



**Politecnico
di Torino**

Politecnico di Torino

Department of Applied Science and Technology (DISAT)

Master of Science's Degree in Physics of Complex Systems

Master's Degree Thesis

New computational approaches to biomolecular kinetics based on machine-learned reaction coordinates

External supervisor

Fabio Pietrucci

Internal supervisor

Alfredo Braunstein

Candidate

Jacopo Dellai

2025/2026

Le vent se lève!

Il faut tenter de vivre!

- Paul Valéry, Le cimetière marin

Contents

1	Introduction	1
1.1	State of the art	4
1.2	Barnase and Barstar	4
2	Methods	8
2.1	Modeling collective variables	8
2.1.1	The overdamped Langevin equation	8
2.1.2	How to compute the free energy and diffusion profiles	12
2.1.3	How to compute the rate	13
2.1.4	Which projection is the best one?	14
2.2	The datasets	15
2.2.1	Protein pair contacts dataset	16
2.2.2	Protein and water pair contacts dataset	17
2.3	Autoencoder	19
2.4	Driving the network toward a minimal rate	24
3	Results	29
3.1	Design of the AE	29
3.1.1	Activation function	29
3.1.2	Batch size	30
3.2	Is the dynamic really overdamped?	30
3.3	Dataset of only protein atom pairs	33
3.4	Including the water information	35
3.5	CVNet	38
4	Conclusions and achievements	40
4.1	Perspectives	41

1. Introduction

Abstract. Despite the key importance of the process of association between two proteins for biology and drug-design, predicting the kinetic rates from theory and simulation remains still an open problem for which no reliable and affordable methods have been developed. In this thesis we study all-atom Molecular Dynamics simulations of the spontaneous association of Barnase and Barstar solvated in water, and by looking at the relaxation toward the bound state we infer the dissociation kinetic rate of the complex using neural network-based collective variables whose behavior is studied using a combination of stochastic processes and statistical inference.

Protein-protein interactions (PPI) are fundamental to the regulation of cellular life, governing essential pathways in signal transduction and immune responses, and control most biological functions, including pathological processes like cancer, Alzheimer or COVID [1]. Despite their paramount biomedical importance, systematically predicting whether two proteins will associate, calculating their binding free energy (ΔG_{bind}), and accurately estimating their association (k_{on}) and dissociation (k_{off}) rates remains an outstanding challenge in computational biophysics. All-atom Molecular Dynamics (MD) simulations play an important role to address this gap [2]. Unlike static structural biological techniques such as X-ray crystallography or cryo-electron microscopy (cryo-EM) [3, 4] which capture frozen ground states, MD simulations capture the full time-dependent, multi-dimensional interaction pathways at the atomic resolution, modelling the native environment (e.g. a water solution) and allowing for the visualization of metastable states. The application of MD to physical-chemical transformations, such as phase transitions, chemical reactions, and biomolecular conformational changes, faces two prominent challenges. The first one is the *timescale gap* between the simulation timestep, on the order of femtoseconds, and the timescale of the previously mentioned transformations, which would

require simulations on the order of seconds to days. The association or dissociation of a stable protein complex, which are typical examples of rare events, occur on timescales ranging from milliseconds to hours. Bridging this gap via brute-force, conventional MD requires an unfeasible amount of computational resources. The second problem is that the configuration space of a typical solvated biomolecular system consists of thousands to millions of degrees of freedom ($3N$, where N is the number of atoms), and extracting information about the mechanism or calculating macroscopic observables from these high-dimensional trajectories represents a profound challenge [5].

While simulation datasets are massive both in the number of particles simulated and the number of configurations saved along a trajectory, both empirical observations and fundamental physics show that the relevant information about long-lived metastable states do not populate the entire high-dimensional space uniformly [6]. Instead, the essential physics governing the transitions between metastable states typically lives on a much lower-dimensional intrinsic manifold. To exploit this property we introduce the **Collective Variables** (CVs), a set of low dimensional variables obtained by projecting the trajectories living in the full $3N$ -dimensional configuration space onto a lower-dimensional space. An ideal CV, also known as the *Reaction Coordinate* (RC), acts as a mapping that compresses the geometric complexity while preserving the core thermodynamic and kinetic properties of the system. If the dynamics along the CV are Markovian (depending on the nature of the physical system and on the time resolution of the simulation [7]) then the act of projecting the high-dimensional dynamics into a lower dimension space provides a rigorous theoretical framework to model the kinetics and the out-of-equilibrium relaxations via the overdamped Langevin equation.

Historically, selecting these CVs was a kind of art form, relying solely on previous knowledge of the complex, on intuition or on heuristic trial-and-error. When dealing with complex processes like PPI, this trial-and-error approach is cumbersome and frequently fails to capture the true, multi-faceted transition state ensemble. In recent years, Machine Learning (ML) has emerged as a conceptually powerful alternative to heuristic CV design, and since the goal is to uncover a compact representation from unlabeled coordinates, the focus is on unsupervised learning. In particular, the interest is in unsupervised algorithms that operate solely on the input coordinates to extract low-dimensional embeddings, which range from linear methods like Principal Component Analysis (PCA) to non-linear neural networks such as autoencoders (AEs) or methods rooted in the Variational Approach to Conformational Dynamics (VAC) and transfer matrix [8]. Despite their successes, standard dimensionality

reduction techniques suffer from a fundamental limitation: their performance is strictly bounded by the quality of the training dataset. As an example, a CV generated by an autoencoder is optimized purely to reconstruct the dataset; datasets, more generally, can be constructed from radically different types of simulations like locally-equilibrated trajectories, transition path sampling, or trajectories under the effect of an artificial biasing force. Therefore, the resulting CV has a strong dependence on the MD protocol details, without rigorous foundations. Because these CVs are not built with physics in mind, there is no guarantee that they represent an optimal or even physically meaningful projection of the actual dynamical process.

This project aims at transcending standard dimensionality reduction by shifting the optimization paradigm from spatial compression to *dynamical minimization*. Building upon the theoretical foundations established by Zhang et al. [9] and developed into a viable algorithm for complex systems by Mouaffac et al. [2], this work recognizes that a mathematically rigorous reaction coordinate for a rare event must be optimized based on its ability to capture the slow, rate-minimizing modes of the system: the exact reaction coordinate describes the slowest relaxation mode toward equilibrium and is uniquely characterized by its minimization of the kinetic transition rates. Therefore, rather than attempting to accurately reconstruct the high-dimensional atomic coordinates, the goal of this work is to determine the low-dimensional CV (in our case a one-dimensional variable) that directly minimizes the kinetic rate of the process as predicted by a Langevin model. To achieve this, we introduce a novel deep learning framework. Our approach differs fundamentally from previous methodologies in two distinct ways:

1. Elimination of pre-selected features: traditional methods often restrict the network’s input to a hand-picked, heavily curated subset of features. In contrast, we feed our neural network a huge amount of raw structural information, leaving it entirely unconstrained to explore the full high-dimensional configuration space.
2. Physically constrained optimization: the unsupervised machine is mathematically penalized unless it identifies the exact coordinate that minimizes the kinetic rate, forcing it to discover the true physical reaction coordinate directly from the sampling dynamics.

We test these methodologies on a prototypical highly challenging example of PPI: the binding between the bacterial ribonuclease **Barnase** and its intracellular inhibitor **Barstar**. The Barnase-Barstar (BB) complex serves as an ideal benchmark due to its extremely rapid and diffusion-limited association, and for the rigidity of each

monomer during association, allowing us to ignore any potential conformational change.

1.1 State of the art

All-atom MD simulations on the BB complex have been an object of interest for the scientific community for well more than a decade now. Considering the extremely slow process of dissociation of the BB complex (approximately 10^5 seconds, roughly one day), observing the spontaneous dissociation in a simulation remains impossible. For this reason, in order to compute the dissociation rate, the main method used by the scientific community was *enhanced sampling*: instead of letting the system spontaneously dissociate, biases in the simulations are introduced to speed up the process. Examples of enhanced sampling for the BB complex are the “tempered binding” method [10], which scales interfacial interactions to accelerate kinetics, and the stratified Potential of Mean Force (PMF) framework developed by [11], the latter relying on a sequence of geometrical and conformational biasing potentials to computationally force dissociation and map the equilibrium standard binding free energy. The work we present in this thesis differs from these established methodologies in that we avoid enhanced sampling in our MD simulations. Instead of accelerating the unbinding process along a predefined reaction coordinate, we initiate the system out of equilibrium with the two proteins in the dissociated state and monitor its *spontaneous* relaxation towards equilibrium. By analyzing these unconstrained relaxation trajectories, we are able to extract the true dissociation rate (k_{off}) without introducing artifacts or terms inherent to enhanced sampling strategies.

1.2 Barnase and Barstar

Barnase is a small protein (110 amino acids) synthesized by the bacterium *Bacillus amyloliquefaciens* whose purpose is to cut the cell’s RNA, thus being lethal for the host cell. Its inhibitor **Barstar** is another small protein (89 amino acids) and it tightly binds to Barnase and blocks it from cutting the RNA [12, 13]. The two main reasons for why a cell would produce Barnase are to *attack* nearby cells by releasing Barnase and to *regulate* the population: when a cell is damaged it stops producing Barstar so that it dies. The tight binding of the BB complex makes it a good school-case example of a complex that shows fast spontaneous association and rare dissociation. It also has the added benefit that the two proteins do not undergo any conformational changes during association, making the study of the process

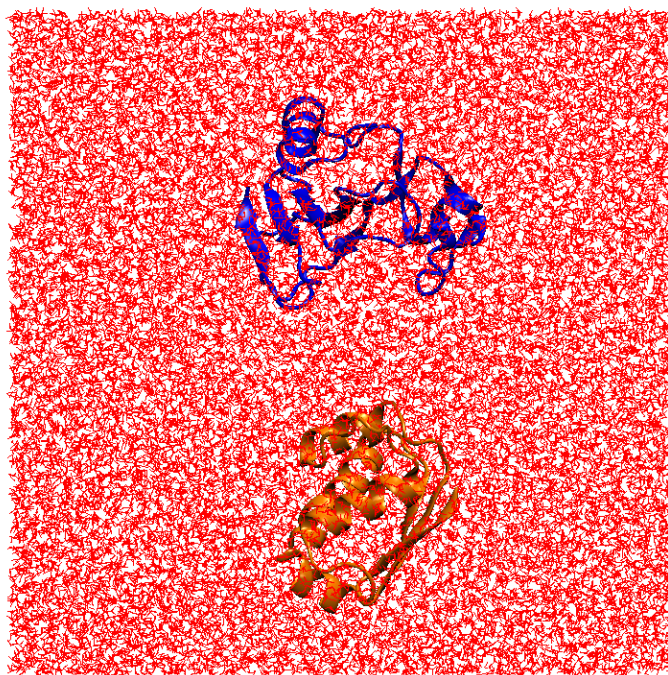


Figure 1.1: First frame of the simulation showing that the two proteins are solvated in a box of 34 795 water molecules (104 385 atoms between hydrogen and oxygen). The box has all sides of equal length 93.339 Å.

less cumbersome. Thermodynamic and kinetic parameters are experimentally well-characterized, with a binding free energy of $\Delta G_b = -19.0 \pm 0.2 \text{ kcal} \cdot \text{mol}^{-1}$ ($-32 \text{ } kT$ at 300 K), and rate constants $k_{\text{on}} = 6.0 \cdot 10^8 \text{ s}^{-1} \cdot \text{M}^{-1}$ (diffusion limited) and $k_{\text{off}} = 8.0 \cdot 10^{-6} \text{ s}^{-1}$ [14, 15].

The MD simulations are performed using, as initial conditions, the information readily available in the Protein Data Bank (PDB: 1BRS). The PDB provides a file containing the x, y, z coordinates of each atom of the complex in the dissociated state (fig. 1.2a). The file is structured in the following way. The first line is:

```
CRYST1  93.339  93.339  93.339  90.00  90.00  90.00  P 1  1
```

CRYST1 defines the cell parameters, which are lengths of 93.339 Å and angles of 90^{circ} (cubic cell). P1 says “no symmetries added”. The second line is:

```
ATOM  1  N  VAL  A  3  51.540  78.260  54.720  1.00  0.00  N
```

ATOM means we're looking at an atom, 1 is the number with which we identify the atom, **N** indicates the atom, in this case nitrogen, and the number of such atom in case of a functional group (for example **H3**, meaning it's the third hydrogen of the functional group). The name of the atom is also repeated in the last column but without the information about of the functional group. **VAL** is the amino acid (or **residue**) the atom belongs to, in this case it's Valine¹. **A** means the atom belongs to protein A, which is Barnase, and **3** is the number of the residue. Then there are the x, y, z coordinates in Å. In conclusion the numbers 1.00 and 0.00 respectively represent the occupancy factor (how much of the atom is present in this position in this frame, 1 represents 100%) and the B-factor (how certain we are that the atom is in this position, the higher its value the most uncertain we are). The PDB files can be used also to visualize the proteins via VMD (Visual Molecular Dynamics, see [16]), using which figs. 1.1 and 1.2 have been obtained.

¹The “free” structure of an amino acid differs from that of an amino acid inside a protein. In the case of Valine, when free it has a NH_2 group, but when in the protein it forms a NH_3 group. For this reason when analyzing the PDB information one may find oddities in the residues compared to the descriptions found in the literature.

