Simulating Chemical Reactivity using Machine Learning Potentials and Umbrella Sampling



Tristan Johnston-Wood

The Queen's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2023

Abstract

Dynamic simulations of reactions enable prediction of mechanisms and explanation of chemical phenomena. Traditional methods of simulation include *ab initio* molecular dynamics (AIMD), which is generally accurate but expensive, and force field methods, which are fast but often unsuitable for exploring bond breaking or forming processes. Machine learning potentials (MLPs) have emerged as a promising tool for exploring dynamics at the accuracy of quantum mechanics and a fraction of the cost of AIMD. However, training often requires many data points and expert knowledge. In this thesis, we introduce strategies to efficiently train and use MLPs to study a range of chemical processes, utilising and extending the umbrella sampling (US) methodology. **Chapter 2** presents a training strategy for the generation of MLPs along with the *gap-train* and *mlp-train* Python packages developed during this thesis, which automate training and enhanced sampling. In **Chapter 3**, we illustrate the use of the ACE MLP to study terpene reactions. The generated MLPs accurately predict free energy and product ratios, providing good agreement with experiment and previous computational studies. We explore a proposed bifurcation of a sesquiterpene, using MLPs to confirm its existence. Nuclear quantum effects are introduced using path integral molecular dynamics and MLPs, highlighting examples where quantum dynamics affect the free energy and product ratios in hydrogen migration reactions. **Chapter 4** presents a series of metrics to quantify the convergence and overlap in US. We study the influence of overlap on free energy for analytic, MLP and protein systems, finding an overlap of 15% is sufficient to converge free energy. Finally, in **Chapter 5**, attempts to develop an adaptive US scheme are presented, along with a Python package, *adaptiveus*, to automate the process.

Acknowledgements

One rarely realises how integral support, discussion and collaboration with others is before embarking on their PhD. However, coming to the end, I owe a great deal to the people mentioned here. First and foremost, I would like to thank Professor Fernanda Duarte, who has provided guidance, detailed feedback and support throughout. During the first half of my DPhil, Tom Young provided support well above the call of duty, teaching me (almost) everything I know about Python and always enthusiastically engaging in scientific discussion, often ending in fiery debates but fruitful ideas. I am now confident I didn't annoy him too much as I still enjoy a glass of wine with him talking science.

I would also like to thank the other members of the Duarte group, especially Alistair for interesting discussions on bonds and orbitals; Matina ($\mu\alpha\lambda\acute{\alpha}\kappa\alpha$), Tomasz and honorary members Harry and Duthie for guidance early on and general vibes later; and Matthew (freelance broadcaster and commentator) for his distracting and brightening presence. Thank you to the members of the MLP subgroup for their ideas and feedback, including Hanwen, Veronika and Khan. Double trouble Aleksy and Martin have provided a plethora of scientific ideas and fun, as has Kate, even allowing me to feature in her mockumentary.

My first year at Oxford saw many memories created and knowledge gained through TMCS, and I would like to thank my fellow classmates for their help as well as Martin Galpin for as many pints as maths lessons. I have seen many Part II students pass through the group, and I am happy to say how much I enjoyed their presence and the life they gave to the group, including Janko, James, and recently Alex, Valdas and Josh. I have also shared in a number of fruitful

academic collaborations including with Volker, Adrian, Gábor, Julien and Steve.

Outside of the group, Lillian has supported me through balancing my DPhil with applications and I wouldn't have had nearly as much fun without her as well as Mike and Ava; all three gifting me invaluable ideas and feedback. Finally, I would like to thank my parents, Ruth and Chris, and sister Marianne, for their unwavering support and continued excitement at *yet another degree*.

Declaration

I, Tristan Johnston-Wood, declare that the thesis I am submitting is entirely my own work except where indicated in the text.

Signature: 

Date: 29th June 2023

Software availability

The Python packages developed as part of this thesis are freely available on *GitHub*. *gap-train* and *mlp-train* were developed in collaboration with Dr Tom Young, while *adaptiveus* was developed by the author.

i. *gap-train*: https://github.com/duartegroup/gap-train

ii. *mlp-train*: https://github.com/duartegroup/mlp-train

iii. *adaptiveus*: https://github.com/tristan-j-wood/adaptive-us

List of Publications

This thesis is partially based on the following publications:

Chapter 3

- Reference [1]: Young, T. A.; Johnston-Wood, T.; Deringer, V. L.; Duarte, F. "A transferable active-learning strategy for reactive molecular force fields", *Chemical science* **2021**, *12*, 10944–10955.

  Contribution: Development and implementation of *gap-train*, MM-MD calculations for training and data analysis.

- Reference [2]: Young, T. A.; Johnston-Wood, T.; Zhang, H.; Duarte, F. "Reaction dynamics of Diels–Alder reactions from machine learned potentials", *Physical Chemistry Chemical Physics* **2022**, *24*, 20820–20827.

  Contribution: Conceptualisation, formal analysis, investigation, software, visualization, writing.

⁂

Contributing-author articles published or submitted for publication during the course of this degree that do not form part of this thesis are as follows:

- Reference [3]: Shaw, R. A.; Johnston-Wood, T.; Ambrose, B.; Craggs, T. D.; Hill, J. G. "CHARMM-DYES: parameterization of fluorescent dyes for use with the CHARMM force field", *Journal of Chemical Theory and Computation* **2020**, *16*, 7817–7824.

  Contribution: Conceptualisation, writing, data analysis, data generation.

- Reference [4]: Chan, H. H.; Moesser, M. A.; Walters, R. K.; Malla, T. R.; Twidale, R. M.; John, T.; Deeks, H. M.; Johnston-Wood, T.; Mikhailov, V.; Sessions, R. B., et al. "Discovery of SARS-CoV-2 M pro peptide inhibitors from modelling substrate and ligand binding", *Chemical science* **2021**, *12*, 13686–13703.

  Contribution: Setup and analysis of QMDFT, writing.

