate student who collaborates with me. He is very smart and seems to always have an idea when we encounter problems in research. I have no doubt that he will have a bright future ahead of him.

- My co-advisor Christopher Jarzynski. I would like to thank him for telling me that there is physics behind reaction coordinate. A 20-minute talk with him motivates my PhD study and eventually becomes the spirit throughout my research.

- My first-year roommate Chih-Chiao Hung, and Su-Kuan Chu. I have many thanks to Chih-Chiao and Su-Kuan. Without them, my life in Maryland wouldn't start so smoothly.

- My MEI speaking and writing teacher Susan Gould. Honestly, I've felt hopelessness, and unfairness when I failed the Maryland microteaching test. I also can't believe how much the English writing class costs. Susan is such a great teacher making you feel that all these things are worth it. I've got an incredible amount of chance to practice speaking and writing from her. She also carefully looks at every mistake we make and customizes her suggestions to optimize our learning. I can't forget those days I studied English in her class.

- Our graduate program coordinator Josiland Chambers. Josiland is truly the best coordinator I've ever met. She has a very well-organized way to handle things. I believe that's why she is so responsive. She is also super supportive to physics graduate students at Maryland. As I can see from my terpmail box, she always gives a definite answer to our question, so we won't have to worry about any uncertainty. Without her, I might just quit my PhD at some point because of our previous coordinator.

In the end, I would like to thank the University of Maryland for giving me the chance to work or learn with these people. They are beyond amazing as what I can say.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Aib9	α-aminoisobutyric acid 9
AI	Artificial Intelligence
ANN	Artificial Neural Network
SGOOP	Spectral Gap Optimization of Order Parameters
CK Test	Champman-Komogorov test
CNT	Classical Nucleation Theory
CV	Collective Variable
DOF	Degree of Freedom
FES	Free Energy Surface
FFS	Forward Flux Sampling
HMM	Hidden Markov Model
KDP	KH_2PO_4
LSTM	Long Short-term Memory
MaxCal	Maximum Caliber
MD	Molecular Dynamics
metaD	Metadynamics
MSM	Markov State Model
Multi-SGOOP	multi-dimensional SGOOP
NN	Nearest-Neighbor
ps-LSTM	path sampling Long Short-term Memory
RC	Reaction Coordinate
RNN	Recurrent Neural Network
SGOOP-d	SGOOP distance
smFRET	single-molecule force spectroscopy
tICA	time-lagged Independent Component Analysis
TPS	Transition Path Sampling
VAMP	Variational Approach for Markov Processes
WTmetaD	Well-tempered metadynamics

Chapter 1: Introduction

1.1 Motivation

Analyzing time series from complex dynamical systems in nature is a common yet challenging task in scientific computation since these time series are usually high-dimensional. In order to apply our physics intuitions to these dynamical systems, it is a common practice to project these time series to certain low-dimensional degrees of freedom, which often introduces complicated memory effects. In this dissertation, we will focus on modeling these memory effects within time series generated from Langevin dynamics, Molecular Dynamics (MD) simulations, and some experimental time series.

Over the past several decades, the fast development of computer science and improved computational power of modern computers have rendered molecular dynamics (MD) simulation a powerful tool to study complex molecular dynamics in physics, chemistry, biology, such as droplet formation, freezing, protein folding and unfolding [1–13]. These problems are of great importance to not only our understanding to soft condensed matters but also to industry such as discovery of new drugs or engineering novel materials [1, 14–19]. Contrary to the importance, they are intrinsically complex in nature and difficult to study due to their many-body characteristic and associated long-range interactions. MD simulations, built upon our understanding of these complex systems at atom-

istic level, integrate Newton's equations of motion, providing us a model to complement experimental observations and gain deeper understanding of underlying mechanisms. For example, MD simulations allow us to simulate crystal growth processes without any impurities, which is a hard-to-prepare environment in experiments. In MD simulations, we can also detect any stochastic events in atomistic precision and on femtosecond timescale, such as a droplet with a few atoms or molecules formed in a supersaturated vapour, which is difficult to detect with today's experimental techniques and technologies.

1.1.1 Timescale limitation of molecular dynamics (MD) simulations

Although MD simulation is a powerful tool to model complex dynamics, it suffers from notorious timescale limitation problem [1, 20]. For the MD simulation considering all atoms, the maximal integration timestep is usually required to be less than 2-5 fs in order to capture the movements of fast fluctuating Hydrogen atoms. The maximal number of iterations needed to reach a millisecond for MD simulation would then be around 10^8, which is still an incredible number for modern computers to execute in realistic time. However, most, if not all interesting and important physics in the complex systems consisting of a large number of atoms happen at the timescale of microseconds, seconds, or even hours. For example, protein can unfold from its folded structure at a frequency of approximately 1000 Hz in water at 25°C, which is equivalent to a millisecond [21]. Such phenomena can rarely be seen in the all-atom MD simulation since MD is too slow to reach their occurring timescale, so we call them "rare events". The previously mentioned important phenomena such as droplet formation, freezing, and crystal nucleation of atoms

or molecules are all examples of rare events.

1.1.2 Dimensionality reduction and memory effect

Visualizing or analyzing MD trajectories using the whole set of phase space variables is also not efficient if our focus is on the dynamics of the rare events, since the phase space is usually very high-dimension and most of the variables do not provide useful information. For example, just simulating a small protein consisting of 10 atoms results in a 60-dimensional phase space, while the positions of most Hydrogen atoms simply fluctuate randomly during simulations.

In fact, it has also been proposed that important physics of some of the rare events that happen in these complex systems can be captured by certain low-dimensional variables [22–24]. Unfortunately, we still don't have a reliable and systematic way to obtain such important low-dimensional representation, and we don't even know how many dimensions we need to consider in arbitrarily complex system.

Even though we completely capture the important low-dimensional variables, arbitrary memory effect can always come into effect from other variables. In other words, our models of any future events can depend on arbitrarily long history even after we include all the low-dimensional important physics from the past.

1.1.3 Projecting to slow Degree of Freedom (DOF) prevents memory

Overcoming the timescale limitation, finding low-dimensional important physics, and preventing memory effects are not completely separate problems. In this section, we

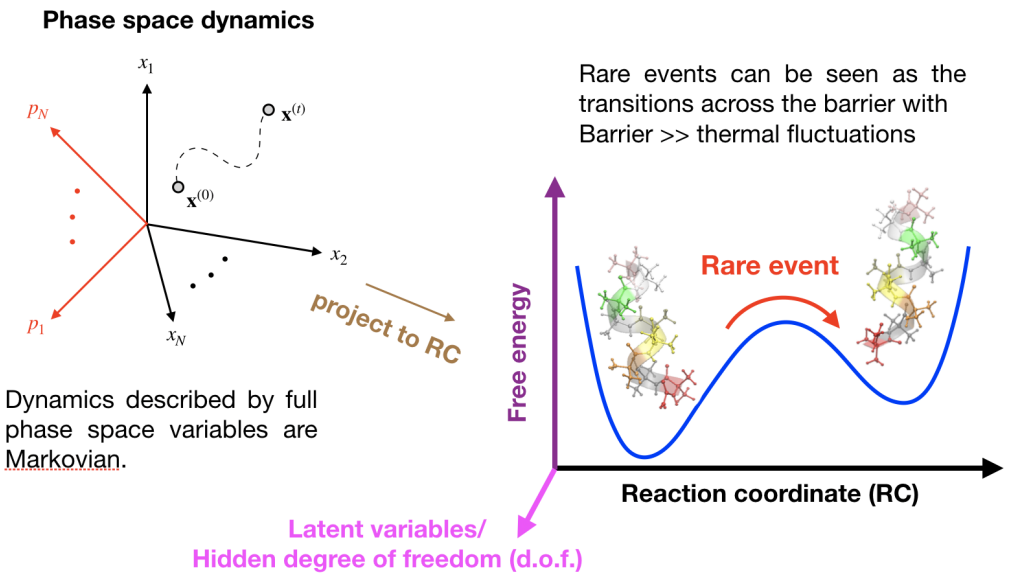


Figure 1.1: **Illustration of the projection from phase space dynamics to reaction coordinate (RC).** This figure shows the projection from phase space dynamics to reaction coordinate.

will see that they are closely related to each other. If we allow ourselves to describe a complex system with all phase space variables including positions and momenta of each atom, we can precisely predict any future event given the current status of the system. In other words, we have a history-independent Markov process, and there is no memory. An all-atom MD simulation is a Markov process guaranteed by the definition because MD simulation can be viewed as the integration of Newton's equations of motions for all phase space variables.

As we described in the previous section, the variables that really matter to the rare event dynamics are often low-dimensional. Therefore, we commonly project the system dynamics to a low-dimensional representation called "reaction coordinates (RC)" [25,26]. Ideally, the RC is a 1-dimensional coordinate which we expect to capture all important physics, including (i) thermodynamic truthfulness: demarcating between the various rele-

vant metastable states present in the actual high-dimensional system, (ii) kinetic truthfulness: preserving pathways for moving between these different states, and (iii) timescale separation: displaying a clean-cut separation of timescales between the relaxation times in the various metastable states, and the time spent in the actual event of crossing from one state to another, as illustrated in Fig. 1.1. An important assumption here is our RC is a slowly varying variable such that everything projected out from RC, which are called latent variables or hidden degrees of freedom, relaxes much faster than RC and can be treated as randomly fluctuating noise [27–29]. This assumption of randomly fluctuating latent variables is then crucial for us to describe the dynamics of RC as a Markov process again.

Therefore, a good RC should be the slowest degree of freedom. The rare events can then be seen as the transitions across the highest barrier on RC, where the barrier height is so high compared to thermal fluctuations.

1.2 Finding optimal reaction coordinate (RC)

As we have mentioned, the memory effect and sampling problem of rare events can be unified by finding an optimal RC. In this section, we will introduce two different approaches based on the properties of RC we discussed in the previous section.

1.2.1 Past-future information bottleneck as the reaction coordinate

The first method for finding an optimal RC is first introduced by Ribeiro, Wang, and Tiwary and called “Reweighted autoencoded variational Bayes for enhanced sam-

pling (RAVE)" [30]. This is a neural network-based method that aims at finding a low-dimensional representation of the current status with maximal predictability of the next time step because the dynamics of RC should be as Markovian as possible. As a result, in RAVE we construct the RC using the full phase space variables from MD simulation at every time step, make a prediction of the next step RC using a type of neural network called variational autoencoder, and then minimize the mean square error between the neural network output and actual RC of the next time step. This approach finds RC by making use of the neural network as an approximate Markov model for RC.

1.2.2 Spectral gap optimization of order parameters (SGOOP)

Another approach for finding such a RC is called "Spectral gap optimization of order parameters (SGOOP)" [26, 31]. In this approach, we first construct a transition probability matrix along any candidate RC based on statistical mechanics and then calculates its eigenspectrum. The eigenvalues of such transition matrix carry information about the timescales of various dynamical processes. The best RC will then produce a transition matrix with a maximal timescale separation between visible slow and hidden fast processes. This timescale separation, also known as the spectral gap, therefore quantifies how well we can approximate the hidden degree of freedom projected out by RC as a random noise.

Although SGOOP and RAVE use different properties we have introduced in Sec. 1.1.3 to find optimal RCs, they should eventually find the same RC in the ideal situations. However, in practice, they can only find approximations of RCs to their aspects. There

are many other methods for finding RCs such as time-lagged Independent Component Analysis (tICA) [32–34] or deep learning-based approach VAMPNets which project MD conformations onto a few collective variables for subsequent clustering [35–37]. Since timescale separation and slow mode finding are more related to the mechansim and performance of metadynamics, we use SGOOP as one of the key methods in my research, and we will revisit it with more details in Chapter. 2 and Chapter. 3.

1.3 Metadynamics: Overcoming timescale limitations

Once we have a good RC that has minimal memory and limits the timescale of rare events, we can perform an enhanced sampling technique called metadynamics [38, 39] to overcome the timescale limitations of MD simulations. In this section, we will briefly introduce metadynamics and show how we obtain useful information from metadynamics. In particular, we will focus on its most mature variant called "Well-tempered metadynamics (WTmetaD)" [38].

1.3.1 Well-tempered metadynamics

When performing metadynamics [38–40], we add the history-dependent Gaussian bias $V(\mathbf{s}, t)$ as a function of biasing variables $\mathbf{s}$ to encourage the system to visit new metastable states. Here specifically we use the well-tempered variant of metadynamics in which the height of the Gaussian is rescaled through a bias factor each time a point is revisited. This allows the bias to converge smoothly. The Gaussian bias $V(\mathbf{s}, t)$ can be

expressed as follows:

$$V(s,t) = \sum_{k\tau<t} h e^{-\frac{V(\mathbf{s},k\tau)}{\Delta T}} \exp\left(-\sum_{i}^{d} \frac{[s_i(t) - s_i(k\tau)]^2}{2\sigma_i^2}\right) \tag{1.1}$$

where $\mathbf{s}$ is a d-dimensional biasing variable and can be written as a function of phase space variables. The h is called initial height, σ_i is the width of Gaussian bias, τ is the time interval between each bias deposition, ΔT is a parameter more commonly written as the bias factor $\gamma = (T + \Delta T)/T$. Higher bias factor then represents a more aggressive manner of adding bias.

Since metadynamics accelerates the rare events and helps the simulations escape from free energy minima by filling up the minima with the Gaussian bias, it is crucial to choose the biasing variables $\mathbf{s}$ carefully if we would like metadynamics to be efficient in sampling the rare events. Especially in practice metadynamics becomes slower than unbiased simulations when we use more than two biasing variables. Therefore, we can use the optimal RC found by SGOOP as our biasing variables because the free energy barrier along its optimal RC minimizes the timescale of rare events not captured by the RC. It is then not hard to imagine adding bias can directly help climbing this barrier.

1.3.2 Reconstructing free energy surface

At the end of metadynamics, the bias potential fills up all free energy minima and converges. Therefore, the simulation becomes freely diffusive in the space of biasing variables. In theory, a relation connecting the free energy with the deposited bias can be derived irrespective of the precise choice of biasing variable $\mathbf{s}$, which will be asymptot-

ically valid at the limit of long simulation time. In practice, however, it helps if $\mathbf{s}$ is as close to the true RC as possible.

It can be shown that in the long time limit, the bias potential will smoothly converge to the rescaled free energy [38, 41]:

$$V(\mathbf{s}, t \to \infty) = -\frac{\Delta T}{T + \Delta T} F(\mathbf{s}) + \text{Constant} \tag{1.2}$$

where it can be seen that by choosing an appropriate ΔT, the bias potential converges to the free energy at temperature $T + \Delta T$. This free energy can also be obtained by calculating the probability mass function via direct counting, where it can be shown that the unbiased probability distribution $P(\mathbf{R})$ of atomic coordinates $\mathbf{R}$ can be obtained by reweighting through the following relation

$$P_M(\mathbf{R}, t) = P(\mathbf{R}) e^{\beta[V(\mathbf{s}(\mathbf{R}), t) - c(t)]} \tag{1.3}$$

where $\beta = 1/(k_B T)$, k_B is the Boltzmann constant, $P_M(\mathbf{R}, t)$ are probability distribution at time t in biased simulation, and the $c(t)$ is given by the following equation:

$$c(t) = \frac{1}{\beta} \log \frac{\int d\mathbf{s} \exp[\frac{\gamma}{\gamma - 1} \beta V(\mathbf{s}, t)]}{\int d\mathbf{s} \exp[\frac{1}{\gamma - 1} \beta V(\mathbf{s}, t)]} \tag{1.4}$$

1.3.3 Infrequent metadynamics

More recently, a simple extension to well-tempered metadynamics was introduced which allows recovering not just static free energies but also unbiased kinetic information

from metadynamics. This protocol has been dubbed "infrequent metadynamics". [41, 42] This method is based on hyperdynamics developed by Voter and Grubmüller [43, 44]. The key idea here is that as long as the bias deposition rate is infrequent enough compared to barrier-crossing timescales, in principle we should be able to reweight the biased timescales from well-tempered metadynamics directly to obtain unbiased kinetics through a simple acceleration factor:

$$\alpha(t) = \frac{\tau}{\tau_M} = \langle e^{\beta V(s(t)))} \rangle_b \tag{1.5}$$

where τ is the unbiased transition time we seek to learn, β is the inverse temperature, and τ_M is the biased transition time we actually observe in metadynamics. $V(s(t))$ is the net bias deposited until time t deposited on order parameter (OP) or reaction coordinate (RC) s. The subscript b means that the average is performed over the biased ensemble. The central assumption in infrequent metadynamics is that the biasing variable does a good job of timescale separation between time spent in the free energy basin and the time spent during barrier crossing. Thus, the need to have a more accurate RC for biasing, that satisfies the criteria of Sec. 1.1.3 becomes even more significant for infrequent metadynamics than in traditional metadynamics. For instance, as we will show in Chapter 2 section, infrequent metadynamics becomes more accurate if the biased variable includes all relevant slow modes with long autocorrelation time, and any hidden modes not considered in the biasing variable are as Markovian (or quickly decorrelating) as possible.

1.4 Limitations of 1-d RC

A systematic approach to overcome timescale limitation and memory effect of rare events is then to find optimal RCs, construct bias on this good RC, and perform metadynamics. Biasing metadynamics is computationally expensive. Therefore, it is a common practice to add bias on grid points when adding 2-dimensional bias although the grid also produces its own problems. Adding bias of three or more dimensions is therefore computationally too expensive to sample rare events. Due to the efficiency problems of metadynamics, we often seek a 1-dimensional RC as the biasing variable. It can also be imagined that infrequent metadynamics only works well for simple 1-dimensional RC. However, 1-dimensional RC can be very misleading for capturing correct kinetic connectivity. For example, in Fig. 1.2, the possibly best 1-dimensional RC we have shown not only projects the system to a free energy profile with incorrect free energy barriers but also displays incorrect connectivity between each state. If we look at such RC projection, we could incorrectly conclude that there is no direct transition from state 1 to 3 or from state 2 to 4. In Chapter. 3, we will show that in some systems the kinetic distance calculation gives incorrect kinetic connectivity if we calculate it on a 1-dimensional RC projection. We will also show how we reproduce the connectivity by calculating the kinetic distance with more RCs. Moreover, biasing along this RC will encourage or discourage unexpected pathways. As we have mentioned, it then becomes challenging to reweight kinetic rates from infrequent metadynamics.

In actual complex molecular systems, such multiple pathways, multiple processes, and complicated mechanisms can commonly exist. For example, Salvalaglio et al. have

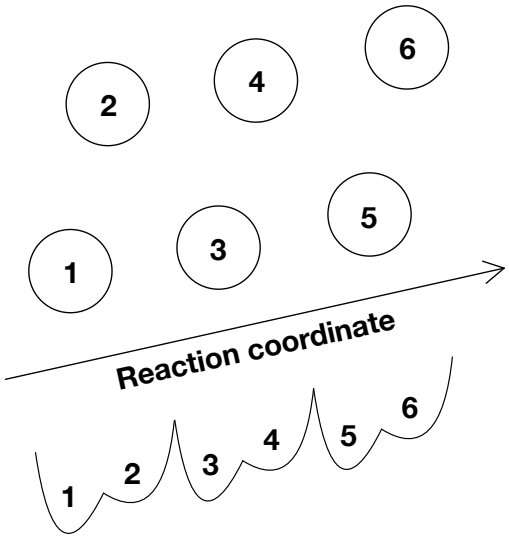


Figure 1.2: **Misleading projection onto reaction coordinate (RC).** Projecting 6-state to a 1-dimensional RC can be misleading. For example, the direct transitions between state 1 and 3 do not exist when projecting along the reaction coordinate. It is also hard to preserve the heights of potential barriers between states.

shown a simulation result that when urea nucleates from aqueous solution, a two-step nucleation mechanism is favorable [45]. In addition to simulations, Lee et al. have also reported experimental observations of multiple pathways of KH_2PO_4 (KDP) nucleation in aqueous solution [46]. For protein folding, Roca et al. have found both experimentally and computationally that the RNA pseudoknot, consisting of two hairpins with differing stabilities, can also fold by parallel pathways [47]. In such systems, it is then very hard to construct a 1-dimensional RC which could distinguish mechanisms along different pathways. For metadynamics, biasing just 1-dimensional RC can also produce physical mechanisms that are unexpected, incorrect, or misleading. In addition, there are situations where obtaining a good RC is simply impossible or not practical. For instance, in a single-molecule force spectroscopy experiment, the coupling to the external probe could induce memory when monitoring the protein motion with a reduced coordinate such as

the extension [48]. Finally, even though we project the systems to a good RC, there could still be memory effects coming from the ignored dimensions. This memory effect can appear as the long-term dependency within time series, where the existing dimensionality reduction techniques would simply fail, and we will really need to model the memory with a more powerful theory. We will discuss how we model this memory in more detail in Chapter. 4.

1.5 Outline of thesis

The rest of this dissertation is organized as follows:

• In Chapter 2, we revisit the classic problem of homogeneous nucleation of a liquid droplet in a supersaturated vapor phase. We consider this at different extents of the driving force, or equivalently the supersaturation, and calculate a reaction coordinate (RC) for nucleation as the driving force is varied. The RC is constructed as a linear combination of three order parameters, where one accounts for the number of liquid-like atoms, and the other two for local density fluctuations. The RC is calculated from biased and unbiased molecular dynamics (MD) simulations using SGOOP. Our key finding is that as the supersaturation decreases, the RC ceases to simply be the number of liquid-like atoms, and instead it becomes important to explicitly consider local density fluctuations that correlate with shape and density variations in the nucleus. All three order parameters are found to have similar barriers in their respective potentials of mean force, however, as the supersaturation decreases the density fluctuations decorrelate slower and thus carry longer memory. Thus at lower supersaturations density fluctuations are non-Markovian

and cannot be simply ignored from the RC by virtue of being noise. Finally, we use this optimized RC to calculate nucleation rates in the infrequent metadynamics framework, and show it leads to more accurate estimate of the nucleation rate with four orders of magnitude acceleration relative to unbiased MD.

• In Chapter 3, we propose a method that could be used to reconstruct interconversion time between two states in the systems with multiple pathways. As we have discussed in Sec. 1.4, when the systems have multiple pathways, multiple processes, and complicated polymorphism, projecting the dynamics onto a 1-dimensional RC can lead to incorrect kinetic connectivity. To deal with this issue, in this work we develop a formalism that learns a multi-dimensional yet minimally complex reaction coordinate (RC) for generic high-dimensional systems. When projected along this RC, all possible kinetically relevant pathways can be demarcated and the true high-dimensional connectivity is maintained. One of the defining attributes of our method lies in that it can work on long unbiased simulations as well as biased simulations often needed for rare event systems. We demonstrate the utility of the method by studying a range of model systems including conformational transitions in a small peptide Ace-Ala_3-Nme, where we show how two-dimensional and three-dimensional reaction coordinate found by SGOOP can capture the kinetics for 23 and all 28 out of the 28 dominant state-to-state transitions respectively.

• In Chapter 4, we will introduce one of the fundamental machine learning or artificial intelligence techniques called recurrent neural network (RNN). As we have also discussed in Sec. 1.4, even though we project the system to a good RC, arbitrary memory effect can still come into effect. Due to its recurrent structure, RNN then serves as a best tool for modeling such arbitrary memory in the complex systems. Here we show that

recurrent networks, specifically long short-term memory networks can also capture the temporal evolution of chemical/biophysical trajectories. Our LSTM model learns a probabilistic model of 1-dimensional stochastic trajectories generated from higher-dimensional dynamics. The model not only captures Boltzmann statistics but also reproduces kinetics across a spectrum of timescales. We demonstrate how training the long short-term memory network is equivalent to learning a path entropy, and that its embedding layer, instead of representing contextual meaning of characters, here exhibits a nontrivial connectivity between different metastable states in the underlying physical system. We demonstrate our model's reliability through different benchmark systems and a force spectroscopy trajectory for multi-state riboswitch.

• In Chapter 5, we propose a method to incorporate known physics into RNN. Often one wishes to supplement the experimentally observed dynamics with prior knowledge or intuition about the system. While the recurrent nature of these networks allows them to model arbitrarily long memories in the time series used in training, it makes it harder to impose prior knowledge or intuition through generic constraints. In this work, we present a path sampling approach based on principle of Maximum Caliber that allows us to include generic thermodynamic or kinetic constraints into recurrent neural networks. We show the method here for LSTM network in the context of supplementing time series collected from all-atom molecular dynamics. We demonstrate the power of the formalism for different applications. Our method can be easily generalized to other generative artificial intelligence models and to generic time series in different areas of physical and social sciences, where one wishes to supplement limited data with intuition or theory based corrections.

Chapter 2: Liquid droplet nucleation of Lennard-Jones particles

2.1 Introduction

The nucleation of one phase from another is considered as the first step of several phase transitions in chemical physics, with relevance to diverse and important problems in science and technology [3–11]. Through experiments, simulations and theory, this problem has been extensively studied over the decades [49]. In spite of so much attention being lavished upon this problem, it continues to be a difficult challenge. For instance, at experimentally accessible supersaturations, i.e., the ratio of the actual vapour pressure and the equilibrium vapour pressure [50,51], the characteristic critical nucleus size is so small that it becomes difficult to observe experimentally. The tens to hundreds of atoms size of the nucleus thus makes it in principle ideal for probing through molecular dynamics (MD) simulations. However, this is easier said than done due to the inherent rare event nature of the problem, where one nucleation event can take seconds, hours or longer, making it far beyond the microsecond timescale available through the fastest supercomputers. This has led to the development of a plethora of sampling schemes that attempt to enhance the process of nucleation in a controllable manner [38, 40, 52–57]. These various sampling methods need the pre-determination of slow degree or degrees of freedom relevant to the nucleation process being studied. This slow degree of freedom which is most informative

of the underlying physics is referred to as the reaction coordinate (RC) [25, 26, 58]. In sampling methods such as metadynamics [38, 39], where one gradually deposits a time-dependent bias to escape free energy minimum, the need to know a reasonably good RC beforehand is well-documented. In a different class of methods such as forward flux sampling (FFS) [56, 57], recent work has started to highlight how FFS can benefit from pre-knowledge of adequate slow order parameters or the RC [59]. Finally in methods such as transition path sampling (TPS) [60, 61] and variants thereof [54, 55], this dependence on pre-knowledge of RC is somewhat mitigated, but instead one becomes reliant on the accuracy of the initial path used in the sampling. In any case, one can say with confidence that any sampling scheme for the study of nucleation can only benefit from a prior sense of an approximate RC for nucleation, with the degree of benefit varying from scheme to scheme.

In this work, we consider what is arguably the simplest of nucleation problems, namely that of the homogeneous nucleation of a liquid droplet in a supersaturated vapor phase at different supersaturation levels [50]. The system is modeled using Lennard-Jones interactions [50, 62]. Even in this simplest of problems, we find that the RC for homogeneous nucleation deviates significantly from standard assumptions made so far in theoretical and simulation approaches [62–64]. Our calculations of the RC are performed using a spectral gap based optimization method "SGOOP", originally proposed by Tiwary and Berne [26, 31, 65, 66], for the automatic construction of RC from different trial order parameters. We find that there exists a supersaturation dependent interplay between size, density and shape of the nucleus. This interplay leads to a non-trivial RC that goes far beyond a spherical, uniformly dense nucleus assumed in classical nucleation

theory (CNT) [67–72]. While we define RC more rigorously in the main text, here we summarize it as a low-dimensional variable permitting a Markovian description of the underlying high-dimensional dynamics [25, 63, 73]. Our key finding is that as the supersaturation decreases, the RC becomes composed of not just the number of atoms in the largest liquid-like cluster, but it also becomes helpful to consider the spatial fluctuations of the aforementioned quantity. These fluctuations display similar barriers as the number of liquid-like atoms, but have a longer autocorrelation time (or equivalently, slower diffusion). This diffusion anisotropy becomes stronger as the supersaturation decreases. In the spirit of works by Szabo, Peters, Hynes and others [25, 74–77], we find that the RC itself starts to align with the direction of slowest diffusion or longest memory, given that the free energy barriers in the directions of various individual order parameters are similar. Finally, we use the optimized RC as a biasing variable in infrequent metadynamics calculations [42], which allow recovering unbiased kinetic rate constants from biased simulations. We find that considering this diffusion anisotropy adjusted RC in infrequent metadynamics leads to more accurate estimates of the nucleation rate across different supersaturations with orders of magnitude speed-up relative to unbiased MD.

This work demonstrates the potential of using methods such as SGOOP in unraveling the subtle aspects of the RC in complex nucleation problems. Such a RC first of all directly gives useful physical insight into the processes at play, but secondly, as we show here it also serves as useful descriptor for performing enhanced sampling simulations including metadynamics and beyond.

2.2 Classical nucleation theory and its limitations

In order to motivate this work and the various order parameters we consider here, we start with a brief description of CNT which has been a basic building block in the study of nucleation. In CNT, the first liquid droplet formed in the vapor is treated as spherical shaped and uniformly dense [67–72]. The nucleation process is then modeled by balancing the surface tension penalty with chemical potential benefit [78]. This simple theory, though it captures qualitatively how nucleation happens, however fails to quantify the true nucleation rate in any practical sense. It is believed that CNT makes several oversimplified assumptions especially incorrectly assuming that the cluster is spherical and uniform [79, 80]. By using numerical and experimental tools, the lack of sphericity and uniform density has indeed been documented in crystal nucleation and in nucleation in more complex systems [81–85]. However such simulations and experiments are expensive, and it has been hard to quantitatively probe such effects even in the simple gas system such as the one used in this work.

2.3 Theory and method

2.3.1 Order parameters

A popular order parameter that goes beyond the spherical nucleus approximation of CNT was introduced by Frenkel and ten Wolde [86]. This order parameter n equals the number of liquid-like atoms in the system in a way that it is still a continuous and differentiable function [86] of atomic coordinates, a necessity for the biased simulations

we perform later. In this definition, an atom is classified as liquid if it has more than 5 neighboring atoms. The number of the neighborhood atoms of the atom with label i, or equivalently the coordination number c_i, is calculated through the use of a switching function as follows:

$$c_i = \sum_{j \neq i} \frac{1 - (r_{ij}/r_c)^6}{1 - (r_{ij}/r_c)^{12}} \tag{2.1}$$

where the summation is carried over all atoms $j \neq i$. The distance r_{ij} between atoms i and j needs to be less than a cut-off r_c to be considered as neighbors. The number of liquid phase atoms n is then calculated using a similar form with threshold value c_l which we take to be 5 here in spirit of Ref. [86]:

$$n = \sum_{i=1}^{N} \frac{1 - (c_l/c_i)^6}{1 - (c_l/c_i)^{12}} \tag{2.2}$$

The above defined n thus captures the total number of liquid-like atoms in the system [86]. It is however oblivious to details such as the density of the clusters in which these atoms are present, if there are more than 1 clusters, the shape of these clusters and other nuances. In order to consider these, we propose including the second and third moments of the distribution of coordination numbers, defined as μ_2^2 and μ_3^3 respectively [83, 87, 88]. These are explicitly calculated as:

$$\mu_2^2 = \frac{1}{N}\sum_{i=1}^{N}(c_i - \bar{c})^2, \quad \mu_3^3 = \frac{1}{N}\sum_{i=1}^{N}(c_i - \bar{c})^3 \tag{2.3}$$

where $\bar{c}$ is the average of the coordination number. In Fig. 2.1 we show a representative

unbiased MD trajectory in (n, μ_2^2) space where it can be seen that at roughly the critical size of $n = 30$ different μ_2^2 can represent clusters with strikingly different profiles in terms of shape, density and compactness.

The distribution of the coordination numbers captures the variations in density of the liquid phase and thus can be used to study the local properties of the liquid droplets. Another nice feature of these moments is that they can easily be generalized for multi-component systems to take into account the variation in density related to specific species [81,84,89]. Our RC is then expressed as a linear combination of these three order parameters n, μ_2^2 and μ_3^3.

2.3.2 Nucleation rate

The process of nucleation is inherently stochastic in nature and satisfies the law of rare events. In other words, different independent observations of nucleation should give a distribution of nucleation times adhering to a Poisson process [90]. If we let $P(t)$ denote the survival probability of not observing any liquid droplets at and until time t, it will satisfy the following relation valid for all Poisson processes:

$$P(t) = e^{-t/\tau} \tag{2.4}$$

where τ is the characteristic time for the first nucleation event, the inverse of which can be interpreted as the nucleation rate. In this work, we find the characteristic time by performing multiple independent simulations starting from the system in gaseous state with randomized velocities (other simulation details in Sec. 2.3.6), and collect the statis-

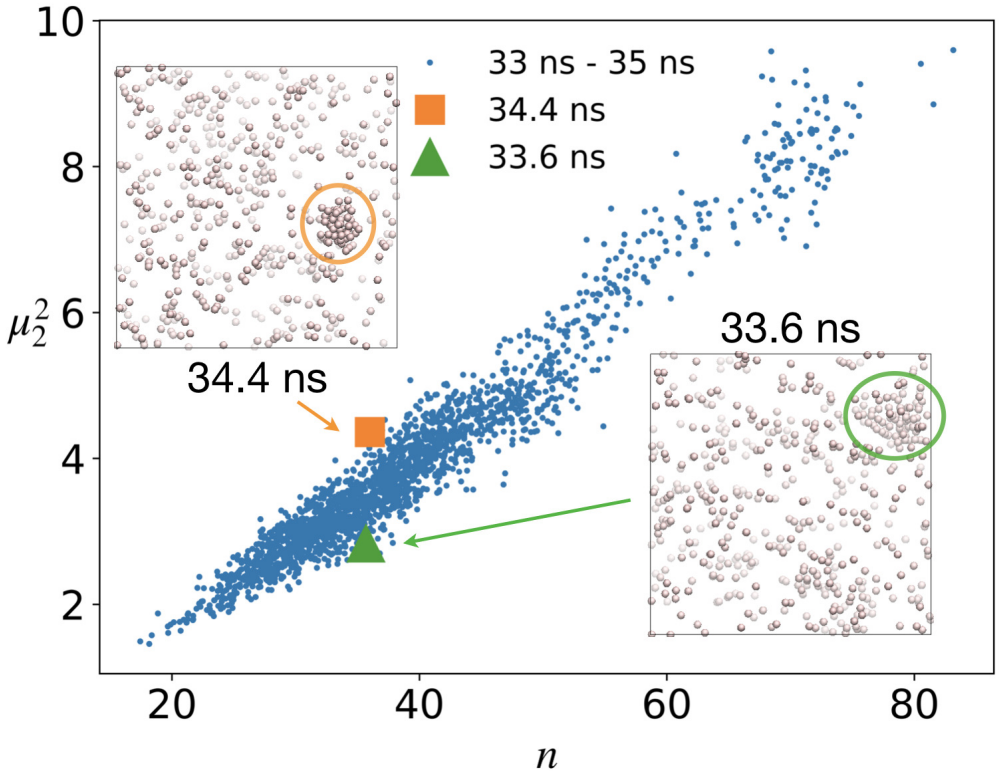


Figure 2.1: **Local density fluctuations of argon nuclei.** Here we show an unbiased MD trajectory in (n, μ_2^2) space at supersaturation 11.43, for a 2 ns interval between 33 ns to 35 ns. The panels are the snapshots at similar n but different μ_2^2, showing clearly how there can be clusters with same n but otherwise very different properties including density an compactness. For instance here, at higher μ_2^2, the cluster is visibly more compact than the one at lower μ_2^2.

tics of transition times until the first nucleation event. The nucleation time is then obtained by performing a Poisson fit to these independent observations following the protocol outlined in Ref. [62, 91].

2.3.3 Reaction coordinate

In order to quantify how these various order parameters n, μ_2^2, and μ_3^3 matter for the process of nucleation, we intend to learn a RC χ as their linear combination. In addition to quantifying exactly how much these order parameters matter for driving nucleation,

this RC will also serve as a crucial input for biased simulations to be performed later in this work. We first carefully define what exactly we mean by RC.

The RC for a given molecular system is traditionally defined as an abstract low-dimensional coordinate that best captures progress along relevant reaction pathway. While this intuitive notion can be formalized and quantified in several different ways, here we use the definition of RC as follows. For a given multidimensional complex system undergoing a certain dynamics, it is an optimal low-dimensional variable such that the multidimensional dynamics of the full system in terms of movement between different metastable states can be mapped into Markovian dynamics between various states viewed as a function of the RC [25,92]. Thus an optimal RC is a low-dimensional mapping which best satisfies (i) thermodynamic truthfulness: demarcating between the various relevant metastable states present in the actual high-dimensional system, (ii) kinetic truthfulness: preserving pathways for moving between these different states, and (iii) timescale separation: displaying a clean-cut separation of timescales between the relaxation times in the various metastable states, and the time spent in the actual event of crossing from one state to another.

2.3.4 SGOOP

To find such a RC, here we use the method "Spectral gap optimization of order parameters (SGOOP)" which we have briefly introduced in Sec. 1.2.2. This method uses the principle of maximum caliber ("MaxCal"), which is similar to path entropy [93–96], to construct a transition probability matrix along any candidate RC, and then calculates

its eigenvalues $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq ... \geq 0$. Here $\lambda_0 = 1$ corresponds to stationary state, while the other eigenvalues carry information about the timescales of various dynamical processes and all have to be real due to detailed balance. The best RC will then produce a transition matrix K with a maximal timescale separation between visible slow and hidden fast processes. This timescale separation, also known as spectral gap, is quantified as the difference $\lambda_n - \lambda_{n+1}$, where n is the number of discernible energy wells along the putative RC. SGOOP needs two key inputs: (i) an estimate of the stationary probability density π along any putative RC, and (ii) some dynamical observables or constraints. With these inputs, the SGOOP transition matrix K can be formulated as follows:

$$K_{mn} = \Lambda \sqrt{\frac{\pi_n}{\pi_m}} \tag{2.5}$$

where π_m is the stationary probability along any putative, spatially-discretized RC χ with m denoting the grid index, and Λ is a dynamical observable we will revisit shortly. K_{mn} gives the rate for moving from grid m to grid n in a small time interval. The input (i), namely the stationary density π can come from unbiased MD at high enough supersaturations, or if the supersaturation is too low to permit unbiased MD, it can come through the use of preliminary metadynamics along a trial RC, followed by reweighting [41]. For input (ii), namely calculation of the dynamic observable needed to constrain the maximum caliber estimate of rate matrix, we run short unbiased MD runs which calculate the mean number of nearest neighbor transitions $\langle N \rangle$ along any putative RC. It is then easy

to show [97] that the dynamical observable Λ in Eq. 2.5 is given by

$$\Lambda = \frac{\langle N \rangle}{\sqrt{\pi_m \pi_n}} \tag{2.6}$$

where

$$\langle N \rangle = \sum_{\substack{(m,n) \\ \forall |m-n|=1}} \pi_m K_{mn}$$

Equivalently [97], if one was to completely by-pass the MaxCal framework, a very similar equation as Eq. 2.5 can be derived by comparing a master equation along χ with a discretized Smoluchowski equation along the same. [98] Then the prefactor Λ becomes:

$$\Lambda = \frac{D_\chi}{2d^2} \tag{2.7}$$

where D_χ is the position-dependent diffusivity along the coordinate χ and d is the grid spacing along χ.

Eqs. 2.5–2.7 collectively show that the rate matrix K, hence the spectral gap, and consequently the optimized RC, depend not just on the free energy barriers that would be encapsulated in the stationary density π, or equivalently in the associated free energy, but that the dynamics of the system as captured in the diffusivity of the various order parameters can also play a significant role in the RC. As we will show later in Sec. 2.4, we find this to be a very important point in the context of liquid droplet nucleation.

WTmetaD parameters							
Label	S	L (nm)	h (kJ/mol)	ω_n	ω_χ	Δt (ps)	γ
S_1	13.65	9.9	0.01	0.5	0.08	25	5
S_2	12.80	10.1	0.05	0.5	0.08	25	8
S_3	11.43	10.5	0.2	0.5	0.08	25	8
S_4	9.87	11.0	0.2	0.5	0.08	25	8
S_5	9.04	11.3	0.2	0.5	0.08	25	8

Table 2.1: **Metadynamics parameters used at different supersaturation levels.** The metadynamics parameters used at different supersaturation levels: S represents the supersaturation level, L is the size of simulation cubic box which we used to control the supersaturation. Gaussian bias kernels of starting height h and width ω were added every Δt, which was kept same for metadynamics irrespective of free energy or kinetics calculation. γ is the bias factor for well-tempered metadynamics [39].

2.3.5 Metadynamics

For high enough supersaturation such as S_1, S_2, S_3 in Table. 2.1, we can perform unbiased simulations directly in reasonable computer time, both for the calculation of nucleation kinetics and for feeding stationary density into SGOOP for constructing the RC. However, for lower supersaturations we need to apply enhanced sampling methods since nucleation becomes a rare event. In this work, we use well-tempered metadynamics [38,39] along a trial RC to obtain preliminary stationary density estimates, and infrequent metadynamics [41,42] to calculate the kinetics of nucleation.

The different parameters of the Gaussian bias are listed in Table. 2.1 in which h is the starting height, ω is the width, Δt is the deposition interval, and γ is the bias factor. The final output of a traditional metadynamics run is the free energy along the variable s or along any other degree of freedom which can be expressed as function of the atomic coordinates of the system.