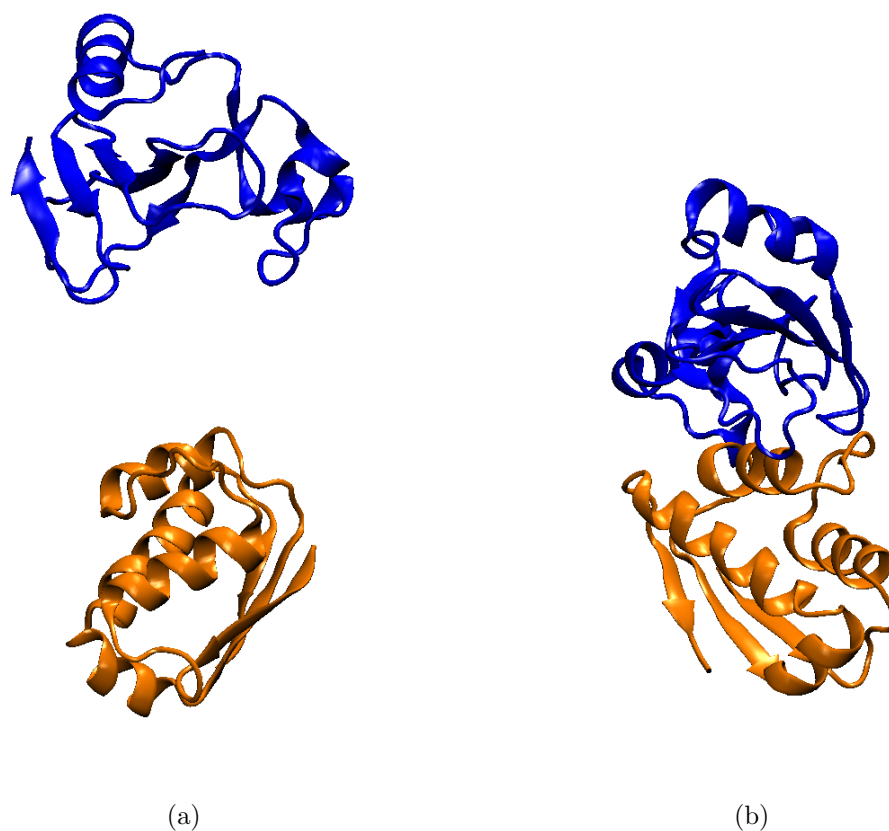


Figure 1.2: First **(a)** and last **(b)** frames of the simulation of BB spontaneous association. Barnase is shown in blue and Barstar in orange. The image is seen as in front of the (x, y) plane. Water molecules have been removed for simplicity. As it is possible to observe, no conformational changes occur during the association.

2. Methods

2.1 Modeling collective variables

In this section we will present the theoretical groundings on the dynamical description of a system using only a subset of all the dynamical variables; such a subset was already called in the introduction the set of collective variables (CV). Then, without entering into full details but providing them when necessary, I will present the theory of kinetic rates, which provides an equation for the dissociation rate. This discussion derives from the reading of the works [17–24].

2.1.1 The overdamped Langevin equation

During the all-atom MD simulations, we collect the data of the position and velocity in three-dimensional space of each atom, and we save just the positions after every interval of time Δt . Then, our goal is to infer a CV knowing the evolution of the system in the position space. In the original fully dimensional phase space, each atom evolves accordingly to Hamilton’s equations of motion, but what can we say about the evolution of the CV found by reducing the dimensionality? To understand the dynamics of a reduced-dimensionality CV, we must track both a chosen slow “relevant” coordinate $x(\mathbf{X})$ (which is the CV) and its conjugate momentum $p(\mathbf{X})$, where $\mathbf{X} = (\mathbf{q}, \mathbf{p})$ represents the fully dimensional phase space. The microscopic system evolves according to the classical Liouville equation

$$\frac{\partial}{\partial t}A(t) = LA(t) \tag{2.1}$$

where $A(t)$ is a generic *dynamical variable* and L is the Liouville operator. Considering that the microscopic system is maintained at thermal equilibrium by a heat bath (as is standard in MD simulations), the full phase space distribution is governed by the canonical ensemble $\rho_{\text{eq}}(\mathbf{X}) \propto e^{-\beta H(\mathbf{X})}$ where $\beta = (kT)^{-1}$. We can define the joint equilibrium probability density of finding the system at a specific value of our

slow variables (x, p) as

$$p_0(x, p) = \int d\mathbf{X} \rho_{\text{eq}}(\mathbf{X}) \delta(x(\mathbf{X}) - x) \delta(p(\mathbf{X}) - p) \quad (2.2)$$

Since the Hamiltonian is separable into a quadratic kinetic term and a configuration-dependent potential, momenta and positions are statistically independent and momenta are exactly Gaussian in the full phase space; assuming the CV's effective mass is approximately constant, its conjugate momentum inherits a position-independent Gaussian distribution, giving the factorization $p_0(x, p) = p_0(x)\psi(p)$, where $p_0(x) \propto \exp(-\beta F(x))$ defines the free energy landscape $F(x)$ (or potential of mean force), and $\psi(p)$ represents the Maxwell-Boltzmann momentum distribution. To isolate the “relevant” dynamics, we introduce Zwanzig's nonlinear projection operator P . Instead of projecting onto a linear subspace, P projects an arbitrary phase space function $G(\mathbf{X})$ onto the space of all functions of our chosen slow variables by utilizing conditional equilibrium averages:

$$\begin{aligned} (PG)(\mathbf{X}) &= \{p_0(x, p)\}^{-1} \int d\mathbf{X}' \rho_{\text{eq}}(\mathbf{X}') \delta(x(\mathbf{X}') - x) \delta(p(\mathbf{X}') - p) G(\mathbf{X}') \\ &= \langle G | x, p \rangle \equiv \langle G \rangle_{x, p} \end{aligned} \quad (2.3)$$

where $x \equiv x(\mathbf{X})$ and $p \equiv p(\mathbf{X})$ denote the instantaneous values of the CV. We also define the complementary orthogonal projection operator $Q = (1 - P)$. Using the standard Dyson operational identity, the time evolution operator e^{tL} can be exactly decomposed as

$$e^{tL} = e^{tL}P + \int_0^t ds e^{(t-s)L} P L e^{sQL} Q + e^{tQL} Q \quad (2.4)$$

We now apply this identity to evaluate the equations of motion for our vector of slow variables $\mathbf{a} = (x, p)^T$, noting that $\frac{d}{dt}\mathbf{a}(t) = e^{tL}L\mathbf{a}$:

$$\frac{d}{dt}\mathbf{a}(t) = e^{tL}P L \mathbf{a}(t) + \int_0^t ds e^{(t-s)L} P L e^{sQL} Q L \mathbf{a}(t) + e^{tQL} Q L \mathbf{a}(t) \quad (2.5)$$

Let's analyze the three resulting terms on the right-hand side of eq. (2.5) piece by piece:

- The Deterministic Drift Term ($e^{tL}P L \mathbf{a}$). For the coordinate x , the action of the operator yields $P L x = \langle \dot{x} \rangle_{x, p} = \frac{p}{m}$, where m is the effective mass of the CV. For the momentum, $P L p = \langle \dot{p} \rangle_{x, p} = \langle -\frac{\partial H}{\partial x} \rangle_{x, p}$. Taking this conditional average over the canonical ensemble explicitly reveals the free-energy gradient:

$$P L p = \frac{1}{\beta} \frac{1}{p_0(x)} \frac{\partial p_0(x)}{\partial x} = -\frac{\partial F(x)}{\partial x} \quad (2.6)$$

Thus, the systematic drift provides the exact non-linear thermodynamic force landscape:

$$e^{tL}PL \begin{pmatrix} x \\ p \end{pmatrix}^T = \begin{pmatrix} \frac{p(t)}{m} \\ -\frac{\partial F(x(t))}{\partial x} \end{pmatrix}^T \quad (2.7)$$

- The Noise Term ($e^{tQL}QL\mathbf{a}$). For the coordinate, since $Lx = \frac{p}{m}$, then $QLx = (1 - P)Lx = \frac{p}{m} - \frac{p}{m} = 0$, meaning no direct random force enters the position evolution. For the momentum equation, we define the orthogonal, rapidly fluctuating random force as

$$\xi(t) = e^{tQL}QLp \quad (2.8)$$

By construction, because $P\xi(t) = 0$, this noise remains completely uncorrelated with our slow variables at $t = 0$.

- The Memory Kernel Term. Since $QLx = 0$, the memory term only impacts the momentum equation. The integrand involves evaluating $PL\xi(s)$; Zwanzig showed this term simplifies to a linear function of velocity

$$PL\xi(s) = -\frac{p}{m} (\beta \langle \xi(0)\xi(s) \rangle_x) = -\frac{p}{m} \gamma(s) \quad (2.9)$$

where $\gamma(s)$ is the memory kernel, directly tied to the noise autocorrelation via the Fluctuation-Dissipation Theorem (FDT)

$$\langle \xi(t)\xi(t') \rangle = kT\gamma(t - t') \quad (2.10)$$

Propagating this term forward via $e^{(t-s)L}$ yields $-\frac{p(t-s)}{m}\gamma(s)$. The reduction of the memory term to a kernel depending only on the time lag and linear in the past momentum is a simplifying assumption rather than an exact consequence of the Zwanzig conditional projector, which in general yields a friction term that is nonlinear in the past coordinate and momentum; see [21, 22].

Combining these components back into eq. (2.5) and substituting $p(t) = m\dot{x}(t)$, we arrive at the generalized Langevin equation (GLE) for a nonlinear collective coordinate moving across an arbitrary free energy landscape

$$m \frac{d^2}{dt^2} x(t) = -\frac{\partial F(x)}{\partial x} - \int_0^t \gamma(t - t') \frac{dx(t')}{dt'} dt' + \xi(t) \quad (2.11)$$

The GLE is non-Markovian due to the memory effects carried by the kernel. However, it's easy to see that these memory effects disappear if we consider the “relevant” variables to evolve with a timescale that is much greater than that of the “irrelevant”

variables. In many applications in physics, chemistry and biology it's typical to have a time resolution Δt that is greater than the memory time scale [7], and so we can accurately describe the projected dynamics with a Markovian Langevin equation. Under the assumption that the process evolves with a characteristic time that is much longer than that of the memory, the memory function $\gamma(t - t')$ can be approximated by a delta function having the same area

$$\gamma(t - t') \cong 2\gamma_0\delta(t - t') \quad (2.12)$$

where γ_0 is the friction opposing to the process. By plugging this into the integral we obtain

$$m \frac{d^2}{dt^2}x(t) \cong -\frac{\partial F(x)}{\partial x} - \gamma_0 \frac{dx(t)}{dt} + \xi(t) \quad (2.13)$$

As a last step, let's assume that the dynamics can be treated as *overdamped*. Physically, this implies a separation of timescales where the characteristic relaxation time $\tau = m/\gamma_0$ is much smaller than the time resolution Δt at which the system is analyzed ($\Delta t \gg m/\gamma_0$). On this coarse-grained timescale, the velocity of the CV decorrelates from itself almost instantly between successive observations. Consequently, the inertial term becomes negligible, and the GLE simplifies to the overdamped Langevin equation (OLE)

$$\frac{d}{dt}x(t) = -\frac{1}{\gamma_0} \frac{\partial F(x)}{\partial x} + \sqrt{\frac{2kT}{\gamma_0}}\eta(t) \quad (2.14)$$

where $\eta(t)$ is normalized Gaussian white noise, i.e. with $\langle \eta(t) \rangle = 0$ and $\langle \eta(t)\eta(t') \rangle = \delta(t - t')$. This result can be extended to a variable-dependent friction, whose inverse $D = \frac{kT}{\gamma_0}$ (Einstein relation) gives the *diffusion*, and the OLE finally reads

$$\frac{d}{dt}x(t) = -\frac{D(x)}{k_B T} \frac{\partial F(x)}{\partial x} + \frac{\partial D(x)}{\partial x} + \sqrt{2D(x)}\eta(t) \quad (2.15)$$

Although this equation is mainly used for its simplicity, it can be shown that both the Markovian and overdamped approximations are reasonable simplifications for the simulated system. These approximations will be tested on our dataset and the results will be presented in section 3.2.

As a last step we introduce the Smoluchowski equation, which is the Fokker-Plank equation that arises from an overdamped process

$$\frac{\partial}{\partial t}P(x, t) = \frac{\partial}{\partial x} \left[D(x)e^{-\beta F(x)} \frac{\partial}{\partial x} \left(e^{\beta F(x)} P(x, t) \right) \right] \quad (2.16)$$

where $P(x, t)$ is the distribution of the variable x that evolves by eq. (2.15).

2.1.2 How to compute the free energy and diffusion profiles

In a typical situation one would possess the knowledge about the diffusion properties and the free energy of the system that is being analyzed, and from there they would be able to study the dynamics of said system. In this case we find ourselves in the opposite situation, since we know the evolution of the system and want to infer the free energy and diffusion. In this section we will present the two methods that have been used in this work, which are the *Kramers-Moyal* (KM) *expansion* and the *optimized Langevin equation* (optLE).