⁂

The following manuscripts were in preparation at the time of completion of this degree:

Chapter 4

- Johnston-Wood, T.; Duarte. F. Free energy and bifurcation dynamics of rearrangement reactions using machine learning potentials.

  Contribution: Conceptualisation, data generation, data analysis, writing.

Contents

1 Introduction **1**

1.1 Computational methods for chemical reactivity . . . 3

1.1.1 A brief history of chemical reactivity prediction . . . 5

1.2 Machine learning potentials . . . 7

1.2.1 Machine learning potentials preamble . . . 9

1.2.1.1 MLP construction . . . 10

1.2.1.2 Representations . . . 10

1.2.2 Neural network potentials . . . 12

1.2.2.1 The PhysNet architecture . . . 13

1.2.3 Nonparametric machine learning potentials . . . 14

1.2.3.1 Gaussian approximation potential . . . 15

1.2.3.2 Atomic cluster expansion . . . 16

1.2.4 Comparative analysis of PhysNet, GAP and ACE . . . 20

1.2.5 Machine learning potential software . . . 21

1.3 Reactions of terpenes and related molecules . . . 22

1.3.1 Isoindenes . . . 23

1.3.2 Terpinenes . . . 24

1.3.3 Indanes . . . 25

1.3.4 Sesquiterpenes . . . 25

1.4 Umbrella sampling . . . 25

1.4.1 Theoretical basis . . . 27

1.4.1.1 The reaction coordinate . . . 28
1.4.1.2 Umbrella sampling dynamics . . . 29
1.4.1.3 The weighted histogram analysis method . . . 31
1.4.2 Convergence testing . . . 36
1.4.3 Umbrella sampling software . . . 37
1.4.4 Adaptive umbrella sampling . . . 38
1.4.4.1 Adaptive US with a barrierless potential . . . 39
1.4.4.2 Adaptive US with ideal overlap of windows . . . 40
1.5 Methods . . . 41
1.5.1 Electronic structure theory . . . 41
1.5.1.1 Preamble . . . 41
1.5.1.2 Hartree-Fock theory . . . 43
1.5.1.3 Density functional theory . . . 45
1.5.2 Molecular Dynamics . . . 49
1.5.2.1 Ensemble theory . . . 49
1.5.2.2 Molecular dynamics integrators . . . 51
1.5.2.3 Thermostats . . . 52

2 Machine Learning Potentials Method Development 54
2.1 Summary . . . 55
2.2 MLP training strategies for molecular systems . . . 56
2.2.1 Validation of MLPs . . . 57
2.2.2 Training a water system using the PhysNet potential . . . 58
2.2.2.1 Water monomer potential . . . 58
2.2.2.2 Water cluster potential . . . 60
2.2.3 Training a GAP using classical MD . . . 63
2.2.4 Active learning training strategy . . . 66
2.3 Development of automated MLP training . . . 68
2.3.1 *gap-train* . . . 69
2.3.2 *mlp-train* . . . 70
2.4 Conclusions . . . 73

3 Reaction Dynamics from MLPs **75**

3.1 Summary . . . 76

3.2 Optimisation of MLP training . . . 77

3.2.1 Training an ACE potential from reactants . . . 78

3.2.2 Training an ACE potential from the TS . . . 79

3.2.2.1 AL via a relaxed PES scan . . . 80

3.2.2.2 AL via climbing image nudged elastic band . . . 80

3.2.2.3 AL via CINEB + TS optimisation . . . 82

3.2.3 Automation of training along a reaction coordinate . . . 84

3.2.3.1 Method development . . . 84

3.2.3.2 Application to indane . . . 85

3.2.4 Choice of MLP selector . . . 87

3.3 Methyl shifts in isoindene . . . 87

3.3.1 Training an isoindene ACE potential . . . 90

3.3.2 Free energy of the isoindene reaction . . . 91

3.3.3 Path integral molecular dynamics of isoindene . . . 96

3.4 Hydride shifts in terpinenes . . . 98

3.4.1 Training a terpinene ACE potential . . . 98

3.4.2 Free energy of the terpinene reaction . . . 99

3.4.3 Path integral molecular dynamics of terpinene . . . 100

3.5 Product ratios in indanes . . . 103

3.5.1 Retraining an ACE potential of indane . . . 103

3.5.2 Uphill dynamics of indane . . . 103

3.5.3 Path integral molecular dynamics of indane . . . 106

3.6 Bifurcating surfaces in sesquiterpenes . . . 108

3.6.1 Training an ACE potential on sesquiterpene . . . 109

3.6.2 Product ratios of sesquiterpene . . . 111

3.7 Conclusions . . . 112

4 Umbrella Sampling Method Development **115**

4.1 Summary . . . 116

4.2 Quantitative measures of window sampling . . . 117

4.2.1 US bin metrics . . . 117
4.2.2 Quantitative overlap of windows . . . 119
4.2.3 Effect of κ on sampling . . . 124
4.3 Convergence of window distributions . . . 127
4.3.1 Convergence of fitted Gaussians . . . 127
4.3.2 Convergence of windows to fitted Gaussians . . . 129
4.4 Free energy dependence on overlap . . . 130
4.4.1 Free energy dependence using analytic systems . . . 131
4.4.1.1 The $U(r) = r^2$ model . . . 131
4.4.1.2 The $U(r) = r^4 - 4r^2 + 4$ model . . . 136
4.4.2 Free energy dependence using MLPs and MD . . . 138
4.4.2.1 The S_N2 test system . . . 138
4.4.2.2 The isoindene test system . . . 143
4.4.2.3 The Aβ42 protofibril test system . . . 146
4.5 Conclusions . . . 147

5 *adaptiveus* Python Module and Adaptive US 149
5.1 Summary . . . 149
5.2 *adaptiveus* Python module . . . 150
5.3 Adaptive US scheme . . . 153
5.3.1 Automatic reference value selection . . . 155
5.3.2 Automatic κ value selection . . . 161
5.4 Conclusions . . . 166