As we mentioned in Chapter. 1, we will calculate nucleation rate using "infrequent metadynamics" [41, 42]. It assumes that the bias deposition rate is infrequent enough compared to barrier-crossing timescales, allowing us to be able to reweight the biased timescales from well-tempered metadynamics to obtain unbiased timescales through the acceleration factor:

$$\alpha(t) = \frac{\tau}{\tau_M} = \langle e^{\beta V(s(t))} \rangle_b \tag{2.8}$$

where τ is the unbiased transition time we seek to learn, β is the inverse temperature, and τ_M is the biased transition time we actually observe in metadynamics. $V(t)$ is the net bias deposited until time t, where the bias is constructed as a Gaussian function of RC s. The central assumption in infrequent metadynamics is that the biasing variable does a good job of timescale separation between time spent in the free energy basin and the time spent during barrier crossing. Thus the need to have a more accurate RC for biasing, that satisfies the criteria of Sec. 2.3.3 becomes even more significant for infrequent metadynamics than in traditional metadynamics. For instance, as we will show in the Results section, infrequent metadynamics becomes more accurate if the biased variable includes all relevant slow modes with long autocorrelation time, and any hidden modes not considered in the biasing variable are as markovian (or quickly decorrelating) as possible. Here we learn such a 1-dimensional RC χ as a linear combination of our order parameters. Our bias potential then becomes $V(\chi, t) = V(w_1 n + w_2 \mu_2^2 + w_3 \mu_3^3, t)$. The weights of the different order parameters (w_1, w_2, w_3), are determined with SGOOP [99].

2.3.6 Model set-up and simulation details

The simulations were performed under the constant number, volume, temperature (NVT) ensemble with N=512 argon atoms and average temperature fixed at 80.7 K. Although the isochoric condition allows us to compare our results with previous simulation works [51], it should be mentioned that this is different from most actual experiments which are performed under isobaric conditions. The volume of the simulation box was set in order to correspond to desired supersaturation levels S detailed in Table. 2.1. The supersaturation is computed as the ratio of the actual vapour pressure p and the equilibrium vapour pressure p_e. The actual vapour pressure was calculated through the thermodynamic relation $p = 2E/3V$, where E is the kinetic energy of the system, while the equilibrium vapour pressure of argon at our range of supersaturation levels is equal to 0.43 bar [50, 62]. In order to compare our results with unbiased nucleation rates in Ref. [50], our cubic box size ranged from 9.5 nm to 11.5 nm. The interaction between atoms were modeled through a Lennard-Jones potential with $\epsilon = 0.99797$ kJ/mol and $\sigma = 0.3405$ nm [50]. The potential was truncated with cutoff at 6.75 σ. The velocity rescale thermostat with time constant of 0.1 ps was used to do temperature coupling. [100] All simulations were performed using GROMACS version 2016.5 [101] patched with PLUMED version 2.4.2 [102].

2.4 Results

2.4.1 RC predicted from SGOOP

We first describe the RC, as introduced and defined in Sec. 2.3.3, that we identify for the condensation of a liquid droplet across different supersaturation values. To learn this RC we have used SGOOP [26,65], with the stationary probability density π estimated through unbiased MD at high supersaturations S_1, S_2 and preliminary metadynamics [39] at low supersaturations S_4, S_4, S_5. The preliminary metadynamics runs were performed biasing n. All runs were complemented with short unbiased MD runs (50 ns) for obtaining dynamical constraints for MaxCal. For every supersaturation, SGOOP is initiated from a given choice of trials weights (w_1, w_2, w_3) for the RC χ expressed as $\chi = w_1 n + w_2 \mu_2^2 + w_3 \mu_3^3$.

A key question that immediately arises is whether at any given supersaturation S there is a unique RC, or if there are multiple possible combinations of the weights (w_1, w_2, w_3) which meet equally well the criteria for an optimal RC described in Sec. 2.3.3. Yet another question which we ask and answer is how transferable is the RC learnt at one supersaturation S across different values of S. To answer the first question, we perform several exhaustive SGOOP trials to estimate the optimized RC, first in the 2-d (n, μ_2^2) space where we do an explicit grid based search over the full space, and then in the 3-d (n, μ_2^2, μ_3^3) space where we start SGOOP from different initial weights. In the latter case, the optimization over weights in SGOOP is performed using a basin hopping algorithm which is a global search algorithm with several stochastic jumps aiding the system from

not getting trapped in local minima.

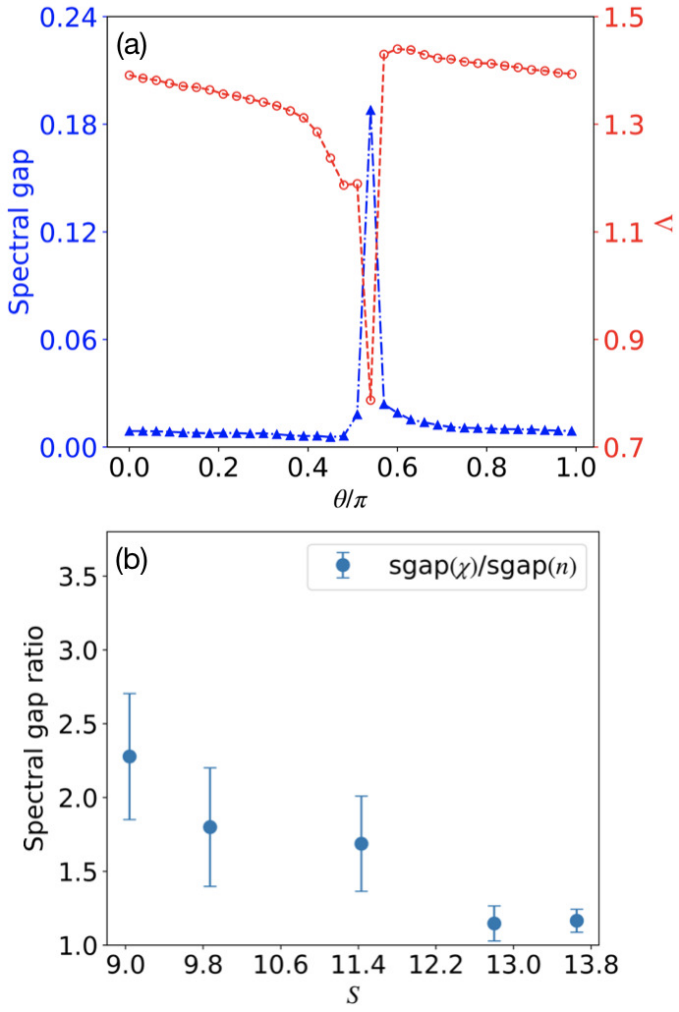


Figure 2.2: **The spectral gaps of the optimized reaction coordinates used in argon simulations.** (a) The spectral gap (sgap) (blue asterisks, left axis) and dynamical prefactor Λ (red circles, right axis) of SGOOP transition rate (Eq. 2.5) along different RC $\chi \equiv \cos(\theta)n + \sin(\theta)\mu_2^2$. Both the maximal spectral gap and minimum Λ take place at $\theta = 0.5\pi$. (b) Mean spectral gap ratio at five different supersaturation levels S_1–S_5: For each supersaturation, we averaged the spectral gap ratios calculated from 20 independent biased runs, and the error bars represent the standard error from the averaged results.

In the 2-d $\chi = w_1 n + w_2 \mu_2^2$ optimization, we do an explicit search among all possible RCs by rotating the putative RC $\chi \equiv \cos(\theta)n + \sin(\theta)\mu_2^2$ in the (n, μ_2^2) space. Here as shown in Fig. 2.2(a) for S = 11.43, we find that the spectral gap profile has a sharp peak when the RC is almost exclusively comprised of μ_2^2, i.e. $\theta \approx \pi/2$ and μ_2^2 has around 8

times higher weight in the RC than n. Such a μ_2^2–heavy RC is obtained irrespective of any S value, showing unequivocally that the second moment μ_2^2 plays a more important role in the RC than n itself. Fig. 2.2(a) also shows the variation of the kinetic pre-factor Λ of Eq. 2.5 with RC choice, and we will revisit this profile in Sec. 2.4.2. Next, we perform optimization in the full 3-d (n, μ_2^2, μ_3^3) space. Here we find that there are many combinations (w_1, w_2, w_3) with similarly enhanced spectral gaps relative to the traditional choice of $\chi = n$, but with a common theme that the second moment and the third moment consistently show up in the optimized RC.

Thus to summarize so far: (a) RC optimization in (n, μ_2^2) leads to RC predominantly comprised of μ_2^2, (b) RC optimization in (n, μ_2^2, μ_3^3) leads to a RC invariably with weights for all 3 variables, but with multiple local maxima in the spectral gap profile. In other words, the RC is quite degenerate, but considering μ_2^2 and μ_3^3 in the RC is important for a more accurate description of the nucleation process. In Fig. 2.3 we also show results from a full grid search over spectral gaps in the (w_1, w_2, w_3) space at at S = 11.43 further illustrating the findings from SGOOP. Here among the first few largest local maxima from 3 different trajectories, we picked $(w_1, w_2, w_3) = (0.15, 0.65, -0.15)$ for use in further calculations across all supersaturations S. In Fig. 2.2 (b), we plot the ratio between the spectral gap along RC = $0.15n + 0.65\mu_2^2 - 0.15\mu_3^3$ and that along RC = n at different S. As can be seen there, at all S values the optimized RC gives higher spectral gaps than just n, and the improvement increases sharply with decreased supersaturation. That is, as the supersaturation decreases the importance of consider shape and density fluctuations in the nuclei become more and more important, which is one of the central findings of this paper. Furthermore, the optimized RC learnt at one supersaturation gives improved spectral gaps

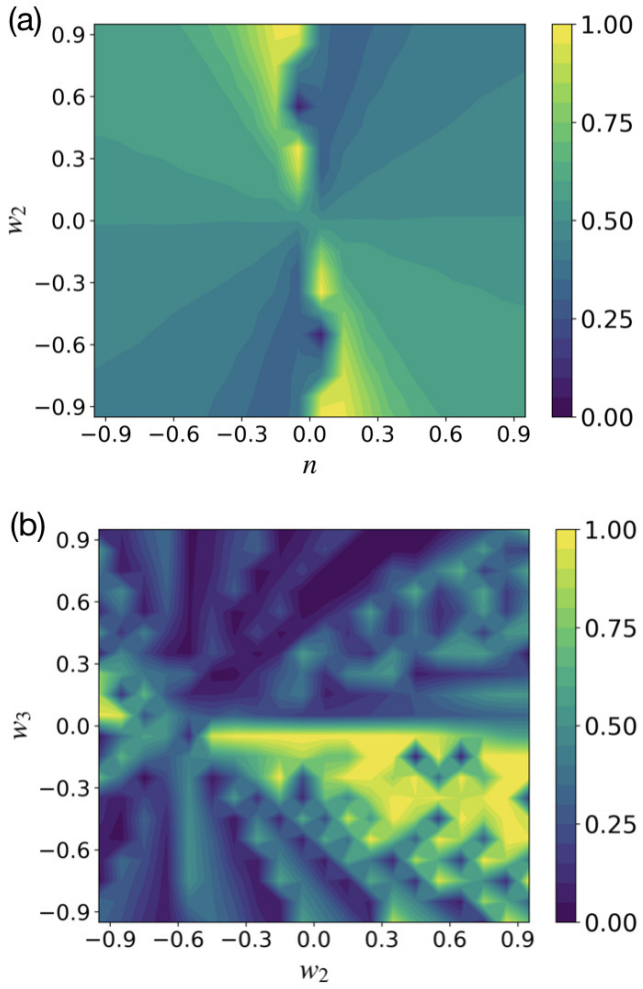


Figure 2.3: **The contour plot of brute force search of spectral gap values.** Spectral gap values evaluated given RC $\chi = w_1 n + w_2 \mu_2^2 + w_3 \mu_3^3$ in the space of weights w_1, w_2, and w_3: (a) The contour plot of spectral gaps as a function of (w_1, w_2) with $w_3 = 0$. (b) The contour plot of spectral gaps as a function of (w_2, w_3) with $n = 0.05$. The spectral gaps were computed using biased trajectory at supersaturation $S = 11.43$. The colorbar in each figure represents the spectral gap values.

at other supersaturations, and hence the RC is transferable across supersaturations. Thus in Sec. 2.4.3 we use the RC $\chi = 0.15n + 0.65\mu_2^2 - 0.15\mu_3^3$ at all supersaturations for enhanced sampling based calculations of the nucleation rate.

2.4.2 Understanding the RC

SGOOP optimizes the RC by finding a low-dimensional projection with highest gap between slow and fast processes. In most cases this amounts to selecting a projection with the highest barrier separating the metastable states. To understand if the RC learnt in Sec. 2.4.1 can be attributed to simply barriers in the free energy profile, or if dynamical concerns such as the prefactor Λ in Eqs. 2.5–2.7 play a role, we construct free energies along various 1-d and 2-d components (totaling 6 combinations) of (n, μ_2^2, μ_3^3). These free energies were obtained by running metadynamics with same parameters defined in Table. 2.1 and bias potentials added along n. We then averaged over 10 independent metadynamics runs with each trajectory reweighted using the free estimator described in Ref. [41].

From the various 1-d and 2-d free energy profiles shown in Fig. 2.4 (a)-(f) for S = 11.43, it is hard to distinguish between the importance of the various order parameters n, μ_2^2,and μ_3^3. The 2-d profiles show that starting from the gas phase (red stars in Fig. 2.4 (a)-(c)), all three order parameters change in a very correlated manner until the barrier is reached and nucleation is essentially complete (n >100). The 1-d free energies along the three order parameters (Fig. 2.4 (d)-(f)) show that the free energy barrier that needs to be overcome is also very similar for each of the 3 order parameters, though there are some systematic differences which we revisit shortly in Fig. 2.4 (g)-(i). Comparing the 1-d free energy along μ_3^3 (Fig. 2.4 (f)) with the corresponding 2-d free energies (Fig. 2.4 (b)-(c)), we can see that unlike n and μ_2^2, the 1-d projection along μ_3^3 does a very poor job of describing the pathway in higher dimension space, further justifying our choice

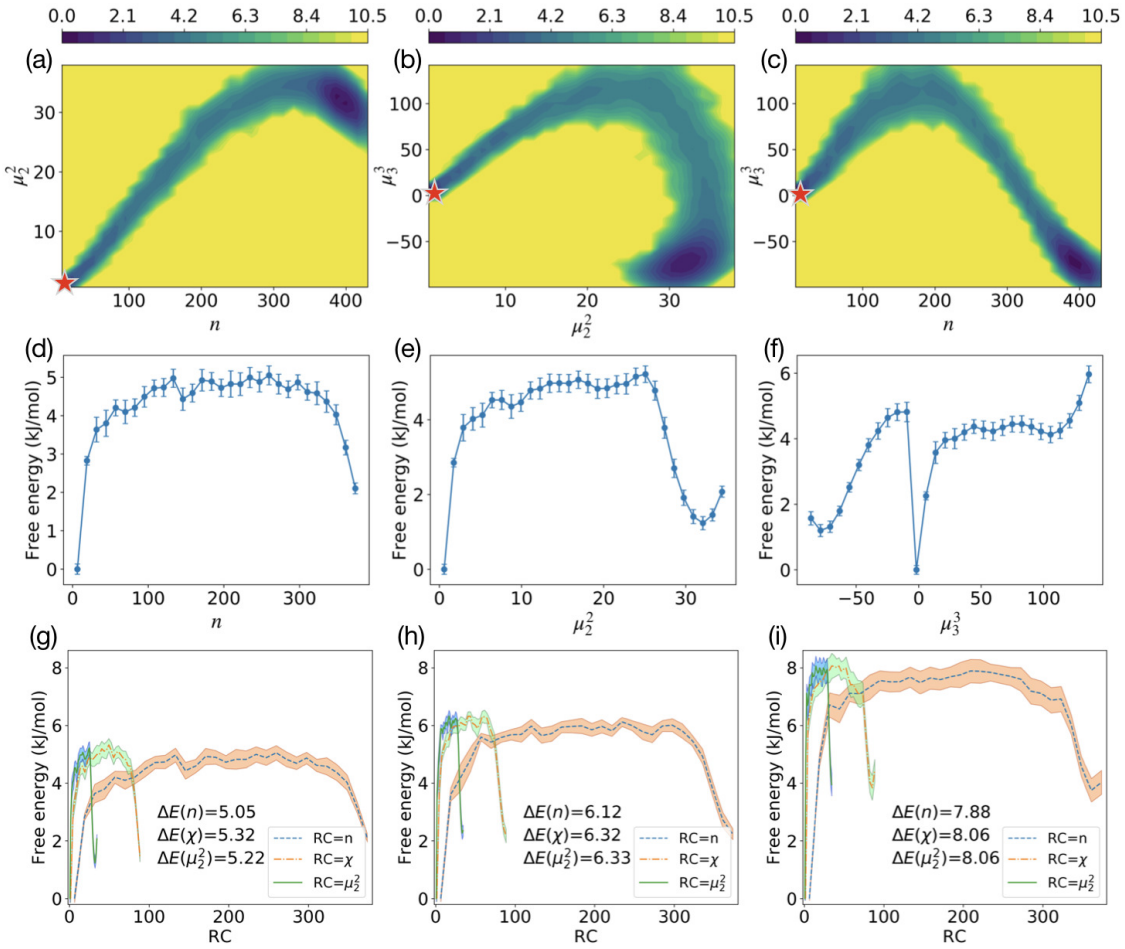


Figure 2.4: **The free energy plots obtained from well-tempered metadynamics.** The free energy plots obtained from well-tempered metadynamics biasing along n: The top three panels are the 2-d free energy surfaces of (a) (n, μ_2^2), (b) (μ^2, μ_3^3), and (c) (n,μ^3) at supersaturation S_3. The starting gaseous state corresponding to each plot is shown with a red star. The middle three panels show the 1-d free energy curves along (d) n, (e) μ_2^2, and (f) μ_3^3 respectively. The profiles and the errorbars are calculated from the averages over 10 independent metadynamics runs at supersaturation $S = 11.43$. The bottom three panels display the 1-d free energy curves from (g) S=11.43, (h) S=9.87, and (i) S=9.04. In each panel, we show the profile averaged over 10 independent metadynamics runs along n, χ, and μ_2^2. The regions between errorbars are filled. The corresponding energy barriers $\Delta E(RC)$ along three different putative RCs are also shown. It can be seen that as S decreases the barrier difference decreases. All energies are in units of kJ/mol.

of RC χ in the previous section with higher weight for μ_2^2 than for μ_3^3. In Fig. 2.4 (g)-(i), we show the free energies along three different RC choices, namely n, μ_2^2 and the optimized $\chi = 0.15n + 0.65\mu_2^2 - 0.15\mu_3^3$, for three different S values. As S is decreased,

invariably there is a small but consistent improvement in the barrier height when viewed as function of χ or μ_2^2, relative to when viewed as function of n. However, firstly this difference is very small (0.25 kJ or 0.1 k_BT), and secondly, it appears to get even smaller with decreasing S (Fig. 2.4 (g)-(i), left to right). Thus, the free energy barrier can not be used to explain the behavior of spectral gap versus supersaturation shown in Fig. 2.2 (b). Here we showed that at all supersaturation levels we considered, the spectral gaps of the optimized RC are better than of n. It was also pointed out that as supersaturation decreases the spectral gap improvement increases. This tells us that the optimized RC works better at lower supersaturation, which is inconsistent with the change in free energy barriers along different order parameters with supersaturation.

Our next step is therefore explaining why SGOOP finds that μ_2^2 has a role to play in χ, and why the advantage in considering μ_2^2 increases with decreasing supersaturation S. In Fig. 2.2(a), we provided a profile of how the prefactor Λ varied with the RC choice and correspondingly with the spectral gap. It can be seen there that the prefactor Λ has a strong inverse correlation with the spectral gap of χ – the maximum spectral gap coincides with minimum Λ. Thus the minuscule increase in barrier height with varying RC is compensated by the slowness of the dynamics along the RC, as captured by Λ or the average number of first neighbor transitions in a unit time.

To gain further insight into this, we calculated time-autocorrelation functions along our three different order parameters (see Fig. 2.5) as higher autocorrelation time represents slower diffusivity. Our calculations show that μ_2^2 and μ_3^3 have longer autocorrelation times than n, and therefore lose memory slower than n [63,64]. Furthermore, the increase in autocorrelation times of the two order parameters μ_2^2 and μ_3^3 relative to n becomes more

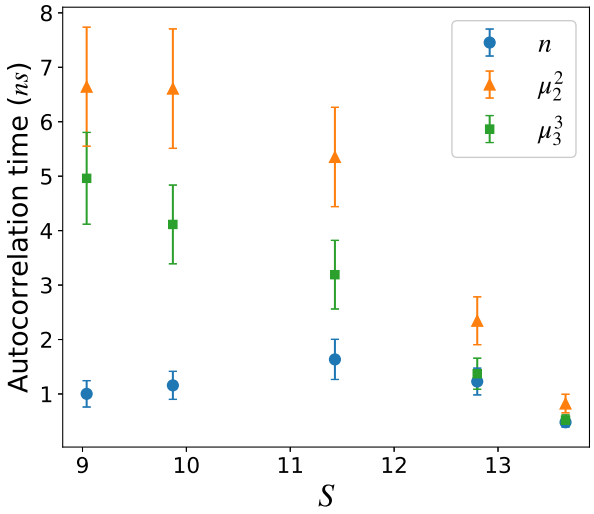


Figure 2.5: **The mean autocorrelation times of different order parameters.** The mean autocorrelation times of the order parameters n (blue circles), μ_2^2 (orange triangles), and μ_3^3 (green squares) calculated from unbiased MD simulations at five different supersaturation levels S_1–S_5. At each supersaturation level, the calculations from 10 independent runs are averaged. The error bars show the standard error of the averaged results.

and more pronounced as the supersaturation S decreases (Fig. 2.5). This is in striking contrast to Fig. 2.4, where we found an opposite trend looking at the free energy barriers along these order parameters.

We therefore conclude this section with the observation that anisotropic diffusion in the space of order parameters becomes an important factor in determining the RC especially at lower supersaturation levels. The longer autocorrelation times are linked to less Markovian behavior, which means μ_2^2 and μ_3^3 carry longer memory than n [63, 64]. Coupled with the finding that all three order parameters have similar barriers in their respective potential of mean force, this means that change in nuclei characteristics such as shape and density become slower as supersaturation S decreases, and it becomes important to explicitly consider this in the construction of a Markovian RC.

Here we would also like to highlight past work by Peters [77] which applied a theoretical model to the study of the interplay between concentration fluctuations and nucleation processes in multicomponent systems. While that work did not compute rates, as we do in the next section here, and also made stringent assumptions such as a radially symmetric concentration profile, our key findings here are similar to theirs. Namely, that in this work, shape variations can drive or inhibit a nucleus from going into the second phase, while in their case [77], certain types of concentration profiles can drive a classically pre-critical nucleus over the nucleation barrier.

2.4.3 Nucleation kinetics

Now that we have identified an optimized RC $\chi = 0.15n + 0.65\mu_2^2 - 0.15\mu_3^3$ with improved spectral gap relative to the Frenkel-ten Wolde parameter n, we perform two sets of enhanced sampling simulations (specifically, infrequent metadynamics) using n and χ as biasing variable respectively. We use Eq. 2.8 to reconstruct the unbiased timescale estimates from these biased runs. At high enough supersaturations we are able to run unbiased MD as well and together with the results of Reguera et al [50] these constitute a valuable set of results to benchmark our findings against. At each supersaturation level, we launched 40 independent metadynamics runs with 20 of them biasing n and the other 20 biasing the optimized RC χ. For each independent run, in order to be able to compare our results with previously published work [50, 62] we defined the nucleation event as when the number of liquid-like atoms n reaches 30 for the first time. Every independent observation of such an event in terms of its metadynamics time was scaled by the accel-

S	RC	$\tau_N(s)$ $[p-value]$	$J(1/\mathrm{cm}^3/\mathrm{s})$	α
S_1	n	$4.16 \pm 0.45 \times 10^{-9}$ [0.17]	$2.48 \pm 0.27 \times 10^{26}$	1.1
	χ	$4.56 \pm 0.31 \times 10^{-9}$ [0.67]	$2.26 \pm 0.15 \times 10^{26}$	1.1
S_2	n	$8.66 \pm 0.85 \times 10^{-9}$ [0.47]	$1.12 \pm 0.11 \times 10^{26}$	2.0
	χ	$7.88 \pm 0.51 \times 10^{-9}$ [0.37]	$1.23 \pm 0.08 \times 10^{26}$	1.6
S_3	n	$1.00 \pm 0.15 \times 10^{-7}$ [0.87]	$8.64 \pm 1.30 \times 10^{24}$	4.0×10^1
	χ	$0.47 \pm 0.07 \times 10^{-7}$ [0.35]	$1.84 \pm 0.27 \times 10^{25}$	6.4×10^1
S_4	n	$1.26 \pm 0.27 \times 10^{-6}$ [0.64]	$5.96 \pm 1.28 \times 10^{23}$	3.7×10^2
	χ	$0.69 \pm 0.12 \times 10^{-6}$ [0.80]	$1.09 \pm 0.19 \times 10^{24}$	1.1×10^3
S_5	n	$1.58 \pm 0.19 \times 10^{-5}$ [0.59]	$4.32 \pm 0.52 \times 10^{22}$	5.7×10^3
	χ	$0.62 \pm 0.15 \times 10^{-5}$ [0.13]	$1.10 \pm 0.27 \times 10^{23}$	7.8×10^3

Table 2.2: **Characteristic nucleation times, nucleation rates, and the corresponding mean acceleration factors of different RCs at different supersaturation levels.** The table shows the characteristic nucleation times τ_N by fitting Eq. 2.4 and the corresponding nucleation rates J. Results are shown as obtained from the simulations biasing along n as well as biasing along the optimized RC χ. The labels corresponds to the supersaturation levels denoted in Table. 2.1. α is the mean acceleration factor for every set of simulations. For the fitted characteristic nucleation times τ_N we have also provided in square brackets the corresponding p-value of the fit when used in Kolmogorov-Smirnov test of Ref. [91].

eration factor (Eq. 2.8) to obtain an unbiased observation of the nucleation time. With these 20 independent estimates of the nucleation time, we can compute the characteristic time (Eq. 2.4) of observing the first nucleation event τ_N by fitting a Poisson distribution to the statistics, where τ_N is the expected value of the fitted Poisson distribution. The corresponding nucleation rates are then calculated through the formula $J = 1/(\tau_N V)$ and $J = 1/(t_N V)$ where V is the volume of system. The results are shown in Table. 2.2 and in Fig. 2.6.

We find that the use of n as a biasing variable in infrequent metadynamics does a remarkably decent job of obtaining nucleation rates (in agreement with the findings of Ref. [62]) even with very significant acceleration factors or computational boost relative to unbiased MD. There is nonetheless further improvement of up to three times that can

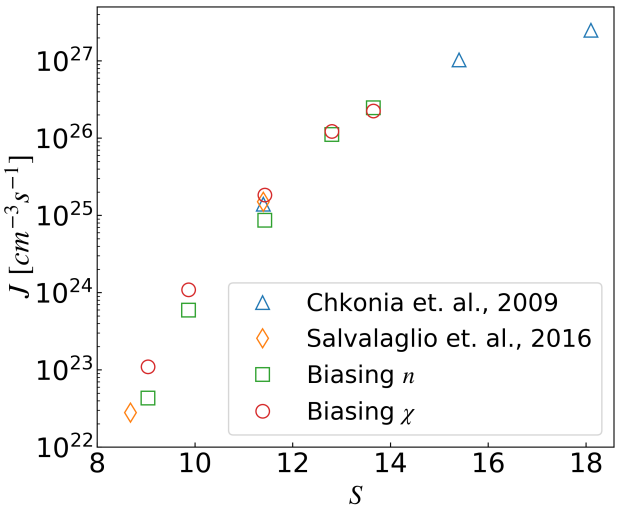


Figure 2.6: **The nucleation rates obtained from infrequent metadynamics at different supersaturation levels.** The nucleation rates calculated from the Poisson fits of reweighted nucleation times obtained from infrequent metadynamics biasing along n and χ (green squares and red circles respectively). The values and their associated error bars are listed in Table. 2.2. We also compare our results with previous works from Ref. [50] and Ref. [62] (blue triangles and red diamonds respectively).

be obtained in the quality of the nucleation rate if the optimized RC is used instead of n, especially as the supersaturation is decreased. In a field where errors can be as high as twenty six orders of magnitude [80], improvement of three times seems minuscule, reflecting that n is after all not that bad of a biasing variable for infrequent metadynamics. Yet, even though the improvement is relatively small compared to the usual standards in nucleation kinetics, it is systematic, robust and indicative of possible usefulness when employed in more complex systems with different competing variables, including but not limited to composition fluctuations [77]. As can be seen from Table. 2.2, the acceleration factor in metadynamics relative to unbiased MD increases steadily as S decreases, reaching almost four orders of magnitude at the lowest S. All reweighted nucleation times, irrespective of whether they came from biasing n or biasing χ give p-values above the

recommended cut-off in the Kolmogorov-Smirnov test from Ref. [91]. At $S = 11.43$, the use of χ as biasing variable instead of n leads to much better agreement with the unbiased estimate of Reguera et al [50], as can be seen in Fig. 2.6. In general, the characteristic times for nucleation from runs biasing the RC χ are significantly lower than that those from biasing n, and roughly speaking this difference increases as S decreases. In addition to the explicit agreement with unbiased estimate of Reguera et al [50] at $S = 11.43$, the lower characteristic time (with similar p-values) can be seen as further evidence of the benefit of biasing χ instead of n. This is because in metadynamics the presence of missing slow degrees of freedom from explicit consideration in the biasing variable typically leads to hysteresis during free energy calculations, or overestimate of the accelerated time through Eq. 2.8, as pointed out in Ref. [91] and Ref. [103].

2.5 Conclusion

In this work, we used new tools [26, 42] to revisit a classic problem in nucleation, namely that of the formation of liquid droplet from gaseous precursor as a function of varying driving force for nucleation, namely supersaturation. Our interest was in (a) constructing a Markovian reaction coordinate (RC) for this process, and (b) testing if there is any gain to be had through the use of a more Markovian RC in enhanced sampling calculations of nucleation kinetics. To answer these questions especially at low supersaturations where access to unbiased trajectories of nucleation is difficult (needed by many other RC optimization methods such as Ref. [104] and Ref. [105]), we use the spectral gap optimization method from Ref. [26] to construct optimized RC from input biased simula-

tions. Our calculations demonstrate unequivocally that it is not sufficient to consider only the typical order parameter used to describe nucleation, namely the number of liquid like atoms in the system. By considering further variables that account for heterogeneity in the system, such as higher moments μ_2^2 and third moment μ_3^3 of the distribution of coordination numbers, we could obtain a much more Markovian RC. Interestingly these various order parameters have nearly identical free energy barriers, and they differ primarily only in associated diffusivities. The importance of these variables further increases with decreasing supersaturation as their associated autocorrelation time increases sharply. In other words, shape and density fluctuations in the nucleating clusters cease to stay rapidly equilibrating variables which can be entirely ignored from a Markovian low-dimensional description of nucleation. We conclude that diffusion anisotropy plays a more important role at lower S, which is supported by our analysis of autocorrelation functions and autocorrelation times. While previous work has demonstrated how infrequent metadynamics can predict nucleation time with only n as the RC, we show in this work that the prediction of nucleation time can be further improved by biasing along an optimized RC. It will be interesting to see if the use of such a more Markovian RC makes improvement in the reliability and efficiency of other enhanced sampling methods such as forward flux sampling. One additional important comment we would like to make here is that while the RC was found to be increasingly more complex as the supersaturation was brought down, there is no guarantee that this trend will continue monotonically as the supersaturation is further decreased. Indeed, in a general setting the rate k for an activated process depends on the diffusivity D only in the pre-exponential but on the free energy barrier ΔG in the exponentiated term, i.e. $k = De^{-\beta \Delta G}$. As the supersaturation decreases, we expect at

some point the increase in nucleation barrier will be so significant that any manifestations of diffusion anisotropy will be washed out, and classical nucleation theory will again start to take hold as has been pointed out for instance by Binder. [106] Our supersaturation values in this work however did not reach this regime.

Finally, it should be mentioned that in this calculation we didn't consider effects due to the finite size of the system, which can be done using the method proposed in Ref. [62], but was not the main objective here. We realize that our findings here might simply be a finite size effect, resulting from the coupling between fluctuations in the density of the parent phase and fluctuations in the size of the growing nucleus of the product phase, which are inherently coupled due to the overall material balance in the simulation box. To check whether our findings might indeed be valid in the thermodynamic limit, in future work we will explore these simulations at different box sizes and with different supersaturation levels. Similarly our findings might change with constant number, pressure and temperature (NPT) simulations. The present work can be redone taking these important nuances into account. Finally, strictly speaking ours was a model system with model parameters. This work is a proof of principle that ideas such as SGOOP for RC optimization are potentially useful for study of nucleation through enhanced sampling or otherwise. In future we will be extending this work to systems such as crystal nucleation, multiple polymorphs, systems with multiple pathways or multiple species, where there will be even more order parameters to be considered. All of these continue to be very difficult yet important problems for understanding nucleation pathways and rates, and we are hopeful our tools will allow us and others to systematically study these.

In the next chapter, we will further discuss the systems where 1-dimensional RC

is simply not sufficient to reproduce correct kinetic connectivity. In such systems, using infrequent metadynamics to reproduce all pathways is not possible. We will first discuss existing methods that have been developed to solve this problem and their limitations in studying rare events. We will then introduce our recently developed methods which could be used to improve the existing methods and applied to reconstruct rare event kinetics.

Chapter 3: SGOOP-based kinetic distance

3.1 Introduction

It has been a problem of longstanding theoretical and practical interest to model reaction pathways and transition mechanisms in generic chemical and biological systems [2, 14, 46, 47, 107–110]. Due to recent progress in high-performance computing, brute-force Molecular Dynamics (MD) simulations with all-atom resolution have enabled a possible way to do such analysis in femtosecond temporal and all-atom spatial precision, making it a useful tool for studying diverse phenomena. However, this leads to a deluge of data resulting from explicit enumeration of all atomic coordinates over a very large number of MD timesteps. To make sense of such high-dimensional trajectories resulting from MD, it is a common practice to project them along low-dimensional coordinates identified with one of many dimensionality reduction schemes [111–114]. However, more often than not in such schemes, one ends up losing the kinetic connectivity of the high-dimensional landscape. This can thus lead to incorrect interpretation of MD trajectories, for example making molecular conformations appear closer to each other than they are and obfuscating interconversion pathways between them [115].

In this work, we develop a formalism that learns a multi-dimensional yet minimally complex reaction coordinate (RC), such that when projected along this RC, all possible ki-

netically relevant pathways can be demarcated and the true high-dimensional connectivity is maintained. The central idea is to calculate the interconversion times between different pairs of metastable states, which can be defined *a priori* or learned on-the-fly [116], and monitor how these distances change by adding additional dimensions to the RC. The procedure is stopped when the interconversion times do not vary with additional RC components. The interconversion times are calculated using the commute distance framework proposed by Noé, Clementi, and co-workers [117, 118]. While such a kinetic or commute distance-based procedure is indeed already recommended best practice in the construction of Markov State Models (MSMs) [34], it is not directly amenable to rare event systems that might be undersampled, or accessible only through biased simulations.

To deal with this issue, in this work we combine the commute distance [117, 118] with the Maximum Caliber based "Spectral Gap Optimization of Order Parameters (SGOOP)" approach [99]. This amounts to inducing a distance metric, which we call "SGOOP-d" that preserves kinetic truthfulness, and can be calculated from long unbiased simulations as well as biased simulations. Such biased simulations are often unavoidable in the study of rare events in chemical and biological physics. Here we use metadynamics [119] as an example of the biasing method to illustrate the usefulness of SGOOP-d while anticipating that the method directly applies to other biasing protocols as well [120]. We demonstrate the utility of the method by studying a range of model systems including conformational transitions in a small peptide Ace-Ala_3-Nme. In this system, for instance, one has a total of at least 28 inter-state transitions. As we show here, with only two component-RC learned from SGOOP-d we do accurately capture most of the 28 pairs of distances, with minimal improvement achieved by adding a 3rd component to the RC. Similar results are

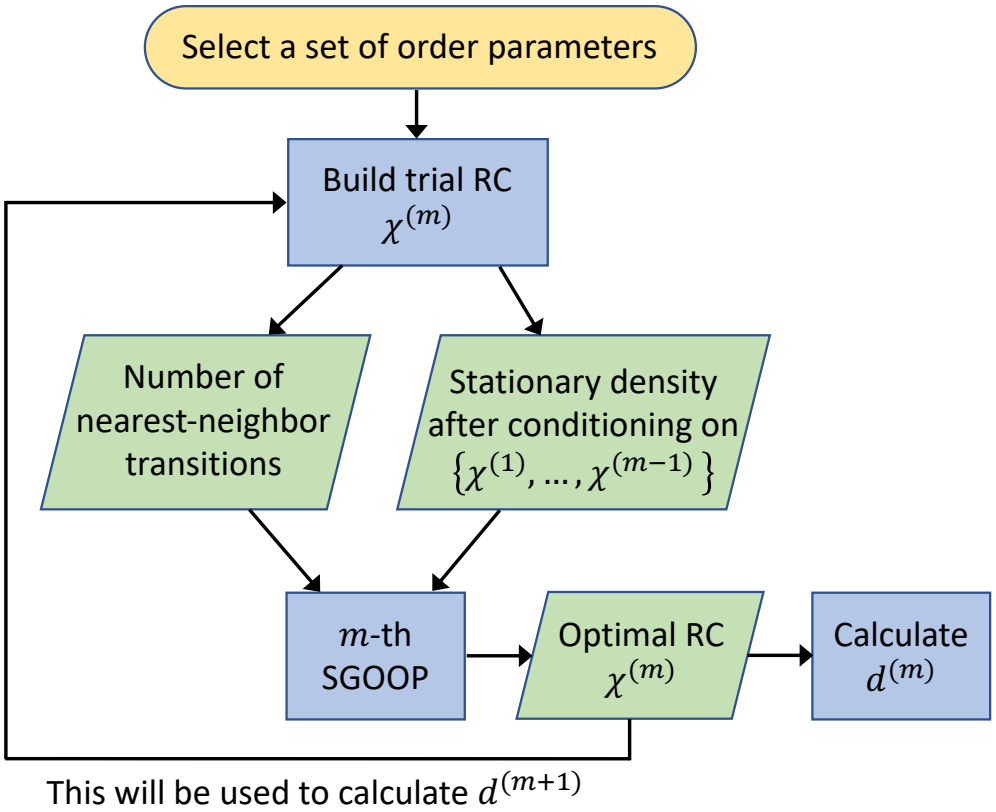


Figure 3.1: **Illustration of the calculation of the m-th RC component found by SGOOP-d.** This flowchart describes the calculation of the m-th RC component $\chi^{(m)}, m \geq 1$ through multi-dimensional spectral gap optimization. For each m we calculate $d^{(m)}$ in Eq. 3.17, which represents the contribution to commute distance on the basis of this m-th component. The optimal RC $\chi^{(m)}$ will be fed to the next SGOOP calculation for finding $d^{(m+1)}$. This iteration will stop when we obtain convergence in state-to-state d^2_{comm} values with addition of RC components. The commute distance d^2_{comm} will be the sum of all the $d^{(m)}$ obtained in the iteration.

obtained on the basis of input trajectories coming from metadynamics simulations biased along pre-selected biasing variables. Open-source software detailing the method has also been released.

3.2 Theory

3.2.1 Commute Distance and Commute Map

Our work builds upon the powerful advances first introduced by Noé, Clementi, and co-workers that allow quantifying a kinetically truthful distance metric between generic molecular configurations [117,118]. One such notion of "kinetic distance" was introduced in Ref. [117], which was then generalized in Ref. [118] as the "commute distance". Both of these distances amount to transformations of the input coordinate space into a new space wherein Euclidean distances directly correspond to interconversion times. Here we summarize the basic ideas which originated from diffusion maps [121,122] but were later generalized to Markovian dynamics [117,118].