The KM expansion gives a series representation of the time evolution of the distribution $P(x, t)$ in terms of the conditional momenta of the small increments $\Delta x = \Delta x(t + \Delta t) - \Delta x(t)$ [25]. Upon defining:

$$M^{(n)}(x) \equiv \frac{1}{n!} \frac{\langle (\Delta x)^n | x(t) = x \rangle}{\Delta t}, \quad n = 1, \dots \quad (2.17)$$

then the KM expansion reads

$$\frac{\partial}{\partial t} P(x, t) = \sum_{n=1}^{\infty} (-\partial_x)^n (M^{(n)}(x) P(x, t)) \quad (2.18)$$

For all Markovian processes, like the OLE, all terms for $n \geq 3$ vanish [18], so that only $M^{(1)}(x)$ (drift) and $M^{(2)}(x)$ (diffusion) are left. By simply evaluating the mean and variance of Δx we can fully characterize the free energy $F(x)$ and diffusion $D(x)$ of the process:

$$\begin{aligned} D(x) &\approx \frac{\langle \Delta x^2 \rangle - \langle \Delta x \rangle^2}{2\Delta t} \equiv \frac{\sigma_x^2}{2\Delta t} \\ f(x) &= \begin{cases} \frac{\langle \Delta x \rangle}{\beta D(x) \Delta t} - \frac{D'(x)}{\beta D(x)} \\ - \frac{dF(x)}{dx} \end{cases} \end{aligned} \quad (2.19)$$

where $f(x)$ is the force acting on the system.

The optLE is an iterative method that computes the free energy and diffusion by maximizing the model likelihood based on an explicit expression of the short-time propagator [7]. Given a trajectory x sampled at a time resolution Δt , the log-likelihood of observing the data under a model specified by the free-energy $F(x)$ and the variable-dependent diffusion $D(x)$ is

$$\ln \mathcal{L}(\theta) = \sum_{i=1}^{N-1} \ln p(x_{i+1}, \Delta t | x_i; \theta) \quad (2.20)$$

where the free energy and diffusion are treated as the parameters θ to be optimized and p is the short-time transition probability density, defined as

$$p(x', t + \Delta t | x, t) \approx \frac{1}{\sqrt{2\pi\mu}} \exp\left(-\frac{(x' - x - \phi)^2}{2\mu}\right) \quad (2.21)$$

with:

$$\phi = a\Delta t + \frac{1}{2}(aa' + Da'')\Delta t^2, \quad \mu = 2D\Delta t + (aD' + 2a'D + DD'')\Delta t^2 \quad (2.22)$$

where a is the drift term in eq. (2.15) and $D \equiv D(x)$. The optLE method differs from KM in that it uses a propagator of order Δt^2 , whereas the expansion of KM stops at order Δt .

2.1.3 How to compute the rate

Once $F(x)$ and $D(x)$ have been reconstructed from the trajectory, via KM expansion or optLE, the physically relevant observable is the *rate* at which the system escapes from one metastable region to another. In our case this is the rate of going from the associated state to the dissociated one (k_{off}) or viceversa (k_{on}). The general theory of kinetic rates for a stochastic process, and in particular for an overdamped (Smoluchowski) particle, can be found in [17]. I reproduce here the derivation, adapted to the notation introduced above.

Consider the CV x evolving on the free energy landscape $F(x)$ according to the OLE eq. (2.15), and suppose we want the typical time needed to go from a point q_0 , sitting in a metastable (reactant) region, to a point b that marks the product state. It is convenient to bound the reactant region by a reflecting boundary at $x = a < q_0$ (physically, this represents the fact that the system cannot escape backward past the bottom of the reactant well) and to place an absorbing boundary at $x = b$ (once the system reaches b the “reaction” is considered complete and we stop measuring the time). The quantity we look for is the *mean first passage time* (MFPT) $\tau(q_0)$ for a particle that starts at $x = q_0$ and reaches b for the first time. The MFPT is most naturally obtained from the backward (adjoint) form of the Smoluchowski equation, rather than from the forward equation used earlier (eq. (2.16)). Starting from the forward operator

$$\mathcal{L}_{\text{FP}}P(x, t) = \frac{\partial}{\partial x} \left[D(x)e^{-\beta F(x)} \frac{\partial}{\partial x} \left(e^{\beta F(x)} P(x, t) \right) \right] \quad (2.23)$$

one can construct its formal adjoint $\mathcal{L}^\dagger$ defined by $\int dx f(x) \mathcal{L}_{\text{FP}}P = \int dx (\mathcal{L}^\dagger f(x))P$ for arbitrary test functions $f(x)$. Two integrations by parts (boundary terms van-

ishing at a and b) give

$$\mathcal{L}^\dagger f(x) = e^{\beta F(x)} \frac{d}{dx} \left[D(x) e^{-\beta F(x)} \frac{df}{dx} \right] \quad (2.24)$$

which is the operator generating the time evolution of any observable when it's the initial condition that is varied rather than the final one. The MFPT $\tau(x)$, viewed as a function of the starting point x , satisfies the inhomogeneous equation

$$\mathcal{L}^\dagger \tau(x) = -1 \quad (2.25)$$

together with the boundary conditions appropriate to our setup:

$$\begin{aligned} \tau(b) &= 0 & (\text{absorbing}) \\ \tau'(a) &= 0 & (\text{reflecting}) \end{aligned}$$

Using the explicit form of $\mathcal{L}^\dagger$ from eq. (2.24), eq. (2.25) becomes

$$\frac{d}{dx} \left[D(x) e^{-\beta F(x)} \frac{d\tau}{dx} \right] = -e^{-\beta F(x)} \quad (2.26)$$

This is now a linear, first-order ODE for the auxiliary function $D(x)e^{-\beta F(x)}\tau'(x)$, and can be integrated twice in closed form, yielding the result

$$\tau(q_0) = \int_{q_0}^b dx \frac{e^{\beta F(x)}}{D(x)} \int_a^x dy e^{-\beta F(y)} \quad (2.27)$$

The corresponding rate constant for the transition is simply its inverse

$$k = \tau^{-1}(q_0) \quad (2.28)$$

evaluated with q_0 taken near the bottom of the reactant well. The result depends only weakly on the precise choice of q_0 (given it's deep inside the well), since the outer integral is dominated by the barrier region where $F(x)$ is largest.

2.1.4 Which projection is the best one?

The last important point to analyze is the following: how good is the projected variable at capturing the behavior of the fully dimensional system? In fact, depending on how the variable is constructed, we see different aspects of the original system, while ignoring others. As an example, let us consider a simple two-dimensional double well alongside the x -direction. If we choose the reduction $(x, y) \mapsto x$, then we are able to still perfectly see the shape of the double well; contrarily, if we chose

the reduction $(x, y) \mapsto y$, we will see a single well, losing all the information on the double well shape. We then need a criterion to evaluate how good is a variable at describing the original system. In particular, we’re interested in a projection that preserves the kinetic properties of the system. Zhang et al. [9] provides the following crucial result: the transition rate of the full dynamics is always smaller than or equal to the one computed using the effective dynamics, with the optimal reaction coordinate yielding a minimal rate [2]. Among all the possible projections, the optimal one is the one that minimizes the kinetic rate; the best projection, the one achieving the true kinetic rate, is called the *committor*. Then, when evaluating a CV, we will look at the kinetic rate it produces, and we will deem optimal the CV that, among the others, will produce the minimal rate.

2.2 The datasets

Given a trajectory dataset, before any machine learning algorithm can be deployed to obtain significant results, one must first choose a specific numerical representation X for our trajectories. This amounts to choosing a set of “features” that adequately describes the system of interest. Let $\mathbf{X}_r \in \mathbb{R}^{N \times D_r}$ be the matrix that represents the “raw” trajectory dataset where N is the number of simulation time-points collected times the number of trajectories and D_r is the number of degrees of freedom of the dataset, and $\mathbf{X} \in \mathbb{R}^{N \times D}$ the matrix representing the “featurized” dataset; featurization is performed by the function $\chi : \mathbb{R}^{D_r} \mapsto \mathbb{R}^D$ that transforms a “raw” datapoint into a featurized one:

$$\mathbf{x}_i = \chi(\mathbf{x}_{i,r}) \quad (2.29)$$

When sufficient prior knowledge of the system is available, it’s convenient to choose a feature space such that $D \ll D_r$ that can appropriately characterize the molecular motion of interest without significant loss of information. Often, the number of degrees of freedom required to encode many physical properties or observables of a molecular system is relatively low [26, 27]. In this work we will present results obtained with two different featurized datasets, each with the same amount of datapoints but different features. Each datapoint represents a “snapshot” of the atoms coordinates in three-dimensional space taken during the MD simulation with a time resolution of $\Delta t = 200$ ps. 250 independent MD trajectories are generated and 175 trajectories result in successful associations (the MD simulations were performed by Mouaffac et al., see [1] for greater details). These trajectories all last a different amount of time in the range $[10^1, 10^2]$ ns, and they are concatenated to create a single dataset with $N = 143\,050$ datapoints. With all these information, we have created and used two datasets.

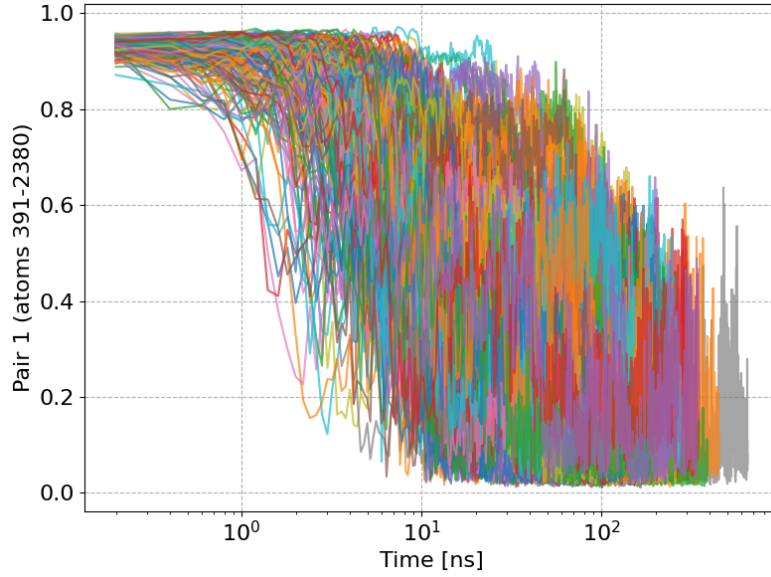


Figure 2.1: Association trajectories projected on the pair 1 CV (atoms 391-2380).

2.2.1 Protein pair contacts dataset

Since we are tracking the process of association of the two proteins, we have deemed useful to consider in the first dataset the distance between key interface atoms during association. The function χ maps the coordinates of atom $i \in A$ (set of interface atoms of barnase) and atom $j \in B$ (set of interface atoms of barstar) into a scalar quantity called *coordination number*:

$$c_{ij} = \chi(r_{ij}) = \frac{1 - \left(\frac{r_{ij}-d_0}{r_0}\right)^n}{1 - \left(\frac{r_{ij}-d_0}{r_0}\right)^m} \quad (2.30)$$

$$r_{ij} = |\mathbf{x}_{i,r} - \mathbf{x}_{j,r}| = \sqrt{(x_{i,r} - x_{j,r})^2 + (y_{i,r} - y_{j,r})^2 + (z_{i,r} - z_{j,r})^2}$$

with $d_0 = 0$, $r_0 = 0.6$ nm, $n = 6$ and $m = 8$ (see [1]). The coordination number has the advantage of easily explaining the association process: when $c_{ij} \sim 1$ then the two atoms are very close, hence association for that pair has happened, and when $c_{ij} \sim 0$ the two atoms are still distant from each other. Although indifferent for the final results, the pair coordination numbers are shifted ($x' = 1 - x$) in order to have trajectories that decrease in time (fig. 2.1). The coordination number is computed for $D = 487$ pairs using the open-source, community-developed PLUMED library [28, 29]. A PLUMED input file is defined in the following way. The file starts with:

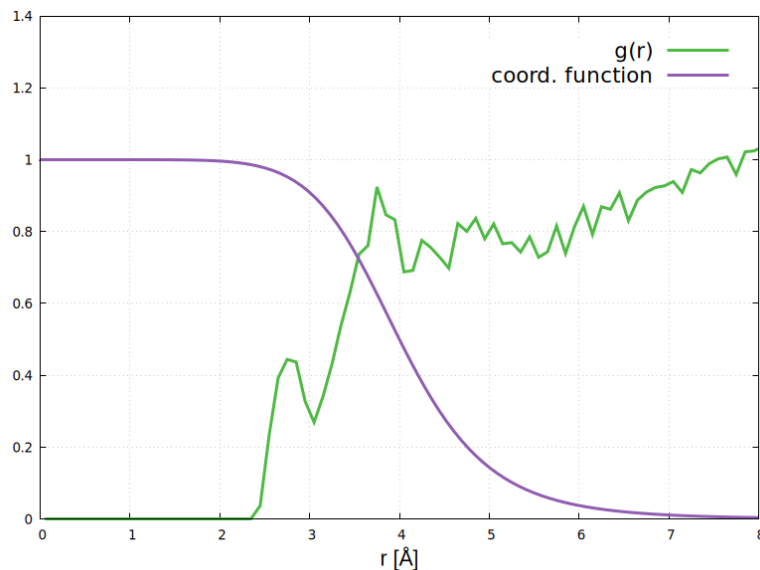


Figure 2.2: Radial distribution function $g(r)$ of the water molecules around BB (green curve). Superimposed it's the coordination function eq. (2.30) of parameters $r_0 = 0.4$, $n = 8$, $m = 16$, showing that with this choice of parameters we fully capture the nearest neighboring water molecules (first spike) and starts to decrease for the second set of neighbors.

```
WHOLEMOLECULES ENTITY0=1-1700 ENTITY1=1701-3134
ox: GROUP ATOMS=3135-107511:4
```

The first line tells PLUMED which atoms belong to barnase (ENTITY0) and which to barstar (ENTITY1), where the numbers of the atoms are the same of the PDB file, and second line tells PLUMED about the waters. Then you need to provide the instruction on how to compute the coordination number:

```
pair1: COORDINATION GROUPA=391 GROUPB=2380 R_0=0.6 NN=6 MM=8 PAIR
```

PLUMED here considers atom 391 of barnase and atom 2380 of barstar and computes their coordination using the specified parameters.