6 Conclusions 168
6.1 Thesis overview . . . 168
6.2 Future outlook . . . 172

Bibliography 174

A Appendices 188
A.1 Molecular dynamics integrators and thermostats . . . 188
A.1.1 Verlet algorithm . . . 188

A.1.2 Velocity Verlet algorithm . . . 189
A.1.3 Berendsen temperature coupling . . . 190
A.1.4 Velocity-rescaling temperature coupling . . . 191
A.1.5 Langevin dynamics . . . 191
A.2 Enhanced sampling methods . . . 192
A.2.1 Metadynamics . . . 192
A.2.2 Replica-exchange . . . 193
A.2.3 Additional methods . . . 193
A.2.4 Umbrella integration . . . 194

B Computational methods **195**
B.1 Methods for Chapter 2 . . . 195
B.2 Methods for Chapter 3 . . . 197
B.3 Methods for Chapter 4 . . . 198
B.4 Methods for Chapter 5 . . . 198

List of Abbreviations

Abbreviation	Definition
ACE	Atomic cluster expansion
ACSF	Atomic-centred symmetry functions
AIMD	Ab initio molecular dynamics
AL	Active learning
ASE	Atomic simulation environment
BO	Born-Oppenheimer
CC	Coupled cluster
CINEB	Climbing image nudged elastic band
DFT	Density functional theory
DFTB	Density functional based tight binding
EST	Electronic structure theory
FF	Force field
GAP	Gaussian approximation potential
GGA	Generalised gradient approximation
GPR	Gaussian process regression
HF	Hartree-Fock
IRC	Intrinsic reaction coordination
KL	Kullback-Leibler
LDA	Local density approximation

Abbreviation	**Definition**
MAD	Mean absolute deviation
MAE	Mean absolute error
MAED	Max atomic environment difference
MD	Molecular dynamics
MLP	Machine learning potential
MLP-MD	Machine learning potential driven molecular dynamics
MM	Molecular mechanics
MMMD	Molecular mechanics molecular dynamics
MO	Molecular orbital
MP2	Møller-Plesset perturbation theory to 2nd order
NEB	Nudged elastic band
NN	[Artificial] neural network
NNP	Neural network potential
NQEs	Nuclear quantum effects
PES	Potential energy surface
PIMD	Path integral molecular dynamics
PMF	Potential of mean force
PTSB	Post transition state bifurcation
QM	Quantum mechanics

Abbreviation	**Definition**
QRRHO	Quasi rigid rotor harmonic oscillator
RDF	Radial distribution function
RMSD	Root mean squared deviation
RPMD	Ring polymer molecular dynamics
SOAP	Smooth overlap of atomic positions
TS	Transition state
US	Umbrella sampling
VB	Valence bond
WHAM	Weighted histogram analysis method

1

Introduction

Accurate simulation of chemical reactivity offers the potential to better understand reaction mechanisms, design new molecules and explain biological phenomena. However, the use of theoretical models for any arbitrary chemical system has not been achieved. This is due in part to the accuracy of the models and the speed at which the simulations can be performed. Dirac's quote that "The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved" (1929) still rings true nearly 100 years later.[5] However, some progress has been made. Approximate solutions to the Schrödinger equation are implemented in many electronic structure packages, allowing for the energy of a relatively small number of atoms

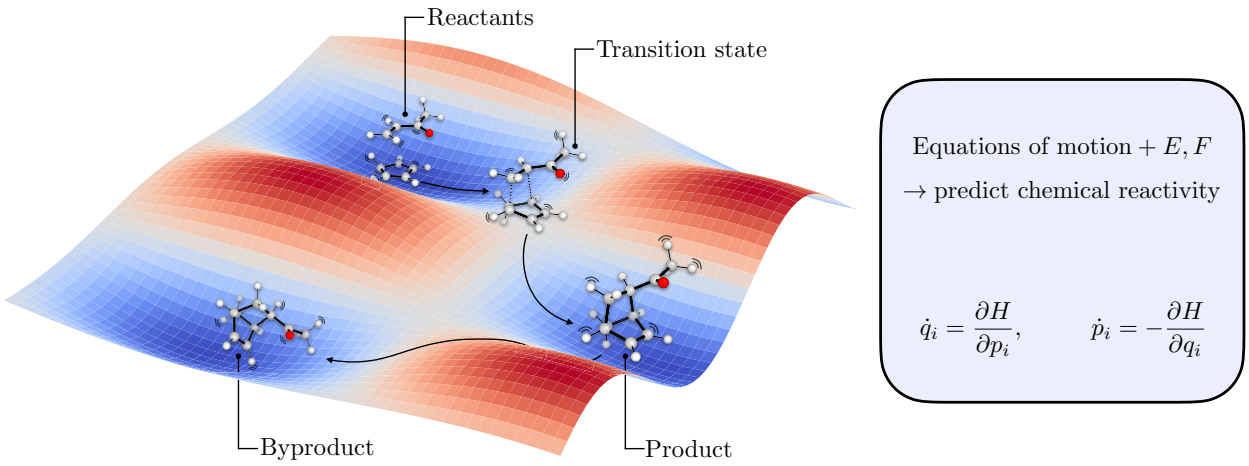


Figure 1.1: Schematic representation of an arbitrary chemical reaction occurring over a potential energy surface. Provided the equations of motion, energy and force are known, dynamics can be performed on this potential energy surface, elucidating reaction mechanics and possible products.

to be accurately computed.[6, 7] On the other end of the spectrum, large-scale simulations of whole proteins can be easily performed for nano- or microseconds using molecular mechanics (MM) packages.[8, 9] A gap exists between these two extrema, however. Many of the systems of interest are too large and complex to simulate both at a satisfying accuracy (chemical accuracy, 1 kcal mol^{-1}) and efficiency $\mathcal{O}$(hours).