We consider a generic dynamical system undergoing Markovian dynamics in a finite-dimensional state space Ω. The local density $\rho_t(\mathbf{x})$, $\forall \mathbf{x} \in \Omega$ can be propagated in time t through

$$\rho_{t+\tau}(\mathbf{y}) = \int_{\mathbf{x}\in\Omega} \rho_t(\mathbf{x}) p_\tau(\mathbf{y}|\mathbf{x}) d\mathbf{x} \equiv \mathcal{P} \circ \rho_t(\mathbf{x}) \tag{3.1}$$

where $p_\tau(\mathbf{y}|\mathbf{x})$ is the transition density of finding the system at state $\mathbf{y}$ at time $t+\tau$ given that we have started it at state $\mathbf{x}$ at time t. Equivalently, Eq. 3.1 defines a Markov operator $\mathcal{P}$ and describes how an initial distribution $\rho_t(\mathbf{x})$ at time t propagates to the distribution $\rho_{t+\tau}(\mathbf{y})$ at a later time $t+\tau$. One usual assumption made here is that there exists a unique

equilibrium distribution $\pi(\mathbf{x})$ which satisfies

$$\pi(\mathbf{x}) = \mathcal{P} \circ \pi(\mathbf{x}) \tag{3.2}$$

At the same time, we can write an equivalent equation for the weighted density $\nu_t(\mathbf{x}) = \rho_t(\mathbf{x})/\pi(\mathbf{x})$

$$\pi(\mathbf{y})\nu_{t+\tau}(\mathbf{y}) = \int p_\tau(\mathbf{y}|\mathbf{x})\pi(\mathbf{x})\nu_t(\mathbf{x})d\mathbf{x} = \mathcal{T} \circ \nu_t(\mathbf{x}) \tag{3.3}$$

where $\mathcal{T}$ is the corresponding backward operator, also called the transfer operator. With this formalism, following the literature on diffusion maps [122] one defines a distance measure $D_\tau^2(\mathbf{x}_1, \mathbf{x}_2)$ between two points $\mathbf{x}_1, \mathbf{x}_2$ in the state space of a random walk as

$$D_\tau^2(\mathbf{x}_1, \mathbf{x}_2) = \int_{\mathbf{y}\in\Omega} \frac{|p_\tau(\mathbf{y}|\mathbf{x}_1) - p_\tau(\mathbf{y}|\mathbf{x}_2)|^2}{\pi(\mathbf{y})} d\mathbf{y} \tag{3.4}$$

This definition can be seen [122] as equivalent to (a) preparing two ensembles initially located at $\mathbf{x}_1$ and $\mathbf{x}_2$, (b) letting them evolve by a lag time τ, and then (c) computing the difference between the subsequently resulting probability distributions. In order to make use of Eq. 3.4, one needs the transition density $p_\tau(\mathbf{y}|\mathbf{x})$. To facilitate its computation [117], we assume that the transfer operator $\mathcal{T}$ has N discrete eigenpairs and assume reversible dynamics/detailed balance $\pi(\mathbf{x})p_\tau(\mathbf{y}|\mathbf{x}) = \pi(\mathbf{y})p_\tau(\mathbf{x}|\mathbf{y})$:

$$p_\tau(\mathbf{y}|\mathbf{x}) = \sum_{j=0}^{N-1} \lambda_j(\tau)\psi_j(\mathbf{x})\pi(\mathbf{y})\psi_j(\mathbf{y}) \tag{3.5}$$

where λ_j and ψ_j are the corresponding eigenvalues and eigenvectors of the transfer operator $\mathcal{T}$. With the orthonormality condition $\int \pi(\mathbf{y})\psi_j(\mathbf{y})\psi_k(\mathbf{y})d\mathbf{y} = \delta_{jk}$, applying Eq. 3.5 to Eq. 3.4 directly leads to:

$$D_\tau^2(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{N-1} (\lambda_j \psi_j(\mathbf{x}_1) - \lambda_j \psi_j(\mathbf{x}_2))^2 \tag{3.6}$$

In Eq. 3.6 the summation starts at $j = 1$ since the $j = 0$ eigenvector for the transfer operator $\mathcal{T}$ is a constant in $\mathbf{x}-$space. By further integrating out the lag time τ in Eq. 3.6, we can make Eq. 3.6 insensitive to the choice of the lag time, and in this way we arrive at the definition of the commute distance d_{comm}^2:

$$\begin{aligned} d_{\text{comm}}^2(\mathbf{x}_1, \mathbf{x}_2) &= \int_0^\infty D_\tau^2(\mathbf{x}_1, \mathbf{x}_2) d\tau \\ &= \sum_{j=1}^{N-1} \left(\sqrt{\frac{t_j}{2}} \psi_j(\mathbf{x}_1) - \sqrt{\frac{t_j}{2}} \psi_j(\mathbf{x}_2) \right)^2 \end{aligned} \tag{3.7}$$

where $t_j = -\frac{\tau}{\ln \lambda_j}$ is the relaxation timescale associated with jth eigenvector. Often one uses the rate $k_j = {t_j}^{-1}$ instead of the timescale [123]. Eq. 3.7 now has a Euclidean distance form and a direct physical meaning: it is approximately the average time the system spends to commute between two states [118]. The distance d_{comm} is thus called the "commute distance", and the associated mapping

$$\mathbf{x} \mapsto (\sqrt{\frac{t_1}{2}}\psi_1, ..., \sqrt{\frac{t_{N-1}}{2}}\psi_{N-1}) \tag{3.8}$$

is called the "commute map".

Assuming that the dynamics in the $\mathbf{x}-$space is Markovian and fully sampled giving access to eigenvalues and eigenvectors of $\mathcal{T}$, we can then use Eq. 3.7 to calculate a Euclidean distance which approximates the commute time in the $\mathbf{x}-$space. It is also worth pointing out that in Eq. 3.7 the timescales follow $t_1 \geq t_2 \geq ...0$, which implies that the commute distance increases monotonically with consideration of further eigenvectors of $\mathcal{T}$, and that there is an increasingly vanishing contribution from every additional eigenvector that we consider. If such a distance can be obtained through Eq. 3.7, it is very useful for analyzing high-dimensional trajectories arising from well-sampled simulations as shown for instance in Ref. [117, 118]. However many if not most real-world applications are characterized by rare events, wherein the system stays trapped in the part of the configuration space it was initiated from and rarely visits other regions. Adequate and reliable sampling of the underlying configuration space thus remains a longstanding challenge in computational chemistry and physics. This implies that the eigenvectors and eigenvalues needed to evaluate the various terms in Eq. 3.7 are simply not available or far from reliable. In fact, the dominant first few components of the commute map could even serve as biasing coordinates along which the sampling could be enhanced through methods such as umbrella sampling, metadynamics, or others. This brings out the inverse nature of the problem wherein constructing an accurate commute distance depends on sufficient sampling of the eigenvalues and eigenvectors of the transfer operator, but the sampling itself could benefit greatly from the knowledge of the commute map.

3.2.2 Calculating commute distances for rare events

In this section, we develop a formalism for obtaining commute distances in poorly sampled rare-event systems where access to $\mathcal{T}$ and its eigenvectors/eigenvalues is not straightforward. The central idea is to perform biased sampling to accelerate the exploration of the configuration space. Here we use metadynamics as the biased sampling method, but the developed formalism should be more generically applicable. While this basic idea is simple, there are, however, at least two major, immediate difficulties when applying Eq. 3.7 with metadynamics or other similar enhanced sampling methods. First, the use of any sort of biasing corrupts the kinetics of the system, critical to calculating accurate eigenvalues and eigenvectors of the transfer operator $\mathcal{T}$. Second, the biasing itself needs access to the slow modes of the system, which are the dominant components of the commute map in Eq. 3.8. In SGOOP, described in Sec. 3.2.2.1 and Sec. 3.2.2.2, we find these slow modes from the transfer operator of such a transition matrix but only look at its dynamics along a 1-d coordinate. We refer to these slow modes as the reaction coordinate (RC) for the system [28, 124]. As mentioned in Sec. 3.2.1 the different components ψ_i of the commute map have a vanishing relevance to the calculation of the commute distance as $i \gg 1$, and thus one can stop after the first few dominant components and bias these components in any biasing method of choice. However, without knowing the commute map, it is hard to calculate the dimensionality and components of the RC which would then be biased.

3.2.2.1 SGOOP for 1-dimensional RC

In this sub-section we summarize the "Spectral Gap Optimization of Order Parameters (SGOOP)" method for optimizing a multi-dimensional RC [97, 99, 116]. In later sections, we use SGOOP to develop an approach that circumvents both of the above-described challenges. Summarily, SGOOP in its original form is a method for obtaining a one-dimensional RC given static and dynamic information about a multi-dimensional system by combining this information in a Maximum Caliber or path entropy framework [125, 126]. SGOOP constructs the RC as a combination of pre-selected candidate order parameters $(s_1, ..., s_d)$, which can be thought of as a set of basis functions using which we are trying to describe our problem. The dimensionality d is kept high enough so that dynamics in the high-dimensional $s-$space is likely Markovian, needed for the formalism described in Sec. 3.2.1. The central ideas behind SGOOP [99] in its original form can be summarized as the following three points:

(i) It uses a reweighting protocol [127] to estimate the equilibrium distribution $P_0(s_1, ..., s_d)$ from an initial metadynamics simulation performed by biasing some trial RC.

(ii) In addition, it uses short unbiased MD simulations to obtain dynamical observables pertaining to the system. These observables could be the position-dependent diffusivity or more typically, the number of nearest-neighbor transitions along some binned trial RCs.

(iii) By combining (i) and (ii) SGOOP constructs the transition rate matrices K which can

then be formulated as follows:

$$K_{mn} = \begin{cases} -\Lambda\sqrt{\frac{\pi_n}{\pi_m}}, & \text{if } n \neq m \\ -\sum_{k\neq m} K_{mk}, & \text{if } n = m \end{cases} \tag{3.9}$$

where $\pi \equiv P_0$ is the stationary probability along any putative, spatially discretized RC χ with n denoting the grid index and Λ is a dynamical observable. As mentioned in point (i), the stationary distribution can be obtained from a long unbiased simulation or from a biased simulation followed by an appropriate reweighting. The dynamical variable Λ, as discussed in point (ii), can be calculated by the number of nearest-neighbor transitions $\langle N \rangle$ defined as

$$\langle N \rangle = \sum_{\substack{(m,n) \\ \forall |m-n|=1}} \pi_m K_{mn} N_{mn} \tag{3.10}$$

where $N_{mn} = 1 \forall |m-n| = 1$ and 0 otherwise. Plugging Eq. 3.9 into Eq. 3.10 we obtain an estimate of Λ as:

$$\Lambda = \frac{\langle N \rangle}{\sum \sqrt{\pi_m \pi_n}} \tag{3.11}$$

The eigenvalues $\{k_j\}$ of the rate matrix K are nonnegative and satisfy $k_0 = 0 < k_1 \leq k_2 \leq ....$ The quantity $e^{-k_{n-1}} - e^{-k_n}$, which is the "spectral gap" of the transfer operator $\mathcal{T}$, can be interpreted as the timescale separation between the n slow mode and all the other hidden faster modes as projected on the corresponding RC. It can be shown that the optimal RC has the maximal spectral gap [97]. Different candidate one-dimensional

RCs are then first ranked in terms of the number of slow modes or metastable states they demarcate, and then in terms of the timescale separation (or the spectral gap) between the slow and fast modes as projected on any RC. The optimal RC maximizes both of these.

3.2.2.2 SGOOP for multi-dimensional RCs and rate matrices

In this section, we will introduce a multi-dimensional version of SGOOP [116] which makes it possible to extend the dimensionality of the RC in SGOOP. Each additional RC component $\chi^{(i)}, i \geq 2$ is constructed in a way that it captures features indiscernible in the previous components through a conditional probability factorization described in Sec. 3.2.2.2. This de-emphasizes the features already captured by the components so identified. With multiple iterations of the SGOOP protocol one can identify a multi-dimensional RC $\chi = \{\chi^{(1)}, \chi^{(2)}, ...\}$. Mathematically this can be written as follows. Once the first RC component $\chi^{(1)}$ has been learned by SGOOP, we focus our attention on the probability distribution P_1 conditional on the knowledge of χ defined as:

$$\begin{aligned} P_1(s_1, ..., s_d) &\equiv P_0(s_1, ..., s_d|\chi^{(1)}) \\ &= \frac{P_0(s_1, ..., s_d)}{P_0(\chi^{(1)})} \end{aligned} \tag{3.12}$$

where we have used that the equilibrium probability $P_0(s_1, ..., s_d, \chi^{(1)}) = P_0(s_1, ..., s_d)$ as $\chi^{(1)}$ is a deterministic function of $(s_1, ..., s_d)$. The next round of SGOOP is then performed on data sampled from P_1 instead of P_0, which yields the second RC component $\chi^{(2)}$ that captures features missed by $\chi^{(1)}$. The procedure can be repeated for further RC components and can be performed using any enhanced sampling method [116]. Here we

illustrate it using metadynamics. By performing well-tempered metadynamics simulation along $\chi^{(1)}$ where one builds a bias $V_b(\chi^{(1)})$, it can be shown that

$$P_0(\chi^{(1)}) \propto e^{-\beta F(\chi^{(1)})} \propto e^{+\beta[\frac{\gamma}{\gamma-1}V_b(\chi^{(1)})]}$$
$$P_1 \equiv \frac{P_0(s_1,...,s_d)}{P_0(\chi^{(1)})} \propto e^{-\beta[F(s_1,...,s_d)+V_b(\chi^{(1)})]} \tag{3.13}$$

where $\beta = 1/k_BT$, γ is the bias factor for well-tempered metadynamics [119], and F is the free energy of the system. Therefore, P_1 is simply the unreweighted/biased probability density obtained by sampling in the presence of bias potential $V_b(\chi^{(1)})$.

We now discuss details of the construction of the rate matrix through SGOOP. Following Eq. 3.9 and Eq. 3.10, the rate matrix along any putative RC χ can be built as follows:

$$K^{(1)}_{mn} = \begin{cases} -\frac{\langle N\rangle}{\sum\sqrt{\pi_n\pi_m}}\sqrt{\frac{\pi_n}{\pi_m}}, & \text{if } n \neq m \\ -\sum_{k\neq m} K^{(1)}_{mk}, & \text{if } n = m \end{cases} \tag{3.14}$$

where $\langle N\rangle$ is the total number of nearest-neighbor transitions per unit time, counted along a suitably discretized RC $\chi = \{\chi_n\}$ with n indicating grid index, $\pi \equiv P_0$ is the corresponding stationary density and 1 in superscript indicates this is the rate matrix along the first component $\chi^{(1)}$ of the RC. For the first round of SGOOP to learn $\chi^{(1)}$, $\langle N\rangle$ is calculated from short unbiased MD simulations. The $K^{(1)}$ matrices are then constructed for different putative RCs and its eigenvalues used to screen for the best RC $\chi^{(1)}$ with highest spectral gap.

For learning the second component $\chi^{(2)}$ and other higher-order components, we generalize Eq. 3.14 as follows [116]:

$$K^{(2)}_{mn} = \begin{cases} -\frac{\langle N \rangle^{(1)}}{\sum \sqrt{\pi^{(1)}_n \pi^{(1)}_m}} \sqrt{\frac{\pi^{(1)}_n}{\pi^{(1)}_m}}, & \text{if } n \neq m \\ -\sum_{k \neq m} K^{(2)}_{mk}, & \text{if } n = m \end{cases} \tag{3.15}$$

In Eq. 3.15, $\pi^{(1)} \equiv P_1$ is defined in Eq. 3.13. $\langle N \rangle^{(1)}$ denotes the average number of first-nearest neighbor transitions along a putative RC observed per unit time, but now measured in the biased simulation performed by sampling from this conditional probability density P_1. The procedure can then be easily generalized for constructing rate matrices $K^{(3)}, K^{(4)}, ...$ for learning further RC components.

3.2.2.3 Commute distance calculation for rare events with SGOOP

Here we use SGOOP to induce a commute distance metric for complex high-dimensional systems that can be calculated from a combination of biased simulations and short unbiased trajectories. Assuming that a satisfactorily large number of components have been included in χ, any two points $\{\mathbf{x}, \mathbf{x}'\} \in \Omega$ can then be mapped without substantial loss of information to its values in the χ space as $\{\chi, \chi'\}$. Whether the dimensionality of the RC χ is indeed sufficient or not is a non-trivial question to answer, which we will address later in this section and in Sec. 3.4. With the RC optimized by SGOOP,

we can then reformulate Eq. 3.7 as

$$
\begin{aligned}
d^2_{\text{comm}}(\mathbf{x}, \mathbf{x}') &= d^2_{\text{comm}}(\chi, \chi') \\
&= \sum_{j=1}^{N-1} \frac{1}{2k_j} \left[\psi_j(\chi) - \psi_j(\chi')\right]^2 \\
&= \sum_{j=1}^{N-1} \frac{1}{2k_j^{(1)}} \left[\psi_j^{(1)}(\chi) - \psi_j^{(1)}(\chi')\right]^2
\end{aligned} \tag{3.16}
$$

In the above equation, we have made use of the mapping $\mathbf{x} \rightarrow \chi$ learned from SGOOP, but otherwise, it still needs the eigenvalues and eigenvectors of the transfer operator $\mathcal{T}$. In the final line, we have introduced a superscript (1) to indicate the case where the first RC $\chi^{(1)}$ learned from SGOOP is indeed sufficient for the system at hand. In such a case, SGOOP yields a Maximum Caliber based rate matrix $K^{(1)}$ for transitions between grid points along suitably discretized $\chi^{(1)}$. Details of the construction of this rate matrix are described in Sec. 3.2.2.2 while illustrative examples are provided in Sec. 3.4. By diagonalizing the rate matrix $K^{(1)}$ we obtain the eigenvalues $k_1^{(1)}, k_2^{(1)}, ...$ and corresponding eigenvectors $\psi_1^{(1)}, \psi_2^{(1)}, ...$ to use in Eq. 3.16.

The above commute distance so obtained can be understood as an estimate of true commute distance using the 1-dimensional projected RC $\chi^{(1)}$. However, as shown in Sec. 3.4 and also emphasized in the literature on numerous occasions [115], a 1-dimensional projection is often not kinetically truthful and does not reflect the connectivity of underlying high-dimensional space. We thus consider additional RC components $\chi^{(m)}$ from the multi-dimensional SGOOP protocol, with eigenvalues $k_1^{(m)}, k_2^{(m)}, ...$ and corresponding eigenvectors $\psi_1^{(m)}, \psi_2^{(m)}, ...$, where $m \geq 1$ denotes which RC component

we are looking at. Each such component induces its own contribution to the commute distance which we add to the contribution of the 1st component $\chi^{(1)}$ in Eq. 3.16 yielding the central equation of this work for a $M-$component RC:

$$d^2_{\text{comm}}(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{M} \sum_{j=1}^{N-1} \frac{1}{2k_j^{(m)}} \left[\psi_j^{(m)}(\chi) - \psi_j^{(m)}(\chi')\right]^2 \equiv \sum_{m=1}^{M} d^{(m)} \tag{3.17}$$

Here $d^{(m)}$ is the contribution to the commute distance arising from the m^{th} RC component, while $k_j^{(m)}$ and $\psi_j^{(m)}$ are the j^{th} eigenvalue and eigenvector of the Maximum Caliber-based transition matrix $K^{(m)}$ calculated along along RC-component $\chi^{(m)}$ (Sec. 3.2.2.2).

We want to mention two important points here. Firstly, for any RC component $\chi^{(m)}$ for $m \geq 1$, the construction of the rate matrix $K^{(m)}$ as detailed in Sec. 3.2.2.2 ensures that the rates are ordered as per $0 < k_1^{(m)} \leq k_2^{(m)} \leq ....$ This leads to a useful property that the commute distance is a strictly monotonically increasing function of adding further RC components as well as further eigenvectors along any RC component. By monitoring how $d^2_{\text{comm}} = d^{(1)} + d^{(2)} + ...$ converges with addition of RC components, we can quantify the dimensionality of the RC needed for a given system at hand. Secondly, the intuitive idea behind going from Eq. 3.16 to Eq. 3.17 is that different eigenvectors are orthogonal to each other allowing for a Euclidean distance measure. This is strictly true for the SGOOP-derived eigenvectors along a given RC component, i.e. the dot product of $\psi_j^{(m)}$ and $\psi_k^{(m)}$ is 0 $\forall j, k, m \geq 0$ as mentioned in Sec. 3.2.2.2. However when comparing $\psi_j^{(m)}(\chi^{(m)})$ and $\psi_k^{(n)}(\chi'^{(n)})$ for $m \neq n$ i.e. for different RC components through multiple rounds of SGOOP [116] this is not strictly true, and thus we expect Eq. 3.17 to be an upper bound for

the commute distance. Note that the error could come from any eigenpair of each SGOOP rate matrix arising from redundant contributions due to different RC components having some aspects of the same dynamical processes. However as we will show later in Sec. 3.4, as long as each optimal RC captures the most important features or slowest processes, in the next round of SGOOP, such optimal RC will efficiently reduce the error from the non-orthogonality, making Eq. 3.17 a good approximation to those important features.

3.3 Model set-up and simulation details

In this section, we will introduce the model potentials and simulations that we will use later in the discussion of Sec. 3.4.

3.3.1 Analytical potentials set-up

The potential $U(x, y)$ governing the model with three metastable states is given by

$$\begin{aligned} U(x,y) = W(x^6 + y^6) - G(x,x_1)G(y,y_1) \\ - G(x,x_2)G(y,y_2) - G(x,x_3)G(y,y_3) \end{aligned} \tag{3.18}$$

where $W = 0.0001$ and $G(x, x_0) = e^{-\frac{(x-x_0)^2}{2\sigma^2}}$ denotes a Gaussian function centered at x_0 with width $\sigma = 0.8$. We also build 4-state model systems, each denoted by 4A, 4B, with

governing interaction potentials U_{A}, U_{B}, U_{C}:

$$\begin{aligned} U_i(x,y) &= W(x^4+y^4) + G_i(x,0.0)G'(y,0.0) \\ &\quad - G_i(x,2.0)G'(y,-1.0) - G_i(x,0.5)G'(y,2.0) \\ &\quad - G_i(x,-0.5)G'(y,-2.0) - G_i(x,-2.0)G'(y,1.0) \end{aligned} \tag{3.19}$$

where $G_i(x,x_0) = e^{-\frac{(x-x_0)^2}{2\sigma_i^2}}$ have widths $\sigma_i = 0.8, 1.0$ for $i = \mathrm{A}, \mathrm{B}$ respectively, while $G'(y,y_0) = e^{-\frac{(y-y_0)^2}{2\sigma'^2}}$ have a fixed width $\sigma' = 1.2$. The configurations corresponding to the model potentials in Eq. 3.18 and Eq. 3.19 are illustrated in Fig. 3.5(a)-(d).

3.3.2 Simulation set-up

The integration timestep for the Langevin dynamics simulation was 0.01 units, and the simulation was performed at $\beta = 2.5$ for 3-state and 4-state potentials, where $\beta = 1/(k_B T)$. The MD results for alanine dipeptide and Ace-Ala$_3$-Nme were obtained using the software GROMACS 5.0.4 [128, 129], patched with PLUMED 2.4 [102] with $2fs$ timestep. The temperature was kept constant at 300K for alanine dipeptide and 400K for Ace-Ala$_3$-Nme using the velocity rescaling thermostat [100]. The metadynamics parameters for each system are listed in Table. 3.1.

In Fig. 3.2 and 3.3, we have shown the free energy profile along ϕ_3 and free energy surface as function of (ϕ_1, ϕ_2). In Fig. 3.4, we have also shown the 3-dimensional free energy surface of Ace-Ala$_3$-Nme and its corresponding metastable molecular configurations. The corresponding dihedral angles for the 8 metastable states are tabulated in Table. 3.2.

Systems	h	ω	Δt/MD step	γ
3-state	0.3 (k_BT)	0.2	200	3.5
4A (Fig. 3.5(b))	0.4 (k_BT)	0.3	200	126
4B (Fig. 3.5(c))	0.4 (k_BT)	0.3	200	6
Alanine dipeptide	1.2 (kJ/mol)	0.2	500	5
Ala3 1-d	1.5 (kJ/mol)	0.25	400	10
Ala3 2-d	1.5 (kJ/mol)	0.2	500	5

Table 3.1: **Metadynamics parameters used for simulation of Langevin dynamics with 3-state, 4-state model potentials, alanine dipeptide, and Ace-Ala$_3$-Nme (Ala3) for SGOOP-d.** The metadynamics parameters used for simulations of Langevin dynamics with 3-state, 4-state model potentials, alanine dipeptide, and Ace-Ala$_3$-Nme (Ala3), where Ala3 1-d corresponds to biasing 1-d SGOOP-RC and Ala3 2-d corresponds to biasing 2-d SGOOP-RC. Gaussian bias kernels of starting height h and width ω are added every Δt MD step. γ is the bias factor for well-tempered metadynamics.

Metastable state	ϕ_1	ϕ_2	ϕ_3
S_1	-1.19	0.94	1.08
S_2	0.94	0.95	1.03
S_3	0.97	-1.54	1.06
S_4	-1.45	-1.57	1.04
S_5	-1.46	1.03	-1.72
S_6	0.98	1.01	-1.71
S_7	0.95	-1.46	-1.82
S_8	-1.25	-1.45	-1.73

Table 3.2: **Reference dihedral angles for the metastable states of Ace-Ala$_3$-Nme.** The reference dihedral angles in radians for the 8 metastable states we used in SGOOP-d to estimate 28 pairs of commute distances. The first and last 4 metastable states are separated by the third dihedral angle ϕ_3, where the first 4 have ϕ_3 near 1 and the last 4 have ϕ_3 near -1.7. The relative positions can be seen in the free energy plots at two different ranges of ϕ_3.

3.4 Results

In this section, we demonstrate the usefulness and reliability of the SGOOP [99, 116] based commute distance [117, 118] protocol developed in Sec. 3.2, which we label "SGOOP-d" for convenience, by applying it to a range of analytical potentials, as well as to small molecules with rare conformational transitions between different metastable

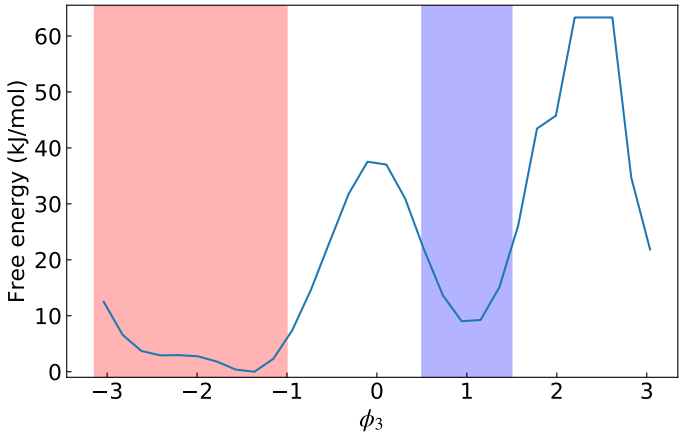


Figure 3.2: **The free energy profile of Ace-Ala$_3$-Nme along ϕ_3.** The free energy of Ace-Ala$_3$-Nme along ϕ_3 obtained by histogramming an unbiased MD trajectory at 400K. The red and blue regions correspond to $\phi_3 \in (-3.14, -1.00]$ and $\phi_3 \in (0.5, 1.5]$. These two ranges of ϕ_3 are integrated out to obtain free energy profiles at each metastable state.

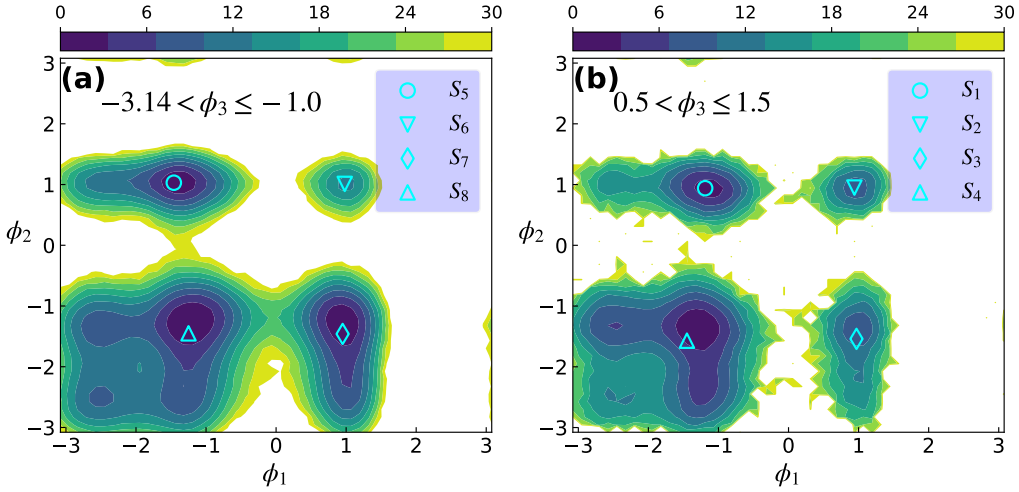


Figure 3.3: **The free energy surface of Ace-Ala$_3$-Nme as function of (ϕ_1, ϕ_2).** Free energy as function of (ϕ_1, ϕ_2) obtained by making histogram of an unbiased MD trajectory where (a) $\phi_3 \in (-3.14, -1.00]$ or (b) $\phi_3 \in (0.5, 1.5]$ are selected and integrated over. The free energy profiles show four metastable states at each basin in Supplementary Figure 3.2.

states. Low-dimensional projections of these high-dimensional potentials can in general lead to a spurious number of barriers and inter-basin connectivity [115, 130]. Here we show how to use SGOOP-d to ascertain the minimal dimensionality of the RC that preserves the kinetic aspects of the underlying high-dimensional landscape. To do so

we calculate the state-to-state commute distances and monitoring how these change and eventually converge with an increase in RC dimensionality. This is done using either biased or long unbiased simulations. We can also use the RC so learned to perform further efficient and reliable biased simulations. We consider different types of unbiased and biased trajectories to demonstrate the general applicability of our proposed framework.

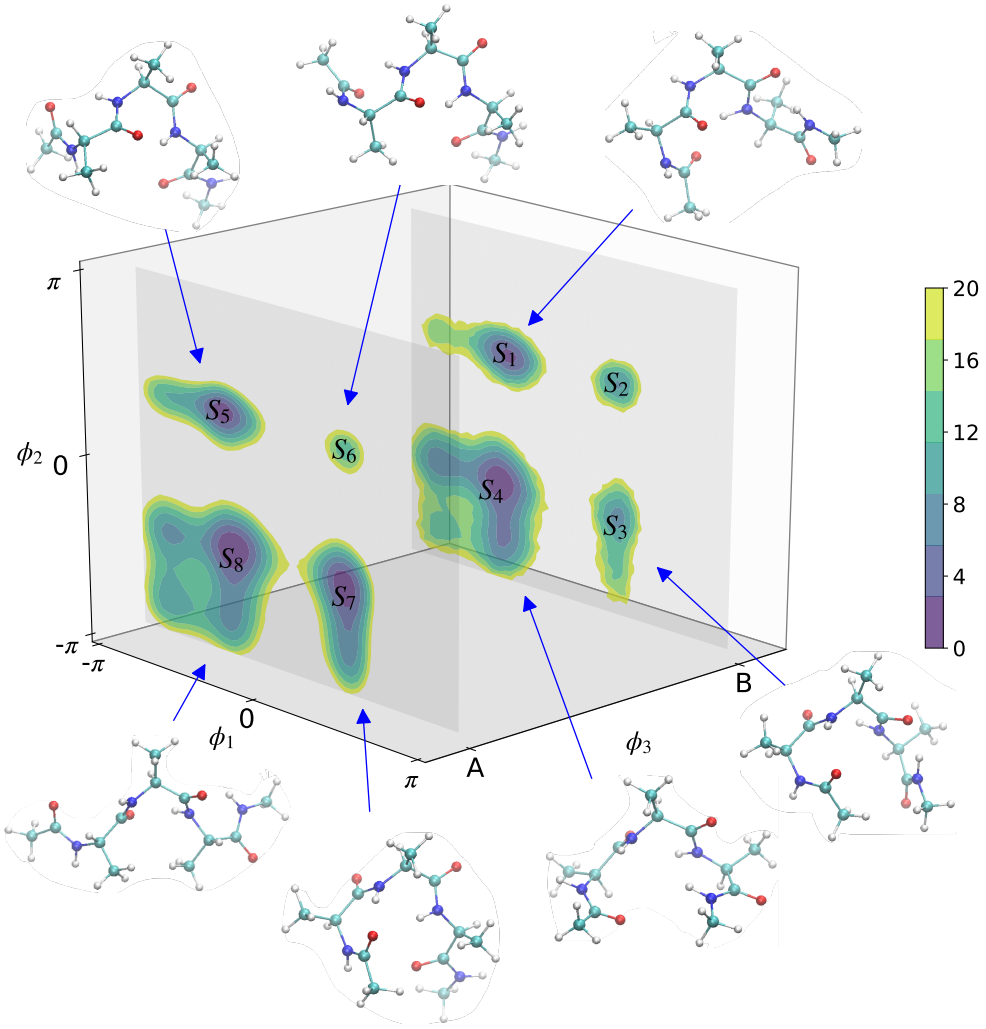


Figure 3.4: **The 3-dimensional free energy of Ace-Ala$_3$-Nme.** This figure shows the 3-d free energy with the state minimum discussed in Fig. 3.7 pointing by their corresponding conformations is shown. The integration of two ranges of ϕ_3 in Supplementary Figure 3.2 and 3.3 are plotted at A and B. Unlike the 2-d free energy plots shown in Supplementary Figure 3.3, the unbiased trajectory used here is not long enough to see S_2. This also shows that S_2 is rare and not always seen in every unbiased trajectory we have generated.

Numerical and computational details of these systems have been provided in Sec. 3.3.

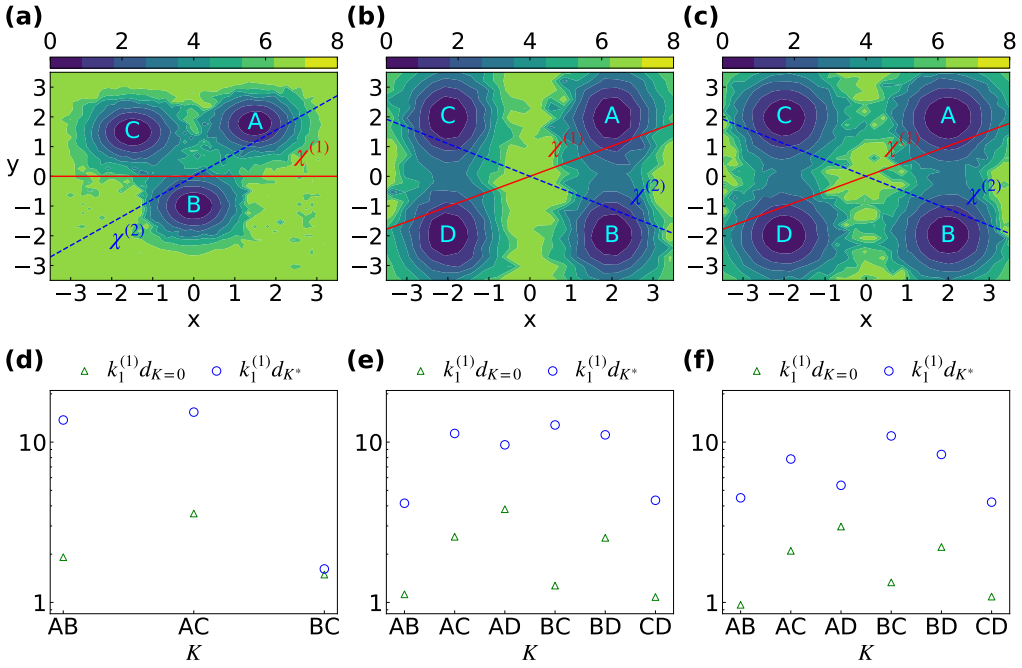


Figure 3.5: **The SGOOP-d analysis of the analytical potentials.** (a)-(c) show the 3-state and 4-state potentials 4A, 4B as sampled during molecular dynamics respectively. In (a)-(c) we have also provided the two RC components $\chi^{(1)}$ (solid red lines) and $\chi^{(2)}$ (dashed blue lines) evaluated using Eq. 3.22. Contours in all plots are separated by $0.89 k_B T$. In (d)-(f) we show the estimated commute distances d_K between different pairs of metastable states (in arbitrary units) at $K = 0$ and $K = K^*$. As explained in Sec. 3.4.1, using $K = K^*$ gives the right kinetic connectivity between different metastable states for each of the model potentials. The results with statistical averages and error bars are shown in SI. Here we only show the result with one pair of RC for each model system in order to show how the second RC component captures the missing features of the first RC component.

3.4.1 Analytical potentials

The analytical potentials used here are originally inspired from Ref. [115]. These are built with two degrees of freedom x and y, but with a varying number of metastable states and barriers separating them. Thus a 1-d projection is not always guaranteed to be kinetically truthful. Specifically we consider a 3-state potential and two 4-state potentials

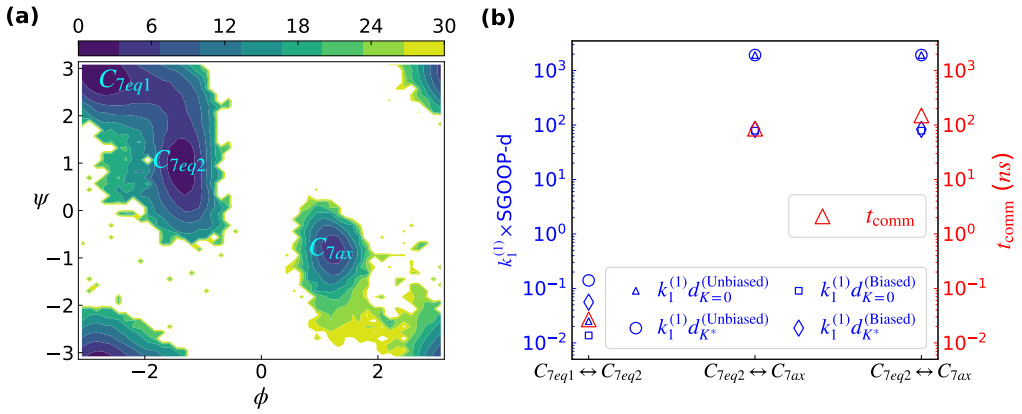


Figure 3.6: **The SGOOP-d analysis of alanine dipeptide.** (a) Free energy surface as a function of ϕ and ψ obtained by reweighting metadynamics simulation biasing along 1-d RC $\chi^{(1)}$ specified in Table. 3.4 . The positions of three metastable states are specified. (b) shows the SGOOP-d $k^{(1)}d_K$ at $K = 0$ using one RC and at $K = K^*$ using two RCs for each pair of metastable states (in arbitrary units) obtained from a long unbiased simulation (blue triangles and blue circles respectively, left axis) and the biased simulation (blue squares, blue diamonds, left axis). In (b), we also provide the estimated commute time t_{comm}(red triangles, right axis) calculated from the long unbiased simulation.

labeled 4A and 4B (Figs. 3.5 (a)-(c)). For each of these, we build inter-state commute distances using one-dimensional and two-dimensional RCs, with different components expressed as linear combinations of x and y. Since the underlying dimensionality is two, here we will demonstrate the results with up to two-dimensional RC. In such a case we can simplify Eq. 3.17 by introducing

$$\hat{d}^{(m)} = k_1^{(m)} d^{(m)} \tag{3.20}$$

and then writing

$$\begin{aligned} d_{\text{comm}}(\mathbf{x}_1, \mathbf{x}_2) &= d^{(1)} + d^{(2)} \\ &= \frac{1}{k_1^{(1)}} \hat{d}^{(1)} + \frac{1}{k_1^{(2)}} \hat{d}^{(2)} \end{aligned} \tag{3.21}$$

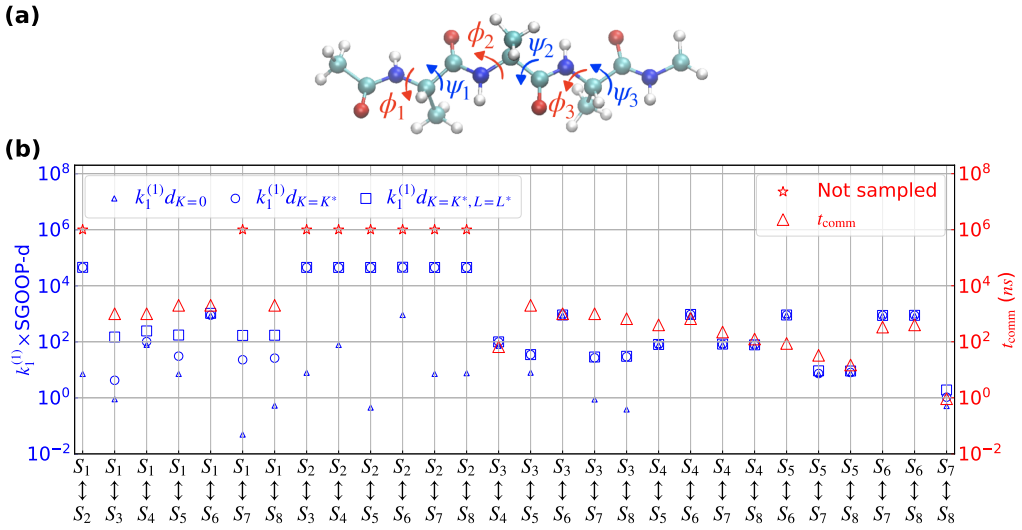


Figure 3.7: **The SGOOP-d analysis of Ace-Ala$_3$-Nme.** In this figure, (a) provides the molecular structure of Ace-Ala$_3$-Nme with the corresponding dihedral angles. The corresponding metastable states and their conformations are detailed in SI. (b) shows the calculation of SGOOP-d which provides the estimated commute distances using one-dimensional, two-dimensional and three-dimensional RC respectively (blue triangles, blue circles and blue squares, left axis). The coefficients of these RCs are shown in Table 3.4. Corresponding to their calculation, these are labelled respectively $k_1^{(1)} d_{K=0}$, $k_1^{(1)} d_{K^*}$ and $k_1^{(1)} d_{K^*,L^*}$ (in arbitrary units) as shown in the legend. (b) also provides the estimated commute time t_{comm} (red triangles, right axis) calculated from long unbiased simulation of Ace-Ala$_3$-Nme. The slowest transitions which are not sampled in the long unbiased simulation are denoted by star markers in the plot. Their commute times are not quantitatively reliable and serve only as guide to the eye.