2.2.2 Protein and water pair contacts dataset

In the second dataset there has been included also the information about the amount of water molecules surrounding each interface atom of both barnase (100 interface atoms) and barstar (99 interface atoms), amounting the number of features to $D = 686$. To count the number of waters, we have used eq. (2.30) where $i \in A$ are now

the interface atoms of barnase or barstar and $j \in B$ are all the oxygens in the water molecules, and then sums all the contributions:

$$c_i = \sum_{j \in B} c_{ij} \quad (2.31)$$

The values of c_i can go beyond 1, and this would create unbalance in the dataset. In order to normalize these values in the interval $[0, 1]$ while still retaining useful information, each datapoint within the same feature is transformed as:

$$x'_i = \frac{x_i - x_{\min}^{\text{all traj}}}{x_{\max}^{\text{all traj}} - x_{\min}^{\text{all traj}}} \quad (2.32)$$

By doing so, not only these features also live in the $[0, 1]$ range as the pair features, but they also decrease along the trajectories, meaning that they retain the same information: values close to 1 mean that the proteins have not associated, values close to 0 indicate association. The parameters of eq. (2.30) are *heuristically* chosen to be $r_0 = 0.4$ nm, $n = 8$ and $m = 16$ by looking at the *radial distribution function* $g(r)$. The radial distribution function (RDF) describes how density varies as a function of the distance from a reference atom. It's constructed by computing all pairwise interatomic distances and organizing them into a discrete histogram, which is then normalized against the baseline of an uncorrelated ideal gas. In three dimensions, this normalization is the density of the system $\rho = N/V$ multiplied by the volume of the spherical shell $4\pi r^2 dr$. Then:

$$g(r) = \frac{\rho(r)}{\rho_0(r)} = \rho(r) \cdot \frac{V}{N4\pi r^2} \quad (2.33)$$

where $\rho(r)dr$ is the probability of finding one atom in a shell between r and $r + dr$. The RDF of the water molecules around BB, shown in fig. 2.2, was computed in the following way. First of all, every atom of the two proteins (except the hydrogens) was used as a center for the measurements. Then, the distance is computed looking at the oxygens of the water molecules. Periodic boundary conditions were enforced and to compute the histogram a resolution of $\delta r = 0.1$ Å was used. From fig. 2.2 we see that the first shell of water molecules sit around $2.5 \div 3$ Å, and the second at 4 Å. Then, if we want to capture just the water molecules that are closer to the proteins, the parameters above indicated have to be chosen.

Dataset split

For the training of the autoencoder the dataset is split into 80% training set and 20% validation set. There is no test set since, at least for the autoencoder, we're

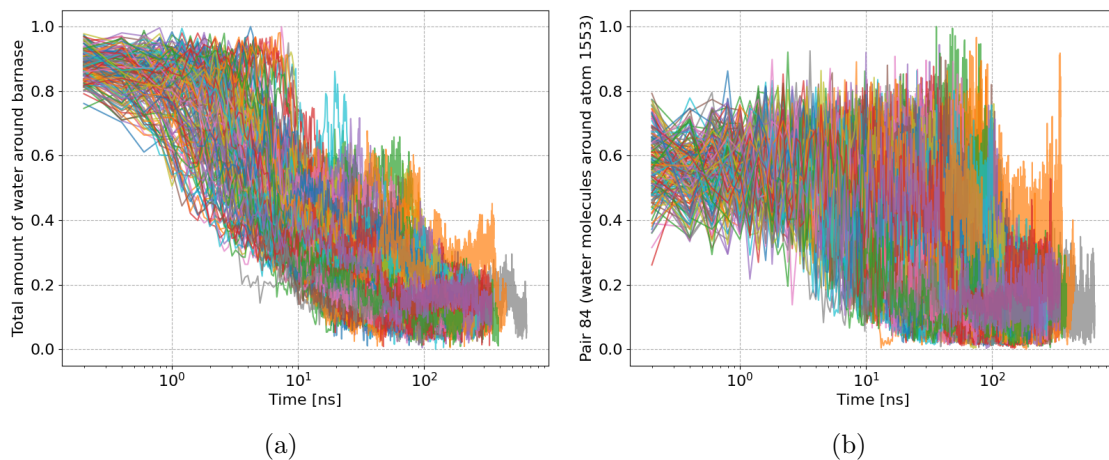


Figure 2.3: **(a)** Evolution of the total number of water molecules around barnase. **(b)** Evolution of water CV 84, representing the amount of water molecules around atom 1553 of barnase.

interested in the task of dimensionality reduction, and not in making predictions or generalizing to new datasets (meaning, new complexes). In a sense, the test will be performed when evaluating if the CV found by dimensionality reduction is able to capture the kinetics of the system. The training set is used to train the model. Since we will use stochastic gradient descent to update the weights of the network, the training loss is updated after each batch is passed. The validation set is the portion of the dataset used to compute the validation loss which informs us whether the model needs tuning or adjustments: if the training and validation losses are similar, then the model is good, whereas a difference in their evolution during training is a symptom of overfitting or underfitting. The validation loss is computed at the end of each epoch once the weights are already updated.

2.3 Autoencoder

As a first task, we want to find a one-dimensional CV by performing pure dimensionality reduction on the dataset using a kind of neural network called *autoencoder* (AE). AEs are a class of neural networks trained on the task of *attempting to copy* the input into the output: given a dataset $X \in \mathbb{R}^{N \times D}$, the AE output is $\hat{X} \in \mathbb{R}^{N \times D}$ which should be as identical as possible to X . The AE consists of two main components, the **encoder** and the **decoder**. The encoder maps the input into *latent-space variables* $q = f_{\theta}(X)$ and the decoder tries to reconstruct the input starting with these latent space variables $\hat{X} = g_{\theta}(q)$. If the AE learns to perfectly copy the input set $g(f(X)) = \hat{X} = X$, then it's not of particular use. Instead, AEs are designed

in such a way that they're unable to achieve perfect copying, getting restricted to copying just approximately. This approximate learning is fundamental for the AE to learn just the important features of the dataset. For this reason, traditionally AEs have been used as *dimensionality reduction* tools: since the model needs to prioritize copying only important aspects, in the process of learning it will produce a lower-dimensional version of the input X which retains all the important information. An AE which produces $q \in \mathbb{R}^d$ through $f : \mathbb{R}^D \mapsto \mathbb{R}^d$, with $d \ll D$, is called an *undercomplete* AE [30]. The learning process is achieved using a loss function which minimizes the difference between input and output. The most common choice of such a loss function is the *Mean Square Error* (MSE)

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^M |\mathbf{x}_i - \hat{\mathbf{x}}_i|^2 \quad (2.34)$$

where $|\cdot|$ indicates the norm-2 of a vector¹ and M is the number of datapoints in the training set. It's easy to show (see [31]) that $\mathcal{L}$ is minimized when $\mathbf{x}_i = \hat{\mathbf{x}}_i \forall i = 1, \dots, M$. When the decoder function g is linear and the loss is MSE, then the AE learns to span the same subspace of PCA. Much more useful however is when both f and g are nonlinear functions, and thus the AE is capable of learning a more powerful *nonlinear* generalization of PCA.

Without entering into great details (for which I recommend the reading of [32]), neural networks (NNs) are machine learning models consisting of connected units called *neurons* that are stacked into *layers*. Each neuron i takes as input a vector of features $\mathbf{x} = (x_1, \dots, x_d)$ and outputs a scalar quantity $a_i(\mathbf{x})$. The functional form of a_i depends on the nonlinearity one wants to encode in their NN, however almost always it takes the form of a *weighted* linear combination of the input features and a nonlinear transformation $\sigma_i(z)$ called the *activation function*. The linear transformation almost always has the form of a dot product between the input vector $\mathbf{x}$ and a vector of weights $\mathbf{w}^{(i)}$ followed by a bias $b^{(i)}$:

$$z^{(i)} = \mathbf{w}^{(i)} \cdot \mathbf{x} + b^{(i)} \quad (2.35)$$

Then, $z^{(i)}$ serves as the input of the nonlinear function $\sigma_i(z)$. This operation is repeated for each neuron in a layer, and the output of a layer serves as the input of the following one. When the information travels only in one direction, meaning there are no connections within a layer nor loops inside the network, the NN is called *feedforward*. The first layer is called the input layer and the last layer is the output

¹ $|\mathbf{x}| = \sqrt{\sum_{i=1}^d x_i^2}$ for $\mathbf{x} \in \mathbb{R}^d$

layer, whereas all the layers in the middle are called hidden layers. The architecture of the NN, meaning the number of neurons in each layer and the number of layers, and the choice of the activation function are dependent on the problem that one needs to tackle. The last ingredient in the construction of a NN is the definition of a *loss function* which, upon minimization, serves to update the weights and biases. The algorithm used to compute the gradients of the loss function using the layered structure of the NN is called *Backpropagation* [33].

Architecture of the AE

The AE is coded using the `mlcolvar` library [34], with information and code publicly available at <https://mlcolvar.readthedocs.io/>. The design of the AE was chosen based on the work proposed in [35]. The AE is chosen symmetrical, meaning that encoder and decoder are identical and mirrored, with 1 hidden layer composed of 100 neurons: Input (D) $\rightarrow$ Hidden 1 (100) $\rightarrow$ Bottleneck (1) $\rightarrow$ Hidden 2 (100) $\rightarrow$ Output (D). Since the output of the bottleneck will be used as the CV, the dimension of the bottleneck will vary with the dimension of the desired CV. A graphical representation of the architecture of the AE is shown in fig. 2.4. To compare different architectures and choose the best, we plot the training and validation loss functions throughout the training procedure and judge basing on criteria such as the stability during training, meaning how smooth the two curves decrease over the epochs, how many epochs does the model take to learn, or if the model overfits. Overfitting is handled by the `early_stopping` procedure, with `min_delta`= 10^{-5} and `patience`= 40. The weights are initialized using He uniform initialization: both weights and biases are sampled from the uniform distribution $U[-\sqrt{k}, \sqrt{k}]$ with $k = (n_l/6)^{-1}$, with n_l the number of neurons in the input layer. Before passing the input data to the network, `mlcolvar` automatically performs a `mean_std` normalization, meaning that for each feature it performs the normalization

$$x' = \frac{x - \mu_i}{\sigma_i} \quad (2.36)$$

Training protocol

The dataset is first split into training and validation sets in the ratio 80/20. The network is then trained by minimizing the MSE and to which we also added L_2 regularization, also known as *ridge regression*. Ridge regression is a technique used to encourage smaller, more evenly distributed weights by adding a penalty on the

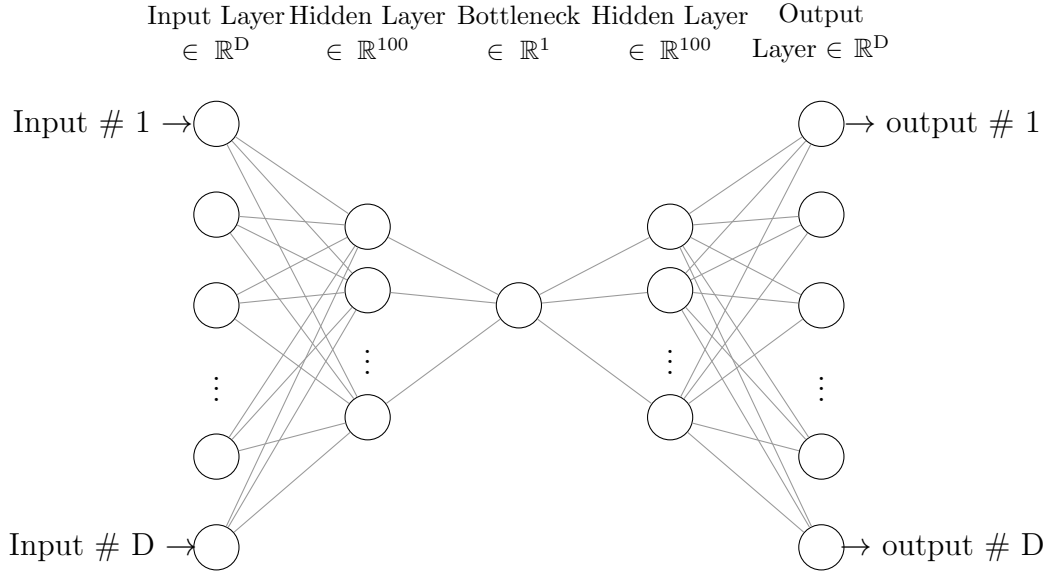


Figure 2.4: Graphical representation of the architecture of the AE.

amplitude of the coefficients. The loss of the AE then reads

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^M |\mathbf{x}_i - \hat{\mathbf{x}}_i|^2 + \gamma |\boldsymbol{\theta}|^2 \quad (2.37)$$

where $\boldsymbol{\theta}$ represents the parameters of the encoder and decoder functions, respectively $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$. The reasons for which we decided to include ridge regression in this problem are:

- suppose the AE learns to produce a spike in response to small variations. Then, by adding L_2 regularization the weights stay small, ensuring that the mapping from input to output is more continuous and stable.
- handling correlated values. Without L_2 , the optimizer might assign an arbitrarily large positive weight to one feature and a large negative weight to a correlated one. L_2 penalizes large weights, forcing the model to distribute the “importance” more evenly and logically across the features.
- numerical stability. L_2 adds curvature in the optimization space, ensuring that the optimizer finds a minimum rather than oscillating.