Since the pioneering work of Doren et al., who used neural networks to describe the potential energy surface (PES) of CO adsorbed onto Ni, machine learning potentials (MLPs) have emerged as a promising solution to the accuracy and efficiency problem.[10] Through learning the 'ground truth' PES of a region of chemical space implicitly, full calculation of the Hamiltonian can be bypassed. This, in theory, can lead to an accuracy equal with the ground truth method, but at speeds much faster than traditional electronic structure methods.

Even with fast evaluations of energies and forces from MLPs, reactions taking on the order of $> \mu$s require enhanced sampling techniques, used to accelerate the exploration of phase space. One such example is umbrella sampling (US), used to generate free energy profiles between reactants and products.[11] This thesis aims to continue the momentum of MLPs through the development of automated packages and by applying MLPs to a range of (bio)chemically

relevant reactions, using US to generate free energies (**Chapters 2** and **3**). We also discuss methods of quantifying sampling in US, along with presenting packages for the automation of adaptive US (**Chapters 4** and **5**).

1.1 Computational methods for chemical reactivity

Computational and theoretical chemistry has proven hugely successful over the past 100 years. Notable examples illustrating the power of combining theoretical models and high performance computing for chemical prediction have been reported. One of the earliest examples from Houk et al., in 1987, involved the correct prediction of the major E:Z products in ring-opening reactions of cyclobutene derivatives using Hartree-Fock (HF), confirmed by experimental studies thereafter, **Fig. 1.2b**.[12] A similar analysis was then reported by Murakami et al., who successfully predicted the E:Z ratios of silyl and stannyl derivatives of 3-methylcyclobutene using the density functional theory (DFT) hybrid functional B3LYP.[13, 14]

Computational chemistry has also played a key role in the analysis of catalytic processes, including the design of catalysts.[15–19] For example, Connon et al. designed an epoxide-activating urea based on similar thiourea catalyst that was known to activate carbonyl compounds.[20] They used B3LYP/6-31+G* to assess the stability of hydrogen-bonded complexes in known thiourea catalysts and proposed a novel urea catalyst that was later experimentally shown to be highly active, **Fig. 1.2c**.

Computational models have also provided insight into reaction mechanisms.[22–25] For example, B3LYP/6-31G* free energy calculations were used by Ball et al. to elucidate the mechanism of hydrosilylation of alkynes catalysed by ruthenium, **Fig. 1.2d**.[21] Previous models were not able to rationalise the reaction mechanism and this computational study was able to do so by calculating the relative stability and electron density of different intermediates. Herein, we highlight the history and application of computational models in the context of chemical reactivity, particularly for organic reactions.

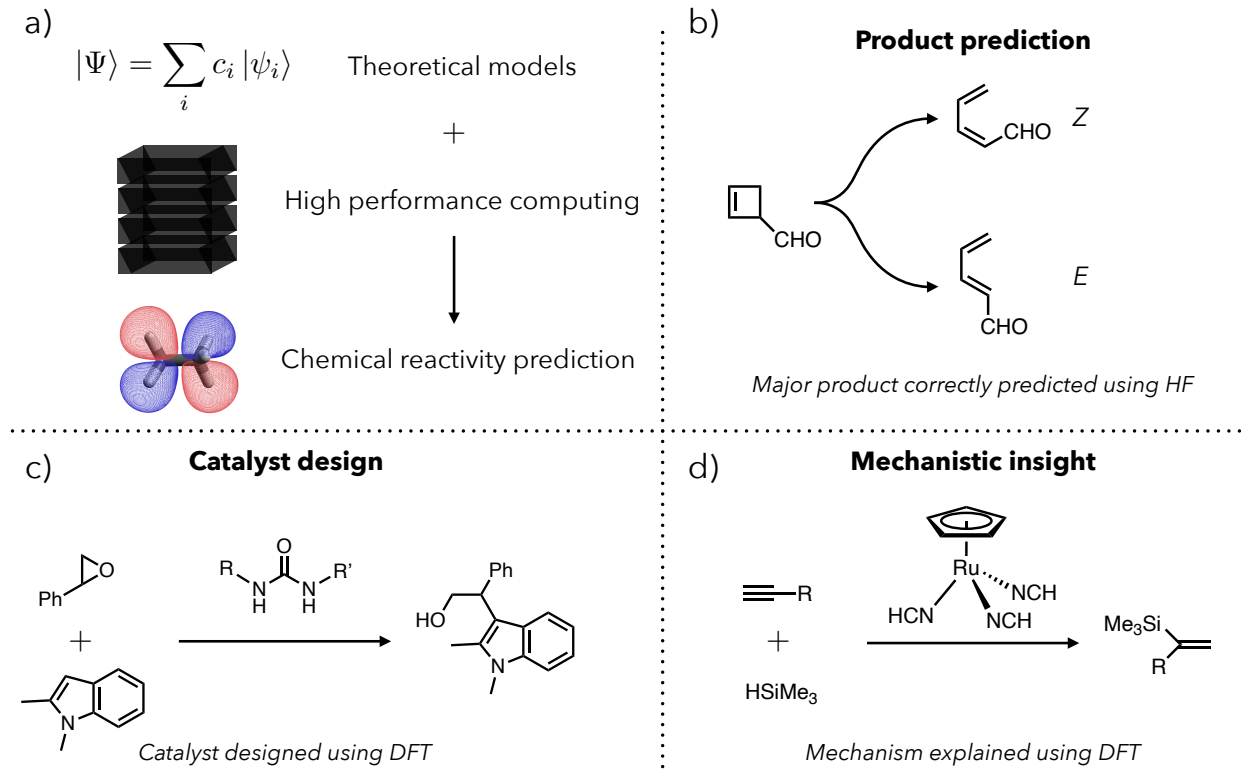


Figure 1.2: Selected examples of computational models for organic systems that were subsequently experimentally validated. These models led to b) the correct prediction of the E:Z major product,[12] c) the design of a novel thiourea catalyst,[20] and d) mechanistic insight on a Ru-catalysed reaction.[21]