To see how good a job the RC components do at reconstructing the state-to-state connectivity, we further parameterize Eq. 3.21 by introducing a $K \equiv \frac{k_1^{(1)}}{k_1^{(2)}}$ for the ratio of eigenvalues, yielding

$$k_1^{(1)} d_{\text{comm}}(\mathbf{x}_1, \mathbf{x}_2) \equiv k_1^{(1)} d_K \equiv \hat{d}^{(1)} + K \hat{d}^{(2)} \tag{3.22}$$

We highlight here that in our framework K is not a free parameter that needs to be tuned. Instead, it can be approximated on the basis of Maximum Caliber based rate matrices

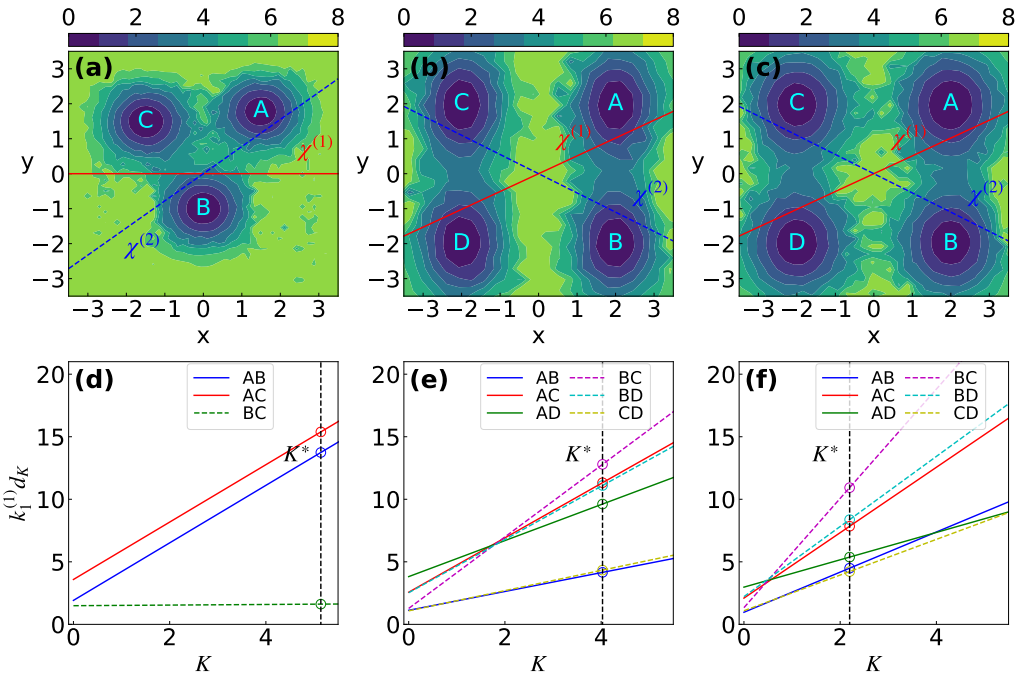


Figure 3.8: **SGOOP-d analysis isn't sensitive to K^*.** (a)-(c) show the analytical 3-state and 4-state potentials 4A, 4B respectively. In (a)-(c) we have also provided the two RC components $\chi^{(1)}$ (solid red lines) and $\chi^{(2)}$ (dashed blue lines) evaluated using Eq. 3.22. Contours in all plots are separated by $0.89 k_B T$. In (d)-(f) the estimated commute distances d_K between different pairs of states (in arbitrary units) are plotted as a function of K, where the benchmark parameter K^* in each case is specified as the vertical black dashed line.

(Sec. 3.2.2.2) as:

$$K^* \equiv \frac{k_1^{(1)}}{k_1^{(2)}} \tag{3.23}$$

where K^* indicates a Maximum Caliber based estimation of K. However, as the Maximum Caliber-based rate estimates are approximate and might depend on the choice of the dynamical constraints and quality of sampling [126], in Fig. 3.8, we also show that the precise value of K^* doesn't have a large effect on the connectivity.

Fig. 3.5 and Table. 3.3 detail the two RC-components $\chi^{(1)}$ and $\chi^{(2)}$ so obtained for the different model potentials. Here using $K = 0$ is equivalent to using only the first

Systems		$\theta^{(1)}/\pi$	$\theta^{(2)}/\pi$
3-state		0.00	0.21
4-state	4A (Fig. 3.5(b))	0.15	0.84
	4B (Fig. 3.5(c))	0.15	0.84

Table 3.3: **First and second components of the reaction coordinate found for the 3-state and 4-state model potentials.** In this table, we have shown the first and second components of the reaction coordinate $\chi^{(1)}$ and $\chi^{(2)}$ found for each model analytical potential. The angles $\theta^{(1)}$ and $\theta^{(2)}$ in the table define $\chi^{(i)} = \cos(\theta^{(i)})x + \sin(\theta^{(i)})y$.

component $\chi^{(1)}$ to determine the commute distance, while increasing non-zero values of K captures increasing contributions from the second component $\chi^{(2)}$ through Eq. 3.22. As can be seen for the 3-state system (Fig. 3.5 (d)), considering only the first component $\chi^{(1)}$ would lead to an erroneous conclusion that the pairs of states AB, AC, and BC are all kinetically equidistant. This is not consistent with the high-dimensional data sampled shown in Fig. 3.5 (a), where the barrier experienced between the states BC is much lower than for AB and AC. By adding the second component $\chi^{(2)}$ to the kinetic distance in Eq. 3.22 using $K = K^*$, we recover this correct picture. Similar conclusions regarding kinetically truthful picture consistent with the data can be drawn for the remaining two 4-state potentials shown in Fig. 3.5. In both Fig. 3.5 (e) and (f), using only the 1-d RC $\chi^{(1)}$, AB, BC, and CD are equally short, while AD is the slowest transition. This erroneous connectivity has been corrected after adding a second component of RC $\chi^{(2)}$, where AB and CD are equally shortest at $K = K^*$. Note that in both Fig. 3.5 (e) and (f) AD is slightly lower which shows the noisy nature in the Maximum Caliber-based estimation of transition rates.

Systems	RCs	Coefficients
Alanine dipeptide	$\chi^{(1)}$	$(0.643, 0.778, -0.133, -0.088, -0.221, -0.165)$
	$\chi^{(2)}$	$(0.827, 1.166, -0.120, 0.578, 0.013, 0.240)$
Ace-Ala$_3$-Nme	$\chi^{(1)}$	$(0.187, -1.127, -0.228, -2.362, 0.230, 1.176)$
	$\chi^{(2)}$	$(1.174, 0.738, 0.132, 0.716, 0.356, 2.827)$
	$\chi^{(3)}$	$(-0.037, -0.839, 0.557, 1.454, 1.693, 1.624)$

Table 3.4: **Reaction coordinates found for alanine dipeptide and Ace-Ala$_3$-Nme.** This table shows the reaction coordinates found for alanine dipeptide and Ace-Ala$_3$-Nme. For alanine dipeptide, two RC components both expressed as $\chi = a\cos\phi + b\sin\phi + c\cos\psi + d\sin\psi + e\cos\theta + f\sin\theta$ with their 6 respective coefficients are listed. For Ace-Ala$_3$-Nme, three RC components all expressed as $\chi = a\cos\phi_1 + b\sin\phi_1 + c\cos\phi_2 + d\sin\phi_2 + e\cos\phi_3 + f\sin\phi_3$ with their 6 respective coefficients are listed.

3.4.2 Alanine dipeptide

The next system we use to illustrate our method is the well-studied alanine dipeptide. Here we consider the molecule as characterized by three dihedral angles ϕ,ψ, and θ. This molecule has three metastable configurations (Fig. 3.6(a)) which can be characterized by using only ϕ and ψ, while θ plays a role in characterizing the transition between the metastable states [131]. Here we express the different RC components as linear combinations of 6 order parameters, namely cosines and sines of the 3 aforementioned dihedrals, with the final optimized coefficients listed in Table. 3.4. The spectral gap in SGOOP is optimized using a basin-hopping algorithm. [132–134] These RC components and associated information are then plugged into Eq. 3.22 to estimate the commute distance d_K. In Figs. 3.6(b)-(c) we show the commute distance so calculated using an input biased trajectory and a benchmark long unbiased trajectory respectively. The biased trajectory was generated by doing well-tempered metadynamics along 1-d RC $\chi^{(1)}$ defined in Table. 3.4. See Sec. 3.3 for further details of both the biased and unbiased simulations.

For this simple system, the commute distances d_K show similar connectivities for

$K = 0$ and $K = K^*$, which shows that one RC is indeed sufficient to describe the system in terms of recovering state-to-state connectivity between all 3 metastable states. Both types of input trajectories show a near degenerate structure with two pairs of states kinetically separated from each other, while one pair is very close.

3.4.3 Ace-Ala$_3$-Nme

In this final section, we demonstrate our method on a more complicated molecular system, namely the peptide Ace-Ala$_3$-Nme with a much larger number of metastable states, and an even larger number of state-to-state transitions [135]. Simulation details are provided in Sec. 3.3.2. As discussed in Ref. [135] the three dihedral angles ϕ_1, ϕ_2, ϕ_3 are sufficient to characterize the $2^3 = 8$ dominant metastable states corresponding to positive and negative parts of the Ramachandran diagram for the 3 central Alanine residues. The RC components used in computing SGOOP-d distances are calculated as a linear combination of cosines and sines of these 3 dihedral angles, thereby amounting to a total of 6 order parameters. We consider the 8 most dominant metastable states labelled S_1,..., S_8 and the associated $\binom{8}{2} = 28$ inter-state transitions. The corresponding dihedral angles for these 8 states are tabulated in Sec. 3.3. Here we consider up to three RC components and demonstrate that after considering 3 components the commute distances converge especially for the slower state-to-state transitions. They are also in agreement with the benchmark calculations on this system through counting transitions in the higher dimensional underlying space from a long unbiased trajectory. The final optimized solutions for all three RC components are shown in Table. 3.4. Here in order to add a third RC

component, we generalize Eq. 3.22 by introducing an additional parameter L:

$$k_1^{(1)} d_{K,L} \equiv \hat{d}^{(1)} + K\hat{d}^{(2)} + L\hat{d}^{(3)} \tag{3.24}$$

Similar to what was done for K^* in Eq. 3.23 we can approximate L^* as

$$L^* \equiv \frac{k_1^{(1)}}{k_1^{(3)}} \tag{3.25}$$

With a long enough unbiased MD trajectory, we can also calculate the commute time t_{comm} between two metastable states through a simple counting protocol (see also Ref. [130]). In Fig. 3.7, we show SGOOP-d distances calculated using Eq. 3.24 with 1, 2, and 3 RC components, and compare them with the corresponding 28 t_{comm} values between the 8 metastable states in the same plot. It can be seen from the plot that with only the use of two RC components SGOOP-d already provides converged estimates of relative inter-state connectivity and commute distances between 23 of the 28 pairs of states based on the visualization of 3-d free energy provided in Fig. 3.4. Here we must point out that there are eight transitions that are not sampled by even the reference long unbiased simulation, although SGOOP-d of those transitions clearly converged. Therefore, the comparison of SGOOP-d with respect to the unobserved transitions may need a more cautious evaluation instead of merely looking at the free energy. However, in order to get the correct connectivity for the remaining 5 pairs of states as well, we have to include the third RC component. We emphasize that in Fig. 3.7 the slowest 8 transitions have been given the same reference commute time for the sake of clarity, as we were unable to observe any

such transition events even in the 1 μs long unbiased simulation. Thus the reference commute times for these states serve as approximate lower bounds to the true values and are denoted by star markers in the plot.

3.5 Conclusion and outlook

In summary, in this work we have developed a computationally efficient formalism labeled "SGOOP-d" and summarized in the flowchart in Fig. 3.1, that can help towards solving a longstanding important problem in chemical physics and physical chemistry. Namely, how many dimensions should a projection from high-dimensions into low-dimensional reaction coordinates (RC) have, so that (1) the projection is kinetically and thermodynamically truthful to the underlying landscape, and (2) these minimal number of components can then be used to perform biasing simulations without fear of missing slow degrees of freedom. The formalism here makes the best of two different approaches, namely commute map [118] and SGOOP. [99] This way it induces a distance metric which we call SGOOP-d that is applicable to biased rare event systems as well as unbiased trajectories with arbitrary quality of sampling. The kinetically truthful RC learned here can then also be used to improve the sampling quality of the biased simulation itself [136] or as a progress coordinate in path-based sampling methods [59, 137–140]. We thus believe that going forward our work represents a useful tool in the study of kinetics in rare event systems with multiple states and interconnecting pathways. In future we will be extending this work to systems such as crystal nucleation, multiple polymorphs, systems with multiple pathways or multiple species, where there will be even more order parameters

to be considered. All of these continue to be very difficult yet important problems for understanding nucleation pathways and rates, and we are hopeful our tools will allow us and others to systematically study these.

Chapter 4: Building Kinetic Model using Simple Language Model

4.1 Introduction

4.1.1 Memory in terms of long-term dependency

In Chapter. 2, we have studied how the RC optimized through SGOOP, a method using traditional statistical mechanics, can capture memory. In Chapter. 3, we have proposed a SGOOP-based kinetic distances to learn multiple RCs and further reduce memory effect coming from missing slow processes. These methods provide a way to systematically find a good representation of complex systems with minimal memory effect which could have been enhanced due to a bad representation. Unfortunately, even though we have found all important RCs that captures the dynamical processes, there could always be memory effect that appears as the system's long-term dependency within the time series. Modeling such long-term dependencies in temporal data has been a longstanding challenge of machine learning. In recent years, many machine learning techniques have been developed to model this long-term dependency in dynamical systems such as chaos, stock market, and speech recognition. In this chapter, we will introduce a special type of neural network which allows us to model such memory.

4.1.2 An overview of recurrent neural network (RNN)

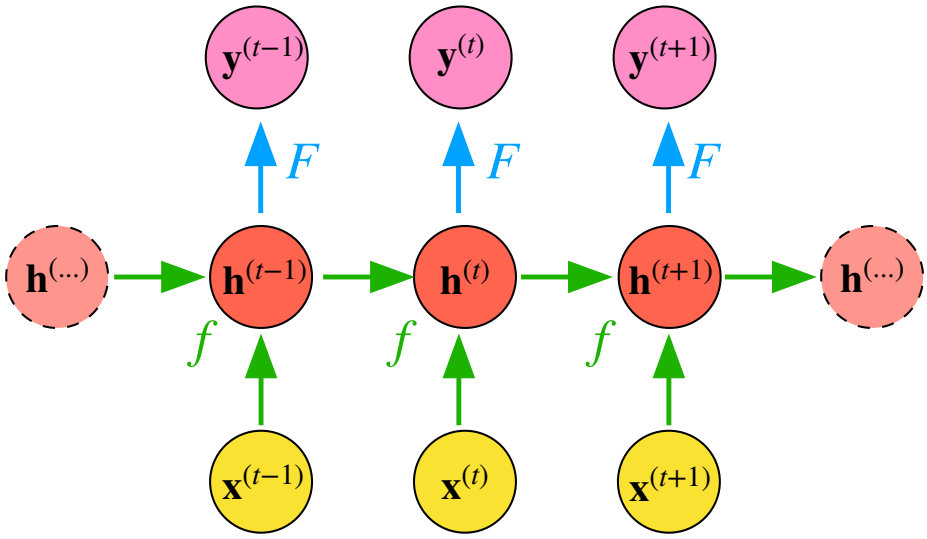


Figure 4.1: **Basic computational graph of RNN.**

Recurrent neural networks (RNNs) are one of the standard and popular machine learning/artificial intelligence (AI) technique developed for modeling temporal sequences, with demonstrated successes including but not limited to modeling human languages [141–147]. Mathematically, RNNs learn the recursive functions which could model the dynamical systems with its long-term dependency captured by the hidden units $\mathbf{h}^{(t)}$, as can be seen in Fig. 4.1. Without loss of generality, RNN can be equivalently mapped to a dynamical equation:

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \theta) \tag{4.1}$$

$$\mathbf{y}^{(t)} = F(\mathbf{h}^{(t)}) \tag{4.2}$$

where θ denotes the parameters in the recursive function. $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ are the input and output state vectors at time step t. In most of the cases, we train the network which parameterizes Eq. 4.2 such that it could predict the next data in a sequence, so $\mathbf{y}^{(t)} = \mathbf{x}^{(t+1)}$. However, learning the long-term dependency requires θ changes effectively during training, which is hard using the traditional backward propagation method. The notorious vanishing gradient problem also makes the training of RNNs even hard. Therefore, at least two modifications of RNNs have been proposed to solve these problem: The reservoir computing which has been used extensively in learning chaotic systems and Long Short-Term Memory (LSTM) which has been used in the rest of this dissertation.

4.1.3 Reservoir computing

As we have mentioned in the previous section, it is difficult for RNNs to learn the mapping from $\mathbf{h}^{(t-1)}$ to $\mathbf{h}^{(t)}$. As a result, an effective simplification of RNNs called reservoir computing [148] which assumes that a randomly connected internal weights have sufficient ability to learn some dynamical systems. Therefore, in reservoir computing, the hidden states are randomly assigned fixed weights and we only need to let the network learn the representation of state which is the output weights, namely the function F in Eq. 4.2. The reservoir computing has been shown to learn chaotic systems [149] very well. Such a capability is already useful for instance in weather forecasting, where one needs extremely accurate predictions valid for a short period of time. However, reservoir computing fails to learn complex molecular dynamics under a certain thermodynamic ensemble since most of the time the system simply fluctuates due to thermal and pressure

bath. To capture the thermal fluctuations which last long yet are important for the systems to overcome the barrier along RCs, we therefore need another method to directly learn the mapping from far past to future.

4.1.4 Long short-term memory (LSTM)

A specific and extremely popular instance of RNNs are long short-term memory (LSTM) [150] neural networks, which possess more flexibility and can be used for challenging tasks such as language modeling, machine translation, and weather forecasting [146, 151, 152]. LSTMs were developed to alleviate the limitation of previously existing RNN architectures wherein they could not learn information originating from far past in time. This is known as the vanishing gradient problem, a term that captures how the gradient or force experienced by the RNN parameters vanishes as a function of how long ago did the change happen in the underlying data [153, 154]. LSTMs deal with this problem by controlling flows of gradients through a so-called gating mechanism where the gates can open or close determined by their values learned for each input. The gradients can now be preserved for longer sequences by deliberately gating out some of the effects. This way it has been shown that LSTMs can accumulate information for a long period of time by allowing the network to dynamically learn to forget aspects of information. Very recently LSTMs have also been shown to have the potential to mimic trajectories produced by experiments or simulations [155], making accurate predictions about a short time into the future, given access to a large amount of data in the past. In this work, we consider an alternate and arguably novel use of RNNs, specifically LSTMs,

in making predictions that in contrast to previous work [149, 155], are valid for very long periods of time but only in a statistical sense. Unlike domains such as weather forecasting or speech recognition where LSTMs have allowed very accurate predictions albeit valid only for short duration of time, here we are interested in problems from chemical and biological physics, where the emphasis is more on making statistically valid predictions valid for extremely long duration of time. This is typified for example through the use of the ubiquitous notion of rate constant for activated barrier crossing, where short-time movements are typically treated as noise, and are not of interest for being captured through a dynamical model.

Here we suggest an alternative way to use LSTM-based language model to learn a probabilistic model from the time sequence along some low-dimensional order parameters produced by computer simulations or experiments of a high-dimensional system. We also show by our computer simulations of different model systems that the language model can produce the correct Boltzmann statistics (as can other AI methods such as Ref. [156, 157]) but also the kinetics over a large spectrum of modes characterizing the dynamics in the underlying data. We highlight here a unique aspect of this calculation that the order parameter our framework needs could be arbitrarily far from the true underlying slow mode, often called reaction coordinate. This in turn dictates how long of a memory kernel must be captured which is in general a very hard problem to solve [136, 158]. Our framework is agnostic to proximity from the true reaction coordinate and reconstructs statistically accurate dynamics in a wide range of order parameters. We also show how the minimization of loss function leads to learning the path entropy of a physical system, and establish a connection between the embedding layer and transition probability. Fol-

lowed by this connection, we also show how we can define a transition probability through embedding vectors. We provide tests for Boltzmann statistics and kinetics for Langevin dynamics of model potentials, MD simulation of alanine dipeptide, and trajectory from single molecule force spectroscopy experiment on a multi-state riboswitch [159] respectively. We also compare our protocol with alternate approaches including Hidden Markov Models. This work thus represents a new usage of a popular AI framework to perform dynamical reconstruction in a domain of potentially high fundamental and practical relevance, including materials and drug design.

4.2 Theory and Method

4.2.1 Mapping MD trajectories to abstract languages

Our central rationale in this work is that molecular dynamics (MD) trajectories, adequately discretized in space and time, can be mapped into a sequence of characters in some languages. By using a character-level language model that is effective in predicting future characters given the characters so far in a sequence, we can learn the evolution of the MD trajectory that was mapped into the characters. The model we use is stochastic since it learns each character through the probability they appear in a corpus used for training. This language model consists of three sequential parts shown schematically in Fig. 4.2. First, there is an embedding layer mapping one-hot vectors to dense vectors, followed by an LSTM layer which connects input states and hidden states at different time steps through a trainable recursive function, and finally a dense layer to transform the output of LSTM to the categorical probability vector.

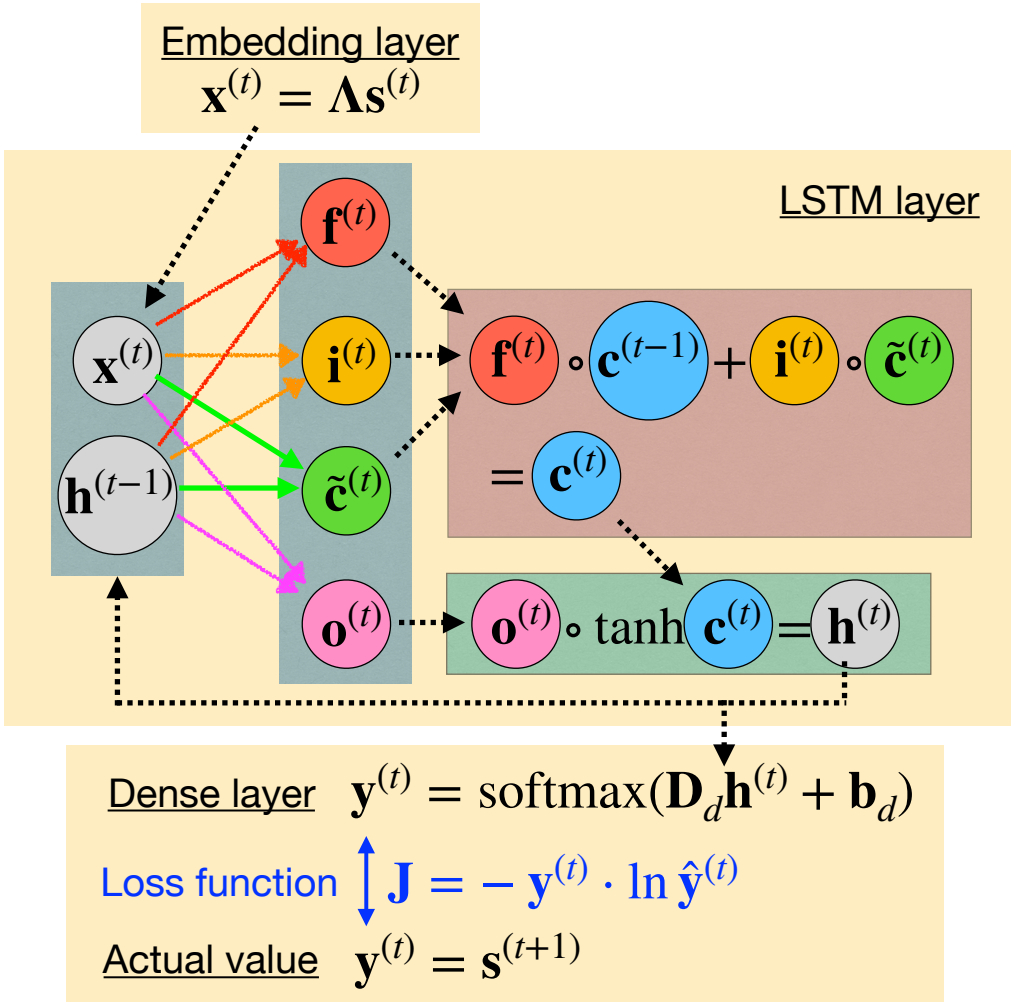


Figure 4.2: **Neural network schematic.** The schematic plot of the simple character-level language model used in this work. The model consists of three main parts: The embedding layer, the LSTM layer, and a dense output layer. The embedding layer is a linear layer which multiplies the one-hot input $\mathbf{s}^{(t)}$ by a matrix and produces an embedding vector $\mathbf{x}^{(t)}$. The $\mathbf{x}^{(t)}$ is then used as the input of LSTM network, in which the forget gate $\mathbf{f}^{(t)}$, the input gate $\mathbf{i}^{(t)}$, the output gate $\mathbf{o}^{(t)}$, and the candidate value $\tilde{\mathbf{c}}^{(t)}$ are all controlled by $(\mathbf{x}^{(t)}, \mathbf{h}^{(t-1)})$. The forget gate and input gate are then used to produce the update equation of cell state $\mathbf{c}^{t)}$. The output gate decides how much information propagates to the next time step. The output layer predicts the probabilities $\hat{\mathbf{y}}^{(t)}$ by parametrizing the transformation from $\mathbf{h}^{(t)}$ to $\hat{\mathbf{y}}$ with learned weights $\mathbf{D}_d$ and learned biases $\mathbf{b}_d$. Finally, we can compute the cross entropy between the predicted probability distribution $\hat{\mathbf{y}}^{(t)}$ and the true probability distribution $\mathbf{y}^{(t)} = \mathbf{s}^{(t+1)}$.

Specifically, here we consider as input a one-dimensional time series produced by a physical system, for instance through Langevin dynamics being undergone by a com-

plex molecular system. The time series consist of data points $\{\xi^{(t)}\}$, where t labels the time step and $\xi \in \mathbb{R}$ is some one-dimensional collective variable or order parameter for the high-dimensional molecular system. In line with standard practice for probabilistic models, we convert the data points to one-hot encoded representations that implement spatial discretization. Thus each data point $\{\xi^{(t)}\}$ is represented by a N-dimensional binary vector $\mathbf{s}^{(t)}$, where N is the number of discrete grid-points. An entry of one stands for the representative value and all the other entries are set to zeros. The representative values are in general finite if the order parameter is bounded, and are equally spaced in $\mathbb{R}$ with in total N representative values. Note that the time series $\{\xi^{(t)}\}$ does not have to be one-dimensional. For a higher-dimensional series, we can always choose a set of representative values corresponding to locations in the higher-dimensional space visited trajectory. This would typically lead to a larger N in the one-hot encoded representations, but the training set size itself will naturally stay the same. We find that the computational effort only depends on the size of training set and very weakly on N, and thus the time spent for learning a higher dimensional time series does not increase much relative to a one-dimensional series.

In the sense of modeling languages, the one-hot representation on its own cannot capture the relation between different characters. Take for instance that there is no word in the English language where the character c is followed by x, unless of course one allows for the possibility of a space or some other letter in between. To deal with this, computational linguists make use of an embedding layer. The embedding layer works as a look-up table which converts each one-hot vector $\mathbf{s}^{(t)}$ to a dense vector $\mathbf{x}^{(t)} \in \mathbb{R}^M$ by the multiplication of a matrix $\mathbf{\Lambda}$ which is called the embedding matrix, where M is called

the embedding dimension

$$\mathbf{x}^{(t)} = \mathbf{\Lambda}\mathbf{s}^{(t)} \tag{4.3}$$

The sequence of dense representation $\mathbf{x}^{(t)}$ accounts for the relation between different characters as seen in the training time series. The $\mathbf{x}^{(t)}$ is then used as the input of the LSTM layer. Each $\mathbf{x}^{(t)}$ generates an output $\mathbf{h}^{(t)} \in \mathbb{R}^L$ from LSTM layer, where L is a tunable hyperparameter. Larger L generally gives better learning capability but needs more computational resources. The LSTM itself consists of the following elements: the input gate $\mathbf{i}^{(t)}$, the forget gate $\mathbf{f}^{(t)}$, the output gate $\mathbf{o}^{(t)}$ the cell state $\mathbf{c}^{(t)}$, the candidate value $\tilde{\mathbf{c}}^{(t)}$, and $\mathbf{h}^{(t)}$ which is the hidden state vector and the final output from the LSTM. Each gate processes information in different aspects [150]. Briefly, the input gate decides which information to be written, the forget gate decides which information to be erased, and the output gate decides which information to be read from the cell state to the hidden state. The update equation of these elements can be written as follows:

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W}_f \mathbf{x}^{(t)} + \mathbf{U}_f \mathbf{h}^{(t-1)} + \mathbf{b}_f) \tag{4.4}$$

$$\mathbf{i}^{(t)} = \sigma(\mathbf{W}_i \mathbf{x}^{(t)} + \mathbf{U}_i \mathbf{h}^{(t-1)} + \mathbf{b}_i) \tag{4.5}$$

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W}_o \mathbf{x}^{(t)} + \mathbf{U}_o \mathbf{h}^{(t-1)} + \mathbf{b}_o) \tag{4.6}$$

$$\tilde{\mathbf{c}}^{(t)} = \tanh(\mathbf{W}_c \mathbf{x}^{(t)} + \mathbf{U}_c \mathbf{h}^{(t-1)} + \mathbf{b}_c) \tag{4.7}$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \circ \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \circ \tilde{\mathbf{c}}^{(t)} \tag{4.8}$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \circ \tanh(\mathbf{c}^{(t)}) \tag{4.9}$$

where σ here denotes sigmoid function, $\mathbf{W}$ and $\mathbf{b}$ are the corresponding weight matrices and bias vectors. The $\tanh(\mathbf{v})$ operates piecewise on each element of the vector $\mathbf{v}$. The operation $\circ$ is the Hadamard product [160].

The final layer in Fig. 4.2 is a simple dense layer with fully connected neurons which converts the output $\mathbf{h}^{(t)}$ of the LSTM to a vector $\mathbf{y}^{(t)}$ in which each entry denotes the categorical probability of the representative value for the next time step $t+1$. The loss function J for minimization during training at every timestep t is then defined as the cross entropy between the output of the model $\hat{\mathbf{y}}^{(t)}$ and the actual probability for the next timestep $\hat{\mathbf{y}}^{(t)}$ which is just the one-hot vector $\mathbf{s}^{t+1}$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{D}_d \mathbf{h}^{(t)} + \mathbf{b}_d) \tag{4.10}$$

$$J = -\sum_{t=0}^{T-1} \mathbf{y}^{(t)} \cdot \ln \hat{\mathbf{y}}^{(t)} = -\sum_{t=0}^{T-1} \mathbf{s}^{(t+1)} \cdot \ln \hat{\mathbf{y}}^{(t)} \tag{4.11}$$

where T is the total length of trajectory, and the final loss function is the sum over the

whole time series. The $\mathrm{softmax}(\mathbf{x})_i = \exp(\mathbf{x}_i)/\sum_j \exp(\mathbf{x}_j)$ is a softmax function mapping $\mathbf{x}$ to a probability vector $\hat{\mathbf{y}}$.

4.2.2 Training the network is equivalent to learning path probability

The central finding of this work, which we will demonstrate through numerical results for different systems, is that a LSTM framework used to model languages can also be used to capture kinetic and thermodynamic aspects of dynamical trajectories prevalent in chemical and biological physics. In this section we demonstrate theoretically as to why LSTMs possess such a capability. Before we get into the mathematical reasoning, we first state our key idea. Minimizing the loss function J in LSTM (Eq. 4.11), which trains the model at time t to generate output $\hat{\mathbf{y}}^{(t)}$ resembling the target output $\mathbf{s}^{t+1}$, is equivalent to minimizing the difference between the actual and LSTM-learned path probabilities. This difference between path probabilities can be calculated as a cross-entropy J' defined as:

$$J' = -\sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)}...\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)}...\mathbf{x}^{(0)}) \tag{4.12}$$

where $P(\mathbf{x}^{(t+1)}, ..., \mathbf{x}^{(0)})$ and $Q(\mathbf{x}^{(t+1)}, ..., \mathbf{x}^{(0)})$ are the corresponding true and neural network learned path probabilities of the system. Eq. 4.12 can be rewritten [161] as the sum of path entropy $H(P)$ for the true distribution P and Kullback-Liebler distance D_{KL} between P and Q: $J' = H(P) + D_{KL}(P||Q)$. Since D_{KL} is strictly non-negative [161] attaining the value of 0 iff $Q = P$, the global minimum of J' happens when $Q = P$ and J' equals the path entropy $H(P)$ of the system [125]. Thus we claim that minimizing the loss function in LSTM is equivalent to learning the path entropy of the underlying

physical model, which is what makes it capable of capturing kinetic information of the dynamical trajectory.

To prove this claim we start with rewriting J in Eq. 4.11. For a long enough observation period T or for a very large number of trajectories, J can be expressed as the cross entropy between conditional probabilities:

$$J = -\sum_{t=0}^{T-1} \sum_{\mathbf{x}^{(t+1)}} P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}...\mathbf{x}^{(0)}) \times \ln Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}...\mathbf{x}^{(0)}) \tag{4.13}$$

where $P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}...\mathbf{x}^{(0)})$ is the true conditional probability for the physical system, and $Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}...\mathbf{x}^{(0)})$ is the conditional probability learned by the neural network. The minimization of Eq. 4.13 leads to minimization of the cross entropy J'.

In the following derivation, we will show how we obtain J' as an estimate of minimizing cross entropy J in Eq. 4.11. To begin with, we start with the cross entropy J:

$$\begin{aligned} J &= -\sum_{t=0}^{T-1} \mathbf{y}^{(t)} \cdot \ln \hat{\mathbf{y}}^{(t)} \\ &= -\sum_{t=0}^{T-1} \sum_{\mathbf{s}^{(t+1)}} P(\mathbf{s}^{(t+1)}|\mathbf{s}^{(t)}, ...\mathbf{s}^{(0)}) \ln \hat{\mathbf{y}}^{(t)} \end{aligned} \tag{4.14}$$

where $P(\mathbf{s}^{(t+1)}|\mathbf{s}^{(t)}, ...\mathbf{s}^{(0)})$ is the conditional probability of the physical system computed from the one-hot vectors of the data. Even if the trajectory has dependency on its long-

term history, as long as trajectory length $T \gg 0$, we can approximate Eq. 4.14 as:

$$J \approx -\sum_{t=0}^{T-1} \sum_{\mathbf{s}^{(t+1)}} \Pr(\mathbf{s}^{(t+1)}|\mathbf{s}^{(t)}, ...\mathbf{s}^{(t-T)}) \ln \hat{\mathbf{y}}^{(t)} \tag{4.15}$$

by letting $\mathbf{s}^{(m)} = 0$ for all negative m. As is typical in character-level language models [162], we assume the embedding dimension M is much greater than the input dimension N and rewrite the above equation as:

$$J = T\left[-\frac{1}{T}\sum_{t=0}^{T-1} \sum_{\mathbf{x}^{(t+1)}} P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}, ...\mathbf{x}^{(t-T+1)}) \ln Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}, ...\mathbf{x}^{(t-T)})\right] \tag{4.16}$$

$$= T\left[\frac{1}{T}\sum_{t=0}^{T-1} \bar{J}^{(t)}(\mathbf{x}^{(t)}, ...\mathbf{x}^{(t-T+1)})\right] \tag{4.17}$$

where $\tilde{J}^{(t)}(\mathbf{x}^{(t)}, ...\mathbf{x}^{(t-T+1)}) \equiv -\sum_{\mathbf{x}^{(t+1)}} P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}, ...\mathbf{x}^{(t-T+1)}) \ln Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}, ...\mathbf{x}^{(t-T+1)})$ is the cross entropy between conditional probabilities. With large enough T, we can also assume ergodicity and convert the time average to ensemble average,

$$J \approx T \sum_{\mathbf{x}^{(T-1)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T-1)}, ..., \mathbf{x}^{(0)}) \bar{J}(\mathbf{x}^{(T-1)}, ...\mathbf{x}^{(0)}) \tag{4.18}$$

$$= -T \sum_{\mathbf{x}^{(T)}} \sum_{\mathbf{x}^{(T-1)}, \mathbf{x}^{(T-2)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T-1)}, ..., \mathbf{x}^{(0)}) P(\mathbf{x}^{(T)}|\mathbf{x}^{(T-1)}..., \mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)}|\mathbf{x}^{(T-1)}..., \mathbf{x}^{(0)}) \tag{4.19}$$

$$= -T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)}, ..., \mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)}|\mathbf{x}^{(T-1)}..., \mathbf{x}^{(0)}) \tag{4.20}$$

The cross entropy J achieves its global minima when Q approaches P :

$$Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)},...\mathbf{x}^{(t-T+1)}) \rightarrow P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)},...\mathbf{x}^{(t-T+1)}) \quad \forall\, t \tag{4.21}$$

We also know that

$$Q(\mathbf{x}^{(T)}|\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)}) = \frac{Q(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)})}{Q(\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)})} \tag{4.22}$$

Plugging Eq. 4.22 in Eq. 4.20,

$$\begin{aligned}
J &= -T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)}|\mathbf{x}^{T-1},...,\mathbf{x}^{(0)}) \\
&= -T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} \left[P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) - P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)})\right] \\
&= -T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) + T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)}) \\
&= J' - J''
\end{aligned}$$

where $Q(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \rightarrow P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)})$ during the minimization. Here we have defined J' and J'' as follows:

$$J' \equiv -T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \tag{4.23}$$

$$J'' \equiv -T \sum_{\mathbf{x}^{(T)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)}) \tag{4.24}$$

$$= -T \sum_{\mathbf{x}^{(T-1)}...\mathbf{x}^{(0)}} P(\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T-1)},...,\mathbf{x}^{(0)}) \tag{4.25}$$

Since the summations run over the state space such that the normalization condition of P and Q holds, according to Gibbs' inequality J' and J'' both are well-defined cross entropies, and their global minima happen when $Q = P$. Therefore, minimizing J leads to minimization of both J' and J'', at which point both J' and J'' are path entropies.

We can also conversely show how Eq. 4.12 reduces to Eq. 4.11 by assuming a stationary first-order Markov process as in Ref. [125]:

$$\begin{aligned} P(\mathbf{x}^{(T)}...\mathbf{x}^{(0)}) &= P(\mathbf{x}^{(T)}|\mathbf{x}^{(T-1)})...P(\mathbf{x}^{(1)}|\mathbf{x}^{(0)})P(\mathbf{x}^{(0)}) \\ Q(\mathbf{x}^{(T)}...\mathbf{x}^{(0)}) &= Q(\mathbf{x}^{(T)}|\mathbf{x}^{(T-1)})...Q(\mathbf{x}^{(1)}|\mathbf{x}^{(0)})Q(\mathbf{x}^{(0)}) \end{aligned} \tag{4.26}$$

where $P(\mathbf{x}_j^{(t+1)}|\mathbf{x}_i^{(t)}) \equiv P_{ij}$ is the transition probability from state $\mathbf{x}_i$ to state $\mathbf{x}_j$ and $P(\mathbf{x}_k^{(0)}) \equiv P_k$ is the occupation probability for the single state $\mathbf{x}_k$. Plugging Eq. 4.26 into Eq. 4.12, and following the derivation in Ref. [125] with the constraints

$$\sum_j P_{ij} = 1 \quad \sum_i P_i P_{ij} = P_j \tag{4.27}$$

we arrive at an expression for the cross-entropy J, which is very similar to the path entropy type expressions derived for instance in the framework of Maximum Caliber [125]:

$$J' = -\sum_i P_i \ln Q_i - T \sum_{lm} P_l P_{lm} \ln(Q_{lm}) \tag{4.28}$$

$$\rightarrow -T \sum_{lm} P(\mathbf{x}_l) P(\mathbf{x}_m|\mathbf{x}_l) \ln Q(\mathbf{x}_m|\mathbf{x}_l) \tag{4.29}$$

In Eq. 4.28 as the trajectory length T increases, the second term dominates in the estimate

of J leading to Eq. 4.29. This second term is the ensemble average of a time-dependent quantity $\tilde{J}(\mathbf{x}_l^{(t)}) \equiv -\sum_m P(\mathbf{x}_m^{(t+1)}|\mathbf{x}_l^{(t)}) \ln Q(\mathbf{x}_m^{(t+1)}|\mathbf{x}_l^{(t)})$. For a large enough T, the ensemble average can be replaced by the time average. By assuming ergodicity [163]:

$$J' = -\sum_{t=1}^{T}\sum_{m} P(\mathbf{x}_m^{(t+1)}|\mathbf{x}_l^{(t)}) \ln Q(\mathbf{x}_m^{(t+1)}|\mathbf{x}_l^{(t)}) \tag{4.30}$$

from which we directly obtain Eq. 4.11. Therefore, under first-order Markovianity and ergodicity, minimizing the loss function J of Eq. 4.11 is equivalent to minimizing J' and thereby learning the path entropy.