The minimization of the loss function and update of the weights is performed using **Adam** [36], a moment-based gradient-based stochastic optimization algorithm. Once the AE is trained, the weights of the model are normalized using **min_max** normalization which works in the following way: you evaluate the average, standard deviation

activation function	tanh
Test set	80%
Validation set	20%
Learning rate	10^{-4}
L_2 strength (γ)	10^{-5}
β_1	0.9
β_2	0.99
ϵ	10^{-8}
batch size	1024
Δt [ps]	400

Table 2.1: Hyperparameters of the AE.

and minimum and maximum values of the bottleneck variable $\mathbf{s}=\text{model}(\mathbf{X})$, and then the weights are normalized in such a way that s has zero mean and values inside the range $[-1, 1]$. Finally, since in the literature the CV is often scaled to have values in $[0, 1]$, we perform the map $q \leftarrow (s + 1)/2$. In conclusion, q alongside the respective time is saved concatenated in a file ready to be used as input for the `optLE`, publicly available at <https://github.com/physix-repo/optLE>, in order to compute the free energy landscape, the diffusion and finally the kinetic rate.

All the hyperparameters of the AE are listed in table 2.1; the choices of these hyperparameters are discussed in section 3.1. While working with the AE, these two problems were encountered:

1. Phase independence of the one-dimensional CV. An annoyance encountered during training is the following: reducing to a single variable is equivalent to projecting the datapoints into a line whose direction is not specified, meaning that learning $z = f(x)$ or $z' = -f(x)$ gives the same result $\hat{x}$. This is however not a real problem, since we can simply check for the sign of the learned variable after the training is complete and eventually flip its sign.
2. Finite precision of floating-point arithmetic. Due to the finite precision of floating-point arithmetic, the CV occasionally exhibits minor numerical drift beyond the physical boundaries ($x < 0$ or $x > 1$). The solution is simply to implement an `if` condition before saving the variable and restore these values into the boundary.

2.4 Driving the network toward a minimal rate

Once we have validated that by pure dimensionality reduction we are able to get a good CV to describe the mechanism of the process (sections 3.3 and 3.4), we want to proceed with replacing the pure dimensionality reduction with a learning procedure constrained on minimizing the kinetic rate. There is no metric or physics inside an autoencoder that pushes the learning process toward finding an “objectively” good variable, independently good from the quality of the dataset. For this reason, we deemed important to build a network that is able to stir the learning of the variable towards a physically-informed one. This thought process is based on the kinetic variational principle, introduced in [2]: the best CV is the one that minimizes the rate of the process as predicted by a Langevin model of the same variable. In the paper, the authors perform a weighted linear combination of proposed collective variables, and the weights are updated by random variations accepted using a Monte Carlo procedure. Our core idea is then not only to replace the linear combination of CVs with a nonlinear one, but also to substitute the Monte Carlo updates with a gradient-based learning process which is done by a neural network trained on minimizing the kinetic rate introduced in eqs. (2.27) and (2.28). The free energy and diffusion are computed using the first two moments of the variable according to the Kramers-Moyal expansion eq. (2.19). In the particular case of the BB complex, the diffusion is found to be almost constant across the CV. For this reason the diffusion is averaged and thus considered a constant quantity

$$D = \langle D(q) \rangle_q \quad (2.38)$$

where the average is performed on the distribution of the latent space variable q . To compute the first two moments of the variable q and the average of $D(q)$, one should compute the probability distribution $p(q)$ which is done by hard binning the trajectories to create a histogram. However, since learning is based on backpropagation, we need every single step of the computations to be differentiable, and hard binning doesn’t satisfy this requisite. For this reason we *soft-bin* using Gaussian kernel-density estimation. In the following we will present the choices made for the construction of the network, discussing which problems were encountered and how they were solved.

Trajectory-level training and validation split

Exactly as done for the AE, the dataset is split into 80% training set and 20% validation. However, the split can no longer be performed frame-by-frame: both $F(q)$ and D are conditional averages over *consecutive* frames, and the rate is only

meaningful if evaluated on trajectories that retain their original chronological order. If we were to shuffle the individual frames across the two sets we wouldn't be able to compute the increments Δq . We therefore split at the level of whole trajectories: out of the 175 available association events, 35 are randomly drawn without replacement with a fixed seed and set aside entirely for validation, while the remaining 140 are used for training. The roles of training/validation sets remain the same as for the AE.

Differentiable estimation of $F(q)$ and D : soft Kramers-Moyal moments

Before computing the free energy and diffusion, we need to create a grid on which to build the histograms. At every forward pass, before any free energy or diffusion is estimated, we rebuild the histogram grid on the fly from the instantaneous output of the network. Let $q_{\min}$ and $q_{\max}$ be the minimum and maximum of the current batch of latent values $q = f_{\theta}(X)$. We define a relative kernel bandwidth

$$\sigma_{\text{eff}} = \sigma_{\text{rel}} (q_{\max} - q_{\min}) \quad (2.39)$$

where σ_{rel} is a fixed hyperparameter, so that the effective smoothing width always occupies the same *fraction* of the current range of q , irrespective of how that range shrinks or grows during training. A buffer of three bandwidths is added on each side to prevent the kernel density estimate from being truncated at the edges,

$$q_{\text{grid}}^{\min} = q_{\min} - 3\sigma_{\text{eff}}, \quad q_{\text{grid}}^{\max} = q_{\max} + 3\sigma_{\text{eff}}, \quad (2.40)$$

and a uniform grid of `N_BINS` points is drawn between these two extrema. Crucially, $q_{\min}$ and $q_{\max}$ are *detached* from the computational graph before being used to build the grid. If we let gradients flow through the definition of the grid itself, the optimizer would be handed a second, competing way of minimizing the loss: instead of reshaping the free energy landscape, it could simply shrink or stretch the range of q to make the histogram boundaries and bin width favorable to a small rate, an artifact with no physical meaning. Treating the grid as a re-computed but non-differentiable structure at every epoch removes this degeneracy while still letting the histogram track the network's evolving output range.

With a grid in hand, we can evaluate the local KM moments of the latent variable q . For every grid point q_j we compute a soft weight for every trajectory frame q_i

$$w_{ij} = \exp\left[-\frac{(q_i - q_j)^2}{2\sigma_{\text{eff}}^2}\right], \quad N_j = \sum_i w_{ij} + \varepsilon, \quad (2.41)$$

with $\varepsilon = 10^{-12}$ protecting empty bins from a division by zero. The first two conditional moments of the frame-to-frame displacement $\Delta q_i = q_{i+1} - q_i$ are then the

weighted averages

$$M^{(1)}(q_j) = \frac{\sum_i w_{ij} \Delta q_i}{N_j}, \quad M^{(2)}(q_j) = \frac{\sum_i w_{ij} \Delta q_i^2}{N_j} \quad (2.42)$$

Since the full dataset is the concatenation of many independent association trajectories, a naive Δq computed across a trajectory boundary would splice the end of one relaxation event onto the beginning of an unrelated one, contaminating the local moments with a spurious, arbitrarily large jump. Then, a mask is introduced to flag the beginning of a new trajectory, avoiding this issue. From the variance of the displacement we obtain the local diffusion

$$D_{\text{local}}(q_j) = \frac{M^{(2)}(q_j) - [M^{(1)}(q_j)]^2}{2 \Delta t} \quad (2.43)$$

which is then averaged using a population-weighted (*confidence*) average

$$D = \frac{\sum_j N_j D_{\text{local}}(q_j)}{\sum_j N_j} \quad (2.44)$$

The value of D is clamped from below at 10^{-10} to avoid the numerical divergence of $1/D$ whenever a noisy or empty bin momentarily produces a vanishing or negative estimation. Considering a constant diffusion, it follows that the mean force, and from it the free energy, are

$$f(q_j) = \frac{M^{(1)}(q_j)}{\beta D \Delta t}, \quad F(q_j) = - \sum_{j' \leq j} f(q_{j'}) \delta q \quad (2.45)$$

where δq is the dynamic grid spacing and the cumulative sum is the discrete analogue of the integral relating force and free energy. The energy profile is shifted so that $F_{\text{min}} = 0$.

The kinetic loss: a differentiable mean first passage time

Given $F(q)$ and D on the current grid, the loss function is a direct discretization of the MFPT. First off, we need to evaluate the values of a , q_0 and b , where q_0 is the bottom of the reactant well, a is the reflecting boundary and b is the absorbing boundary. We take q_0 to be the grid point at which the estimated free energy is minimal, $q_0 = \arg \min_j F(q_j)$, consistently with the theoretical requirement that the reactant reference point sit deep inside the metastable well (section 2.1.3). For a and b , instead of simply taking the global extremal values of q (single outlier frames could otherwise displace the integration boundary and destabilize the loss from epoch to

epoch), we average the ten most extreme values on each side and map a, b onto their nearest grid indices, paying attention that $a < q_0$. With these in hand, the inner integral of eq. (2.27) is computed using a cumulative sum, and then the outer integral with trapezoid integration. The extrema of integration of the outer integral, instead of fixed q_0 and b , are chosen as $\text{start} = \min(q_0, b)$ and $\text{end} = \max(q_0, b)$: these are adaptive extrema which make the integral well-defined regardless of whether the learned latent variable q happens to increase or decrease along the association pathway during a given epoch. The scalar $k = 1/\tau$, the reconstructed kinetic rate itself, is the loss function that is minimized: unlike the AE, whose loss (eq. (2.37)) had no direct physical interpretation beyond a reconstruction error, here every value that is backpropagated through the network is, at every epoch, a legitimate estimate of the escape rate implied by the current CV.

CVNet: architecture and weight transfer from the autoencoder

Rather than designing a new network from scratch, the new network (which for simplicity will be referred to as **CVNet**) is built by stripping the decoder off the already-trained AE and keeping only encoder. The parameters are not re-initialized with the He-uniform scheme, but they are copied directly from the converged AE, so that training starts from a CV already known to produce a reasonable free energy barrier and rate. This warm start is motivated by the much harsher optimization landscape of the kinetic loss compared to a plain MSE: computing the rate is a series of nested compositions of soft-histogram weights, exponentials and cumulative sums, all several differentiation steps removed from the network weights, and a random initialization offers no guarantee that the very first forward pass would even produce a sensible free energy profile (e.g. a monotonic one, with a clearly identifiable q_0) to compute a gradient from. Starting from the parameters of the AE, which are already-known to produce good results, instead guarantees the loss landscape is explored from a physically reasonable CV, and lets us interpret whatever displacement the kinetic loss subsequently induces in the weights as the effect of injecting the kinetic constraint on top of an already legitimate reduction.

Optimization protocol

Training uses full-batch gradient descent: unlike the AE, which benefited from mini-batching precisely because MSE reconstruction error decomposes into an unbiased sum over independent samples, here every single gradient step needs the *entire* training ensemble to populate the local bins with enough statistics; a mini-batch would starve of frames the tails of the q distribution (in particular the region near

grid size (N_BINS)	40
σ_{rel}	0.01
$\beta = (kT)^{-1}$	1
epochs	15 000
validation trajectories	35 / 175
train trajectories	140 / 175

Table 2.2: Hyperparameters of the CVNet.

the transition barrier, which is by construction the least visited one), injecting large amounts of noise into $F(q)$ and D exactly where the integrals of the rate are most sensitive. One epoch therefore corresponds to a single Adam step computed on the full 140-trajectory training set.

Extracting and saving the final collective variable

Once training is complete, the CV q is obtained by evaluating the model on the full dataset. As with the AE, the raw output of the bottleneck is affected by the same sign ambiguity discussed in section 2.3 (q and $-q$ describe the same physics), so we again fix the convention that q should decrease from the dissociated to the associated state by comparing the average of the first ten and last ten frames of the full dataset and flipping the sign of q if the CV does not already decrease. The same min-max normalization procedure of the AE is also applied here, rescaling q into $[-1, 1]$ using the range actually spanned by the kinetically-trained network, and by applying the map $q \leftarrow (q + 1)/2$ we get the CV in the desired range $[0, 1]$. The resulting time series is saved alongside the simulation time to a two-column file, in the same format used for the AE-based CV. Table 2.2 summarizes the hyperparameters used in the CVNet (alongside those already used for the AE, table 2.1).

3. Results

3.1 Design of the AE

In this section we will discuss the results that led to the choices in the design of the AE. Most of the times, when dealing with the construction of a neural network, there are no rules that dictate the choice of things like the hyperparameters or the activation function, and optimization typically relies on systematic hyperparameter exploration to isolate the ideal configuration. We will thus present here the choices we made for the design of the AE.