1.1.1 A brief history of chemical reactivity prediction

Predicting chemical reactivity is inherently a problem concerned with quantum mechanics (QM). Theoretical and computational chemistry was borne out of the foundations of early 20th century QM. Heitler and London were the first to introduce QM into chemistry, proposing a valence bond (VB) theory for H, He and H_2.[26] Pauling extended the use of VB theory to rationalise the structures of more complex molecules, such as ethane, ethene and trimethylamine.[27] This work led in part to the 1954 Nobel Prize in Chemistry. During the 1940s and 1950s, molecular orbital (MO) theory was further developed, and many of the contributions from Coulson to MO theory are still used in organic chemistry to explain reactivity, **Fig. 1.3a**.[28–30]

Alongside these developments, the exponential increase of computer power has enabled the implementation of many of these methods in high performance computers.[31] Pople pioneered the development of computational methods, making them accessible to a broader community through the widely distributed Gaussian suite of programmes.[7] His contributions were recognised with the 1998 Nobel Prize for Chemistry, alongside Kohn for the development of DFT.[32, 33] DFT emerged as an alternative to wavefunction theory, providing a (partial) solution to the electron correlation problem experienced by previous methods, **Fig. 1.3b**.[34, 35]

Alongside QM methods, classical approaches have played a key role in understanding the properties of molecules, **Fig. 1.3c**. These methods have facilitated fast computation of different properties where the accuracy of QM was not required.[36] One of the earliest MD simulations of several hundred interacting classical particles was carried out by Wainwright and Alder in 1959, where the trajectories of hard-sphere (square well potential) particles were obtained for a few thousand collisions.[37] In 1964, an important step was made where Rahman conducted an MD simulation of liquid argon with a soft sphere potential (Lennard-Jones potential).[38] At the same time, more complex molecules were studied using time-independent molecular mechanics, such as the calculation of the strain energy of cycloalkanes by Wiberg in 1965.[39]

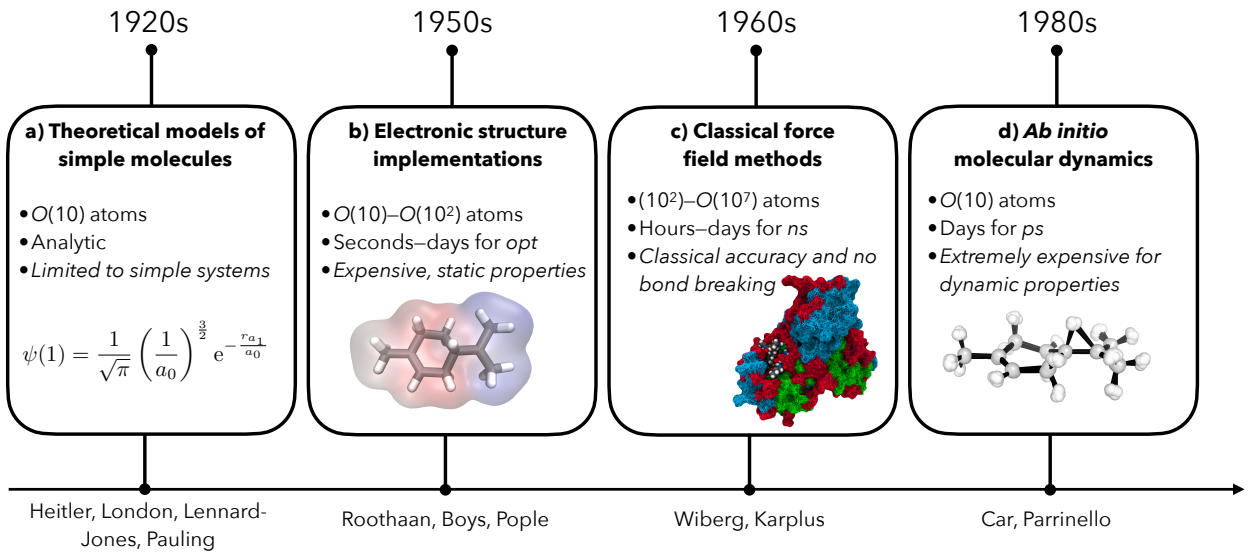


Figure 1.3: Timeline of developments in computational chemistry for modelling molecular systems. The typical number of atoms, computational cost and main limitations are given for each method. Selected pioneers in each field are shown, but the total number of contributors is vast.

Since then, there have been many MD studies carried out on larger systems, such as proteins and nucleic acids, which have been simulated for biological timescales.[40] Similarly, conformational searches of small molecules can be rapidly carried out, e.g., an alanine dipeptide in water.[41] One limitation of these force field (FF) methods is that, other than a few exceptions, they are not reactive, and so cannot be applied to study bond breaking and forming processes.[42]

As computer processing power increased, more expensive calculations could also be run. One relevant example to this work is *ab initio* molecular dynamics (AIMD) simulations, whereby dynamics are propagated using forces computed "on the fly" from electronic structure calculations, **Fig. 1.3d**.[43, 44] AIMD enables one to compute free energies accounting for anharmonic effects and explore dynamic processes, where static calculations do not.[45] Parrinello and Car pioneered AIMD in their 1985 paper, unifying classical MD and Born-Oppenhimer electronic structure calculations.[46] They tested the methodology on a dynamical annealing run of eight atoms of Si, measuring the time evolution of a number of observables, including total energy per atom and lattice constant.

AIMD has been used to model chemical reactivity in a variety of systems, including elu-

cidation of the S_N1/S_N2 mechanism of glycosylation[47] in different solvents, reactivity of organometallic complexes[48], time-dependent dynamics in condensed matter systems[49] and the simulation of liquids such as ions in water.[50] Despite its promise, AIMD is expensive and inaccessible when considering either large systems ($\gtrapprox \mathcal{O}(100)$ atoms) or when high levels of theory are required (e.g., MP2 or coupled cluster with large basis sets).