4.2.3 Embedding layer captures kinetic distances

In word embedding theory, the embedding layer provides a measure of similarity between words. However, from the path probability representation, it is unclear how the embedding layer works since the derivation can be done without embedding vectors $\mathbf{x}$. To have an understanding to Q_{lm} in the first-order Markov process, we first write the conditional probability $Q_{lm} = Q(\mathbf{x}_m^{(t+1)}|\mathbf{x}_l^{(t)})$ explicitly with softmax defined in Eq. 4.10 and embedding vectors $\mathbf{x}$ defined in Eq. 4.3:

$$\begin{aligned} Q_{lm} &= \frac{\exp(\mathbf{s}_m^{(t+1)} \cdot (\mathbf{D}_d \mathbf{h}^{(t)} + \mathbf{b}_d))}{\sum_k \exp(\mathbf{s}_k \cdot (\mathbf{D}_d \mathbf{h}^{(t)} + \mathbf{b}_d))} \\ &= \frac{\exp(\mathbf{s}_m^{(t+1)} \cdot (\mathbf{D}_d f_{\boldsymbol{\theta}}(\mathbf{x}^{(t)}) + \mathbf{b}_d))}{\sum_k \exp(\mathbf{s}_k \cdot (\mathbf{D}_d f_{\boldsymbol{\theta}}(\mathbf{x}^{(t)}) + \mathbf{b}_d))} \end{aligned} \tag{4.31}$$

where f is the recursive function $\mathbf{h}^{(t)} = f_{\boldsymbol{\theta}}(\mathbf{x}^{(t)}, \mathbf{h}^{(t-1)}) \approx f_{\boldsymbol{\theta}}(\mathbf{x}^{(t)})$ which is defined with the update equation in Eq. 4.4-4.9. In Eq. 4.31, $\boldsymbol{\theta}$ denotes various parameters including all

weight matrices and biases, and the summation index k runs over all possible states. Now we can use multivariable Taylor's theorem to approximate $f_{\boldsymbol{\theta}}$ as the linear term around a point $\mathbf{a}$ as long as $\mathbf{a}$ is not at any local minimum of $f_{\boldsymbol{\theta}}$:

$$f_{\boldsymbol{\theta}}(\mathbf{x}^{(t)}) \approx f_{\boldsymbol{\theta}}(\mathbf{a}) + \mathbf{A}_{\boldsymbol{\theta}}(\mathbf{x}^{(t)} - \mathbf{a}) \tag{4.32}$$

where $\mathbf{A}_{\boldsymbol{\theta}}$ is the L by M matrix defined to be $(\mathbf{A}_{\boldsymbol{\theta}})_{ij} = \frac{\partial (f_{\boldsymbol{\theta}})_i}{\partial x_j}|_{\mathbf{x}=\mathbf{a}}$. Then Eq. 4.31 becomes

$$Q_{lm} = \frac{\exp(C_m^{(t+1)}) \exp(\mathbf{s}_m^{(t+1)} \cdot \mathbf{D}_d \mathbf{A}_{\boldsymbol{\theta}} \mathbf{x}_l^{(t)})}{\sum_k \exp(C_k) \exp(\mathbf{s}_k \cdot \mathbf{D}_d \mathbf{A}_{\boldsymbol{\theta}} \mathbf{x}_l^{(t)})} \tag{4.33}$$

where $C_i^{(t+1)} = \mathbf{s}_i^{(t+1)} \cdot [\mathbf{D}_d(f_{\boldsymbol{\theta}}(\mathbf{a}_l) + \mathbf{A}_{\boldsymbol{\theta}}\mathbf{a}_l) + \mathbf{b}_d]$. We can see in Eq. 4.33 how the embedding vectors come into the transition probability. Specifically, there is a symmetric form between output one-hot vectors $\mathbf{s}_m^{(t+1)}$ and the input one-hot vectors $\mathbf{s}^{(t)}$, in which $\mathbf{x}^{(t)} = \mathbf{\Lambda}\mathbf{s}^{(t)}$ and $\mathbf{\Lambda}$ is the input embedding matrix, $\mathbf{D}_d\mathbf{A}_{\boldsymbol{\theta}}$ can be seen as the output embedding matrix, and $C_i^{(t+1)}$ is the correction of time lag effect. While we don't have an explicit way to calculate the output embedding matrix so defined, Eq. 4.33 motivates us to define the following ansatz for the transition probability:

$$Q_{lm} = Q(\mathbf{x}_m|\mathbf{x}_l) = \frac{\exp(\mathbf{x}_m \cdot \mathbf{x}_l)}{\sum_k \exp(\mathbf{x}_k \cdot \mathbf{x}_l)} \tag{4.34}$$

where $\mathbf{x}_m$ and $\mathbf{x}_l$ are both calculated by the input embedding matrix $\mathbf{\Lambda}$. The expression in Eq. 4.34 is thus a tractable approximation to the more exact transition probability in Eq. 4.33. Furthermore, we show through numerical examples of test systems that our ansatz for Q_{lm} does correspond to the kinetic connectivity between states. That is, the

LSTM embedding layer with the transition probability through Eq. 4.34 can capture the average commute time between two states in the original physical system, irrespective of the quality of low-dimensional projection fed to the LSTM [117, 118, 164].

4.3 Model set-up and simulation details

In this section, we will introduce the model potentials and simulations that we will use later in the discussion of Sec. 4.4.

4.3.1 Model potential details

All model potentials have two degrees of freedom x and y. Our first two models (shown in Fig. 4.8**a** and Fig. 4.8**b**) have three metastable states with governing potential $U(x, y)$ given by

$$\begin{aligned} U(x,y) = W(x^6 + y^6) - G(x,x_1)G(y,y_1) \\ - G(x,x_2)G(y,y_2) - G(x,x_3)G(y,y_3) \end{aligned} \tag{4.35}$$

where $W = 0.0001$ and $G(x, x_0) = e^{-\frac{(x-x0)^2}{2\sigma^2}}$ denotes a Gaussian function centered at x_0 with width $\sigma = 0.8$. We also build a 4-state model system with governing interaction

potential:

$$
\begin{aligned}
U(x,y) = W(x^4 + y^4) + G(x, 0.0)G(y, 0.0) \\
- G(x, 2.0)G(y, -1.0) - G(x, 0.5)G(y, 2.0) \\
- G(x, -0.5)G(y, -2.0) - G(x, -2.0)G(y, 1.0)
\end{aligned} \tag{4.36}
$$

The different local minima corresponding to the model potentials in Eq. 4.35 and Eq. 4.36 are illustrated in Fig. 4.8. We call these as linear 3-state, triangular 3-state, and 4-state models respectively. The free energy surfaces generated from the simulation of Langevin dynamics [165] with these model potentials are shown in Figs. 4.8**a-c**.

4.3.2 Molecular dynamics details

The integration timestep for the Langevin dynamics simulation was 0.01 units, and the simulation was performed at $\beta = 9.5$ for linear 3-state and 4-state potentials and $\beta = 9.0$ for triangular 3-state potential, where $\beta = 1/(k_B T)$. The MD trajectory for alanine dipeptide was obtained using the software GROMACS 5.0.4 [128, 129], patched with PLUMED 2.4 [102]. The temperature was kept constant at 450K using the velocity rescaling thermostat [100].

4.3.3 Representative trajectories and data pre-processing

In this section we provide representative trajectories obtained for the different systems described in the main text, where Fig. 4.3 is for linear 3-state model potential, Fig. 4.4 is for triangular 3-state model potential, Fig. 4.5 is for 4-state model potential,

Fig. 4.6 is for alanine dipeptide, and Fig. 4.7 is for force spectroscopy trajectory. We also show how we removed ephemeral/spurious transitions by smoothening the binned trajectory before feeding into the LSTM model, in order to make the learning process more stable. The emphemoral/spurious transitions happened due to the use of only 3 labels for 3-state model systems and 4 labels for 4-state model system when projecting on the x-axis. It is not related to the learning quality of LSTM. With more labels or finer binning, we can avoid producing those spurious states and train the LSTM without smoothening binned trajectory before feeding into the model. This can be seen from the example of alanine dipeptide (Fig. 4.6), where we did not smoothen the trajectory.

4.4 Results

To demonstrate our ideas, here we consider a range of different dynamical trajectories. These include three model potentials, the popular model molecule alanine dipeptide, and trajectory from single molecule force spectroscopy experiments on a multi-state riboswitch [159]. The sample trajectories of these test systems and the data preprocessing strategies are shown in Sec. 4.3. When applying our neural network to the model systems, the embedding dimension M is set to 8 and LSTM unit L set to 64. When learning trajectories for alanine dipeptide and riboswitch, we took $M = 128$ and $L = 1024$. All time series were batched into sequences with a sequence length of 100 and the batch size of 64. For each model potential, the neural network was trained using the method of stochastic gradient descent for 20 epochs until the training loss becomes smaller than the validation loss, which means an appropriate training has been reached. For alanine dipeptide, 40

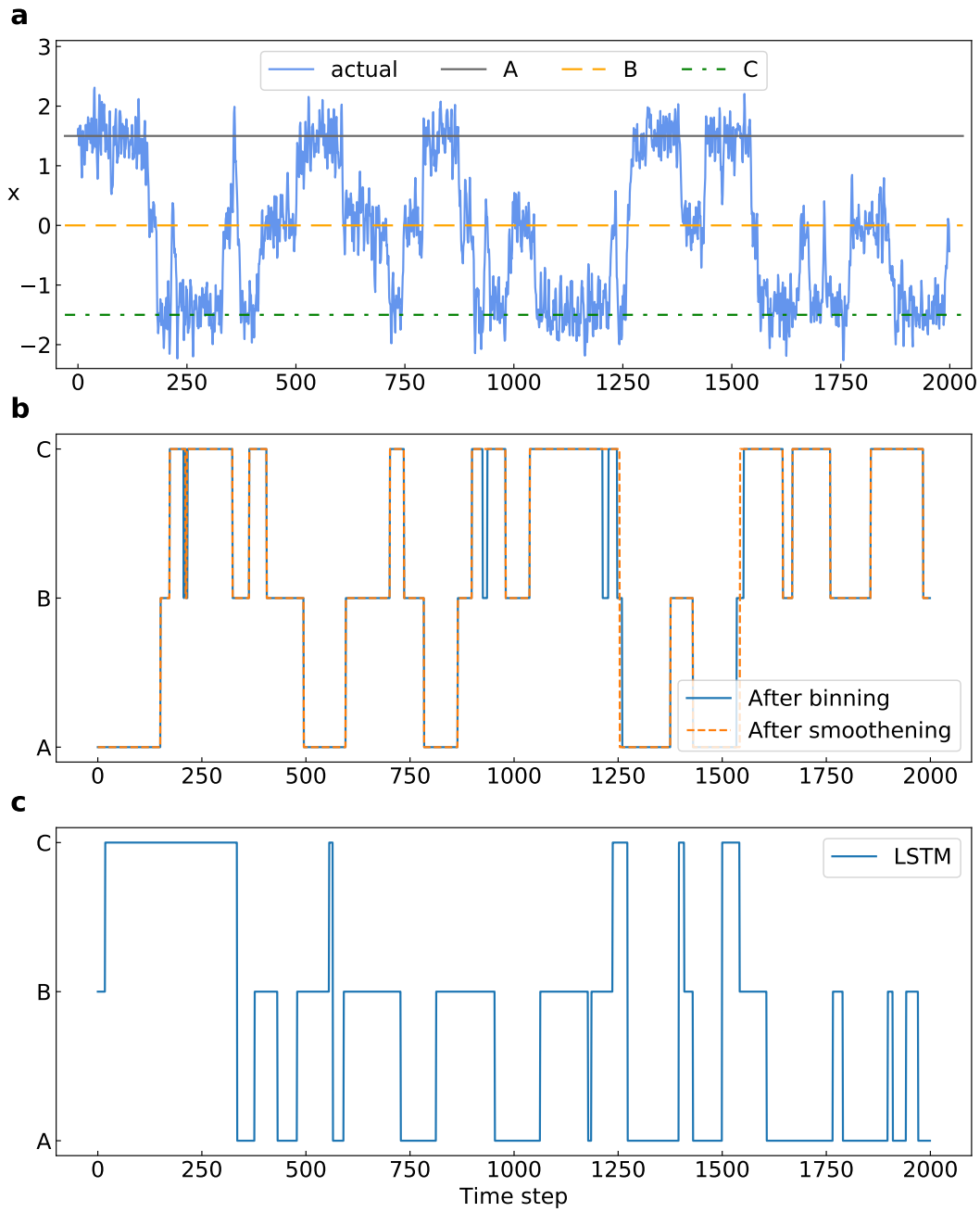


Figure 4.3: **Trajectories for linear 3-state model potential.** **a** The actual trajectory with location of metastable states shown by horizontal solid and dashed lines. **b** The trajectory after spatial discretization, where the trajectory now consists of a sequence of labels representing metastable states. To make the learning process more stable, we removed ephemeral/spurious transitions by smoothening before feeding into the LSTM model. **c** The trajectory generated by our LSTM model.

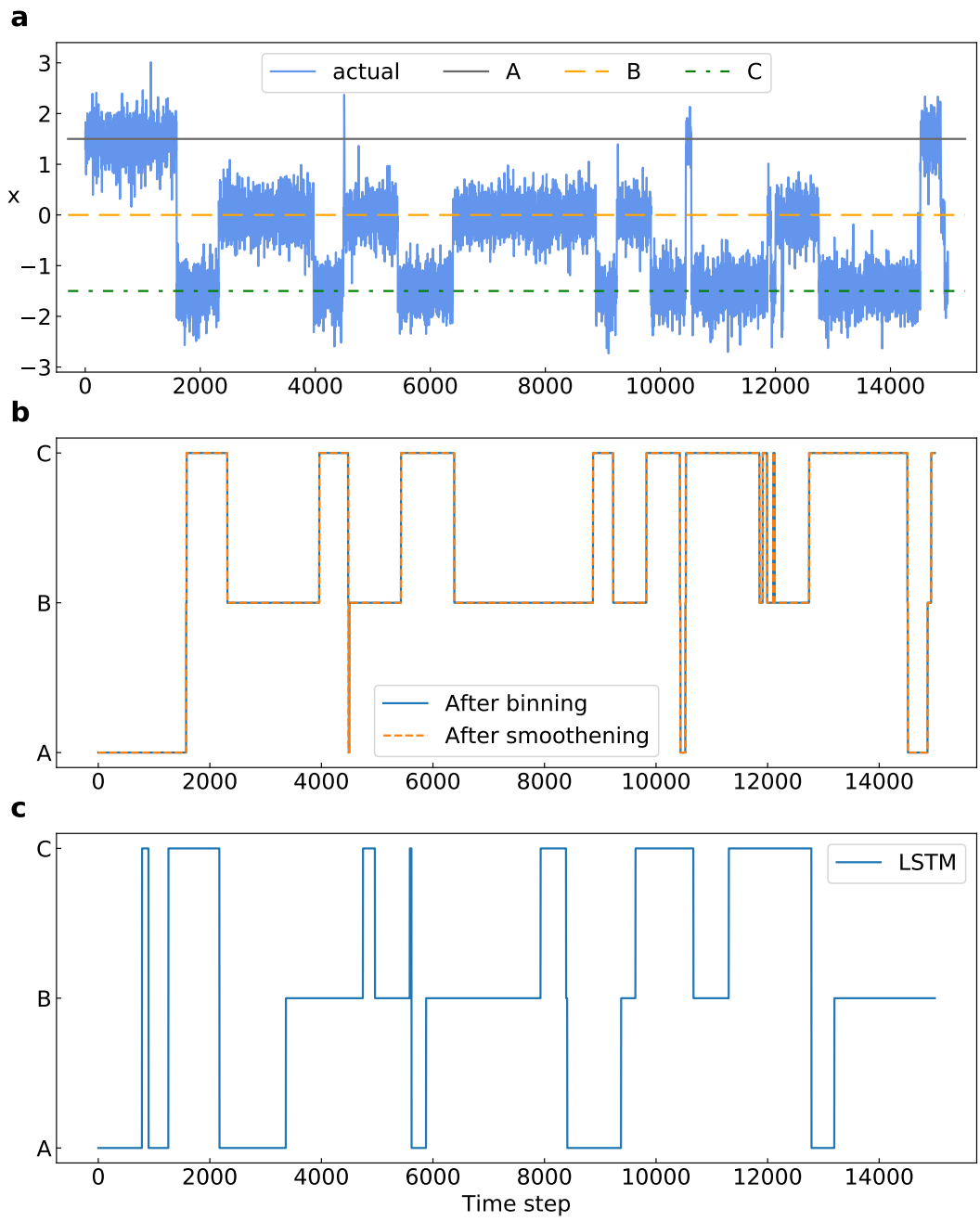


Figure 4.4: **Trajectories for triangular 3-state model potential. a** The actual trajectory with location of metastable states shown by horizontal solid and dashed lines. **b** The trajectory after spatial discretization, where the trajectory now consists of a sequence of labels representing metastable states. To make the learning process more stable, we removed ephemeral/spurious transitions by smoothening before feeding into the LSTM model. **c** The trajectory generated by our LSTM model.

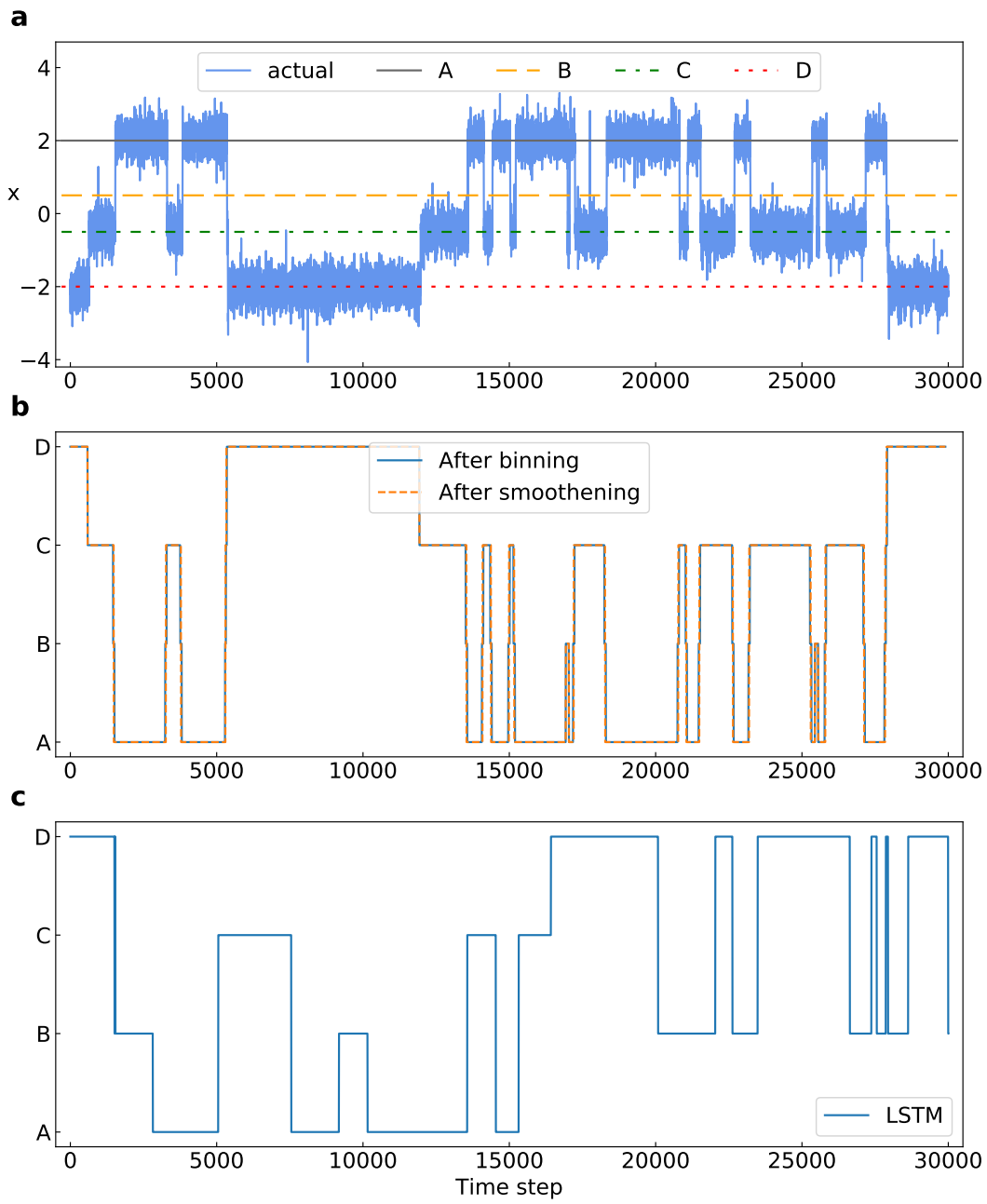


Figure 4.5: **Trajectories for 4-state model potential.** **a** The actual trajectory with location of metastable states shown by horizontal solid and dashed lines. **b** The trajectory after spatial discretization, where the trajectory now consists of a sequence of labels representing metastable states. To make the learning process more stable, we removed ephemeral/spurious transitions by smoothening before feeding into the LSTM model. **c** The trajectory generated by our LSTM model.

training epochs were used. Our neural network was built using TensorFlow version 1.10.

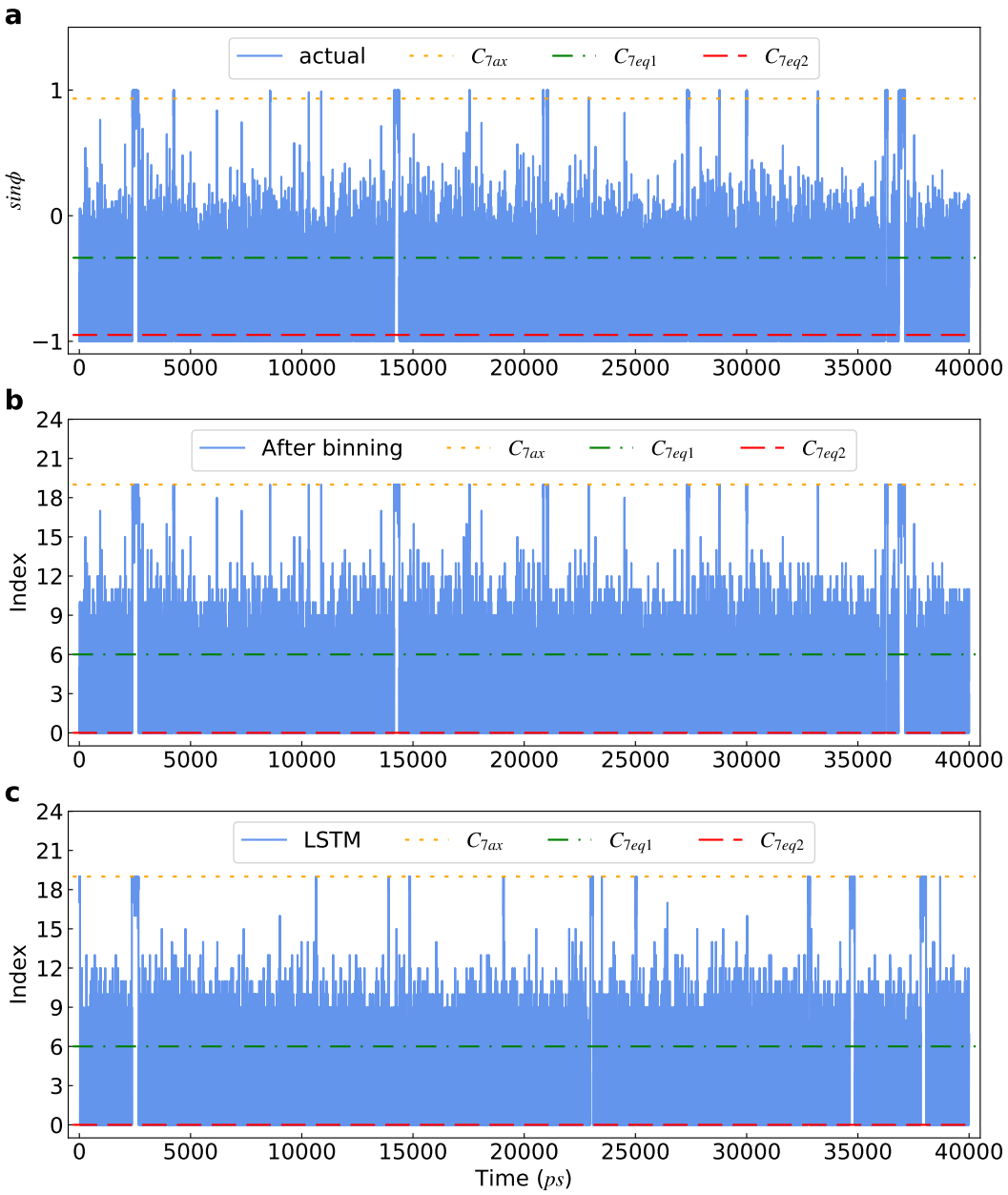


Figure 4.6: **Trajectories along** $\sin\phi$ **for alanine dipeptide. a** The actual MD trajectory with location of metastable states shown by horizontal dashed lines. **b** The trajectory after spatial discretization into 20 indexed positions. **c** The predicted trajectory generated by our LSTM model.

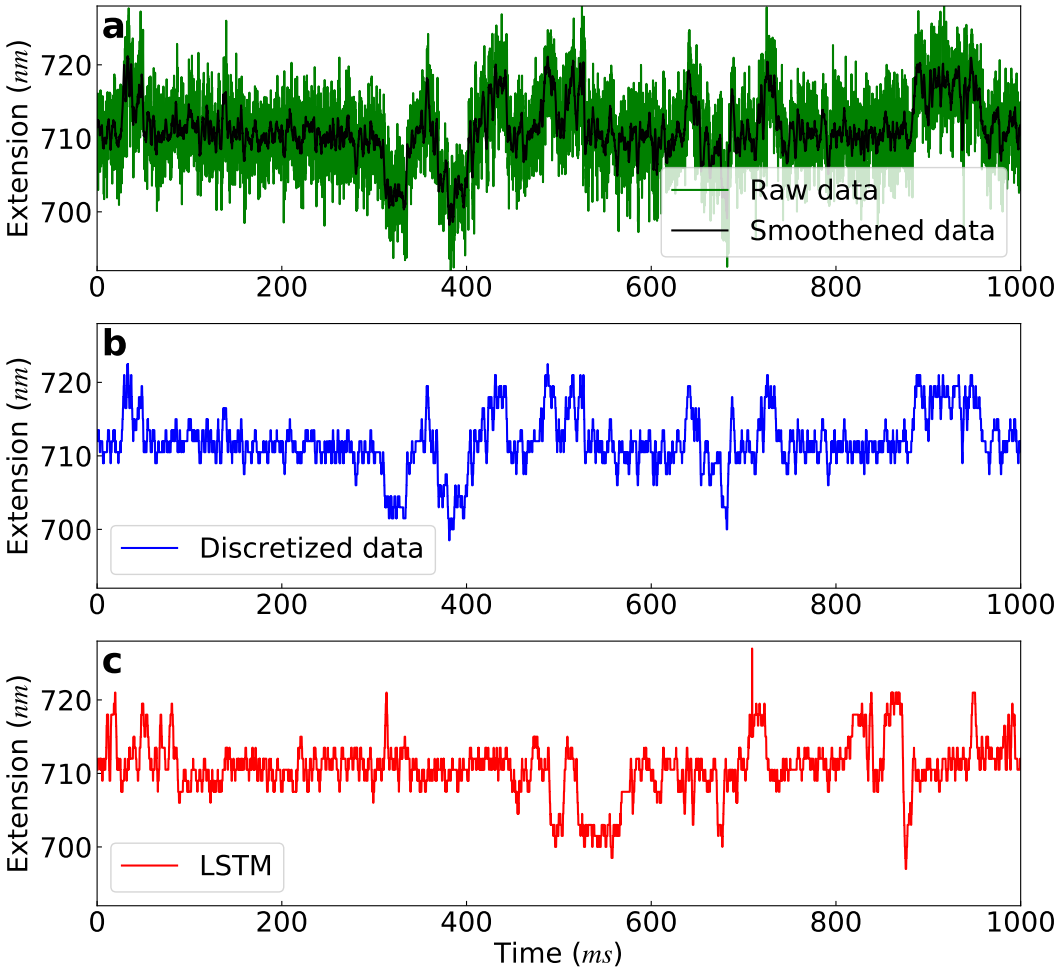


Figure 4.7: **Trajectories for single molecule force spectroscopy experiment on riboswitch. a** The original trajectory and the corresponding smoothened trajectory. **b** The trajectory after spatial discretization into 34 indexed positions. **c** The predicted trajectory generated by the LSTM model.

4.4.1 Boltzmann statistics and kinetics for model potentials

The first test we perform for our LSTM set-up is its ability to capture the Boltzmann weighted statistics for the different states in each model potential. This is the probability distribution P or equivalently the related free energy $F = -\frac{1}{\beta}\log P$, and can be calculated by direct counting from the trajectory. As can be seen in Fig. 4.8, the LSTM does an excellent job of recovering the Boltzmann probability within error bars.

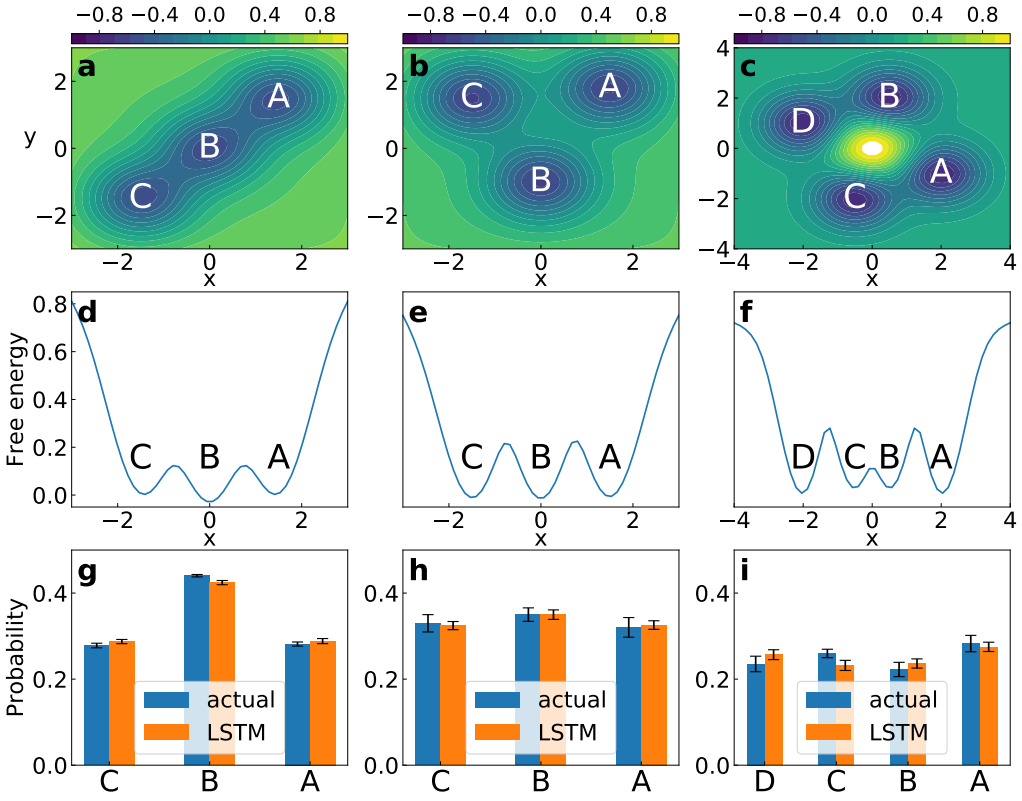


Figure 4.8: **Boltzmann statistics for model systems.** The analytical free energy generated from **a** linear 3-state, **b** triangular 3-state, **c** symmetric 4-state model potentials and **d**, **e**, **f** are the corresponding 1-dimensional projections along x-direction. In the bottom, we compare the Boltzmann probabilities of **g** linear 3-state, **h** triangular 3-state, and **i** symmetric 4-state models for every labeled state generated from actual MD simulation and from our long short-term memory (LSTM) network. The errorbars are calculated as standard errors.

Next we describe our LSTM deals with a well-known problem in analyzing high-dimensional data sets through low-dimensional projections. One can project the high-dimensional data along many different possible low-dimensional order parameters, for instance x, y or a combination thereof in Fig. 4.8. However most such projections will end up not being kinetically truthful and give a wrong impression of how distant the metastable states actually are from each other in the underlying high-dimensional space. It is in general hard to come up with a projection that preserves the kinetic properties of the high-dimensional space. Consequently, it is hard to design analysis or sampling

methods that even when giving a time-series along a sub-optimal projection, still capture the true kinetic distance in the underlying high-dimensional space.

Here we show how our LSTM model is agnostic to the quality of the low-dimensional projection in capturing accurate kinetics. Given that for each of the 3 potentials the LSTM was provided only the $x-$trajectory, we can expect that the chosen model potentials constitute different levels of difficulties in generating correct kinetics. Specifically, a one-dimensional projection along x is kinetically truthful for the linear 3-state potential in Fig. 4.8**a** but not for the triangular 3-state and the 4-state potentials in Figs. 4.8**b** and **c** respectively. For instance, Fig. 4.8**e** gives the impression that state C is kinetically very distant from state A, while in reality for this potential all 3 pairs of states are equally close to each other. Similar concerns apply to the 4-state potential.

In Figs. 4.9 and 4.10**a**-**c** and **d**-**f** we compare the actual versus LSTM-predicted kinetics for moving between different metastable states for different model potentials, for all pairs of transitions in both directions (i.e. for instance A to B and B to A). Specifically, Fig. 4.9**a**-**c** and Fig. 4.9**d**-**f** shows results for moving between the 3 pairs of states in the linear and triangular 3-state potentials respectively. Fig. 4.10 shows results for the 6 pairs of states in the 4-state potential. Furthermore, for every pair of state, we analyze the transition time between those states as a function of different minimum commitment or commit time, i.e. the minimum time that must be spent by the trajectory in a given state to be classified as having committed to it. A limiting value, and more specifically the rate at which the population decays to attain to such a limiting value, corresponds to the inverse of the rate constant for moving between those states [166, 167]. Thus here we show how our LSTM captures not just the rate constant, but time-dependent fluctuations

in the population in a given metastable state as equilibrium is attained. The results are averaged over 20 independent segments taken from the trajectories of different trials of training for the 3-state potentials and 10 independent segments for the 4-state potential.

As can be seen in Figs. 4.9 and 4.10, the LSTM model does an excellent job of reproducing well within errorbars the transition times between different metastable states for different model potentials irrespective of the quality of the low-dimensional projection. Firstly, our model does tell the differences between linear and triangular 3-state models (Fig. 4.9) even though the projected free energies along the x variable input into LSTM are same (Fig. 4.8). The number of transitions between states A and C is less than the others; while for triangular configuration, the numbers of transitions between all pairs of states are similar. The rates at which the transition count decays as a function of commitment time is also preserved between the input data and the LSTM prediction.

The next part of our second test is the 4-state model potential. In Fig. 4.10 we show comparisons for all 6 pairs of transitions in both forward and reverse directions. A few features are immediately striking here. Firstly, even though states B and C are perceived to be kinetically proximal from the free energy (Fig. 4.8), the LSTM captures that they are distal from each other and correctly assigns similar kinetic distance to the pairs B,C as it does to A,D. Secondly, there is asymmetry between the forward and backward directions (for e.g. A to D and D to A, indicating that the input trajectory itself has not yet sufficiently sampled the slow transitions in this potential. As can be seen from Fig. 4.8**c** the input trajectory has barely 1 or 2 direct transitions for the very high barrier A to D or B to C. This is a likely explanation for why our LSTM model does a bit worse than in the other two model potentials in capturing the slowest transition rates, as well as the higher error

bars we see here. In other words, so far we can conclude that while our LSTM model can capture equilibrium probabilities and transition rates for different model potentials irrespective of the input projection direction or order parameter, it is still not a panacea for insufficient sampling itself, as one would expect.

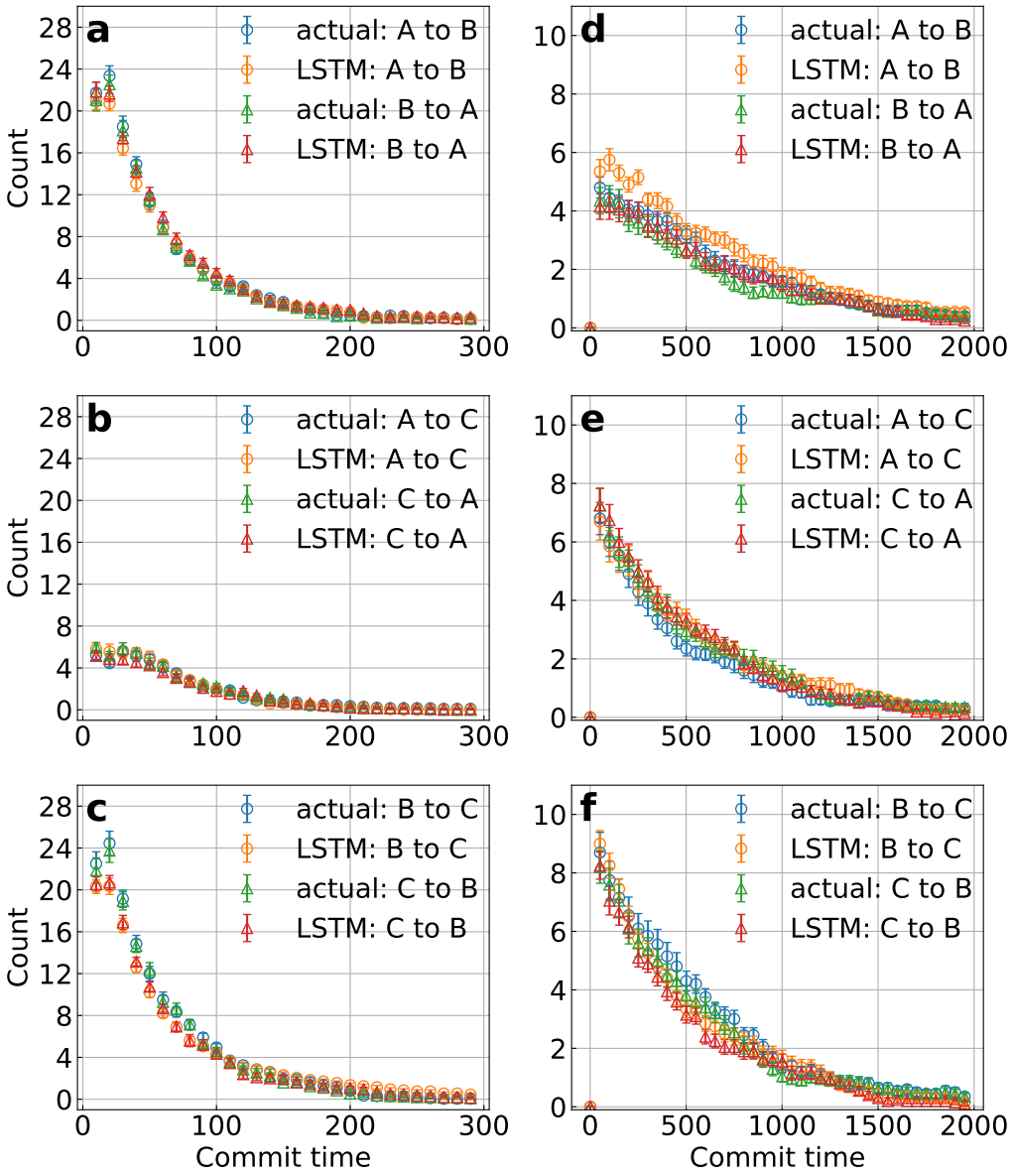


Figure 4.9: **Kinetics for 3-state model systems.** Number of transitions between different pairs of metastable states as a function of commitment time defined in **Results**. The calculations for linear and triangular configurations are shown in **a**-**c** and **d**-**f** respectively. Error bars are illustrated and were calculated as standard errors.