3.1.1 Activation function

Three different activation functions were tested, `ReLU`= $\max\{0, x\}$, `softplus`= $\ln(1 + e^x)$ and `tanh`, and their performances were compared by looking at the evolution of their training and validation losses and by looking at the performance of the produced CV: the CVs are compared by looking at their free energy, computed using the optLE method, and rates. By looking at fig. 3.1a, we can see that with ReLU activation we achieve the lowest train loss, but tanh has the most stable learning evolution with training and validation losses being nearly identical. Then, if we look at the free energy associated to the evolution of their respective CVs fig. 3.1a and their kinetic rates, we can see that tanh achieves the highest free energy barrier ~ 27 kT with a rate $k = 3.24 \cdot 10^{-4} \text{ s}^{-1}$, whereas both softplus and ReLU achieve a lower barrier and higher rate, respectively ~ 17 kT with a rate $k = 1.32 \cdot 10^1 \text{ s}^{-1}$ and ~ 15 kT with a rate $k = 8.67 \cdot 10^1 \text{ s}^{-1}$. Then, tanh is deemed the best activation function and it's used for each neuron of the hidden layers. No activation is set on the bottleneck to avoid constraining the CV: forcing a single neuron through a `tanh` limits its range and risks saturation, whereas by unconstraining the bottleneck we allow the single latent variable to take any value necessary to represent the input, leading to a much better reconstruction.

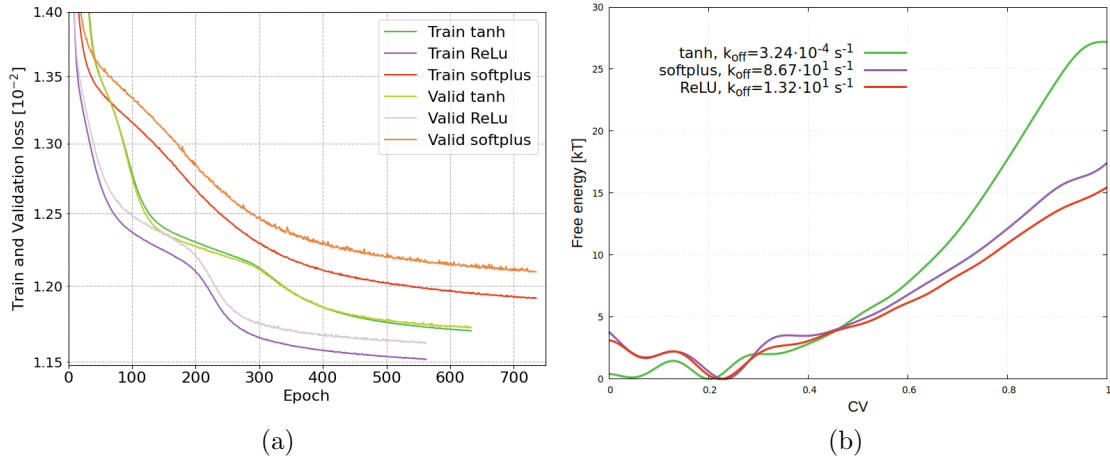


Figure 3.1: **(a)** Detail of the evolution of the training and validation losses of the AE trained with tanh, ReLU and softplus activation functions. **(b)** Free energy profiles (computed with optLE) and kinetic rates of the CVs produced with the three different activation functions.

3.1.2 Batch size

To choose the batch size, the AE was trained with 4 different values $\{1, 256, 512, 1024\}$ with `tanh` as the activation function and the loss curves were compared (fig. 3.2a). From the comparison, it's immediately noticeable that a batch size of 1, also known as *pure* stochastic gradient descend, traps the learning in a local minimum from which the model is unable to escape. Let's then look at the three remaining choices (fig. 3.2b). All three models reach approximately the same value of the loss. The model with batch size of 256 yields the fastest learning with few more than 300 epochs but its validation loss is noisy towards the end, a symptom that the batch size is too small. The model with batch size of 512 shows a very different evolution of training and validation losses, meaning that the model needs to be refined. The model with batch size 1024 takes the most amount of epochs, but it's the most stable (i.e. the evolution of training/validation loss is approximately the same) and shows much less noise than the other two models. For these reasons, the selected batch size is 1024.

3.2 Is the dynamic really overdamped?

As already discussed in section 2.1, the whole theoretical setup and in particular the kinetic variational principle are based on the assumption of an overdamped dynamic. This assumption, although reasonable, must be verified. We present here the

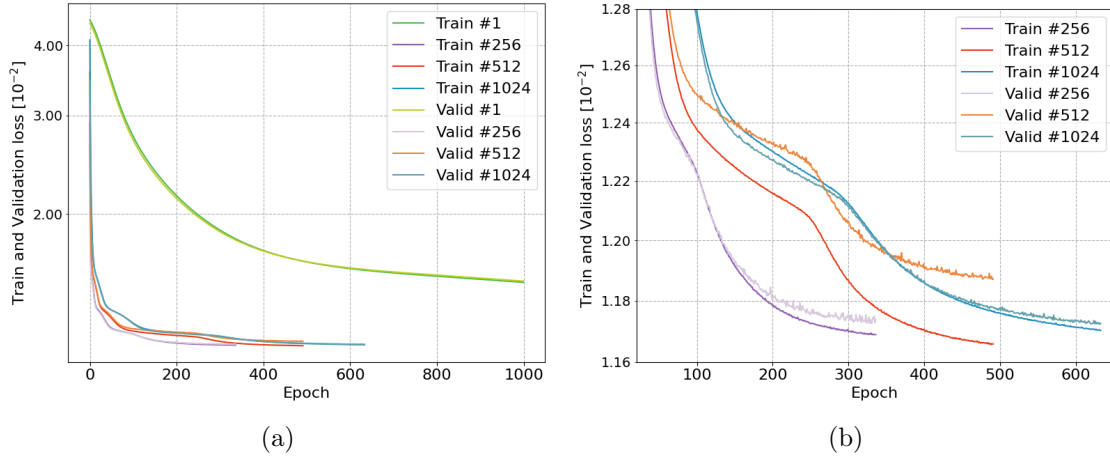


Figure 3.2: **(a)** Comparison of the training and validation losses for different values of the batch size, indicated in the legend as #n. **(b)** Detailed look at the loss curves for batch sizes 256, 512 and 1024.

two tests that have been performed to verify the overdamped (and thus, Markovian) nature of the system. The first test is a macroscopic one (see [1]) and it consists in verifying that the rate becomes independent of the time resolution Δt , i.e. the time interval between two datapoints, for a certain value of Δt greater than the autocorrelation time of the CV velocity, meaning that the system has no memory (non-Markovian) effects. The second test is a microscopic one (see [7]) and it consists in studying the properties of the noise of the Langevin dynamics: since we assume overdamped dynamics, characterized by delta-correlated Gaussian noise, we compute the autocorrelation function (to check delta-correlation) of the noise and the distribution of the noise (to check if the noise is Gaussian distributed with zero mean and unitary variance) for different time resolutions, to see which one best captures the Markovian behavior.

The dataset used throughout this work starts with a time resolution $\Delta t = 200$ ps, and thus we have performed the analyses with values 200 ps, 400 ps and 600 ps, done by removing respectively none, one and two datapoints after each other, being careful to not remove the initial and final ones of each trajectory. A problem of removing datapoints to test higher values of the time resolution is that we also lose important information, especially at the beginning of each trajectory where we have few datapoints representing the unbounded state: higher values of Δt would probably result in a better overdamped approximation, but we would lose on the physical side. The AE was trained with the three dataset featuring the different time resolutions, using the optLE method we computed the free energy and diffusion profiles

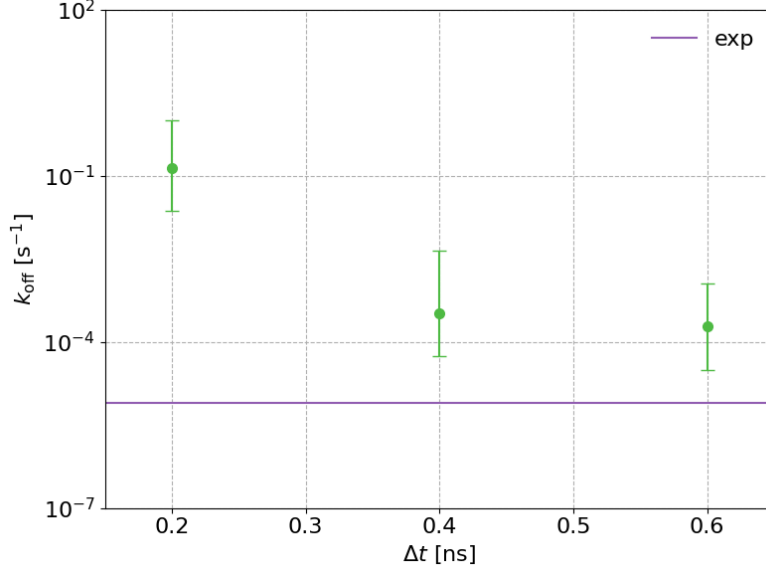


Figure 3.3: Values of the kinetic dissociation rate and error bars for time resolutions $\Delta t = 200, 400$ and 600 ps.

and finally the rates (fig. 3.3): as we can see, the rate becomes independent of the Δt starting from 400 ps, whereas at 200 ps there seem to be some memory effects. This result is in accordance with the one in [1]. To characterize the noise we use the Euler-Maruyama integrator

$$G = \frac{\Delta q + (\beta D F' - D') \Delta t}{\sqrt{2D\Delta t}} \quad (3.1)$$

which gives the value of the effective noise $G(t)$ for each displacement of the CV $\Delta q = q(t + \Delta t) - q(t)$. If the dynamics are faithfully captured by an overdamped Markovian model at resolution Δt then $G(t)$ must behave as a memory-less, zero-mean, unit-variance Gaussian white noise, meaning its normalized autocorrelation function

$$C(t) = \frac{\langle G(0)G(t) \rangle}{\langle G^2 \rangle} \quad (3.2)$$

is expected to go to zero for any non-zero lag time $t \geq \Delta t$. When subjected to this frame-by-frame test, the reconstructed noise exhibits distinct deviations from ideal Gaussian white noise behavior. As shown in the autocorrelation profiles (fig. 3.4a), $C(t)$ drops abruptly after the first step but fails to decay fully to zero. Instead, it enters a persistent, non-vanishing plateau of $C(t) \sim 0.05$ for 200 ps and ~ 0.15 for 400 and 600 ps that extends across long correlation times. The flat nature of this plateau, without any exponential or power-law decay, strongly indicates that this

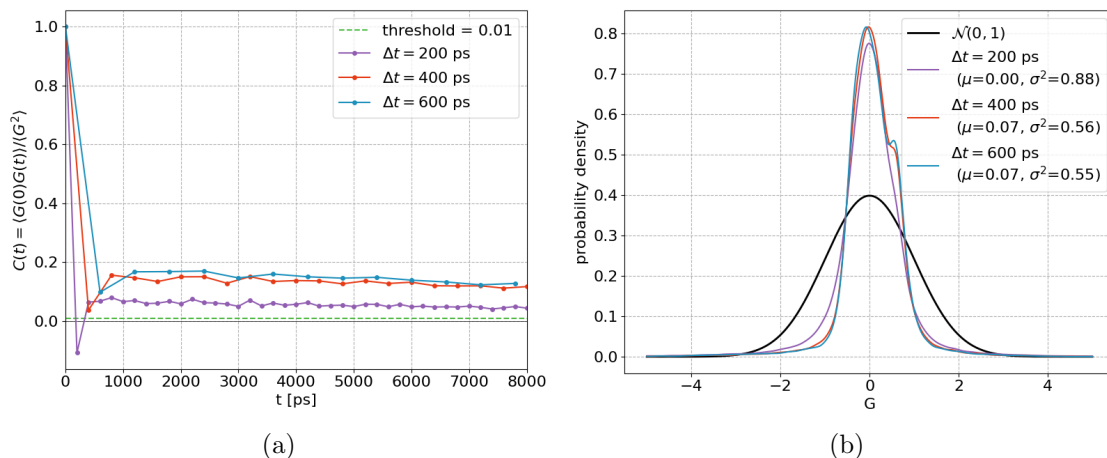


Figure 3.4: **(a)** Autocorrelation function for each time resolution Δt , all plateauing at values different from zero. **(b)** Distribution (with mean and variance) of the noise of the three processes compared the (theoretical) Gaussian distribution.

artifact doesn't arise from a physical memory effect. Rather, this seems more like a systematic error. One possible reason for this error is the lack of samples in the dissociated region, and an overwhelming abundance in the associated one, skewing the averages. Another possible reason for this offset is that is that $\langle G \rangle \neq 0$ for $\Delta t = 400, 600$ ps, as we can see from fig. 3.4b. Despite this non ideal microscopic behavior, macroscopically the kinetic rate exhibits robust stability and independence from the choice of time resolution.