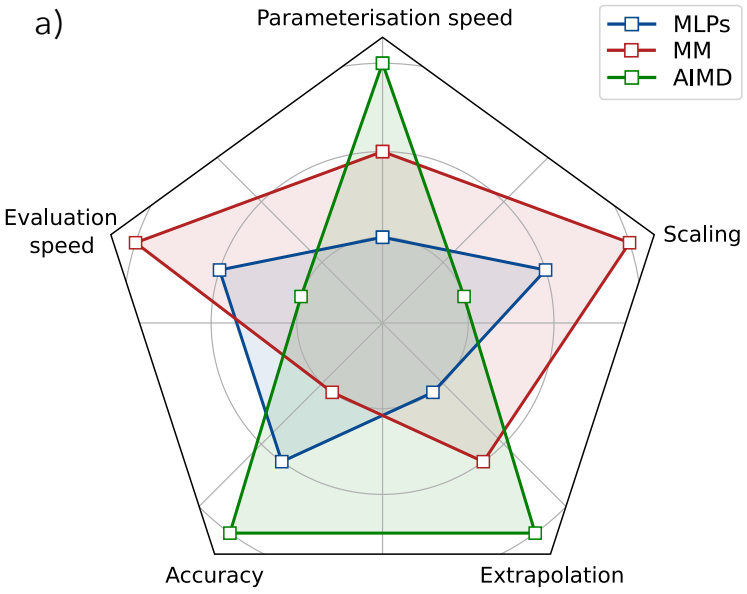
1.2 Machine learning potentials

To reach the timescales and accuracy required for the study of chemical reactions, a paradigm shift is required. FF-based methods are efficient and can easily reach the ns or μs timescales.[8, 9]. However, they are often unsuccessful at describing reactivity to a sufficient accuracy, e.g., chemical accuracy.[42] On the other hand, accurate *ab initio* methods are expensive[6, 7] and static calculations are unable to account for anharmonic effects in many cases.[51] AIMD can offer a solution to this issue but at the cost of either limited sampling or high computational expense. Interatomic MLPs have emerged as a powerful alternative to AIMD, enabling one to compute energies and forces with *ab initio* accuracy while approaching the efficiency of MM methods.

In **Fig. 1.4**, we present a qualitative comparison of MLPs to MM and AIMD. From the radar plot, we can see that MLPs often do not outperform other models in any particular category, but their advantage comes from lying between the expensive AIMD and inaccurate MM. Two areas that need improvement are training, where more efficient and automated methods are required, and extrapolation, so that predictions can be made outside a known region. The former is addressed in **Chapter 2**, where we utilise PhysNet (detailed in **Section 1.2.2.1**) and the Gaussian approximation potential (GAP) (detailed in **Section 1.2.3.1**) to develop an automated training strategy.

MLPs have been used to study the physical and chemical properties of a range of systems, including the free energy from S_N2 dynamics[58], the time gap in percyclic reactions[59], potential energies of organic molecules using neural networks[60], lithium intercalation in

a)



b)

Technique	Method	Scaling
MLPs	ACE[52]	$\mathcal{O}(N^1)$ (> elements)
	GAP[53]	$\mathcal{O}(N^1) \rightarrow \mathcal{O}(N^2)$
	PhysNet[54]	$\mathcal{O}(N^1)$ + encoding
MM	GROMACS[8]	$\approx \mathcal{O}(N^1)$
AIMD	DFT[55]	$\mathcal{O}(N^3) \rightarrow \mathcal{O}(N^4)$
	MP2[56]	$\mathcal{O}(N^4) \rightarrow \mathcal{O}(N^5)$
	CC[57]	$\mathcal{O}(N^7) \rightarrow \mathcal{O}(N^8)$

Figure 1.4: Comparisons of different methods for performing molecular dynamics simulations. a) Qualitative schematic representation of inputs and outputs on an arbitrary scale (with a higher value indicating better performance). b) Approximate scaling of specific methods for evaluating dynamics. Note that exact scaling depends on a number of factors, including approximations, implementations, computer hardware, and hence these values are only approximate.

carbon nanotubes[61], conformers of a range of organic molecules[52] and protein-ligand binding.[62] In **Chapter 3**, we study an array of terpene organic reactions (listed in **Section 1.3**) using the atomic cluster expansion (ACE) potential (detailed in **Section 1.2.3.2**).

1.2.1 Machine learning potentials preamble

In general, the goal of an interatomic MLP is to map the nuclear coordinates, $\mathbf{R}$, and nuclear charge, $\mathbf{Z}$, to a quantum mechanical observable, such as energy, E, through

$$f(\mathbf{R}, \mathbf{Z}) \rightarrow E(\mathbf{R}, \mathbf{Z}). \tag{1.1}$$

If the potentials are used for dynamics, then force, F, is a desirable property, calculated from the derivative of energy, $-\partial E/\partial R$. Accurate prediction of E and F enable molecular dynamics (MD) simulations to be performed and therefore ensemble properties to be calculated, such as free energy. Crucially, because MLPs are trained on QM data, bond breaking/forming events can be modelled. They are typically significantly more efficient per single point calculation than the corresponding ground truth method, as the Schrödinger equation does not have to be solved explicitly.