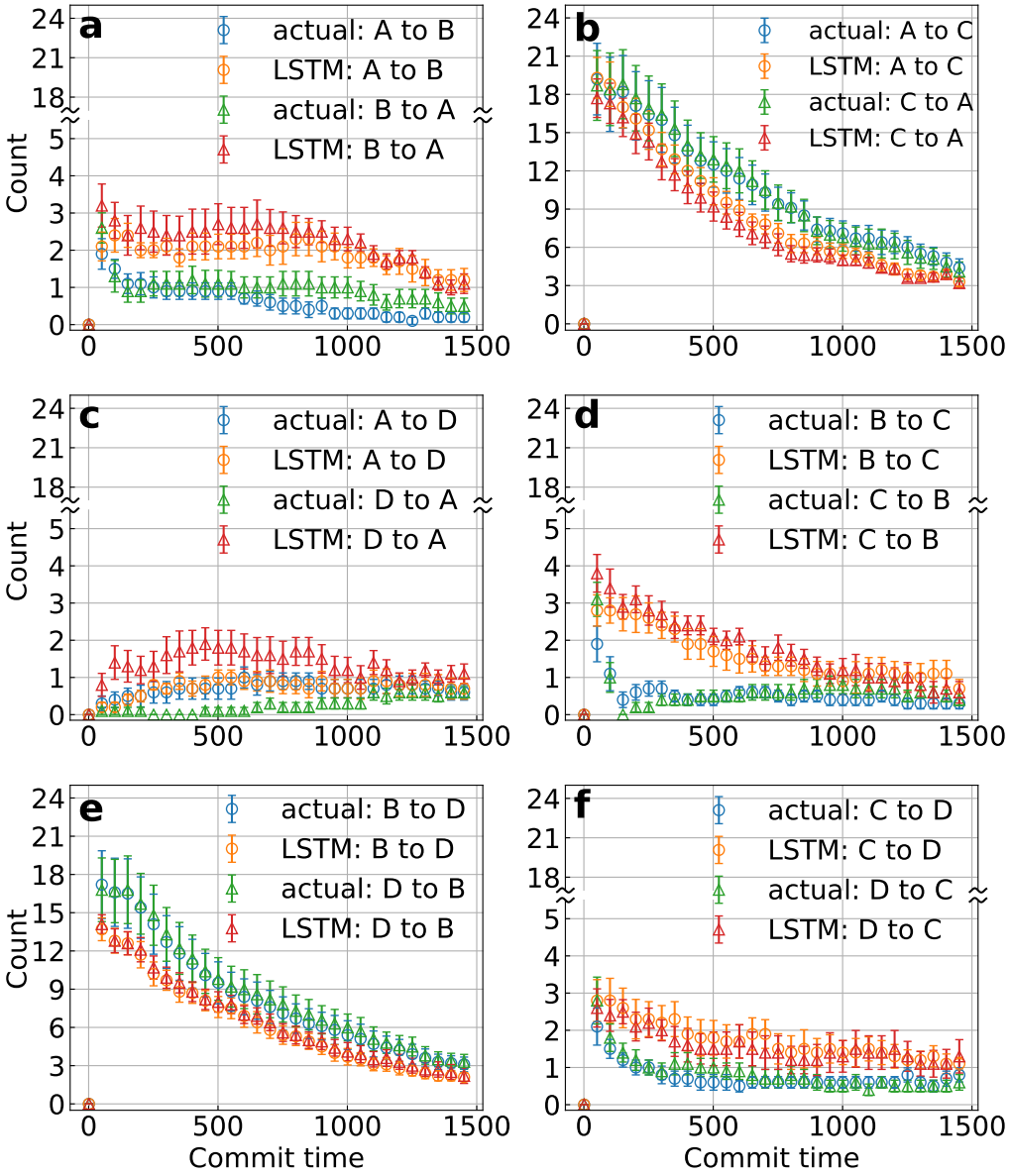


Figure 4.10: **Kinetics for 4-state model system.** Number of transitions between different pairs of metastable states as a function of commitment time defined in **Results** for 4-state model system. Error bars are illustrated and were calculated as standard errors.

4.4.2 Boltzmann statistics and kinetics for alanine dipeptide

Finally, we apply our LSTM model to the study of conformational transitions in alanine dipeptide, a model biomolecular system comprising 22 atoms, experiencing thermal fluctuations when coupled to a heat bath. The structure of alanine dipeptide is shown

Alanine dipeptide			
CVs	Label	C_{7eq} to C_{7ax} (ps)	C_{7ax} to C_{7eq} (ps)
$\sin\phi$	actual	5689.22 $\pm$ 962.366	107.93 $\pm$ 11.267
	LSTM	5752.16 $\pm$ 710.399	103.81 $\pm$ 14.268
$\sin\psi$	actual	5001.42 $\pm$ 643.943	105.70$\pm$ 13.521
	LSTM	4325.01 $\pm$ 526.293	81.68 $\pm$ 10.288

Table 4.1: **Kinetics for alanine dipeptide.** Inverse of transition rates for conformational transitions in alanine dipetide calculated from actual MD trajectories of LSTM model. Here we show the calculation along two different CVs: $\sin\phi$ and $\sin\psi$.

in Fig. 4.11**a**. While the full system comprises around 63 degrees of freedom, typically the torsional angles ϕ and ψ are used to identify the conformations of this peptide. Over the years a large number of methods have been tested on this system in order to perform enhanced sampling of these torsions, as well as to construct optimal reaction coordinates [28, 119, 168, 169]. Here we show that our LSTM model can very accurately capture the correct Boltzmann statistics as well as transition rates for moving between the two dominant metastable states known as C_{7eq} and C_{7ax}. Importantly, the reconstruction of the equilibrium probability and transition kinetics, as shown in Fig. 4.11 and Table 4.1 is extremely accurate irrespective of the choice of one-dimensional projection time series fed into the LSTM. Specifically, we do this along $\sin\phi$ and $\sin\psi$, both of which are known to quite distant from an optimized kinetically truthful reaction coordinate [116, 158], where again we have excellent agreement between input and LSTM-predicted results.

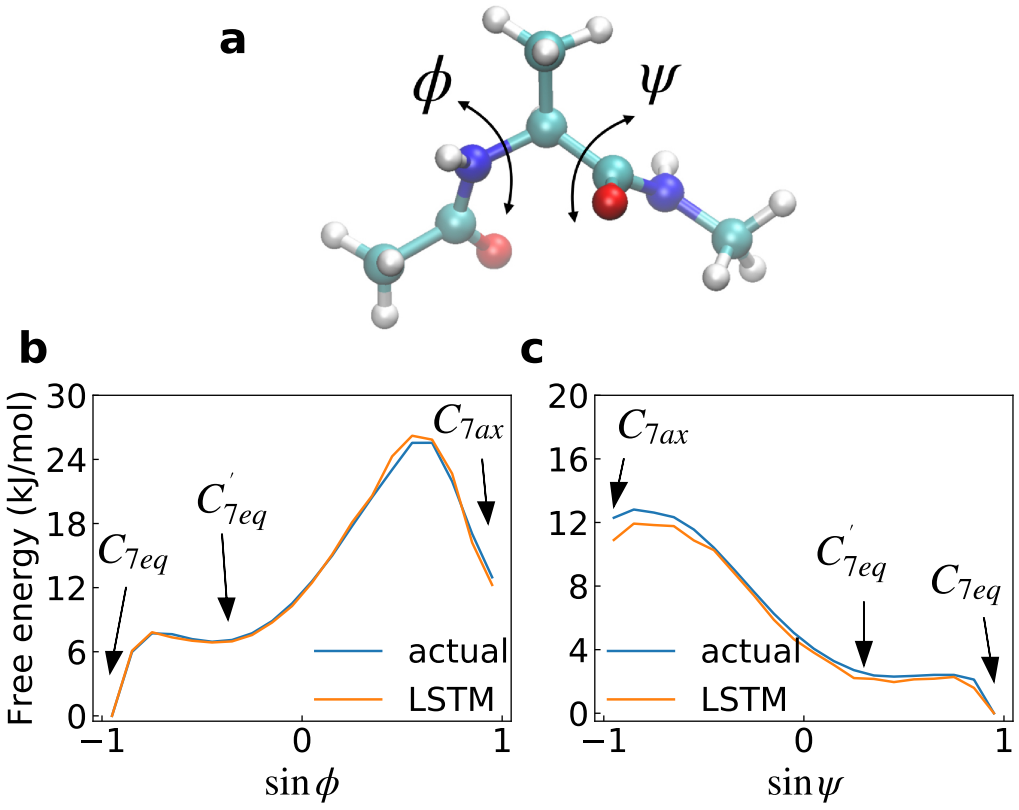


Figure 4.11: **Boltzmann statistics for alanine dipeptide.** **a** The molecular structure of alanine dipeptide used in the actual MD simulation. The torsional angles ϕ and ψ as the collective variables (CVs) are shown. **b** and **c** The 1-dimensional free energy curves along $\sin\phi$ and $\sin\psi$ are calculated using actual MD data and the data generated from LSTM.

4.4.3 Learning from single molecule force spectroscopy trajectory

In this section, we use our LSTM model to learn from single molecule force spectroscopy experiments of a multi-state riboswitch performed with a constant force of 10.9 pN. The data points are measured at 10 kHz (i.e., every 100 μs). Other details of the experiments can be found in Ref. [159]. The trajectory for a wide range of extensions starting 685 nm up to 735 nm was first spatially discretized into 34 labels, and then converted to a time series of one hot vectors, before being fed into the LSTM model. The results are shown in Fig. 4.12. In Fig. 4.12**a**, we have shown an agreement between a profile of

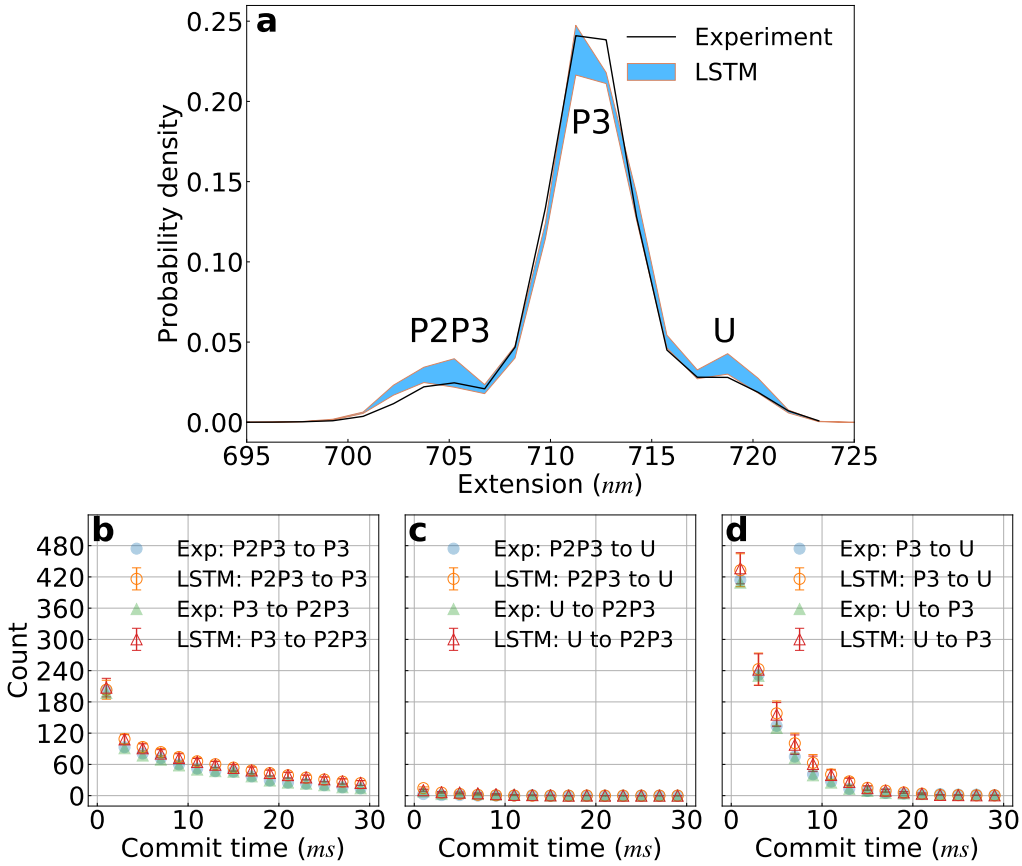


Figure 4.12: **Boltzmann statistics and kinetics for riboswitch.** Using LSTM model to learn thermodynamics and kinetics from a folding and unfolding trajectory taken from a single molecule force spectroscopy measurement [159]: **a** Comparison between the probability density learned by the LSTM model and calculated from the experimental data. The regions between errorbars defined as standard errors are filled with blue color. **b**-**d** Commit time plots calculated by counting the transitions in the trajectory generated by LSTM and the experimental trajectory. The commit time is the minimum time that must be spent by the trajectory in a given state to be classified as having committed to it. Error bars are illustrated and were calculated as standard errors.

probability density averaged over 5 independent training sets with the probability density calculated from the experimental data. Starting from the highest extension, the states are fully unfolded (U), longer intermediate (P3) and shorter intermediate (P2P3) [159]. From Fig. 4.12**b**-**c**, we see that the LSTM model captures the kinetics for moving between all 3 pairs of states for a very wide range of commitment times.

4.4.4 Embedding layer based kinetic distance

In Eq. 4.33, we derived a non-tractable relation for conditional transition probability in the embedding layer, and then through Eq. 4.34 we introduced a tractable ansatz in the spirit of Eq. 4.33. Here we revisit and numerically validate Eq. 4.34. Specifically, given any two embedding vectors $\mathbf{x}_l$ and $\mathbf{x}_m$ calculated from any two states l and m, we estimate the conditional probability Q_{lm} using Eq. 4.34. We use Q_i to denotes the Boltzmann probability predicted by the LSTM model. We then write down the interconversion probability k_{lm} between states l and m as:

$$k_{lm} = Q_l Q_{lm} + Q_m Q_{ml} \equiv 1/t_{lm} \tag{4.37}$$

From inverting this rate we then calculate an LSTM-kinetic time as $t_{lm} \equiv 1/k_{lm} = 1/(Q_l Q_{lm} + Q_m Q_{ml})$. In Fig. 4.13, we compare t_{lm} with the actual transition time τ_{lm} obtained from the input data, defined as

$$\tau_{lm} = T/\langle N_{lm} \rangle \tag{4.38}$$

Here N_{lm} is the mean number of transitions between state l and m. As this number varies with the precise value of commitment time, we average N_{lm} over all commit times to get $\langle N_{lm} \rangle$. These two timescales t_{lm} and τ_{lm} thus represent the average commute time or kinetic distance [117, 118] between two states l and m. To facilitate the comparison between these two very differently derived timescales or kinetic distances, we rescale and

shift them to lie between 0 and 1. The results in Fig. 4.13 show that the embedding vectors display the connectivity corresponding to the original high-dimensional configuration space rather than those corresponding to the one-dimensional projection. The model captures the correct connectivity by learning kinetics, which is clear evidence that it is able to bypass the projection error along any degree of freedom. The result also explains how is it that no matter what degree of freedom we use, our LSTM model still gives correct transition times. As long as the degree of freedom we choose to train the model can be used to discern all metastable states, we can even use Eq. 4.34 to see the underlying connectivity. Therefore, the embedding vectors in LSTM can define a useful distance metric which can be used to understand and model dynamics, and are possibly part of the reason why LSTMs can model kinetics accurately inspite of quality of projection and associated non-Markvoian effects.

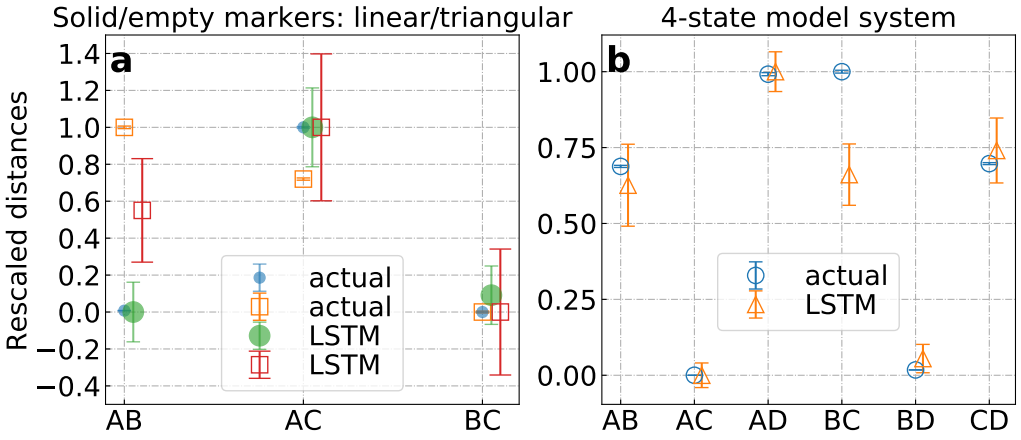


Figure 4.13: **Analysis of embedding layers for model systems.** Our analysis of the embedding layer constructed for **a** the linear and triangular 3-state and **b** the 4-state model systems. In **a**, we use solid circle and empty square markers respectively to represent linear and triangular 3-state model potentials. In each plot, the data points are shifted slightly to the right for clarity. The distances marked actual and LSTM represent rescaled mean transition times as per Eqs. 4.38 and 4.37 respectively. Error bars were calculated as standard errors over 50 different trajectories.

4.4.5 Comparing with Markov state model and Hidden Markov Model

In this section, we briefly compare our LSTM model with standard approaches for building kinetic models from trajectories, namely the Markov state model (MSM) [34] and Hidden Markov model (HMM) [170–172]. Compared to LSTM, the MSM and HMM have smaller number of parameters, making them faster and more stable for simpler systems. However, both MSM and HMM require choosing an appropriate number of states and lag time [34, 172, 173]. Large number of pre-selected states or small lag time can lead to non-Markovian behavior and result in an incorrect prediction. Even more critically, choosing a large lag time also sacrifices the temporal precision. On the other hand, there is no need to determine the lag time and number of states using the LSTM network because LSTM does not rely on the Markov property. Choosing hyperparameters such as M and L may be comparable to choosing number of hidden states for HMM, while very similar values of M and L worked for systems as different as MD trajectory of alanine dipeptide and single molecule force spectroscopy trajectory of a riboswitch. At the same time, LSTM always generates the data points with the same temporal precision as it has in the training data irrespective of the intrinsic timescales it learns from the system. In Fig. 4.14, we provide the results of using HMM and MSM for the riboswitch trajectory with the same binning method and one-hot encoded input, to be contrasted with similar plots using LSTM in Fig. 4.12. Indeed both MSM and HMM achieve decent agreement with the true kinetics only if the commit time is increased approximately beyond 10 ms, while LSTM as shown in Fig. 4.12 achieved perfect agreement for all commit times. From this figure, it can be seen that the LSTM model achieves an expected agreement

Model	System	Model training (s)	Simulate data (s)
LSTM	linear 3-state	103	53.4
	triangular 3-state	103	213.7
	4-state	183	642.0

Table 4.2: Computational effort for a LSTM run.

Model	τ	Find t_I (s)	CK test (s)	Model building (s)	Simulate data (s)
MSM	1	23.9±0.08	32.6±0.36	3.0±0.04	84.4±3.3
	5		35.6±0.36	2.9±0.04	15.4±0.22
HMM	1	48.6±1.89	597.0±7.39	79.4±1.34	290.6±4.01
	5		134.2±1.00	157.2±0.52	57.8±0.77

Table 4.3: **Computational efforts for Makov state model (MSM) and Hidden Markov Model (HMM) when analyzing trajectories from model systems.** The computational efforts for MSM and HMM when analyzing trajectories from model systems. τ and t_I are the lag time used for building the models and the implied timescale of the system. The recorded times are the sum for linear 3-state, triangular 3-state, and 4-state. The results are averaged over 5 independent runs.

with as fine of a temporal precision as desired, even though we use 20 labels for alanine dipeptide and 34 labels for experimental data to represent the states. The computational efforts needed for the various approaches (LSTM, MSM and HMM) are also provided in the Table. 4.2-4.3, where it can be seen that LSTM takes similar amount of effort as HMM. The package we used to build the MSM and HMM is PyEMMA with version 2.5.6 [174]. The models were built with lag time=0.5ms for MSM and lag time=3ms for HMM, where the HMM were built with number of hidden states=3.

4.5 Conclusion and outlook

In summary we believe this work demonstrates potential for using AI approaches developed for natural language processing such as speech recognition and machine translation, in unrelated domains such as chemical and biological physics. This work represents a first step in this direction, wherein we used AI, specifically LSTM flavor of recur-

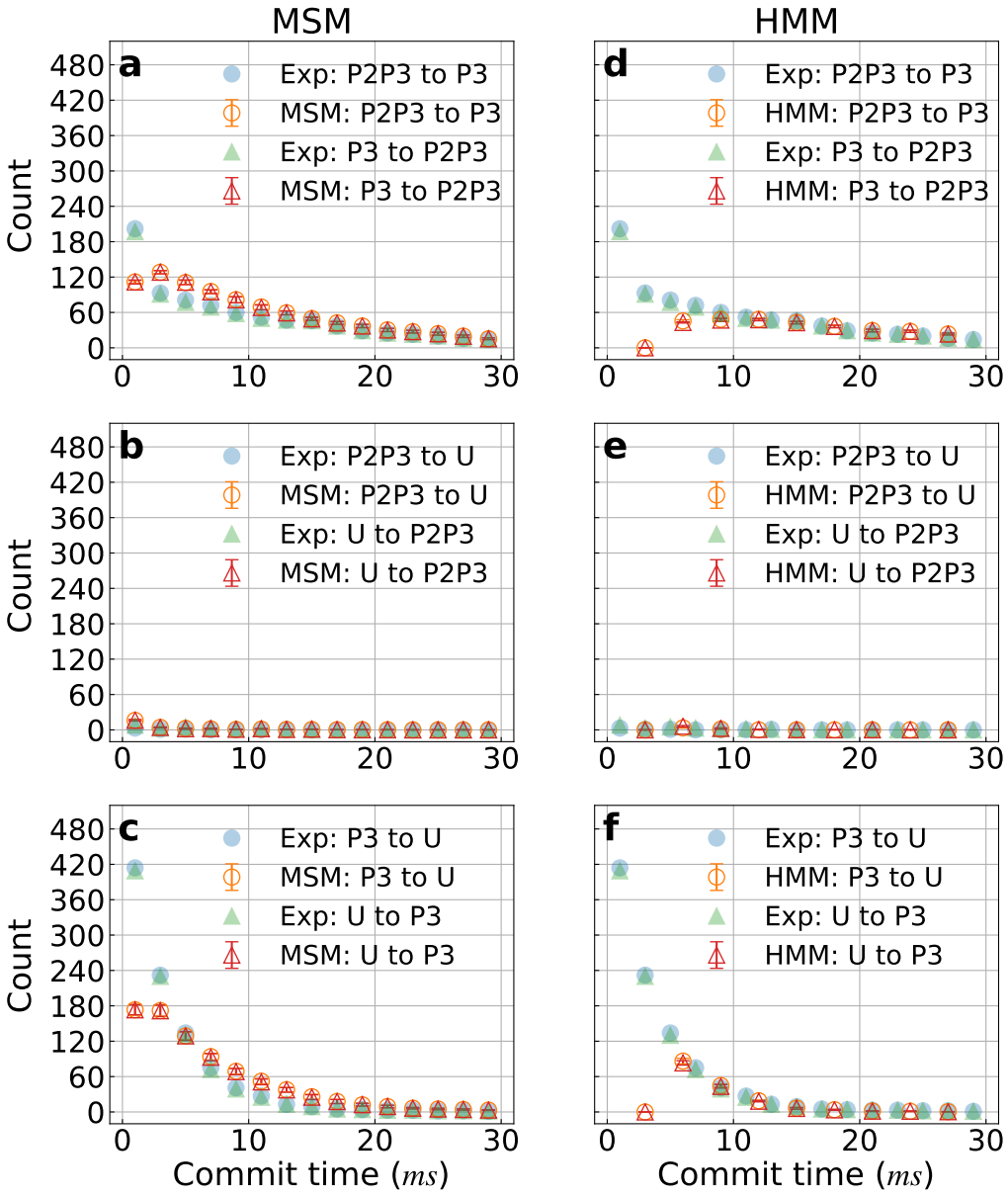


Figure 4.14: **Riboswitch kinetics through alternate approaches.** Number of transitions between different pairs of metastable states as a function of commitment time defined in **Results** for the single molecule spectroscopy trajectory as learned by MSM (left column) and HMM (right column). Associated error bars calculated as standard errors are also provided.

rent neural networks, to perform kinetic reconstruction tasks that other methods [123,175] could have also performed. We would like to argue that demonstrating the ability of AI approaches to perform tasks that one could have done otherwise is a crucial first step.

In future works we will exploring different directions in which the AI protocol developed here could be used to perform tasks which were increasingly non-trivial in non-AI setups. More specifically, in this work we have shown that a simple character-level language model based on LSTM neural network can learn a probabilistic model of a time series generated from a physical system such as an evolution of Langevin dynamics or MD simulation of complex molecular models. We show that the probabilistic model can not only learn the Boltzmann statistics but also capture a large spectrum of kinetics. The embedding layer which is designed for encoding the contextual meaning of words and characters displays a nontrivial connectivity and has been shown to correlate with the kinetic map defined for reversible Markov chains [117, 118, 176]. An interesting future line of work for the embedding layer can be to uncover different states when they are incorrectly represented by the same reaction coordinate value, which is similar to finding different contextual meaning of the same word or character. For different model systems considered here, we could obtain correct timescales and rate constants irrespective of the quality of order parameter fed into the LSTM. As a result, we believe this kind of model outperforms traditional approaches for learning thermodynamics and kinetics, which can often be very sensitive to the choice of projection. Finally, the embedding layer can be used to define a new type of distance metric for high-dimensional data when one has access to only some low-dimensional projection. In the next chapter, we will introduce a simple yet powerful method that allows the LSTM model to extrapolate new physics. We hope that this work represents a first step in the use of RNNs for modeling, understanding and predicting the dynamics of complex stochastic systems found in biology, chemistry and physics.

Chapter 5: Path sampling of recurrent neural networks by incorporating known physics

5.1 Introduction

In Chapter. 4, we have proposed using a neural network-based language model built upon long short-term memory (LSTM) to model long-term dependency within time series obtained from MD simulations and single-molecule force spectroscopy experiments. We show that our simple character-level language model can capture correct kinetic connectivity. However, LSTM can't learn physics that it doesn't see in the training data. In this chapter, we will introduce a simple yet powerful approach to overcome this limitation. We will then show how we let LSTM extrapolate physics that it doesn't see in the training data.

Artificial neural networks (ANNs) and modern-day Artificial Intelligence (AI) seek to mimic the considerable power of a biological brain to learn information from data and robustly perform a variety of tasks, such as text and image classifications, speech recognition, machine translation, and self-driving cars [143–145, 147, 152, 177–180]. In recent years, ANNs have been shown to even outperform humans in certain tasks such as playing board games and weather prediction [146, 181, 182]. Closer to physical sci-

ences, ANNs have been used to make predictions of folded structures of proteins [183], accelerate all-atom molecular dynamics (MD) simulations [156, 158, 184, 185], learn better order parameters in complex molecular systems [186–189], and many other exciting applications. While the possible types of ANNs is huge, in this work we are interested in Recurrent Neural Networks (RNNs). These are a class of ANNs that incorporate memory in their architecture allowing them to directly capture temporal correlations in time series data. [149, 190] Furthermore, RNN frameworks such as long short-term memory (LSTM) neural networks [150] can account for arbitrary and unknown memory effects in the time series being studied. These features have made RNNs very popular for many applications such as weather, stock market prediction and dynamics of complex molecular systems [130, 146, 147, 191]. In such applications, the assumption of independence between data points at different time steps is also invalid, and furthermore events that occurred at an arbitrary time in the past can have an effect on future events [130, 146, 147, 149].

In spite of their staggering success, one concern applicable to RNNs and ANNs in general is that they are only able to capture the information present in their training datasets, unless additional knowledge or constraints are incorporated. Since a training dataset is limited by incomplete sampling of the unknown, high-dimensional distribution of interest, this can cause a model to overfit and not precisely represent the true distribution [192]. For instance, in the context of training MD simulations, partial sampling when generating a training dataset is almost unavoidable. This may come from only being able to simulate dynamics on a particular timescale that is not long enough to completely capture characteristics of interest [193] or simply thermal noise which could manifest as a misleading violation of detailed balance [194]. In such cases, enforcing a constraint cor-

responding to the characteristic of interest when training an RNN-based model is critical for accurately modeling the true underlying distribution of data.

Given the importance of this problem, numerous approaches have been proposed in the recent past to add constraints to LSTMs, which we summarize in Sec. 5.2.1. However, they can generally only deal with very specific types of constraints, complicated further by the recurrent or feedback nature of the networks [195–197]. In this work we provide a generalizable, statistical physics based approach to add a variety of constraints to LSTMs. To achieve this, we use ideas of path sampling combined with LSTM, facilitated through the principle of Maximum Caliber. Our guiding principle is our previous work [130] where we show that training an LSTM model is akin to learning path probabilities of the underlying time series. This facilitates generating a large number of trajectories in a controlled manner and in parallel, that conform to the thermodynamic and dynamic features of the input trajectory. From these, we select a sub-sample of trajectories that are consistent with the desired static or dynamical knowledge. The bias due to sub-sampling is accounted for using the Maximum Caliber framework [126] by calculating weights for different possible trajectories. A new round of LSTM is then trained on these sub-sampled trajectories that in one-shot combines observed time series with known static and dynamical knowledge. This framework allows for constrained learning without incorporating an explicit constraint within the loss function. We demonstrate the usefulness of our approach on several problems, including constraining the dynamics of the 3-state Markov model, correcting the predictions of left and right helix states of a synthetic peptide α-aminoisobutyric acid 9 (Aib9) by LSTM on long timescale, and predicting the transition times of slow modes by incorporating and constraining small structural fluctuations.

5.2 Theory

5.2.1 Previous approaches to add constraints to LSTM networks and their limitations

A naive way of applying constraints when training LSTMs is incorporating a term within the loss function in Eq. 4.11 whose value decreases as the model's adherence to the constraint increases. This approach has been successfully applied for instance in the context of 4-D flight trajectory prediction [195]. A limitation of this naive approach is that the desired constraint must have an explicit mathematical formulation parameterized by the RNN's raw output, so that the value of the regularization term in the constraint can be adjusted through training. In the case of LSTMs, the raw output of the model passed through a softmax layer is equivalent to the probability of a future event conditioned on an observed past event. Formulating mathematical constraints solely in terms of such conditional probabilities has been done for specific constraints [195] and can be very challenging in general. Alternative more nuanced approaches to enforcing constraints in LSTMs have also been employed specific to the particular application. For example, when applying LSTMs to generate descriptions of input images, Ref. [196] constrained part of speech patterns to match syntactically valid sentences by incorporating a part of speech tagger, that tags words as noun, verb etc. within a parallel LSTM language model architecture. The success of this approach relies on being able to reliably introduce more information to the model through the predictive part of the speech tagger. In applying LSTMs to estimating geomechanical logs, Ref. [197] incorporated a physical constraint

by adding an additional layer into the LSTM architecture to represent a known intermediate variable in physical models. The success of this approach as well relies on utilizing a known physical mechanism involved in the specific engineering problem.

5.2.2 Our approach: Path-sampled LSTM

The approaches described in Sec. 5.2.1 while useful in the specific contexts for which they were developed, are not generally applicable to different constraints. For instance, when combining experimental time series for molecular systems with known theoretical knowledge, the constraints are often meaningful only in an ensemble-averaged sense. This per definition involves replicating many copies of the same system. With dynamical constraints involving rates of transitions, the problem is arguably even harder as it involves averaging over path ensembles. Our statistical physics based approach deals with these issues in a self-contained manner, facilitated by our previously derived connections between LSTM loss functions and path entropy [130]. The key approach to constraining the neural networks with desired physical properties is to sample a subset from predicted trajectories generated from the trained LSTM models. The sampling is performed in a way such that the subset satisfies desired thermodynamic or dynamic constraints. For a long enough training set, we have shown in our previous work [130] that LSTM learns the path probability, and thus a trained LSTM generates copies of the trajectory from the correct path ensemble.

Our key idea behind constraining recurrent neural networks with desired physical properties is to sample a subset from predicted trajectories generated from the trained

LSTM models. The sampling is performed in a way such that the subset satisfies desired thermodynamic or dynamic constraints. For a long enough training set, we have shown in our previous work [130] that LSTM learns the path probability, and thus a trained LSTM generates copies of the trajectory from the correct path ensemble. In other words, the final output vector $\hat{\mathbf{y}}^{(t)}$ will learn how to generate $P_\Gamma \equiv P(\mathbf{x}^{(0)}...\mathbf{x}^{(T)})$, where P_Γ is the path probability associated to a specific path Γ in the path ensemble characterized by the input trajectory fed to the LSTM. The principle of Maximum Caliber or MaxCal [96, 126, 198] provides a way to build dynamical models that incorporate any known thermodynamic or dynamic i.e. path-dependent constraints into this ensemble. Per MaxCal [126], one can derive P_Γ by maximizing the following functional called Caliber:

$$\mathcal{C} = \sum_\Gamma P_\Gamma \ln \frac{P_\Gamma}{P_\Gamma^{\mathrm{U}}} - \sum_i \lambda_i \left(\sum_\Gamma s_i(\Gamma) P_\Gamma - \bar{s}_i \right) \tag{5.1}$$

where λ_i is the Lagrange multiplier associated to the i-th constraint that helps enforce path-dependent static or dynamical variables $s_i(\Gamma)$ to desired path ensemble averaged values $\bar{s}_i$. With appropriate normalization conditions for probabilities, maximizing Caliber in Eq. 5.1 relates the constrained path probability P_Γ^* to the reference or unconstrained path probability P_Γ^{U} as follows:

$$P_\Gamma^* \propto e^{-\sum_i \lambda_i s_i(\Gamma)} P_\Gamma^{\mathrm{U}} \tag{5.2}$$

From Eq. 5.2, it is easy to show that for two dynamical systems labelled A and B that only differ in the ensemble averaged values for some j-th constraint being $\bar{s}_j^{\mathrm{A}}$ and $\bar{s}_j^{\mathrm{B}}$,

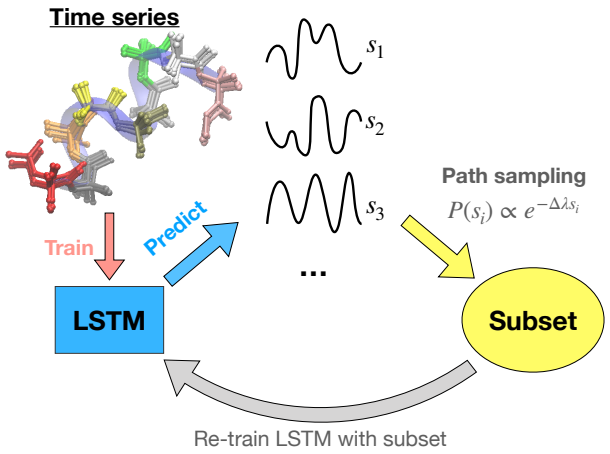


Figure 5.1: **Procedure for path sampling LSTM** This schematic plot shows the workflow for constraining some static or dynamical variable $s(\Gamma_i)$, given an unconstrained LSTM model. The workflow begins with generating numerous predicted trajectories from the constraint-free LSTM model. The corresponding variables that we seek to constrain can be calculated from the predicted trajectories and are denoted by $s(\Gamma_1), s(\Gamma_2), s(\Gamma_3)$ in the plot. We then perform a path sampling and select a smaller subset of trajectories in a biased manner that conforms to the desired constraints, with a probability $P(s(\Gamma_i)) \propto e^{-\Delta\lambda s(\Gamma_i)}$, where $\Delta\lambda$ is solved by the Eq. 5.4. The subset is then used as a new dataset to train the LSTM model.

then their respective path probabilities for some path Γ are connected through:

$$P_\Gamma^{\mathrm{B}} \propto e^{-\Delta\lambda_j s_j(\Gamma)} P_\Gamma^{\mathrm{A}} \tag{5.3}$$

where $\Delta\lambda_j = \lambda_j^{\mathrm{B}} - \lambda_j^{\mathrm{A}}$.

With this formalism at hand, we label our observed time series as the system A and its corresponding path probability as P_Γ^{A}. This time series or trajectory has some thermodynamic or dynamical j-th observable equaling $\bar{s}_j^{\mathrm{A}}$. On the basis of some other knowledge coming from theory, experiments or intuition, we seek this observable to instead equal $\bar{s}_j^{\mathrm{A}}$. In accordance with Ref. [130] we first train a LSTM that learns P_Γ^{A}. Our objective now is to train a LSTM model that can generate paths with probability P_Γ^{B} with

desired, corrected value of the constraint. For this we use Eq. 5.3 to calculate $\Delta\lambda$. This is implemented through the following efficient numerical scheme. We write down the following set of equations:

$$\begin{aligned} \bar{s}_j^{\mathrm{B}} &= \sum_{\Gamma} P_{\Gamma}^{\mathrm{B}} s_j(\Gamma) \\ &= \frac{\sum_{k\in\Omega} s_j(\Gamma_k) e^{-\Delta\lambda_j s_j(\Gamma_k)}}{\sum_{k\in\Omega} e^{-\Delta\lambda_j s_j(\Gamma_k)}} \end{aligned} \tag{5.4}$$

where Ω is the set of labelled paths sampled from the path probability P_{Γ}^{A}. By solving for $\Delta\lambda_j$ from Eq. 5.4 we have the sought P_{Γ}^{B}. In practice, this is achieved through the procedure depicted in Fig. 5.1, where the LSTM model trained with time series for the first physical system is used to generate a collection of predicted paths with a distribution proportional to path probability P_{Γ}^{A}. A re-sampling with an appropriate estimate of $\Delta\lambda_j$ is then performed to build a subset. This value is obtained by computing the right hand side of the second line in Eq. 5.4 over the resampled subset such that correct desired value of the constraint is obtained. This subset denotes sampling from the desired path probability P_{Γ}^{B} and is used to re-train a new LSTM that will now give desired $\bar{s}_j^{\mathrm{B}}$. The method can be easily generalized to two or more constraints. For example, in order to solve for two constraints, we can rewrite Eq. 5.3 as

$$P_{\Gamma}^{\mathrm{B}} \propto e^{-\Delta\lambda_j s_j(\Gamma) - \Delta\lambda_k s_k(\Gamma)} P_{\Gamma}^{\mathrm{A}} \tag{5.5}$$

where $\Delta\lambda_j$ and $\Delta\lambda_k$ are two unknown variables to be solved with two equations for the ensemble averages $\bar{s}_j^{\mathrm{B}}$ and $\bar{s}_k^{\mathrm{B}}$.

Henceforth, we refer to the unconstrained version of LSTM as simply LSTM and the constrained version introduced here as ps-LSTM for "path sampled" LSTM.

5.2.3 Solving state-to-state transitions of Markov processes

In this section we develop useful, exact results for constraining state-to-state transitions in Markov processes that serve as useful benchmarking. It has been shown that if constraining pairwise statistics, maximizing Eq. 5.1 with appropriate normalization conditions yields the Markov process [199]

$$P_{\Gamma}^{*} = p_{i_0} \prod_{k=0}^{T-1} p_{i_k i_{k+1}} \tag{5.6}$$

where $p_{i_k i_{k+1}}$ are the time-independent transition probabilities defined by the Markov transition matrix. For such simple Markovian dynamics, we can easily solve for the outcome transition kernel by the $\Delta\lambda$ chosen.

Now we suppose we would like to adjust the frequency of transition from state m to state n. With Eq. 5.6, following Ref. [199], we can rewrite Eq. 5.3 as

$$\prod_{k=0}^{T-1} p^{\mathrm{B}}_{i_k i_{k+1}} \propto e^{-\Delta\lambda \sum_{k=0}^{T-1} \delta_{i_k,m}\delta_{i_{k+1},n}} \prod_{k=0}^{T-1} p^{\mathrm{A}}_{i_k i_{k+1}} \tag{5.7}$$

where δ_{ij} is the Kronecker delta, equalling 1 when $i = j$ and 0 otherwise. Therefore, it can then be shown that

$$p^{\mathrm{B}}_{mn} \propto e^{-\Delta\lambda} \cdot p^{\mathrm{A}}_{mn} \tag{5.8}$$

Eq. 5.8 with predetermined $\Delta\lambda$ can be used to predict our numerical results.

Based on Eq. 5.12 and the equations in the Appendix, we can analyze the difference in transition kernel:

$$p_{mn}^{\text{ps-LSTM}} \propto e^{-\frac{\Delta\lambda}{L_{\text{traj}}}(\delta_{m0}\delta_{n1}+\delta_{m1}\delta_{n0}+\delta_{m1}\delta_{n2}+\delta_{m2}\delta_{n1})} \cdot p_{mn}^{\text{LSTM}} \tag{5.9}$$

where $\Delta\lambda$ we used is -56.1.

5.3 Model set-up and simulation details

In this section, we will introduce the model potentials and simulations that we will use later in the discussion of Sec. 5.4.

5.3.1 Markov dynamics details

The time series used for training with the 3 state Markov dynamics was generated through random sampling using the transition probability matrix shown below in Eq. 5.10. The length of the time series used for input was 300000 units. The transition probability matrix P used to generate 3-state Markov dynamics is given by:

$$P = \begin{bmatrix} 0.9300 & 0.0667 & 0.0033 \\ 0.0667 & 0.8667 & 0.0667 \\ 0.0033 & 0.0667 & 0.9300 \end{bmatrix} \tag{5.10}$$

Predictions with the ps-LSTM when trained with the 3-state Markov Dynamics

yielded the following transition matrix when constraining the first nearest neighbor transition rate

$$P = \begin{bmatrix} 0.8886 \pm 0.00039 & 0.1078 \pm 0.00039 & 0.0036 \pm 0.00004 \\ 0.0833 \pm 0.00057 & 0.8380 \pm 0.00105 & 0.0787 \pm 0.00056 \\ 0.0084 \pm 0.00009 & 0.1029 \pm 0.00047 & 0.8887 \pm 0.00050 \end{bmatrix} \tag{5.11}$$

5.3.2 Molecular dynamics and neural network details

The MD trajectory for Aib9 was obtained using the software GROMACS 5.0.4 [128, 129], patched with PLUMED 2.4 [102]. The Aib9 molecule consists of 129 atoms solvated with 1540 TIP3P [200, 201] water molecules. CHARMM36m30 all atom force field is used to parametrize the Aib9. Molecular dynamics (MD) was performed to generate the time-series, with temperature 500K using the Nose-Hoover thermostat [202] and the pressure maintained at ambient pressure with Parrinello-Rahman barostat [203]. The molecular dynamics integration time step is $2fs$.