3.3 Dataset of only protein atom pairs

Once we have fully characterized the AE (the full set of hyperparameters shown in table 2.1), we can train it using the two datasets discussed in section 2.2. Let's start by presenting the results using the dataset containing only information about atom pair interactions. Figure 3.5 represents the free energy and diffusion, computed using the optLE method, associated to the CV found by training the AE. The solid line represents the profile obtained by training the model on the whole dataset. The dotted lines represent the best and worst performances obtained by training the model on sub-samples of the dataset; in particular, the model was trained on multiple datasets (10) obtained by removing a random number ($20 \div 40$) of randomly chosen trajectories. In conclusion, the AE produces a CV whose kinetic dissociation rate is $k_{\text{off}} = 3.24 \cdot 10^{-4} \text{ s}^{-1}$ with a block-average error bar in the range $4.35 \cdot 10^{-3} - 5.60 \cdot 10^{-5} \text{ s}^{-1}$. The estimated dissociation rate is not compatible with

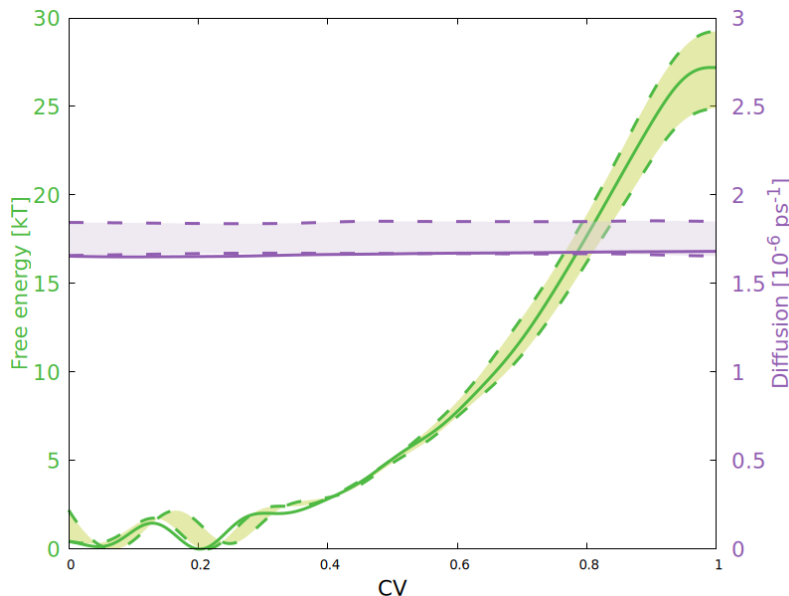


Figure 3.5: Free energy and diffusion, computed using optLE, of the CV obtained with the AE trained with the pairs-only dataset for a time resolution $\Delta t = 400$ ps. The solid line corresponds to the training performed with the entire dataset, while dashed lines represent a training in which a random number of trajectories was removed.

the experimental one as one can see from fig. 3.3. One problem to be underlined here is the approximate nature of what we are considering; the all-atom MD simulation, although performed as accurately as possible, uses *empirical* force-fields, unavoidably making the interatomic forces approximate.

Another important result to be presented here is the failure of the KM expansion to capture the free energy and diffusion of the system. In fact, using the same variable produced by the AE, the KM expansion gives a free energy barrier of ~ 10 kT (almost 17 kT lower than that obtained with optLE) and a diffusion of $5 \cdot 10^{-6} \text{ ps}^{-1}$ (same order of magnitude of optLE); the rate is $k_{\text{off}} = 8.52 \cdot 10^3 \text{ s}^{-1}$. We attribute two reasons to the failure of the KM expansion in this system:

1. scarcity of points in the “dissociated” region and abundance in the “associated” one. The MD simulations we performed are unbiased, meaning that the two proteins are left to spontaneously associate. As already explained, this is a very fast process for the BB complex, meaning we are left with a spare number of datapoints in the initial “dissociated” state. This is also reflected by the free energy: a high value of the free energy reflects a region that is, from a

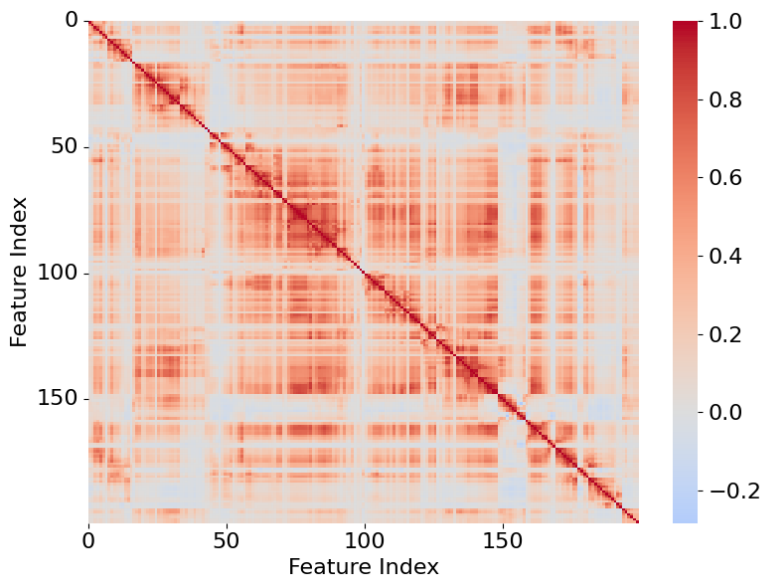


Figure 3.6: Heatmap representation of the feature-feature covariance matrix of the water-only dataset.

probabilistic viewpoint, rarely visited. When we then compute the averages for the KM expansion, we are left with a region that hardly if not at all contributes, skewing the results toward the vastly more populated “associated” region.

2. the time resolution problem. The KM expansion (eq. (2.19)) is the discretization of the Taylor expansion of the Smoluchowski equation, and thus requires small time increments to be well-defined. However, as discussed thoroughly, to get a good overdamped regime and loose memory effects, we need a time resolution $\Delta t \geq 400$ ps.

Both of these problems, especially the second one, are maybe mitigated in the optLE method, which first of all is second order in Δt , and instead of computing localized moments it maximizes a global log-likelihood function over entire trajectories.

3.4 Including the water information

In this section we present the results obtained using the dataset made of $D_1 = 487$ features of pair interactions (atom-atom distance) and $D_w = 199$ features about the amount of water around each interface atom of Barnase and Barstar. With this dataset we train the AE, obtain the CV and get (fig. 3.8) a free energy barrier of

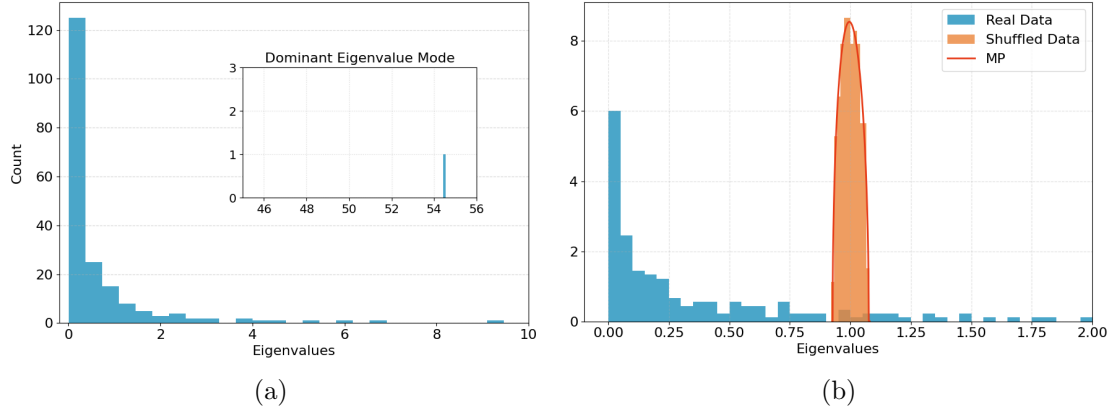


Figure 3.7: **(a)** (non normalized) distribution of the eigenvalues of the covariance matrix, showing a bulk towering at 0 and a single eigenvalue $\lambda \approx 54.5$ which correspond to the dominant mode. **(b)** (normalized) eigenvalue distribution compared to the distribution obtained when the dataset is shuffled, meaning that correlations are broken; superimposed it's the (theoretical) Marchenko-Pastur distribution.

$\sim 24\text{kT}$, a diffusion of $1.8 \cdot 10^{-6} \text{ ps}^{-1}$ and a kinetic rate $k_{\text{off}} = 6.72 \cdot 10^{-3} \text{ s}^{-1}$ with a block-average error bar in the range $3.04 \cdot 10^{-2} - 4.46 \cdot 10^{-3} \text{ s}^{-1}$. It's immediately observed that the results obtained with this dataset are worse than those of the atom pairs-only dataset, with a barrier lower of $\sim 3 \text{ kT}$ and the rate being one order of magnitude greater. The addition of the information about water then doesn't contribute to finding the CV, and in fact hampers the results. It's then interesting to ask ourself if the water dataset brings any real information. To this end, we will make use of the theory of random matrices [37, 38].

Let us just consider the water dataset $X_w \in \mathbb{R}^{N \times D_w}$ and let's compute the feature-feature covariance matrix

$$C_{ij} = \frac{1}{N} \sum_{n=1}^N x_i^n x_j^n \quad (3.3)$$

where x_i^n is the value of feature i at time n , $C_{ij} > 0$ if features i, j tend to follow the same *trend*, $C_{ij} < 0$ if one follows a trend and the other follows it in the opposite direction, and $C_{ij} = 0$ when the two features are uncorrelated. The full covariance matrix is shown through a heatmap in fig. 3.6, and from it we can see that the features are mostly positively correlated: this isn't surprising considering the features we are looking at; it's expected that if the amount of water molecules around a certain interface atom decreases, this will be true also for its neighboring interface atoms, hence the positive correlations. Since the features are truly strongly correlated, we can immediately disregard the idea that our dataset is pure noise (null model). To confirm it, let's compute the eigenvalues of C and plot the histogram of

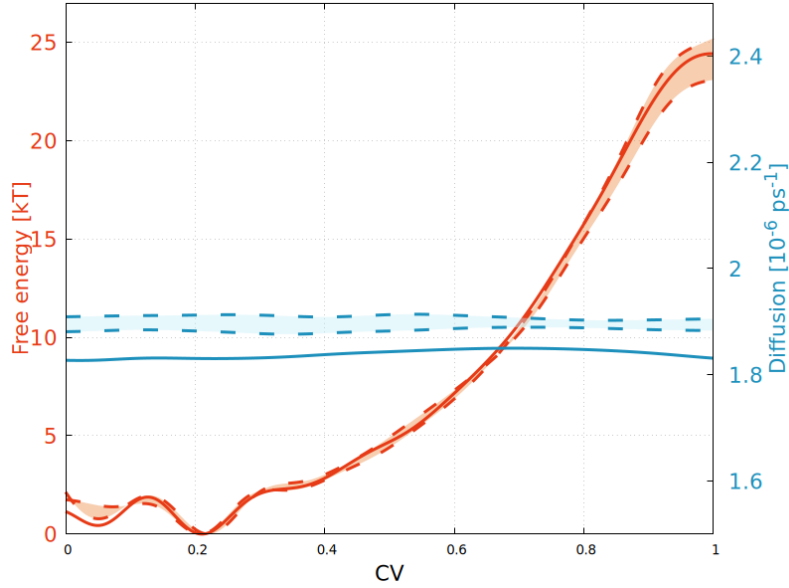


Figure 3.8: Free energy and diffusion, computed using optLE, of the CV obtained with the AE trained with the pairs and water dataset for a time resolution $\Delta t = 400$ ps. The solid line corresponds to the training performed with the entire dataset, while dashed lines represent a training in which a random number of trajectories was removed.

their distribution (fig. 3.7a): in the case of pure noise we would obtain a *Marchenko-Pastur* (MP) distribution (plotted next to the distribution of the eigenvalues of C , obtained by shuffling the values along the columns and in fact breaking the correlations), whereas we see a clear bulk of eigenvalues near zero and a clear highest eigenvalue at around 54.5. The bulk near zero represents a huge set of redundant features, whereas the few large eigenvalues represent directions of strong shared variance, with the highest eigenvalue having a 27.4% explained variance. To better quantify the redundancy suggested by the bulk near zero, we computed the effective rank of C

$$R_{\text{eff}} = \frac{(\sum_{i=1}^D \lambda_i)^2}{\sum_{i=1}^D \lambda_i^2} = 10.9 \quad (3.4)$$

This confirms quantitatively that, despite the nominal dimensionality of $d = 199$, the system's variance is effectively captured by only about eleven independent directions, the remaining features being approximately linear combinations of these. In conclusion, the inclusion of the water dataset leads to a degradation in the AE performance. This counterintuitive behavior can be attributed to two distinct, yet complementary, physical and architectural factors:

1. As shown by the empirical eigenvalue spectrum, the water dataset exhibits severe information redundancy, with the vast majority of the 199 features collapsing into a near-zero bulk. By looking at fig. 2.3b we can see that while the information on the water molecules around a single atom is noisy, the information on the de-solvation around the whole protein is very clear. Physically we can say that the highly-informative modes (associated to the higher valued eigenvalues) capture the global de-solvation process, while the remaining orthogonal degrees of freedom represent highly stochastic, localized thermal fluctuations of individual water molecules. When this raw high-dimensional dataset is fed into the AE, the MSE treats all input dimensions “democratically”. As a consequence, the network’s capacity is heavily penalized by trying to reconstruct ~ 188 dimensions of uncoordinated microscopic noise rather than isolating the low-dimensional physical manifold. The true learning signal is effectively diluted, trapping the network in an unfavorable optimization landscape.
2. Alternatively, the information provided by this specific dataset may offer no complementary insights into the actual RC of the binding transition. Because the physical variance of the water network is almost entirely compressed into a single eigenvalue ($\lambda \approx 54.5$), the dataset acts almost as a one-dimensional descriptor tracking the global, macroscopic de-solvation of the interface. If this global de-solvation mode represents a trivial process or lacks the nonlinear topological resolution necessary to characterize the transition state, its inclusion provides no predictive power. In this scenario, the AE is forced to process a high-dimensional feature space that yields only a single meaningful collective mode that may already be implicitly captured by the core conformational degrees of freedom of the proteins.