One assumption often made with MLPs is the locality of representations; it is assumed that the local environment within some cutoff, r_{cut}, is sufficient to accurately predict the physical properties of the system.[63] While there are nonlocal systems that do not obey this principle,[64] in general, many of the molecular systems of interest can be modelled under this assumption.[65] As such, the total energy of a system of N atoms can be written as a sum of the site energies,

$$E = \sum_{i}^{N} E_i. \tag{1.2}$$

The parameters used to weight the functions in an MLP, such as the coefficients in a linear expansion, are typically optimised using a loss function, $\mathcal{L}$, of the following form, **Eqn. 1.3**.

$$\mathcal{L} = w_E f(E - E^{\text{ref}}) + w_F f(F - F^{\text{ref}}) + R \tag{1.3}$$

Here, w weights the contributions of energy, E, and force, F. The functions, $f(x)$, take the difference between the predicted value and the 'ground truth' reference value, and may be linear, quadratic or otherwise. Finally, R is a regularisation function, which ensures a smooth function is obtained to avoid overfitting.[53]

1.2.1.1 MLP construction

Broadly speaking, MLPs can be split into three parts: representation, fitting and validation. First, the input data, in many cases Cartesian coordinates and nuclear charges, are converted into mathematical 'representations' that uphold the appropriate symmetries, **Section 1.2.1.2**. Second, a training set is generated from which the MLP learns regions of the PES. This is achieved by choosing a 'ground truth' method to compare the estimated energies and forces from the MLP with those obtained at the, e.g., DFT level, **Section 1.5.1**. Finally, the trained MLP is used to predict physical properties and propagate dynamics, **Section 1.5.2**, which can in turn be used to validate the potential.

1.2.1.2 Representations

An important property of MLPs is that they must be invariant to translation, rotation and permutation, since the Hamiltonian, **Eqn. 1.47**, is invariant to these transformations.[66] Therefore, the choice of representation (how the coordinates and charges are mapped mathematically to the MLP architecture) is an important factor in designing or choosing an appropriate MLP. The work from Behler and Parrinello, in the context of neural network potentials (NNPs), represented atoms through atom-centered symmetry functions (ACSFs), with the radial function given in **Eqn. 1.4**.[67]

$$G_i^{\text{atom, rad}} = \sum_{j=1}^{N} \mathrm{e}^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}) \tag{1.4}$$

This function is a sum of Gaussians dependent on the distance between the atom i and its neighbours j. The Gaussians are controlled by the parameters η and R_s, and are multiplied by the smooth-decaying cutoff function, $f_c(R_{ij})$. A similar ACSF for the angular components is defined in terms of the angle between atoms i, j, k.[67] This approach has maintained popularity, with similar representations used in the ANI[68] and PhysNet NNPs, **Section 1.2.2.1**[54].

An alternative approach to ACSFs developed by Bartok, Kondor and Csányi places Gaussians centered at atom i and all neighbours, j, **Eqn. 1.5**.[66] This representation, called the smooth overlap of atomic positions (SOAP), can be viewed as a three-dimensional representation of the atomic environment,

$$\rho_i(\mathbf{r}) = \sum_j \exp\left[-\frac{|\mathbf{r} - \mathbf{r}_{ij}|^2}{2\sigma^2}\right] f_{\text{cut}}(r_{ij}). \tag{1.5}$$

Here, the density, ρ, between atom i and j within a given cutoff r_{ij} is described. The cutoff function, $f_{\text{cut}}(r_{ij})$, ensures a smooth transition when r_{ij} is reached. SOAP has found use in GAP[69] (**Section 1.2.3.1**), and the SOAP kernel can be shown to be equivalent to the $n = 3$ and $n = 4$ terms in ACE (**Section 1.2.3.2**).[70]

The ACE framework, described in **Section 1.2.3.2**, utilises a linear combination of modified atomic basis functions, comprising a radial, **Eqn. 1.6**, and angular basis that are invariant to translation, rotation and permutation.

$$R_{nl(r)} = \sum_{k=1} c_{nlk} g_k(r), \tag{1.6}$$

where c_{nlk} define expansion coefficients and the radial basis built on the Chebyshev polynomials and cutoff distance, r_c.

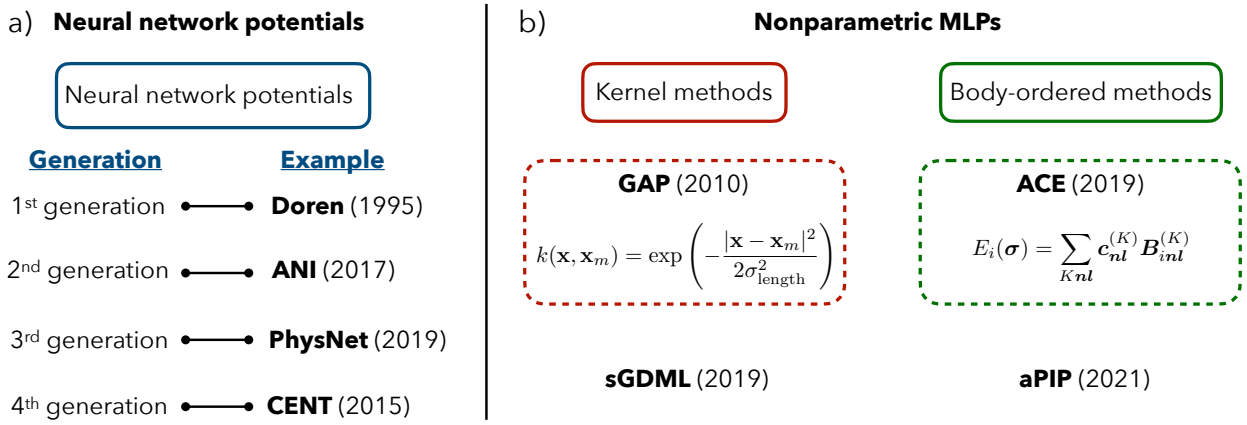


Figure 1.5: Type and timeline of selected MLPs since the first MLP by Doren in 1995.[10]

1.2.2 Neural network potentials

The first use of MLPs in chemical systems can be attributed to the work of Doren in 1995, who used a neural network potential (NNP) to study the adsorption of CO on Ni.[10] Since then, MLPs have grown at an accelerating rate. For example, the number of publications per year containing the phrase 'machine learning potentials' has increased from ≈ 40 in 2010 to over 500 in 2022.[71] An overview of selected NNPs is given in **Fig. 1.5a**, along with some nonparametric MLPs (described in **Section 1.2.3**) in **Fig. 1.5b**.