For training an LSTM to learn the 3-state Markov dynamics, we took the embedding dimension $M = 8$ and the LSTM unit $L = 128$. The time series were batched into sequences with a sequence length of 35 and the batch size of 64. The models were trained with the method of stochastic gradient descent for 10 epochs. After sampling 100 LSTM predictions of length 100 with $\Delta\lambda$ = -56.1 defined in Sec. 5.2.2, a ps-LSTM was retrained with the same hyperparameters except for increasing epochs to 350. For training LSTM to learn Aib9, we discretized the input into 32 states, took the embedding dimension $M = 32$ and the LSTM unit $L = 64$. The time series were batched into sequences

with a sequence length of 100 and the batch size of 64. The model was trained with the method of stochastic gradient descent for 40 epochs. The subsets consist of 10 selected trajectories for constraining the Aib9 MD simulations.

5.4 Results

5.4.1 Three-state Markovian dynamics

For the first illustrative example, we apply LSTM to a 3 state model system following Markovian dynamics for moving between the 3 states. This system, comprising states labelled 0, 1 and 2 is illustrated in Fig. 5.2(a). Fig. 5.2(a) also shows the state-to-state transition rates for the unconstrained system. We then seek to constrain the average number of transitions per unit time between states 0,1 and 1,2 as defined below

$$\langle N \rangle = \frac{1}{L_{\text{traj}}}(N_{0\leftrightarrow 1} + N_{1\leftrightarrow 2}) \tag{5.12}$$

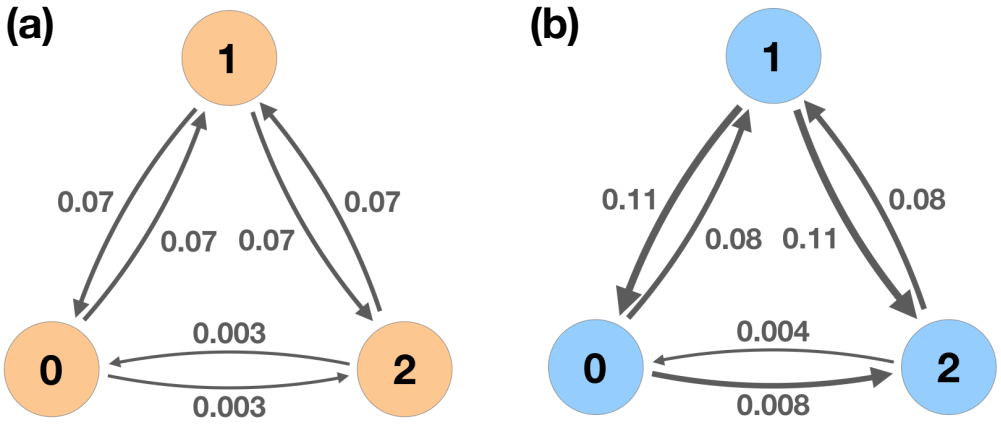
where L_{traj} is the length of trajectory and $N_{0\leftrightarrow 1}$ and $N_{1\leftrightarrow 2}$ are the number of times a transition occurs between states 0 and 1 or states 1 and 2 respectively. This example can then be directly compared with the analytical result Eq. 5.8 derived in Sec. 5.2.3, thereby validating the findings from ps-LSTM.

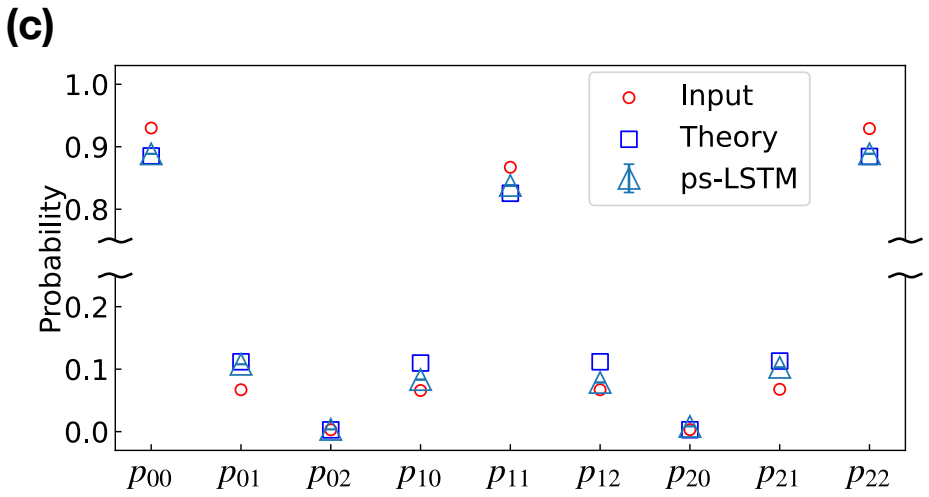
Given the transition kernel shown in Fig. 5.2 (a), we generate a time series that conforms to it. Following Sec. 5.2.2, we train ps-LSTM using this time series and the constraint on $\langle N \rangle$ described in Eq. 5.12. As per the Markovian transition kernel we have $\langle N \rangle = 0.0894$, while we seek to constrain it to 0.13. In other words, given a time series

we want to increase the number of transitions per unit time between 2 of the 3 pairs of states. In Fig. 5.2 (b), we show the transition kernel obtained from the time series generated by ps-LSTM via direct counting. Fig. 5.2 (c) provides values of $\langle N \rangle$ from the analytical transition kernel provided in Sec. 5.3.1 and those generated from the constraint-free LSTM and ps-LSTM. In particular, we would like to highlight that when enforcing a faster rate of state-to-state transitions sampling to increase the average number of nearest neighbor transitions, the transition matrix of ps-LSTM predictions show correspondingly increased rates of transition without completely destroying the original kinetics of the system. Using Eq. 5.9 provided in Sec. 5.2.3, we can predict the new transition kernel given by ps-LSTM. The comparison is also shown in Fig. 5.2.

5.4.2 MD simulations of α-aminoisobutyric acid 9 (Aib9)

For our second, more ambitious application, we study the 9-residue synthetic peptide α-aminoisobutyric acid 9 (Aib9) [204, 205]. Aib9 undergoes transitions between fully left-handed (L) helix and fully right-handed (R) helix forms. This is a highly collective transition involving concerted movement of all 9 residues. During this global transition, there are many alternate pathways that can be taken, connected through a network of several lowly-populated intermediate states [204, 205]. This makes it hard to find a good low-dimensional coordinate along which the dynamics can be projected without significant memory effects [204, 205]. The problem is further accentuated by the presence of numerous high-energy barriers between the metastable states that result in their poor sampling when studied through all-atom MD. For example, through experimental





(d)

Markov 3-state	Input	LSTM	ps-LSTM, $\langle N \rangle$=0.13
$\langle N \rangle$	0.0894	0.0934±0.0015	0.1271±0.0001

Figure 5.2: **3-state Markovian system: ps-LSTM and analytical predictions.** Here we show results of applying ps-LSTM to the 3 state Markovian system where we constrain $\langle N \rangle$. In (a), we provide the input transition kernel without constraints. In (b), we show the transition kernel obtained from ps-LSTM generated time-series via direct counting, where we achieve a $\langle N \rangle$ close to the target $\langle N \rangle$=0.13. In (c), we show the comparison of the transition probabilities from state-m to n, p_{mn}, between the input trajectory used to train our newtork, the predicted values given from analytical results in Eq. 5.9, and the actual transition probability obtained via direct counting using the 200 predictions by ps-LSTM. The calculated values for $\langle N \rangle$ are shown in (d) for LSTM as the average of 100 predictions and for ps-LSTM as the average of 200 predictions.

measurements [206] and enhanced sampling simulations [204, 205], the achiral peptide should show the same equilibrium likelihood of existing in the L and R forms. However, due to force-field inaccuracies [204] and insufficient sampling, MD simulations typically are too short to obtain such a result. In the first type of constraint, which enforces static or equilibrium probabilities, we show how our ps-LSTM approach can correct the time series obtained from such a MD simulation to enforce the symmetric helicity. In a second type of dynamical constraint, we show how we can enforce a desired local transition rate between different protein conformations.

5.4.2.1 Equilibrium constraint on Aib9

We first discuss results for enforcing the constraint of symmetric helicity on Aib9, shown in Fig. 5.3. Here we have defined the free energy $F = -k_B T \ln P$, where k_B and T are the Boltzmann constant and temperature, and P is the equilibrium probability calculated by direct counting from a respective time series. In Figs. 5.3 (a)-(c) we have projected free energies from different methods along the summation χ of the 5 inner dihedral angles ϕ, which allows us to distinguish the L and R helices. We define $\chi \equiv \sum_{i=3}^{7} \phi_i$ and note that $\chi \approx 5.4$ and $\chi \approx -5.4$ for L and R respectively [205]. In order to have a reference to be compared with, we perform the simulation at temperature 500K under ambient pressure. As can be seen from Fig. 5.3(b), we are able to see a symmetric free energy profile after $100ns$.

For LSTM to process the time series for χ as done in Ref. [130], we first spatially discretize χ into 32 labels or bins. To quantify the symmetry between left- and right-

handed populations, we define a symmetry parameter κ:

$$\kappa = \frac{\sum_{i=0}^{i=15} P_i}{\sum_{i=16}^{i=32} P_i} \tag{5.13}$$

where P_i denotes equilibrium probability for being found in bin label i. For symmetric populations we expect $\kappa \approx 1$. In Fig. 5.3 (a), we show the free energy from the first $20ns$ segment of time series from MD. This $20ns$ time series is then later used to train our LSTM model. It can be seen that the insufficient amount of sampling results in an incorrect asymmetry of populations between L and R helix states with $\kappa \approx 0.5$. We first train a constraint-free LSTM on this trajectory following Ref. [130] with which we generate a $200ns$ time series for χ. Fig. 5.3(b) shows how a longer $200ns$ MD trajectory would have been sufficient to converge to a symmetric free energy with $\kappa \approx 1$. However, Fig. 5.3(b) also shows the population along χ measured from the LSTM generated time series, which preserves the initially asymmetry that it witnessed in the original training trajectory.

In Fig. 5.3(c) we show the results from using ps-LSTM where we apply the constraint $\kappa = 1$. For this, we let the constraint-free LSTM model generate 200 indepdendent time series of length $20ns$ long and used the method from Sec. 5.2.2 to enforce the constraint $\kappa = 1$ for $200ns$ long time series. We calculate κ values from the different predicted time series and use Eq. 5.4 to solve for an appropriate $\Delta\lambda$ needed for $\langle\kappa\rangle = 1$. We then perform path sampling with a biased probability $\propto e^{-\Delta\lambda}$ to select 10 trajectories from the 200 predictions. These 10 time series were then used to construct a subset and train a new ps-LSTM. As can be seen in Fig. 5.3(c), ps-LSTM captures the correct symmetric

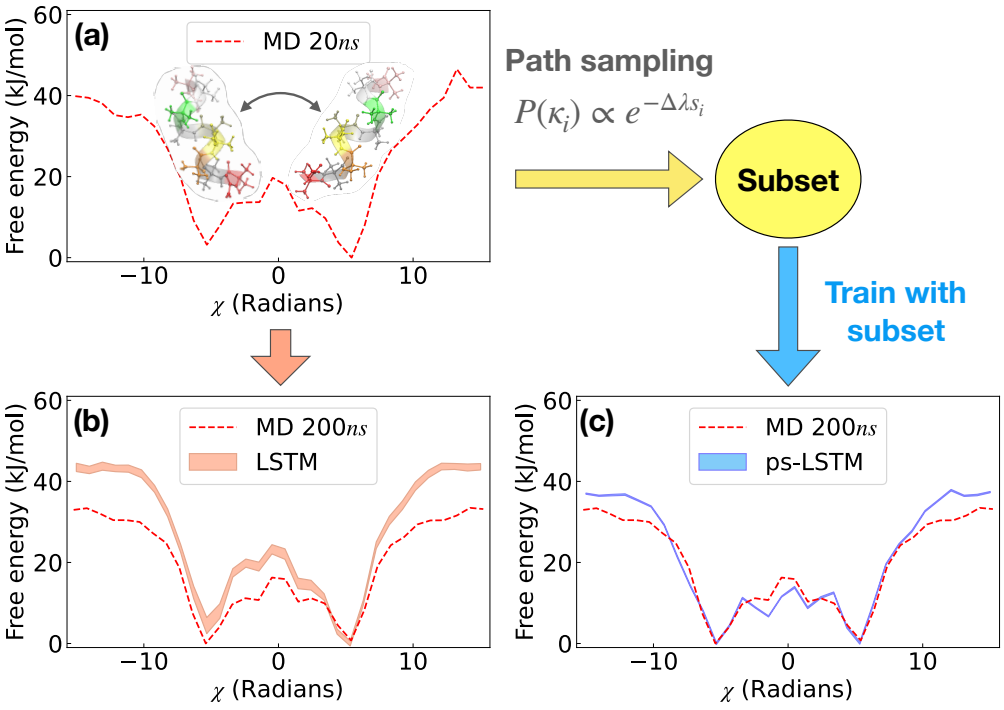
free energy profile giving $\kappa = 1$. Interestingly, ps-LSTM also significantly reduces the deviations from the reference free energy at $|\chi| > 10$. In Fig. 5.4, we have also provided the eigenspectrum of the transition matrix and shown that relative to LSTM, ps-LSTM pushes the kinetics for events across timescales in the correct direction. In Fig. 5.3(d), we show the κ calculated from the trajectories of $20ns$ and $200ns$ MD simulations of Aib9 and from the predicted $200ns$ trajectories of LSTM and ps-LSTM.

5.4.2.2 Dynamical constraint on Aib9

Our second test is performed to enforce a dynamical constraint, i.e. one that explicitly depends on the kinetics of the system [97]. Specifically, we constrain the ensemble averaged number of nearest neighbor transitions per unit time $\langle N \rangle$ along the sum of dihedral angle χ introduced in Sec. 5.4.2.1. $\langle N \rangle$ is defined as

$$\langle N \rangle = \frac{1}{L_{\text{traj}}} \sum_i N_{i,i+1} \tag{5.14}$$

where L_{traj} is the length of trajectory, and $N_{i,i+1}$ equals 1 if the values of χ at times i and $i+1$ are separated only by a single bin, otherwise 0. The nearest neighbor transitions can be seen as a quantification of diffusivity when comparing the form of transition rate matrix from the discretized Smoluchowski equation to the one derived from principle of Maximum Caliber [97]. In Fig. 5.5 (a), we show a free energy profile calculated from a $100ns$ MD trajectory. As can be seen here, this trajectory is long enough to give symmetric populations for the L and R helix states. We find that the averaged number of nearest neighbor transitions $\langle N \rangle$ for this trajectory is approximately 0.4. In Fig. 5.5 (a)



(d)

Aib9	MD 20ns	MD 200ns	LSTM	ps-LSTM
κ	0.478	1.149	0.867±0.385	1.209±0.143

Figure 5.3: **Comparing predictions at 200ns for different values of the symmetry parameter κ.** Here we show that ps-LSTM learns the correct symmetry κ. The original training data is a 20ns Aib9 trajectory generated from MD simulation at 500K, where (a) shows its calculated free energy profile has an asymmetry of population between L and R helix states. The snapshots of L and R configurations at $\chi = 5.2$ and $\chi = -5.31$ are also displayed as insets above the free energy profile. Training LSTM model with this asymmetric data and using it to predict what would happen at 200ns leads to the result shown in (b), where the LSTM predictions retain and even enhance the undesired free energy asymmetry while the free energies calculated from a longer 200ns trajectory shows the desired symmetric profile. In (c), we show that ps-LSTM trained as described in Sec. 5.4.2.1 can not only predict the correct symmetry, but also deviate less from the true free energy calculated from the reference 200ns data. The table in (d) shows the κ values defined in Eq. 5.13 for different trajectories. The free energy profiles and the κ values in (b) and (c) are averaged over 10 independent training processes. The corresponding error bars are filled with transparent colors.

we have also shown the free energy from a 200ns long MD simulation which we use later for comparison. In Fig. 5.5 (b), we show trajectory generated from training constraint-free LSTM [130] which follows the same Boltzmann statistics and kinetics as the input

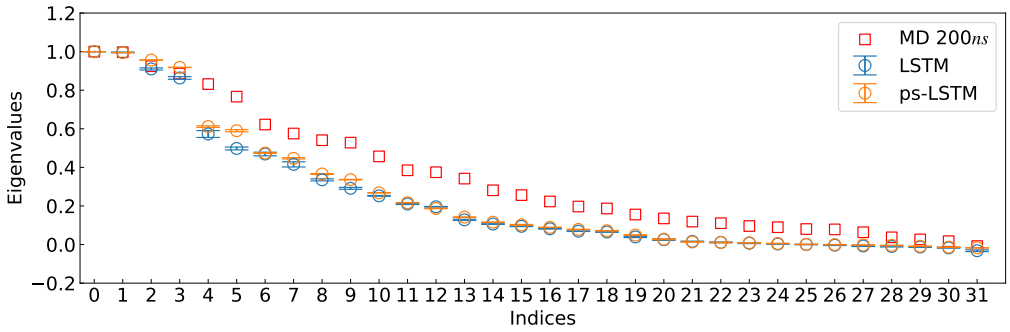


Figure 5.4: **Eigenspectrum of transition probability matrices.** In this plot, we show the eigenspectrum of the transition probability matrices calculated via direct counting from the reference 200ns MD simulation of Aib9 (red squares), the LSTM prediction (blue circles), and the ps-LSTM prediction (orange circles). We can see that both ps-LSTM and LSTM capture the first four slow modes. While there are some deviations at 4th and 5th modes, the deviations mainly come from the training of LSTM. The ps-LSTM method performs path sampling from the predicted trajectories of LSTM therefore it simply captures the errors from the LSTM itself.

trajectory.

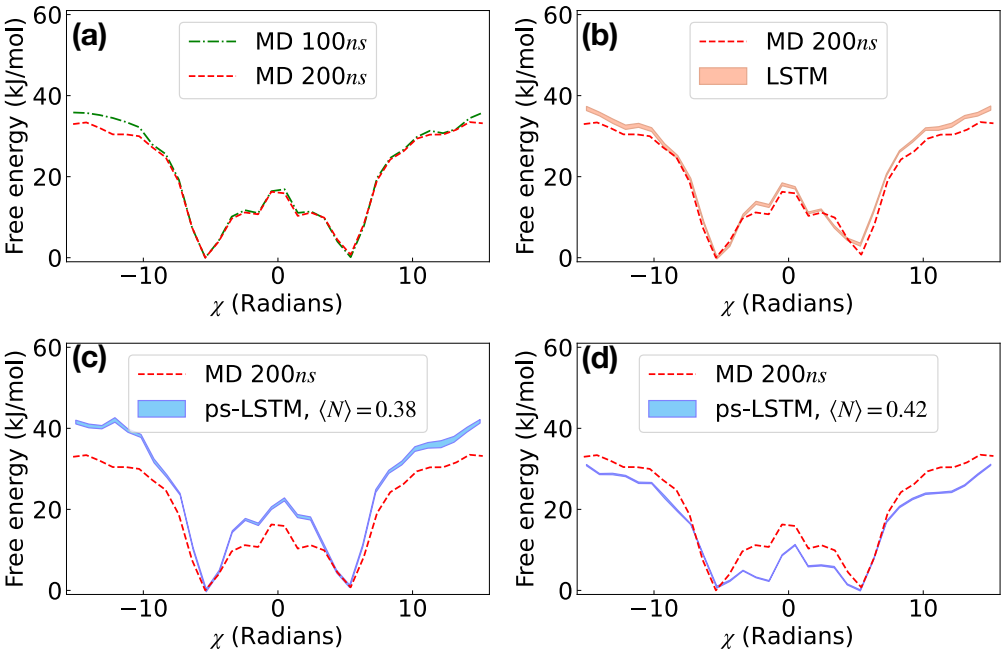
In order to constrain $\langle N \rangle$, we generate 800 independent time series from the constraint-free LSTM, and sample a subset consisting of 10 time series. With an appropriate $\Delta\lambda$, our path-sampled subsets are constrained to two different $\langle N \rangle$ values and used for training two distinct ps-LSTMs. In Fig. 5.5(c) and (d), we have shown the free energy profiles corresponding to ps-LSTM predictions trained on subsets with $\langle N \rangle = 0.38$ and $\langle N \rangle = 0.42$. As can be seen, compared to the actual 200ns MD simulation of Aib9, the potential wells of L and R helix become narrower for $\langle N \rangle = 0.38$ and wider for $\langle N \rangle = 0.42$, which is the direct effect of changing fluctuations via nearest-neighbor transitions. Moreover, the potential barriers along χ become higher for $\langle N \rangle = 0.38$ and become lower for $\langle N \rangle = 0.42$. In Fig. 5.5 (e), we provide the averaged transition times τ from L to R helix states and vice versa, where we can also see that the transition times do become longer for smaller $\langle N \rangle$ and shorter for larger $\langle N \rangle$, which is the expected result for decreased and increased

diffusivities respectively [25, 97].

To summarize so far, the results from constraining $\langle N \rangle$ show that through the path sampling method, ps-LSTM extrapolates the phenomena affected by changing small fluctuations, which was not provided in the training data set.

5.5 Conclusion and outlook

In this work, we proposed a method integrating statistical mechanics with machine learning in order to add arbitrary knowledge in the form of constraints to the widely used long short-term memory (LSTM) neural network used for predicting generic time series in diverse problems. These models are trained on available time series for the system at hand, which often have errors of different kinds. These errors could arise from either poor sampling due to rareness of the underlying events, or simply reprsent instrumentation errors. Using high fidelity artificial intelligence tools [130, 157, 207] to generate computationally cheaper copies of such time series is then prone to preserving such errors. Thus, it is extremely important to introduce systematic constraints that introduce prior knowledge in the LSTM network used to replicate the time series provided in training. The recurrent nature of the LSTM and the non-Markovianity of the time series make it hard to impose such constraints in a trainable manner. For this, here our approach involves path sampling method with the principle of Maximum Caliber, and is called ps-LSTM. We demonstrated its usefulness on illustrative examples with varying difficulty levels and knowledge that is thermodynamic or kinetic in nature. Finally, as our method relies only on data post-processing and pre-processing, it should be easily generalized to other neural



(e)

Aib9	MD 100ns	MD 200ns	LSTM	ps-LSTM, $\langle N \rangle$=0.38	ps-LSTM, $\langle N \rangle$=0.42
$\langle N \rangle$	0.397	0.399	0.405±0.001	0.372±0.001	0.422±0.001
$\tau_{L \to R}$ (ps)	2326±390	2083±230	1934±297	4107±807	920±63
$\tau_{R \to L}$ (ps)	2273±322	2083±193	1912±254	4062±822	922±64

Figure 5.5: **Comparing predictions at 200ns for different values of the dynamical constraint $\langle N \rangle$.** In this plot, we show the free energy profiles calculated from (a) the 100ns trajectory in the training set, (b) both the actual 200ns trajectory and direct prediction from LSTM, (c) the reference 200ns trajectory and ps-LSTM prediction with constraint of nearest-neighbor (NN) transitions $\langle N \rangle = 0.38$, and (d) the reference 200ns trajectory and prediction with constraint $\langle N \rangle = 0.42$. The table above lists the kinetic constraint $\langle N \rangle$ calculated from corresponding trajectories. The averaged transition time $\tau_{R \to L}$ and $\tau_{L \to R}$ in picoseconds were calculated by counting the numbers of transitions in each trajectory. For reference MD, the error bars were calculated by averaging over transition time in a single 100ns or 200ns trajectory, while for the predictions from LSTM and ps-LSTM, the error bars were averaged over 10 independent predictions with the transition time for each predicted trajectory calculated in the same way as MD trajectories. The free energy profiles and the first NN values $\langle N \rangle$ in (b), (c), and (d) are averaged over 10 independent training processes. The corresponding error bars are filled with transparent colors.

network models such as transformers and others [207, 208], and for modeling time series from arbitrary experiments.

Chapter 6: Conclusion and Outlook

In this dissertation, we examine time series produced from various complex dynamical systems including simulations of Langevin dynamics, molecular dynamics of nucleation and protein conformational changes, and single-molecule force spectroscopy experiment of riboswitch folding dynamics. These dynamical systems are intrinsically high-dimensional, making them hard to analyze. In addition, most important rare events happen at the timescale of minutes or even hours and so are rarely observed in MD due to MD's timescale limitation of milliseconds. Therefore, many enhanced sampling methods such as metadynamics have been developed to help sampling such rare events. In this dissertation, we use metadynamics as our main enhanced sampling method to sample rare events such as liquid-droplet nucleation of Lennard-Jones particles and crystal nucleation of urea. However, it is still challenging to study the kinetics and mechanisms of rare events as these enhanced sampling methods often rely on the low-dimensional representation called reaction coordinates (RCs). In the first chapter, we show, with various example, that the reaction coordinate is often constructed using order parameters that are important to the rare event kinetics. We discuss that the reaction coordinate should capture important physics of the rare events, and the rare event dynamics described by reaction coordinates should ideally follow Markov process. Unfortunately, there isn't a systematic

way to find reaction coordinate given any complicated systems. Inappropriately chosen RCs not only introduces complicated memory effect but also leads to incorrect kinetic connectivity between states. In this dissertation, we have shown how incorrect kinetic connectivity and such memory effect can cause underestimate of kinetic rates in droplet nucleation and even completely wrong kinetic pathways in Langevin dynamics of model potentials.

In the first part of this dissertation, we study systems where we assumed we have a complete set of order parameters which have already capture all important physics for its dynamics. We first show how inappropriate reaction coordinate constructed from these order parameters can lead to memory effect. We then focus on finding optimal RCs which could capture all slowly varying physics and minimize such memory effect. For instance, in the problem of liquid-droplet nucleation of Lennard-Jones particles, we consider RC as a 1-dimensional linear combination of three order parameters, where one the number of liquid-like atoms and the other two for local density fluctuations. Using our RC finding method Spectral Gap Optimization of Order Parameters (SGOOP), we find that as the supersaturation decreases, the RC ceases to simply be the number of liquid-like atoms, and instead, it becomes important to explicitly consider local density fluctuations that correlate with shape and density variations in the nucleus. Thus, at lower supersaturation levels, density fluctuations are non-Markovian and cannot be ignored from the RC by virtue of being noise. We then use SGOOP to find an optimal RC which explicitly considers these slow variables. By performing metadynamics with bias potential built as a function of this optimal RC, we show that we systematically improve the rate estimation. The improvement becoming more significant in the low supersaturation regime again shows that

the the local density and shape fluctuations become more important order parameters.

For more complicated systems, 1-dimensional RC can lead to incorrect kinetic connectivity. Therefore, in Chapter. 3, we developed an approach to find SGOOP-based kinetic distances or SGOOP-d. The approach allows us to systematically find additional RCs which capture missing slow degrees of freedom and further minimize the memory effect. One key feature of our method is that it can work with unbiased as well as biased simulations. We not only use the method to find a sufficient number of RCs but also reproduce the correct kinetic connectivity for alanine dipeptide and 23 and all 28 out of the 28 dominant state-to-state transitions in Ace-Ala_3-Nme. Our next step is to use this method to study crystal nucleation of urea from aqueous solution, where complicated polymorphism can exist. Our goal is to see if SGOOP-d can not only reproduce the correct kinetic connectivity but also predict the most probable nucleation pathways.

We also consider the application of the Artificial Intelligence (AI)-based method to build kinetic models with arbitrary memory as a long-term dependency, which constitutes the second part of this dissertation. In this direction, we apply a simple language model built upon a special type of Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM) to model memory within time series obtained from not only MD simulations but also single-molecule force spectroscopy experiments. Our model captures Boltzmann statistics and also reproduces kinetics across a spectrum of timescales. We also demonstrate how training the LSTM is equivalent to learning a path entropy, and that its embedding layer, instead of representing the contextual meaning of characters, reproduces kinetically truthful connectivity between different metastable states. In Chapter. 6, we further improve LSTM by incorporating known physics using the path sampling

method. We demonstrate our method by applying it to various examples with different levels of difficulty. In the first example of 3-state Markov model, we show that our path sampling method can reproduce the correct transition probability predicted by theory. In the second example, we use path sampling method to constrain the LSTM to learn correct chiral symmetry. In the third and final example, we show how we can path sampling LSTMs to extrapolate new physics, where we let LSTM learn and predict the trajectories constrained at a different diffusion-related parameter.

In summary, our work combines statistical mechanics and machine learning and applies to generic time series from high-dimensional complex dynamical systems. For example, we have been working on using our methods in Chapter. 2 and 3 to study kinetic pathways in the crystal nucleation of urea from aqueous solution. SGOOP-d itself also presents an approach for systematically finding the sufficient dimension of RCs. We are also working on using our methods in Chapter. 4 and 5 to study open quantum systems, which also shows the interdisciplinary nature of our work including but not limited to classical, quantum chemical, biological physics, and machine learning. Several preliminary results of open quantum systems are shown in Appendix A.

Appendix A: Learning quantum jump dynamics from open quantum systems using path sampling LSTM

A.1 Introduction

Simulating the dynamics of quantum systems is a long-lasting challenge in the physics community. In the real world, typically, the quantum-mechanical systems are governed by non-unitary evolutions because they are surrounded by a dissipative environment. The general form of the quantum Markov process is the Lindblad equation [209]. However, due to the non-unitarity, we need to solve a partial differential equation of density matrix instead of quantum states (vectors), which is classically challenging. However, we can only measure observables in experiments where the density matrices are inaccessible. Here, we combine the quantum jump approach [210,211] and ps-LSTM to generate quantum trajectories that provide correct expectation values of observable.

In this section, we will show a more challenging example, an open quantum system that consists of a single two-level atom experiencing interactions from initially seven photons, where the photons continuingly dissipate to the environment via a certain dissipation rate (see Fig. A.1). This example is intrinsically hard because the system has a Hilbert space with 20 dimensions yet we only let LSTM see the individual quantum

trajectories of a 1-dimensional observable. Although the quantum trajectories produced by Monte Carlo simulation in Hilbert space are Markovian, the dimensionality reduction from high-dimensional Hilbert space to the observable results in non-Markovian trajectories. In this example, we will let LSTM learn a dissipative observable which is the number of photons. We will then show how we can use our ps-LSTM method to learn to predict trajectories of observable with dissipation rate $\gamma = 0.2$ given training data consisting of only trajectories generated from simulations with $\gamma = 0.1$.

The time-evolution of an open quantum system with $\dim \mathcal{H} = N$ is governed by Lindblad Master equation [209, 212–214]:

$$\dot{\rho} = -\frac{i}{\hbar}[H, \rho] + \sum_{i=1}^{N^2-1} \gamma_i \left(L_i \rho L_i^\dagger - \frac{1}{2} \left\{ L_i^\dagger L_i, \rho \right\} \right). \tag{A.1}$$

where ρ is the density matrix, H is a Hamiltonian of the system, and L_i are commonly called the Lindblad or jump operators of the system. γ_i is the dissipation rate corresponding to jump operator L_i. For convenience, we choose the natural unit where $\hbar = 1$. For a large system, directly solving (A.1) is a formidable task. Therefore, an alternative approach is to perform Monte Carlo (MC) quantum-jump method [215–217], which requires us to generate a large enough number of trajectories to produce correct expectation values of observables. Our training data for LSTM is therefore a set of quantum jump trajectories generated by the Monte Carlo quantum jump algorithm. Here we consider a simple two-level atom coupled to a leaky single-mode cavity through a dipole-type inter-

action [218]:

$$H_{\text{sys}} = \omega_1 a^\dagger a + \omega_2 \sigma_+ \sigma_- + g(\sigma_- a^\dagger + a\sigma_+) \tag{A.2}$$

The $a, a^\dagger$ and σ_-, σ_+ are the annihilation and creation operators of photon and spin, respectively. Suppose above system is surrounded in a dissipative system which induces single-photon loss of cavity. In the quantum jump picture, we can write down following non-Hermitian Hamiltonian

$$H = H_{\text{sys}} - \frac{i\gamma}{2} a^\dagger a \tag{A.3}$$

where there is only one dissipation channel which is called photon emission with jump operator $\sqrt{\gamma}a$.

We use the built-in Monte Carlo solver in the qutip package [210, 211] with a pre-selected dissipation rate γ to generate a bunch of quantum jump trajectories of the cavity photon number n_t. It is important to note that although the Lindbladian and quantum jump method are Markov processes in Hilbert space, the quantum jump trajectories of n_t learned by LSTM do not need to be Markovian in a coarse-grained state space $\langle n \rangle$.

In an approxmated sense, the dissipation rate γ appears as a parameter controlling the classically exponential decay of $\langle n_t \rangle$:

$$\langle n_t \rangle^{\text{theory}} \approx n_0 e^{-\gamma t} \tag{A.4}$$

therefore, given the values of γ and t, we can estimate the corresponding $\langle n_t \rangle^{\text{theory}}$. This

$\langle n_t \rangle^{\text{theory}}$ will later be used as the constraint variable for ps-LSTM.

In general, the Lindbladian equation describes the time-evolution of a $N \times N$ matrix which is computationally challenging. However, the averaged trajectory of the observables, i.e. $\langle n_t \rangle$, is typically governed by a set of differential equations whose number of coefficients is much less than N^2. Previous work [219] has already demonstrated that standard LSTM can learn the feature of decaying pattern from the averaged trajectory $\langle n_t \rangle$, while it is definitely more useful yet challenging for the LSTM to learn the probabilistic model from the individual quantum trajectories n_t and generate the stochastic trajectories with the correct expectation of the observable at every single time step since learning such stochastic trajectories allows us to do ps-LSTM and generate trajectory of observable with a different dissipation rate.

A.2 Results

Here we demonstrate how to apply ps-LSTM trained by individual trajectories from one dissipation rate to generate quantum trajectories with another dissipation rate. The parameters of Hamiltonian Eq. (A.2) are $\omega_1 = \omega_2 = 2\pi$, and $g = \frac{\pi}{2}$. As what we did in the previous example, we first spatially discretize n_t, which is the trajectories generated from the actual Monte Carlo quantum jump simulations with $\gamma = 0.1$, into 20 bins. We then let LSTM learn such trajectories and generate a set of predictions given only the starting condition of $n_t = 7$, as shown in Fig. A.1(c). For training LSTM to learn the quantum jump trajectories, we took the embedding dimension M = 16 and the LSTM unit L = 64. The time series were batched into sequences with a sequence length of 100 ana batch

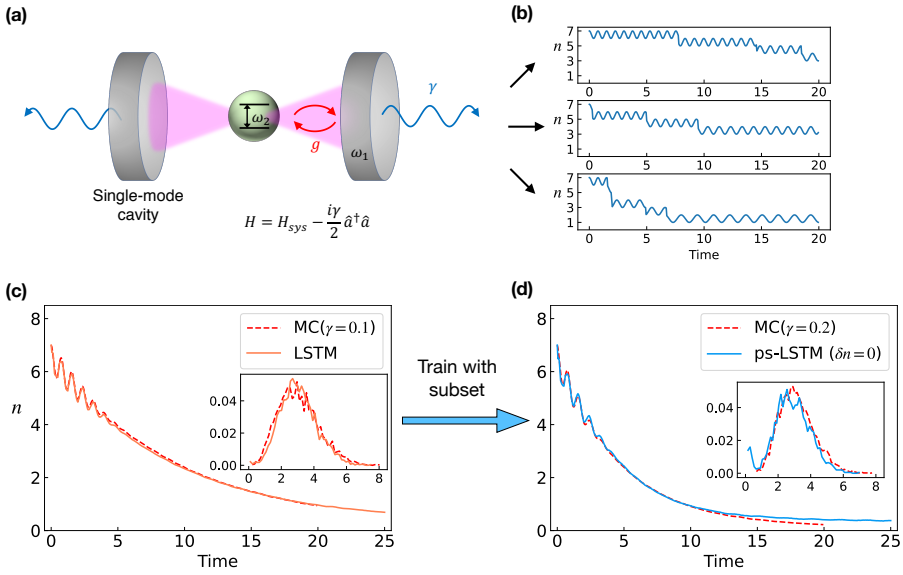


Figure A.1: **Path sampling quantum jump trajectories generated from LSTM** (a) The schematic plot of the open quantum system we simulate. The system consists of a two-level atom surrounded by the cavity, where the initial state is chosen to be $7 \otimes \uparrow$. The cavity photons not only experience interaction with the atom but also interact with the environment via continuously dissipating photons to the environment. The system can be described by the Hamiltonian written below the plot, where system Hamiltonian H_{sys} is just Eq. A.2 with $\omega_1 = \omega_2 = 2\pi$, $g = \frac{\pi}{2}$. ω_1 and ω_2 is the frequency of the cavity and two-level atom, respectively. (b) Some example trajectories from the quantum jump simulations which we used to train LSTM and ps-LSTM. (c) The expectation of photon number $\langle n \rangle$ as a function of time obtained by averaging over 2000 MC simulations with $\gamma = 0.1$ and 2000 predictions generated by LSTM. The inserted panel shows the distribution of the variance calculated over each trajectory. The calculation from MC is shown by the red dashed curve and LSTM by the orange solid curve. (d) The expectation of photon number $\langle n \rangle$ as a function of time obtained by averaging over 2000 MC simulations with $\gamma = 0.2$ and 2000 predictions generated by ps-LSTM. The inserted panel shows the distribution of the variance calculated over each trajectory. The calculation from MC is shown by a red dashed curve and ps-LSTM by the blue solid curve.

size of 64. The models were trained with the method of stochastic gradient descent for 20 epochs. After sampling 20,000 LSTM predictions of length 500, a ps-LSTM was retrained with the same hyperparameters except for increasing sequence length to 140. In Fig. A.1(d), it can be seen that these predictions from LSTM follow the correct evolution curve averaged from the actual Monte Carlo quantum jump simulation with $\gamma = 0.1$.

Now we will constrain our LSTM model to learn a different dissipation rate $\gamma = 0.2$. In order to use ps-LSTM to sample $\gamma = 0.2$, we use Eq. A.4 to estimate the corresponding $\langle n \rangle_t^*$ within the time interval $t \in (5, 7)$. Following the similar spirit of s-ensemble, we define a dynamical variable δn, where

$$\delta n = \frac{1}{\Delta t} \sum_{j=1}^{K} \sum_{s}^{t+\Delta t} \| n_s^j - \langle n_s \rangle^{\text{theory}} \|^2 \tag{A.5}$$

where $\langle n \rangle^{\text{theory}}$ is calculated from Eq. A.4 with $\gamma = 0.2$. K is the number of subsamples, which was chosen to be 2000. $t = 5$ and $\Delta t = 2$ are chosen such that minimizing δn leads to a curve fit of exponential decay in classical regime. The ps-LSTM is then performed by constraining $\delta n = 0$. Constraining LSTM to learn a different γ is very challenging if we only let LSTM learn the averaged trajectory as in Ref. [219], since the oscillating feature within the first 5 time units is a quantum mechanical effect and is hard to capture by simply changing γ.

However, by performing path sampling, we show that by constraining only the δn in classical regime, ps-LSTM produced the correct quantum dynamics it captures from the quantum jump trajectories, which can be seen in Fig. A.1(d). It is also worth noting that we actually perform a more challenging task in the prediction, where we let LSTM and ps-LSTM predict 5 time units more than the trajectories in the training set. That said, LSTM and ps-LSTM still give the prediction of $\langle n_t \rangle$ for $t > 20$, wherein it captures that the cavity photon number has been mostly dissipated and the averaged photon number does not change.

A.3 Method

Here, we start to explain the quantum jump algorithm [210, 211]. We monitored the environment continuously which along with a series of quantum jumps conditioned on the increase in information gained via the environmental measurements. In general, this evolution is governed by the Schrödinger equation with a non-Hermitian effective Hamiltonian

$$H_{\text{eff}} = H_{\text{sys}} - \frac{i}{2}\sum_i C_i^\dagger C_i, \tag{A.6}$$

where $C_i = \sqrt{\gamma_i} L_i$.

Here, the strictly negative non-Hermitian part of Eq. A.6 gives rise to a reduction in the norm of the wave function, that to first-order in a small time δt is given by

$$\langle\psi(t+\delta t)|\psi(t+\delta t)\rangle = 1 - \delta p, \tag{A.7}$$

where

$$\delta p = \delta t \sum_n \left\langle \psi(t)|C_n^+ C_n|\psi(t)\right\rangle. \tag{A.8}$$

The wave function at time t undergoes a jump operator C_n into a state corresponding to the measurement:

$$|\psi(t+\delta t)\rangle = \frac{C_n\,|\psi(t)\rangle}{\langle\psi(t)|C_n^+ C_n|\psi(t)\rangle^{1/2}}. \tag{A.9}$$

The probability of collapse due to the ith-operator C_i is given by

$$P_i(t) = \frac{\langle \psi(t)|C_i^+ C_i|\psi(t)\rangle}{\delta p}. \tag{A.10}$$

To simply simulating the first order differential equation for large system is quite difficult. So we use the following algorithm [210, 211]. Here, we illustrate the steps for MC evolution, first we start from a pure state $|\psi(0)\rangle$.