3.5 CVNet

In this final section, we will present the results for the CV obtained by training the CVNet just for the pairs only dataset, using the hyperparameters in table 2.2. As discussed in section 2.4, the CVNet is trained on minimizing the kinetic rate, treated as the loss function of the network, computed using the free energy and diffusion found with the KM expansion method. However, as we’ve seen in section 3.3, the KM expansion is expected to perform badly for this system. The training process showed a strong instability towards values of the learning rate greater than 10^{-4} , diverging after few epochs for 10^{-3} or higher. For this reason the learning rate was set to get smaller ($10^{-4} \rightarrow 10^{-7}$) through the epochs. This made training extremely

slow, and required a huge amount of epochs. The resulting rate after a training of 15 000 epochs is $k_{\text{off}} = 3.07 \text{ s}^{-1}$. This result improves on the one obtained from the AE using KM expansion, but it's 10^4 times higher than that obtained with optLE. What is probably happening is that the network is not learning how to reduce the rate, since it becomes higher with respect to the optLE one, but rather how to “fix” the problems with the rate computed with free energy and diffusion of the KM expansion.

4. Conclusions and achievements

In this thesis, we investigated the data-driven characterization of the kinetics of protein-protein association by applying machine learning to all-atom, unconstrained molecular dynamics simulations of the Barnase-Barstar complex. By moving away from traditional enhanced sampling simulations that require artificial biasing forces, we showed that out-of-equilibrium relaxation trajectories could, in principle, yield precise kinetic rate constants when paired with robust statistical inference frameworks. Let’s summarize what we have investigated and the achievements of this work:

1. Optimization of the autoencoder: we built and tested an autoencoder for non-linear dimensionality reduction utilizing the `mlcolvar` library. Through systematic hyperparameter exploration, a symmetric, undercomplete autoencoder with tanh activation function and a batch size of 1024 was shown to have the most stable training process and capable of achieving the best one-dimensional collective variable, obtained upon projection of the high-dimensional configuration space of the Barnase-Barstar complex.
2. Validation of overdamped Kinetics: we tested the underlying Markovian and overdamped assumptions essential to modeling kinetics via stochastic differential equations. A macroscopic time resolution test revealed that memory effects effectively vanish at a timescale resolution of $\Delta t \geq 400$ ps. A microscopic analysis of the effective noise highlighted minor systematic deviations from pure Gaussian white noise possibly due to sampling imbalances in the unbound state, though macroscopic observables remained robust.
3. Performance of different free energy and diffusion estimators: using the collective variable derived from interfacial pair-contact features, the optimized Langevin equation (optLE) method gave us a free energy barrier of ~ 27 kT , yielding a dissociation rate of $k_{\text{off}} = 3.24 \cdot 10^{-4} \text{ s}^{-1}$. Conversely, the Kramers-Moyal expansion failed catastrophically, underestimating the free energy barrier by 17 kT . This failure was attributed to data scarcity in the “dissociated”

regions during spontaneous association and the finite-discretization errors inherent to large operational values of Δt .

4. Quantifying solvation redundancy: adding the information of the amount of water molecules around the proteins during the association process into the training dataset resulted in a counterintuitive degradation of the collective variable, lowering the reconstructed free energy barrier and accelerating dissociation rate by an order of magnitude. To understand this phenomenon, we analyzed the feature-feature covariance matrix of the water-only dataset. The empirical eigenvalue distribution revealed an effective rank of only $R_{\text{eff}} = 10.9$ out of 199 nominal dimensions. This quantitative evaluation verified that the water feature space is highly redundant, meaning the autoencoder may have underperformed because its optimization landscape was dominated by uninformative collective water fluctuations that obscured important local interactions.
5. Challenges in kinetic backpropagation (CVNet): finally, we introduced CVNet, an innovative architecture that eliminated the decoder from the autoencoder to directly optimize the encoder’s weights against a physical, differentiable Mean First Passage Time loss. While CVNet successfully adjusted the mapping to significantly lower the rate relative to the raw structural KM baseline ($k_{\text{off}} = 3.07 \text{ s}^{-1}$), the resulting rate remained four orders of magnitude faster than the global likelihood baseline. This structural divergence indicates that when constrained by localized stochastic operators (like soft-binned KM moments) in sparse data regimes, the neural network acts primarily to regularize the numerical deficiencies of the localized estimator rather than isolating the underlying global rate-minimizing coordinate.

4.1 Perspectives

As a final effort I present here some promising ideas, for future research, that build upon the methodology and findings established in this work. These perspectives aim at addressing current limitations and enhancing the accuracy and resolution of data-driven kinetic predictions.

Higher-dimensional CVs

Throughout this study, the projection of the dynamics was restricted to a one-dimensional CV. This low-dimensional framework has its theoretical justification in the work done by Lu et al. [39] who formally proved that accurate rates can be

obtained from an overdamped Langevin model when projecting a high-dimensional system into on a singular optimal CV, namely the committor. However, even with this theoretical support, projecting the phase space of a system of approximately one hundred thousand atoms into a single variable may be a too simple and crude of an approximation in numerical implementations. Furthermore, it is also well known that complex biomolecular processes rarely progress along a singular, well-defined transition path, but often follow different pathways: for the BB complex this is well documented, and in fact BB arrives at the associated state via three distinct binding pathways (see [1]). A one-dimensional CV isn't capable of capturing such multiple pathways, inevitably discarding critical kinetic information. In the context of protein folding, Belkacemi et al. [35] showed that choosing a CV with dimension $d > 1$ is optimal for capturing complex conformational landscapes. Although protein folding and protein-protein association are sensibly different processes, extending our neural network framework to multi-dimensional latent spaces represents a natural and possibly necessary progression to capture the ensemble of binding pathways.

Replace KM with optLE in the CVNet

A major result of this work was to show that the KM expansion fails to accurately resolve the free energy barriers and kinetic rates of the BB complex possibly due to data scarcity in the transition region. Conversely, the optLE framework showed remarkable robustness against these sampling deficiencies. A key objective for future work is therefore to substitute the KM estimator with the optLE framework within the CVNet optimization pipeline. Implementing this transition introduces a substantial algorithmic challenge. While the KM framework yields explicit, analytical expressions for drift and diffusion coefficients that are readily differentiable via standard backpropagation, optLE is inherently rooted in a stochastic, Monte Carlo-based optimization procedure. Replacing a deterministic, gradient-based loss function with a non-differentiable, sample-based optimization routine prevents the direct calculation of analytical gradients with respect to the neural network weights.

Bibliography

- [1] Line Mouaffac, Karen Palacio Rodriguez, Alessandro Barducci, Peter Bolhuis, Isabelle Callebaut, and Fabio Pietrucci. Predicting the thermodynamics and kinetics of protein-protein interaction from spontaneous association simulations. *ChemRxiv*, 2025(0829), 2025.
- [2] Line Mouaffac, Karen Palacio-Rodriguez, and Fabio Pietrucci. Optimal reaction coordinates and kinetic rates from the projected dynamics of transition paths. *Journal of Chemical Theory and Computation*, 19(17):5701–5711, 2023. PMID: 37550088.
- [3] Werner Kühlbrandt. The resolution revolution. *Science*, 343:1443 – 1444, 2014.
- [4] Jesse Durham, Jing Zhang, Ian Humphreys, Jimin Pei, and Qian Cong. Recent advances in predicting and modeling protein-protein interactions. *Trends in biochemical sciences*, 48, 04 2023.
- [5] Fabio Pietrucci. Strategies for the exploration of free energy landscapes: Unity in diversity and challenges ahead. *Reviews in Physics*, 2:32–45, 2017.
- [6] Aldo Glielmo, Brooke E. Husic, Alex Rodriguez, Cecilia Clementi, Frank Noé, and Alessandro Laio. Unsupervised learning methods for molecular simulation data. *Chemical Reviews*, 121(16):9722–9758, 2021. PMID: 33945269.
- [7] Karen Palacio-Rodriguez and Fabio Pietrucci. Free energy landscapes, diffusion coefficients, and kinetic rates from transition paths. *Journal of Chemical Theory and Computation*, 18(8):4639–4648, 2022. PMID: 35899416.
- [8] Luigi Bonati, GiovanniMaria Piccini, and Michele Parrinello. Deep learning the slow modes for rare events sampling. *Proceedings of the National Academy of Sciences*, 118(44):e2113533118, 2021.

-
- [9] Wei Zhang, Carsten Hartmann, and Christof Schütte. Effective dynamics along given reaction coordinates, and reaction rate theory. *Faraday Discussions*, 195:365–394, 12 2016.
 - [10] Albert C. Pan, Daniel Jacobson, Konstantin Yatsenko, Duluxan Sritharan, Thomas M. Weinreich, and David E. Shaw. Atomic-level characterization of protein–protein association. *Proceedings of the National Academy of Sciences*, 116(10):4244–4249, 2019.
 - [11] James C. Gumbart, Benoît Roux, and Christophe Chipot. Efficient determination of protein–protein standard binding free energies from first principles. *Journal of Chemical Theory and Computation*, 9(8):3789–3798, 2013.
 - [12] Robert W. Hartley. Barnase and barstar: Expression of its cloned inhibitor permits expression of a cloned ribonuclease. *Journal of Molecular Biology*, 202(4):913–915, 1988.
 - [13] Robert W. Hartley. Barnase and barstar: two small proteins to fold and fit together. *Trends in Biochemical Sciences*, 14(11):450–454, 1989.
 - [14] Gideon Schreiber and Alan R. Fersht. Interaction of barnase with its polypeptide inhibitor barstar studied by protein engineering. *Biochemistry*, 32 19:5145–50, 1993.
 - [15] Gideon Schreiber and {Alan R.} Fersht. Rapid, electrostatically assisted association of proteins. *Nature Structural Biology*, 3(5):427–431, 1996.
 - [16] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.
 - [17] R. Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, 2001.
 - [18] Hannes Risken. *The Fokker-Planck equation : methods of solution and applications*. Springer, 1984.
 - [19] William Coffey, Yuri Kalmykov, and John Waldron. *The Langevin Equation: With Applications to Stochastic Problems in Physics, Chemistry and Electrical Engineering*. World Scientific, 09 2004.
 - [20] H. Mori. Transport, Collective Motion, and Brownian Motion. *Progress of Theoretical Physics*, 33(3):423–455, March 1965.

- [21] Cihan Ayaz, Laura Scalfi, Benjamin A. Dalton, and Roland R. Netz. Generalized langevin equation with a nonlinear potential of mean force and nonlinear memory friction from a hybrid projection scheme. *Phys. Rev. E*, 105:054138, May 2022.
- [22] Hadrien Vroylandt and Pierre Monmarché. Position-dependent memory kernel in generalized langevin equations: Theory and numerical estimation. *The Journal of Chemical Physics*, 156(24):244105, 06 2022.
- [23] Hugues Meyer, Thomas Voigtmann, and Tanja Schilling. On the non-stationary generalized langevin equation. *The Journal of Chemical Physics*, 147(21):214110, 12 2017.
- [24] Hugues Meyer, Thomas Voigtmann, and Tanja Schilling. On the dynamics of reaction coordinates in classical, time-dependent, many-body processes. *The Journal of Chemical Physics*, 150(17):174118, 05 2019.
- [25] Line Mouaffac. *Modeling the dynamics of complex systems with stochastic processes and machine-learned coordinates*. Theses, Sorbonne Université, September 2025.
- [26] Stefano Piana and Alessandro Laio. Advillin folding takes place on a hypersurface of small dimensionality. *Phys. Rev. Lett.*, 101:208101, Nov 2008.
- [27] Alex Rodriguez, Maria d’Errico, Elena Facco, and Alessandro Laio. Computing the free energy without collective variables. *Journal of Chemical Theory and Computation*, 14(3):1206–1215, 2018. PMID: 29401379.
- [28] Gareth A. Tribello, Massimiliano Bonomi, Davide Branduardi, Carlo Camilloni, and Giovanni Bussi. Plumed 2: New feathers for an old bird. *Computer Physics Communications*, 185(2):604–613, 2014.
- [29] Massimiliano Bonomi, Davide Branduardi, Giovanni Bussi, Carlo Camilloni, Davide Provati, Paolo Raiteri, Davide Donadio, Fabrizio Marinelli, Fabio Pietrucci, Ricardo A. Broglia, and Michele Parrinello. Plumed: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications*, 180(10):1961–1972, 2009.
- [30] Jeffrey Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic Programming and Evolvable Machines*, 19, 10 2017.
- [31] Umberto Michelucci. An introduction to autoencoders, 2022.

- [32] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810:1–124, May 2019.
- [33] David E. Rumelhart and David Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9(1):75–112, 1985.
- [34] Luigi Bonati, Enrico Trizio, Andrea Rizzi, and Michele Parrinello. A unified framework for machine learning collective variables for enhanced sampling simulations: `jti/mlcolvari/tti`. *The Journal of Chemical Physics*, 159(1), 2023.
- [35] Zineb Belkacemi, Marc Bianciotto, Hervé Minoux, Tony Lelièvre, Gabriel Stoltz, and Paraskevi Gkeka. Autoencoders for dimensionality reduction in molecular dynamics: Collective variable dimension, biasing, and transition states. *The Journal of Chemical Physics*, 159(2):024122, 07 2023.
- [36] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [37] Marc Potters and Jean-Philippe Bouchaud. *A First Course in Random Matrix Theory: for Physicists, Engineers and Data Scientists*. Cambridge University Press, 2020.
- [38] Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. *Introduction to Random Matrices*. Springer International Publishing, 2018.
- [39] Jianfeng Lu and Eric Vanden-Eijnden. Exact dynamical coarse-graining without time-scale separation. *The Journal of Chemical Physics*, 141(4):044109, 07 2014.