- Choose a random number $r_1 \in [0, 1]$, representing the probability that a quantum jump occurs.
- Choose a random number $r_2 \in [0, 1]$, used to select which collapse operator was responsible for the jump.
- Integrate the Schrödinger equation, using the effective Hamiltonian Eq. A.6 until a time τ such that the norm of the wave function satisfies $\langle \psi(\tau)\ |\psi(\tau)\rangle = r_1$, at which point a jump occurs.
- The resultant jump projects the system at time τ into one of the renormalized states given by Eq. A.9. The corresponding collapse operator L_n is chosen such that n is the smallest integer satisfying: $\sum_{i=1}^{n} P_n(\tau) \geq r_2$ where the individual P_n are given by Eq. A.10.
- Using the renormalized state from step III as the new initial condition at time τ, draw a new random number, and repeat the above procedure until the final simulation time is reached.

List of Publications

Part of this thesis is based on work published by the author. Here we present the references that each chapter is based on and also mention other relevant publications of the author that do not appear in this thesis.

- Chapter. 2 is based on
  - "Reaction coordinates and rate constants for liquid droplet nucleation: quantifying the interplay between driving force and memory", Sun-Ting Tsai, Zachary Smith, Pratyush Tiwary, J. Chem. Phys. **151** (15), 154106 (2019)
- Chapter. 3 is based on
  - "SGOOP-d: Estimating kinetic distances and reaction coordinate dimensionality for rare event systems from biased/unbiased simulations", Sun-Ting Tsai, Zachary Smith, Pratyush Tiwary, J. Chem. Theory Comput. **17**, 11, 6757-6765 (2021)
  - "On the distance between A and B in molecular configuration space", Sun-Ting Tsai and Pratyush Tiwary, Molecular Simulation. 1-8 (2020)
  - "Multi-dimensional spectral gap optimization of order parameters (SGOOP) through conditional probability factorization", Zachary Smith, Sun-Ting Tsai,

Debabrata Pramanik, Pratyush Tiwary, J. Chem. Phys. **149** (23), 234105 (2018)

- Chapter. 4 is based on
  - "Learning Molecular Dynamics with Simple Language Model built upon Long Short-Term Memory Neural Network", Sun-Ting Tsai, En-Jui Kuo, Pratyush Tiwary, Nat. Commu. **11** 5115 (2020)
- Chapter. 5 is based on
  - "Path sampling of recurrent neural networks by incorporating known physics", Sun-Ting Tsai, Eric Fields, Pratyush Tiwary, arXiv preprint arXiv:2203.00597.
- Other relevant publications that do not appear in this thesis
  - "Toward Automated Sampling of Polymorph Nucleation and Free Energies with the SGOOP and Metadynamics", Ziyue Zou, Sun-Ting Tsai, Pratyush Tiwary, J. Phys. Chem. B **125**, 47, 13049–13056 (2021)
  - "Crumple-Origami Transition for Twisting Cylindrical Shells", Li-Min Wang, Sun-Ting Tsai, Chih-yu Lee, Pai-Yi Hsiao, Jia-Wei Deng, Hung-Chieh Fan Chiang, Yicheng Fei, Tzay-Ming Hong, Phys. Rev. E **101** (5), 053001 (2020)
  - "Kinetics of Ligand–Protein Dissociation from All-Atom Simulations: Are We There Yet?", João Marcelo Ribeiro, Sun-Ting Tsai, Debabrata Pramanik, Yihang Wang, Pratyush Tiwary, Biochemistry **58** (3), 156-165 (2018)

Bibliography

[1] Scott A Hollingsworth and Ron O Dror. Molecular dynamics simulation for all. *Neuron*, 99(6):1129–1143, 2018.

[2] Haiyang Niu, Pablo M Piaggi, Michele Invernizzi, and Michele Parrinello. Molecular dynamics simulations of liquid silica crystallization. *Proceedings of the National Academy of Sciences*, 115(21):5348–5352, 2018.

[3] Thorsten Bartels-Rausch. Chemistry: Ten things we need to know about ice and snow. *Nature*, 494(7435):27, 2013.

[4] Benjamin J Murray, Theodore W Wilson, Steven Dobbie, Zhiqiang Cui, Sardar MRK Al-Jumur, Ottmar Möhler, Martin Schnaiter, Robert Wagner, Stefan Benz, Monika Niemand, et al. Heterogeneous nucleation of ice particles on glassy aerosols under cirrus conditions. *Nature Geoscience*, 3(4):233, 2010.

[5] Peter Mazur. Cryobiology: the freezing of biological systems. *Science*, 168(3934):939–949, 1970.

[6] Jason R Cox, Lori A Ferris, and Venkat R Thalladi. Selective growth of a stable drug polymorph by suppressing the nucleation of corresponding metastable poly-

morphs. *Angewandte Chemie International Edition*, 46(23):4333–4336, 2007.

[7] Deniz Erdemir, Alfred Y Lee, and Allan S Myerson. Polymorph selection: the role of nucleation, crystal growth and molecular modeling. *Current opinion in drug discovery & development*, 10(6):746–755, 2007.

[8] E Dendy Sloan. Fundamental principles and applications of natural gas hydrates. *Nature*, 426(6964):353, 2003.

[9] EG Hammerschmidt. Formation of gas hydrates in natural gas transmission lines. *Industrial & Engineering Chemistry*, 26(8):851–855, 1934.

[10] James D Harper, Charles M Lieber, and Peter T Lansbury Jr. Atomic force microscopic imaging of seeded fibril formation and fibril branching by the alzheimer's disease amyloid-β protein. *Chemistry & biology*, 4(12):951–959, 1997.

[11] Samuel IA Cohen, Sara Linse, Leila M Luheshi, Erik Hellstrand, Duncan A White, Luke Rajah, Daniel E Otzen, Michele Vendruscolo, Christopher M Dobson, and Tuomas PJ Knowles. Proliferation of amyloid-β42 aggregates occurs through a secondary nucleation mechanism. *Proceedings of the National Academy of Sciences*, 110(24):9758–9763, 2013.

[12] Martin Karplus and John Kuriyan. Molecular dynamics and protein function. *Proceedings of the National Academy of Sciences*, 102(19):6679–6685, 2005.

[13] Harold A Scheraga, Mey Khalili, and Adam Liwo. Protein-folding dynamics: overview of molecular simulation techniques. *Annu. Rev. Phys. Chem.*, 58:57–83, 2007.

[14] Pratyush Tiwary, Jagannath Mondal, and Bruce J Berne. How and when does an anticancer drug leave its binding site? *Science advances*, 3(5):e1700014, 2017.

[15] Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of medicinal chemistry*, 59(9):4035–4061, 2016.

[16] Boran Ma and Monica Olvera de la Cruz. A perspective on the design of ion-containing polymers for polymer electrolyte applications. *The Journal of Physical Chemistry B*, 125(12):3015–3022, 2021.

[17] Annalisa Cardellini, Felipe Jiménez-Ángeles, Pietro Asinari, and Monica Olvera de la Cruz. A modeling-based design to engineering protein hydrogels with random copolymers. *ACS nano*, 15(10):16139–16148, 2021.

[18] Alex Bunker and Tomasz Róg. Mechanistic understanding from molecular dynamics simulation in pharmaceutical research 1: drug delivery. *Frontiers in Molecular Biosciences*, page 371, 2020.

[19] Jacob D Durrant and J Andrew McCammon. Molecular dynamics simulations and drug discovery. *BMC biology*, 9(1):1–9, 2011.

[20] Yi Isaac Yang, Qiang Shao, Jun Zhang, Lijiang Yang, and Yi Qin Gao. Enhanced sampling in molecular dynamics. *The Journal of chemical physics*, 151(7):070902, 2019.

[21] Ugo Mayor, Christopher M Johnson, Valerie Daggett, and Alan R Fersht. Protein folding and unfolding in microseconds to nanoseconds by experiment and simu-

lation. *Proceedings of the National Academy of Sciences*, 97(25):13518–13522, 2000.

[22] Andrea Amadei, Antonius BM Linssen, and Herman JC Berendsen. Essential dynamics of proteins. *Proteins: Structure, Function, and Bioinformatics*, 17(4):412–425, 1993.

[23] Rainer Hegger, Alexandros Altis, Phuong H Nguyen, and Gerhard Stock. How complex is the dynamics of peptide folding? *Physical review letters*, 98(2):028102, 2007.

[24] Angel E García. Large-amplitude nonlinear motions in proteins. *Physical review letters*, 68(17):2696, 1992.

[25] Alexander Berezhkovskii and Attila Szabo. One-dimensional reaction coordinates for diffusive activated rate processes in many dimensions. *J. Chem. Phys.*, 122(1):014503, 2005.

[26] Pratyush Tiwary and B. J. Berne. Spectral gap optimization of order parameters for sampling complex molecular systems. *Proc. Natl. Acad. Sci.*, 113(11):2839, 2016.

[27] Polina V Banushkina and Sergei V Krivov. Optimal reaction coordinates. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 6(6):748–763, 2016.

[28] Ao Ma and Aaron R Dinner. Automatic method for identifying reaction coordinates in complex systems. *J. Phys. Chem. B*, 109(14):6769–6779, 2005.

[29] Robert T McGibbon, Brooke E Husic, and Vijay S Pande. Identification of simple reaction coordinates from complex dynamics. *The Journal of Chemical Physics*, 146(4):044109, 2017.

[30] João Marcelo Lamim Ribeiro, Pablo Bravo, Yihang Wang, and Pratyush Tiwary. Reweighted autoencoded variational bayes for enhanced sampling. *J. Chem. Phys.*, 149(7):072301, 2018.

[31] Zachary Smith, Debabrata Pramanik, Sun-Ting Tsai, and Pratyush Tiwary. Multi-dimensional spectral gap optimization of order parameters (sgoop) through conditional probability factorization. *J. Chem. Phys.*, 149(23):234105, 2018.

[32] Lutz Molgedey and Heinz Georg Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical review letters*, 72(23):3634, 1994.

[33] Yusuke Naritomi and Sotaro Fuchigami. Slow dynamics of a protein backbone in molecular dynamics simulation revealed by time-structure based independent component analysis. *The Journal of Chemical Physics*, 139(21):12B605_1, 2013.

[34] Brooke E Husic and Vijay S Pande. Markov state models: From an art to a science. *J. Am. Chem. Soc.*, 140(7):2386, 2018.

[35] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1):1–11, 2018.

[36] 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, 2021.

[37] 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):214114, 2019.

[38] Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. *Phys Rev Lett*, 100(2):020603, 2008.

[39] Omar Valsson, Pratyush Tiwary, and Michele Parrinello. Enhancing important fluctuations: Rare events and metadynamics from a conceptual viewpoint. *Ann. Rev. Phys. Chem.*, 67(1):159, 2016.

[40] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proc Natl Acad Sci*, 99(20):12562, 2002.

[41] Pratyush Tiwary and Michele Parrinello. A time-independent free energy estimator for metadynamics. *J. Phys. Chem. B*, 119(3):736–742, 2014.

[42] Pratyush Tiwary and Michele Parrinello. From metadynamics to dynamics. *Phys. Rev. Lett.*, 111(23):230602, 2013.

[43] Arthur F. Voter. Hyperdynamics: Accelerated molecular dynamics of infrequent events. *Phys. Rev. Lett.*, 78(20):3908, 1997.

[44] Helmut Grubmüller. Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Phys. Rev. E*, 52(3):2893, 1995.

[45] Matteo Salvalaglio, Claudio Perego, Federico Giberti, Marco Mazzotti, and Michele Parrinello. Molecular-dynamics simulations of urea nucleation from aqueous solution. *Proceedings of the National Academy of Sciences*, 112(1):E6–E14, 2015.

[46] Sooheyong Lee, Haeng Sub Wi, Wonhyuk Jo, Yong Chan Cho, Hyun Hwi Lee, Se-Young Jeong, Yong-Il Kim, and Geun Woo Lee. Multiple pathways of crystal nucleation in an extremely supersaturated aqueous potassium dihydrogen phosphate (kdp) solution droplet. *Proceedings of the National Academy of Sciences*, 113(48):13618–13623, 2016.

[47] Jorjethe Roca, Naoto Hori, Saroj Baral, Yogambigai Velmurugu, Ranjani Narayanan, Prasanth Narayanan, D Thirumalai, and Anjum Ansari. Monovalent ions modulate the flux through multiple folding pathways of an rna pseudoknot. *Proceedings of the National Academy of Sciences*, 115(31):E7313–E7322, 2018.

[48] Andrew GT Pyo and Michael T Woodside. Memory effects in single-molecule force spectroscopy measurements of biomolecular folding. *Physical Chemistry Chemical Physics*, 21(44):24527–24534, 2019.

[49] Gabriele C Sosso, Ji Chen, Stephen J Cox, Martin Fitzner, Philipp Pedevilla, Andrea Zen, and Angelos Michaelides. Crystal nucleation in liquids: Open questions and future challenges in molecular dynamics simulations. *Chemical reviews*, 116(12):7078–7116, 2016.

[50] Guram Chkonia, Judith Wölk, Reinhard Strey, Jan Wedekind, and David Reguera. Evaluating nucleation rates in direct simulations. *The Journal of chemical physics*, 130(6):064505, 2009.

[51] Matteo Salvalaglio, Pratyush Tiwary, Giovanni Maria Maggioni, Marco Mazzotti, and Michele Parrinello. Overcoming time scale and finite size limitations to compute nucleation rates from small scale well tempered metadynamics simulations. *The Journal of chemical physics*, 145(21):211925, 2016.

[52] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.

[53] Shankar Kumar, John M Rosenberg, Djamal Bouzida, Robert H Swendsen, and Peter A Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of computational chemistry*, 13(8):1011–1021, 1992.

[54] Titus S van Erp, Daniele Moroni, and Peter G Bolhuis. A novel path sampling method for the calculation of rate constants. *The Journal of chemical physics*, 118(17):7762–7774, 2003.

[55] Daniele Moroni, Titus S van Erp, and Peter G Bolhuis. Investigating rare events by transition interface sampling. *Physica A: Statistical Mechanics and its Applications*, 340(1-3):395–401, 2004.

[56] Rosalind J Allen, Daan Frenkel, and Pieter Rein ten Wolde. Simulating rare events in equilibrium or nonequilibrium stochastic systems. *The Journal of chemical physics*, 124(2):024102, 2006.

[57] Rosalind J Allen, Chantal Valeriani, and Pieter Rein ten Wolde. Forward flux sampling for rare event simulations. *J. Phys.: Cond. Matt.*, 21(46):463102, 2009.

[58] Alexander Berezhkovskii and Attila Szabo. One-dimensional reaction coordinates for diffusive activated rate processes in many dimensions. *J. Chem. Phys.*, 122(1):014503–014506, 2005.

[59] Ryan S DeFever and Sapna Sarupria. Contour forward flux sampling: Sampling rare events along multiple collective variables. *J. Chem. Phys.*, 150(2):024103–024112, 2019.

[60] Christoph Dellago, Peter G Bolhuis, and David Chandler. Efficient transition path sampling: Application to lennard-jones cluster rearrangements. *The Journal of chemical physics*, 108(22):9236–9245, 1998.

[61] PeteráG Bolhuis et al. Sampling ensembles of deterministic transition pathways. *Faraday Discussions*, 110:421–436, 1998.

[62] Matteo Salvalaglio, Pratyush Tiwary, Giovanni Maria Maggioni, Marco Mazzotti, and Michele Parrinello. Overcoming time scale and finite size limitations to compute nucleation rates from small scale well tempered metadynamics simulations. *J. Chem. Phys.*, 145(21):211925–211936, 2016.

[63] J Kuipers and GT Barkema. Non-markovian dynamics of clusters during nucleation. *Physical Review E*, 79(6):062101, 2009.

[64] IJ Ford. Statistical mechanics of nucleation: a review. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 218(8):883–899, 2004.

[65] Pratyush Tiwary and B. J. Berne. Predicting reaction coordinates in energy landscapes with diffusion anisotropy. *J. Chem. Phys.*, 147(15):152701, 2017.

[66] Pratyush Tiwary and B. J. Berne. How wet should be the reaction coordinate for ligand unbinding? *J. Chem. Phys.*, 145(5):054113, 2016.

[67] A Weber and M Volmer. Keimbildung in übersättigten gebilden. *Zeitschrift für Physikalische Chemie*, 119:277, 1926.

[68] L Farkas. Keimbildungsgeschwindigkeit in übersättigten dämpfen. *Zeitschrift für physikalische Chemie*, 125(1):236–242, 1927.

[69] Richard Becker and Werner Döring. Kinetische behandlung der keimbildung in übersättigten dämpfen. *Annalen der Physik*, 416(8):719–752, 1935.

[70] Ya B Zeldovich. On the theory of new phase formation: cavitation. *Acta Physicochem., USSR*, 18:1, 1943.

[71] Humphrey J Maris. Introduction to the physics of nucleation. *Comptes Rendus Physique*, 7(9-10):946–958, 2006.

[72] V. I. Kalikmanov. *Classical Nucleation Theory*, pages 17–41. Springer Netherlands, Dordrecht, 2013.

[73] Alexander Berezhkovskii and Attila Szabo. Time scale separation leads to position-dependent diffusion along a slow coordinate. *J. Chem. Phys.*, 135(7):074108, 2011.

[74] Gert van der Zwan and James T Hynes. Reactive paths in the diffusion limit. *J. Chem. Phys.*, 77(3):1295–1301, 1982.

[75] James T Hynes. Chemical reaction dynamics in solution. *Annual Review of Physical Chemistry*, 36(1):573–597, 1985.

[76] Baron Peters, Peter G Bolhuis, Ryan G Mullen, and Joan-Emma Shea. Reaction coordinates, one-dimensional smoluchowski equations, and a test for dynamical self-consistency. *J. Chem. Phys.*, 138(5):054106, 2013.

[77] Baron Peters. On the coupling between slow diffusion transport and barrier crossing in nucleation. *The Journal of chemical physics*, 135(4):044107, 2011.

[78] M Rao, BJ Berne, and MH Kalos. Computer simulation of the nucleation and thermodynamics of microclusters. *The Journal of Chemical Physics*, 68(4):1325–1336, 1978.

[79] Hui Wang, Harvey Gould, and W Klein. Homogeneous and heterogeneous nucleation of lennard-jones liquids. *Physical Review E*, 76(3):031604, 2007.

[80] VI Kalikmanov, J Wölk, and T Kraska. Argon nucleation: Bringing together theory, simulations, and experiment. *The Journal of chemical physics*, 128(12):124506, 2008.

[81] Pieter Rein ten Wolde and Daan Frenkel. Enhancement of protein crystal nucleation by critical density fluctuations. *Science*, 277(5334):1975–1978, 1997.

[82] Federica Trudu, Davide Donadio, and Michele Parrinello. Freezing of a lennard-jones fluid: From nucleation to spinodal regime. *Physical review letters*, 97(10):105701, 2006.

[83] Daniele Moroni, Pieter Rein Ten Wolde, and Peter G Bolhuis. Interplay between structure and size in a critical crystal nucleus. *Physical review letters*, 94(23):235703, 2005.

[84] Soon Gu Kwon, Galyna Krylova, Patrick J Phillips, Robert F Klie, Soma Chattopadhyay, Tomohiro Shibata, Emilio E Bunel, Yuzi Liu, Vitali B Prakapenka, Byeongdu Lee, et al. Heterogeneous nucleation and shape transformation of multicomponent metallic nanostructures. *Nature materials*, 14(2):215, 2015.

[85] Jihan Zhou, Yongsoo Yang, Yao Yang, Dennis S Kim, Andrew Yuan, Xuezeng Tian, Colin Ophus, Fan Sun, Andreas K Schmid, Michael Nathanson, et al. Observing crystal nucleation in four dimensions using atomic electron tomography. *Nature*, 570(7762):500, 2019.

[86] Pieter Rein ten Wolde and Daan Frenkel. Computer simulation study of gas–liquid nucleation in a lennard-jones system. *The Journal of chemical physics*, 109(22):9901–9918, 1998.

[87] Gareth A Tribello, Federico Giberti, Gabriele C Sosso, Matteo Salvalaglio, and Michele Parrinello. Analyzing and driving cluster formation in atomistic simulations. *Journal of chemical theory and computation*, 13(3):1317–1327, 2017.

[88] Gareth A Tribello, Michele Ceriotti, and Michele Parrinello. A self-learning algorithm for biased molecular dynamics. *Proc. Natl. Acad. Sci.*, 107(41):17509–17514, 2010.

[89] Shawn M Kathmann, Gregory K Schenter, and Bruce C Garrett. Multicomponent dynamical nucleation theory and sensitivity analysis. *J. Chem. Phys.*, 120(19):9133–9141, 2004.

[90] Sidney I Resnick. *Adventures in stochastic processes*. Springer Science & Business Media, 2013.

[91] Matteo Salvalaglio, Pratyush Tiwary, and Michele Parrinello. Assessing the reliability of the dynamics reconstructed from metadynamics. *J. Chem. Theor. Comp.*, 10(4):1420, 2014.

[92] Alexander Berezhkovskii and Attila Szabo. Time scale separation leads to position-dependent diffusion along a slow coordinate. *The Journal of chemical physics*, 135(7):074108, 2011.

[93] Purushottam D Dixit, Abhinav Jain, Gerhard Stock, and Ken A Dill. Inferring transition rates of networks from populations in continuous-time markov processes. *J. Chem. Theor. Comp.*, 11(11):5464, 2015.

[94] Edwin T Jaynes. The minimum entropy production principle. *Ann. Rev. Phys. Chem.*, 31(1):579, 1980.

[95] Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A. Dill. Principles of maximum entropy and maximum caliber in statistical physics. *Rev. Mod. Phys.*, 85(3):1115, 2013.

[96] Purushottam D Dixit, Jason Wagoner, Corey Weistuch, Steve Pressé, Kingshuk Ghosh, and Ken A Dill. Perspective: Maximum caliber is a general variational principle for dynamical systems. *The Journal of chemical physics*, 148(1):010901, 2018.

[97] Pratyush Tiwary and BJ Berne. Predicting reaction coordinates in energy landscapes with diffusion anisotropy. *The Journal of chemical physics*, 147(15):152701, 2017.

[98] DJ Bicout and Attila Szabo. Electron transfer reaction dynamics in non-debye solvents. *J. Chem. Phys.*, 109(6):2325–2338, 1998.

[99] Pratyush Tiwary and BJ Berne. Spectral gap optimization of order parameters for sampling complex molecular systems. *Proceedings of the National Academy of Sciences*, 113(11):2839–2844, 2016.

[100] Giovanni Bussi, Davide Donadio, and Michele Parrinello. Canonical sampling through velocity rescaling. *The Journal of chemical physics*, 126(1):014101, 2007.

[101] Erik Lindahl, Berk Hess, and David Van Der Spoel. Gromacs 3.0: a package for molecular simulation and trajectory analysis. *Molecular modeling annual*, 7(8):306–317, 2001.

[102] Carlo Camilloni et. al. Massimiliano Bonomi, Giovanni Bussi. Promoting transparency and reproducibility in enhanced molecular simulations. *Nature methods*, 16:670–673, 2019.

[103] Rodrigo Casasnovas, Vittorio Limongelli, Pratyush Tiwary, Paolo Carloni, and Michele Parrinello. Unbinding kinetics of a p38 map kinase type ii inhibitor from metadynamics simulations. *J. Am. Chem. Soc.*, 139(13):4780, 2017.

[104] Baron Peters and Bernhardt L Trout. Obtaining reaction coordinates by likelihood maximization. *The Journal of chemical physics*, 125(5):054108, 2006.

[105] Robert B Best and Gerhard Hummer. Reaction coordinates and rates from transition paths. *Proc. Natl. Acad. Sci.*, 102(19):6732, 2005.

[106] K Binder. Nucleation barriers, spinodals, and the ginzburg criterion. *Physical Review A*, 29(1):341, 1984.

[107] J Juraszek, G Saladino, TS Van Erp, and FL Gervasio. Efficient numerical reconstruction of protein folding kinetics with partial path sampling and pathlike variables. *Physical Review Letters*, 110(10):108106, 2013.

[108] Sun-Ting Tsai, Zachary Smith, and Pratyush Tiwary. Reaction coordinates and rate constants for liquid droplet nucleation: Quantifying the interplay between driving force and memory. *The Journal of chemical physics*, 151(15):154106, 2019.

[109] 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. *J. Chem. Phys.*, 134(17):174105, 2011.

[110] Daniel Nagel, Anna Weber, and Gerhard Stock. Msmpathfinder: Identification of pathways in markov state models. *Journal of Chemical Theory and Computation*, 16(12):7874–7882, 2020.

[111] Stefano Piana, Kresten Lindorff-Larsen, and David E Shaw. Protein folding kinetics and thermodynamics from atomistic simulation. *Proceedings of the National Academy of Sciences*, 109(44):17845–17850, 2012.

[112] Robert B Best and Gerhard Hummer. Reaction coordinates and rates from transition paths. *Proc. Natl. Acad. Sci.*, 102(19):6732–6737, 2005.

[113] 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, 2015.

[114] Gareth A Tribello and Piero Gasparotto. Using dimensionality reduction to analyze protein trajectories. *Frontiers in molecular biosciences*, 6:46, 2019.

[115] Alexandros Altis, Moritz Otten, Phuong H Nguyen, Rainer Hegger, and Gerhard Stock. Construction of the free energy landscape of biomolecules via dihedral angle principal component analysis. *The Journal of chemical physics*, 128(24):06B620, 2008.

[116] Zachary Smith, Debabrata Pramanik, Sun-Ting Tsai, and Pratyush Tiwary. Multidimensional spectral gap optimization of order parameters (sgoop) through conditional probability factorization. *J. Chem. Phys.*, 149(23):234105, 2018.

[117] Frank Noé and Cecilia Clementi. Kinetic distance and kinetic maps from molecular dynamics simulation. *J. Chem. Theor. Comp.*, 11(10):5002–5011, 2015.

[118] Frank Noe, Ralf Banisch, and Cecilia Clementi. Commute maps: Separating slowly mixing molecular configurations for kinetic modeling. *J. Chem. Theor. Comp.*, 12(11):5620–5630, 2016.

[119] Omar Valsson, Pratyush Tiwary, and Michele Parrinello. Enhancing important fluctuations: Rare events and metadynamics from a conceptual viewpoint. *Ann. Rev. Phys. Chem.*, 67:159–184, 2016.

[120] Pratyush Tiwary and Axel van de Walle. A review of enhanced sampling approaches for accelerated molecular dynamics. In *Multiscale Materials Modeling for Nanomechanics*, pages 195–221. Springer, 2016.

[121] Boaz Nadler, Stephane Lafon, Ioannis Kevrekidis, and Ronald R Coifman. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in neural information processing systems*, pages 955–962, 2006.

[122] Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven 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, 2005.

[123] Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *J. Chem. Phys.*, 139(1):07B604_1, 2013.

[124] Andreas Bittracher, Ralf Banisch, and Christof Schütte. Data-driven computation of molecular reaction coordinates. *The Journal of Chemical Physics*, 149(15):154103, 2018.

[125] Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A Dill. Principles of maximum entropy and maximum caliber in statistical physics. *Rev. Mod. Phys.*, 85(3):1115, 2013.

[126] Kingshuk Ghosh, Purushottam D Dixit, Luca Agozzino, and Ken A Dill. The maximum caliber variational principle for nonequilibria. *Annual review of physical chemistry*, 71:213–238, 2020.

[127] Pratyush Tiwary and Michele Parrinello. A time-independent free energy estimator for metadynamics. *The Journal of Physical Chemistry B*, 119(3):736–742, 2015.

[128] Herman JC Berendsen, David van der Spoel, and Rudi van Drunen. Gromacs: a message-passing parallel molecular dynamics implementation. *Comp. Phys. Commun.*, 91(1-3):43–56, 1995.

[129] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.

[130] Sun-Ting Tsai, En-Jui Kuo, and Pratyush Tiwary. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nature communications*, 11(1):1–11, 2020.

[131] Pratyush Tiwary and Michele Parrinello. From metadynamics to dynamics. *Physical review letters*, 111(23):230602, 2013.

[132] David Wales et al. *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press, 2003.

[133] David J Wales and Jonathan PK Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[134] Zhenqin Li and Harold A Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proceedings of the National Academy of Sciences*, 84(19):6611–6615, 1987.

[135] Jakub Rydzewski and Omar Valsson. Multiscale reweighted stochastic embedding (mrse): Deep learning of collective variables for enhanced sampling. *arXiv preprint arXiv:2007.06377*, 2020.

[136] Giovanni Bussi and Alessandro Laio. Using metadynamics to explore complex free-energy landscapes. *Nat. Rev. Phys.*, pages 1–13, 2020.

[137] Daniel M Zuckerman and Lillian T Chong. Weighted ensemble simulation: review of methodology, applications, and software. *Annual review of biophysics*, 46:43–57, 2017.

[138] Ron Elber. Milestoning: An efficient approach for atomically detailed simulations of kinetics in biophysics. *Annual review of biophysics*, 49:69–85, 2020.

[139] Lane W Votapka, Benjamin R Jagger, Alexandra L Heyneman, and Rommie E Amaro. Seekr: simulation enabled estimation of kinetic rates, a computational tool to estimate molecular kinetics and its application to trypsin–benzamidine binding. *The Journal of Physical Chemistry B*, 121(15):3597–3606, 2017.

[140] Hao Jiang, Amir Haji-Akbari, Pablo G Debenedetti, and Athanassios Z Panagiotopoulos. Forward flux sampling calculation of homogeneous nucleation rates from aqueous nacl solutions. *The Journal of chemical physics*, 148(4):044505, 2018.

[141] R Rico-Martinez, K Krischer, IG Kevrekidis, MC Kube, and JL Hudson. Discrete-vs. continuous-time nonlinear signal processing of cu electrodissolution data. *Chem. Engg. Commun.*, 118(1):25–48, 1992.

[142] N Gicquel, JS Anderson, and IG Kevrekidis. Noninvertibility and resonance in discrete-time neural networks for time-series processing. *Physics Letters A*, 238(1):8–18, 1998.

[143] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, and Horst Bunke. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 31(5):855–868, 2008.

[144] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE Intl. Conf. Acous. Sp. Sig. Proc.*, pages 6645–6649, 2013.

[145] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Preprint at https://arxiv.org/abs/1406.1078*, 2014.

[146] SHI Xingjian, Zhourong Chen, Hao Wang, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Adv. in Neur. Inf. Proc. Sys.*, pages 802–810, 2015.

[147] Kai Chen, Yi Zhou, and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. In *IEEE Intl. Conf. Big Data*, pages 2823–2824, 2015.

[148] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Comp. Sci. Rev.*, 3(3):127–149, 2009.

[149] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120(2):024102, 2018.

[150] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neur. Comp.*, 9(8):1735–1780, 1997.

[151] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[152] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *Preprint at https://arxiv.org/abs/1410.8206*, 2014.

[153] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[154] Joshua C Agar, Brett Naul, Shishir Pandya, and Stefan van Der Walt. Revealing ferroelectric switching character using deep recurrent neural networks. *Nat. Commun.*, 10(1):1–11, 2019.

[155] Mohammad Javad Eslamibidgoli, Mehrdad Mokhtari, and Michael H Eikerling. Recurrent neural network-based model for accelerated trajectory analysis in aimd simulations. *Preprint at https://arxiv.org/abs/1909.10124*, 2019.

[156] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[157] Hythem Sidky, Wei Chen, and Andrew L Ferguson. Molecular latent space simulators. *Chem. Sci.*, 2020.

[158] Yihang Wang, João Marcelo Lamim Ribeiro, and Pratyush Tiwary. Past–future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nature communications*, 10(1):1–8, 2019.

[159] Krishna Neupane, Hao Yu, Daniel AN Foster, Feng Wang, and Michael T Woodside. Single-molecule force spectroscopy of the add adenine riboswitch relates folding to regulatory mechanism. *Nucl. Acid. Res.*, 39(17):7677–7687, 2011.

[160] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[161] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[162] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3159–3166, 2019.

[163] Calvin C Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proc. Natl. Acad. Sci.*, 112(7):1907–1911, 2015.

[164] Sun-Ting Tsai and Pratyush Tiwary. On the distance between a and b in molecular configuration space. *Mol. Sim.*, pages 1–8, 2020.

[165] Giovanni Bussi and Michele Parrinello. Accurate sampling using langevin dynamics. *Phys. Rev. E*, 75(5):056707, 2007.

[166] Peter Hänggi, Peter Talkner, and Michal Borkovec. Reaction-rate theory: fifty years after kramers. *Rev. Mod. Phys.*, 62(2):251, 1990.

[167] Bruce J Berne, Michal Borkovec, and John E Straub. Classical and modern methods in reaction rate theory. *J. Phys. Chem.*, 92(13):3711–3725, 1988.

[168] Matteo Salvalaglio, Pratyush Tiwary, and Michele Parrinello. Assessing the reliability of the dynamics reconstructed from metadynamics. *J. Chem. Theor. Comp.*, 10(4):1420–1425, 2014.

[169] Peter G Bolhuis, Christoph Dellago, and David Chandler. Reaction coordinates of biomolecular isomerization. *Proc. Natl. Acad. Sci.*, 97(11):5877–5882, 2000.

[170] Sean R Eddy. What is a hidden markov model? *Nat. Biotechnol.*, 22(10):1315–1316, 2004.

[171] Sean A McKinney, Chirlmin Joo, and Taekjip Ha. Analysis of single-molecule fret trajectories using hidden markov modeling. *Bioph. Jour.*, 91(5):1941–1951, 2006.

[172] Mario Blanco and Nils G Walter. Analysis of complex single-molecule fret time trajectories. In *Methods in enzymology*, volume 472, pages 153–178. Elsevier, 2010.

[173] Gregory R Bowman, Kyle A Beauchamp, George Boxer, and Vijay S Pande. Progress and challenges in the automated construction of markov state models for full protein systems. *J. Chem. Phys.*, 131(12):124101, 2009.

[174] Martin K Scherer, Benjamin Trendelkamp-Schroer, Fabian Paul, Guillermo Pérez-Hernández, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank Noé. Pyemma 2: A software package for estimation, validation, and analysis of markov models. *J. Chem. Theor. Comp.*, 11(11):5525–5542, 2015.

[175] John D Chodera and Frank Noé. Markov state models of biomolecular conformational dynamics. *Current opinion in structural biology*, 25:135–144, 2014.

[176] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[177] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[178] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Victoria Kemi Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4, 2018.

[179] Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):1–14, 2021.

[180] Michael Milford. C. elegans inspires self-driving cars. *Nature Machine Intelligence*, 2(11):661–662, 2020.

[181] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[182] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[183] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[184] João Marcelo Lamim Ribeiro, Pablo Bravo, Yihang Wang, and Pratyush Tiwary. Reweighted autoencoded variational bayes for enhanced sampling (rave). *The Journal of chemical physics*, 149(7):072301, 2018.

[185] Luigi Bonati, Yue-Yu Zhang, and Michele Parrinello. Neural networks-based variationally enhanced sampling. *Proceedings of the National Academy of Sciences*, 116(36):17641–17647, 2019.

[186] Philipp Geiger and Christoph Dellago. Neural networks for local structure detection in polymorphic systems. *The Journal of chemical physics*, 139(16):164105, 2013.

[187] Carl S Adorf, Timothy C Moore, Yannah JU Melle, and Sharon C Glotzer. Analysis of self-assembly pathways with unsupervised machine learning algorithms. *The Journal of Physical Chemistry B*, 124(1):69–78, 2019.

[188] Luigi Bonati, Valerio Rizzi, and Michele Parrinello. Data-driven collective variables for enhanced sampling. *The journal of physical chemistry letters*, 11(8):2998–3004, 2020.

[189] Jutta Rogal, Elia Schneider, and Mark E Tuckerman. Neural-network-based path collective variables for enhanced sampling of phase transformations. *Physical Review Letters*, 123(24):245701, 2019.

[190] Herbert Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach*, volume 5. GMD-Forschungszentrum Informationstechnik Bonn, 2002.

[191] Adil Moghar and Mhamed Hamiche. Stock market prediction using lstm recurrent neural network. *Procedia Computer Science*, 170:1168–1173, 2020.

[192] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2), 2019.

[193] Christopher R Weinberger and Garritt J Tucker. Multiscale materials modeling for nanomechanics. 2016.

[194] Wei Wang, Siqin Cao, Lizhe Zhu, and Xuhui Huang. Constructing markov state models to elucidate the functional conformational changes of complex

biomolecules. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 8(1):e1343, 2018.

[195] Zhiyuan Shi, Min Xu, and Quan Pan. 4-d flight trajectory prediction with constrained lstm network. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7242–7255, 2021.

[196] Liang Yang, Haifeng Hu, Songlong Xing, and Xinlong Lu. Constrained lstm and residual attention for image captioning. *ACM Trans. Multimedia Comput. Commun*, 16(3), 2020.

[197] Yuntian Chen and Dongxiao Zhang. Physics-constrained deep learning of geomechanical logs. *EEE Transactions on Geoscience and Remote Sensing*, 58(8):5932–5943, 2020.

[198] Z Faidon Brotzakis, Michele Vendruscolo, and Peter G Bolhuis. A method of incorporating rate constants as kinetic constraints in molecular dynamics simulations. *Proceedings of the National Academy of Sciences*, 118(2), 2021.

[199] Hao Ge, Steve Pressé, Kingshuk Ghosh, and Ken A Dill. Markov processes follow from the principle of maximum caliber. *The Journal of chemical physics*, 136(6):064108, 2012.

[200] Alex D MacKerell Jr, Donald Bashford, MLDR Bellott, Roland Leslie Dunbrack Jr, Jeffrey D Evanseck, Martin J Field, Stefan Fischer, Jiali Gao, H Guo, Sookhee Ha, et al. All-atom empirical potential for molecular modeling and dy-

namics studies of proteins. *The journal of physical chemistry B*, 102(18):3586–3616, 1998.

[201] William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of chemical physics*, 79(2):926–935, 1983.

[202] Denis J Evans and Brad Lee Holian. The nose–hoover thermostat. *J. Chem. Phys.*, 83(8):4069, 1985.

[203] Michele Parrinello and Aneesur Rahman. Polymorphic transitions in single crystals: A new molecular dynamics method. *Journal of Applied physics*, 52(12):7182–7190, 1981.

[204] Mithun Biswas, Benjamin Lickert, and Gerhard Stock. Metadynamics enhanced markov modeling of protein dynamics. *The Journal of Physical Chemistry B*, 122(21):5508–5514, 2018.

[205] Shams Mehdi, Dedi Wang, Shashank Pant, and Pratyush Tiwary. Accelerating all-atom simulations and gaining mechanistic understanding of biophysical systems through state predictive information bottleneck. *arXiv preprint arXiv:2112.11201*, 2021.

[206] Virgiliu Botan, Ellen HG Backus, Rolf Pfister, Alessandro Moretto, Marco Crisma, Claudio Toniolo, Phuong H Nguyen, Gerhard Stock, and Peter Hamm. Energy transport in peptide helices. *Proceedings of the National Academy of Sciences*, 104(31):12749–12754, 2007.

[207] Wenqi Zeng, Siqin Cao, Xuhui Huang, and Yuan Yao. A note on learning rare events in molecular dynamics using lstm and transformer. *arXiv preprint arXiv:2107.06573*, 2021.

[208] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[209] Goran Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, 1976.

[210] J Robert Johansson, Paul D Nation, and Franco Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[211] Paul D Nation and JR Johansson. Qutip: Quantum toolbox in python. *online at http://qutip. org*, 2011.

[212] H.P. Breuer, F. Petruccione, and S.P.A.P.F. Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2002.

[213] Daniel Manzano. A short introduction to the lindblad master equation. *Aip Advances*, 10(2):025106, 2020.

[214] Carlos Alexandre Brasil, Felipe Fernandes Fanchini, and Reginaldo de Jesus Napolitano. A simple derivation of the lindblad equation. *Revista Brasileira de Ensino de Física*, 35(1):01–09, 2013.

[215] Klaus Mølmer, Yvan Castin, and Jean Dalibard. Monte carlo wave-function method in quantum optics. *JOSA B*, 10(3):524–538, 1993.

[216] Jean Dalibard, Yvan Castin, and Klaus Mølmer. Wave-function approach to dissipative processes in quantum optics. *Phys. Rev. Lett.*, 68:580–583, Feb 1992.

[217] R. Dum, P. Zoller, and H. Ritsch. Monte carlo simulation of the atomic master equation for spontaneous emission. *Phys. Rev. A*, 45:4879–4887, Apr 1992.

[218] Edwin T Jaynes and Frederick W Cummings. Comparison of quantum and semiclassical radiation theories with application to the beam maser. *Proceedings of the IEEE*, 51(1):89–109, 1963.

[219] Kunni Lin, Jiawei Peng, Feng Long Gu, and Zhenggang Lan. Simulation of open quantum dynamics with bootstrap-based long short-term memory recurrent neural network. *The Journal of Physical Chemistry Letters*, 12(41):10225–10234, 2021.