Initially, all MLPs were constructed using NNPs. These first generation NNPs (1995-2006) were limited to a small number of atoms (low dimensional) as the decomposition of energy into atomic sites, **Eqn. 1.2**, was not yet developed.[72–74]

The seminal work of Behler and Parrinello in 2007 marked the beginning of the second generation of MLPs, where a many-body descriptor of the local atomic neighbour environment was introduced, enabling the total energy to be written as a sum of atomic energies.[67] Early examples of this type of NNPs include PairHDNNP, based on atom pairs as descriptors[75], and PIP-NN that uses permutation invariant polynomials as inputs.[76]

Because of the locality in NNPs, brought on by local functions with cutoff radii, third generation NNPs attempted to augment the short-range energy by considering long-range electrostatics or dispersion.[77] These NNPs are trained on reference charges from the ground truth method as well as energies and forces. Examples include PhysNet, outlined in **Section**

1.2.2.1, trainable on charges and dipoles,[54] and TensorMol.[78] However, third generation NNPs are not commonly used due to the computational cost of computing the electrostatic term explicitly and the assumption that many long-range interactions are screened.[79]

Finally, fourth generation NNPs account for nonlocal effects caused by atoms and charges outside of the cutoff radius. These potentials are not commonly used for the same reasons as the third generation NNPs[79], with examples including CENT[80] and 4G-HDNNP.[81] Other recent NNPs include the second generation, high dimensional NNPs SchNet[82] and ANI.[60, 68] The former utilises learnable descriptors while the latter uses predefined descriptors. In this work, we explore different training strategies using PhysNet, **Chapter 2**, described as follows.

1.2.2.1 The PhysNet architecture

NNPs use linear functions as building blocks for the output vector,

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{1.7}$$

where the gradients $\mathbf{W}$ and intercepts $\mathbf{b}$ are learned by the potential. To create nonlinear functions suitable for describing PESs, the linear functions are stacked, with each embedded function parsed through a nonlinear activation function, σ,

$$\mathbf{y} = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2. \tag{1.8}$$

This process is repeated until the desired number of hidden layers is reached. NNPs are parametric, meaning they have a fixed number of parameters, independent of the training dataset size.[83]

PhysNet is constructed from stacked linear functions and ACSFs, **Eqn. 1.4**.[54] Nuclear charges are embedded into a vector and Cartesian coordinates are parsed through a pairwise radial distribution function with an appropriate cutoff function. These inputs are then parsed

through a number linear layers, namely interaction, residual and output.

The interaction layer contains the physical aspects of the system, i.e., radial basis functions, as well as messaging between an atom and its local environment. The residual block allows skipping of one or more layers, alleviating any vanishing gradient problems caused by deep layers. Finally, the output layer passes the previous blocks through additional residual blocks and computes the output. The final prediction comprises a weighted sum of output vectors, over multiple modules, N_{module}, which share the same structure but different parameters. The sum is weighted and scaled by the learnable parameters, $\mathbf{s}_{Z_i}$ and $\mathbf{c}_{Z_i}$, which depend on the nuclear charge, Z_i, enabling the vastly different ranges of atomic properties for distinct elements to be spanned.

$$\mathbf{y}_i = \mathbf{s}_{Z_i} \circ \left(\sum_{m=1}^{N_{\text{module}}} \mathbf{y}_i^m \right) + \mathbf{c}_{Z_i} \tag{1.9}$$

The loss function in PhysNet can depend on only energies and forces, or can include total charge and dipole as well, **Eqn. 1.10**.

$$\mathcal{L} = w_E |E - E^{\text{ref}}| + \frac{w_F}{3N} \sum_{i=1}^{N} \sum_{\alpha=1}^{3} \left| -\frac{\partial E}{\partial r_{i,\alpha}} - F_{i,\alpha}^{\text{ref}} \right| + \mathcal{L}_{\text{nh}} \tag{1.10}$$

Unlike Gaussian process regression (GPR), **Section 1.2.3.1**, the loss function is not analytically invertable, and thus the number of cycles in the optimisation, or epochs, is an input hyperparameter. Finally, the regularisation term, $\mathcal{L}_{\text{nh}}$, the 'nonhierarchicality penalty' is based on the energies between different layers, detailed in the work by Unke and Meuwly.[54]

1.2.3 Nonparametric machine learning potentials

Before the third generation of NNPs were developed, alternative MLP architectures were being developed, such as the kernel-based GAP method. In kernel methods, the descriptor for a new configuration is used in a kernel *function*, measuring how similar this configuration is to the others in the dataset. Unlike many NNPs, these MLPs are nonparametric and the

number of parameters depends on dataset size.[83] Kernel methods have grown in popularity due to their ability to accurately describe complex PESs with small datasets.[84] Here, we briefly describe GAP, which is based on GPR.

1.2.3.1 Gaussian approximation potential

The GAP framework differs from NNPs in that it is based on GPR, a method for fitting a nonlinear function to a dataset.[53, 69]

The goal of GPR is to construct a smooth, regular function, such as a PES, $y(\mathbf{x})$, which takes in a d-dimensional vector, $\mathbf{x}$, and outputs a single scalar value. To estimate this function, we make N observations of $y(\mathbf{x})$, forming a dataset. Next, we make the approximation that our true function, $y(\mathbf{x})$, can be defined as a linear combination of basis functions,

$$y(\mathbf{x}) \approx \tilde{y}(\mathbf{x}) = \sum_{m=1}^{M} c_m k(\mathbf{x}, \mathbf{x}_m). \tag{1.11}$$

Here, the basis functions, $k(\mathbf{x}, \mathbf{x}_m)$, are placed at various locations along the input space, $\mathbf{x}_m$, and c_m are the weights of these basis functions, **Fig. 1.6**. One example for a basis function is the Gaussian kernel, **Eqn. 1.12**, with hyperparameter σ^2_{length}, which is used to adjust the width of each basis function.

$$k(\mathbf{x}, \mathbf{x}_m) = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_m|^2}{2\sigma^2_{\text{length}}}\right) \tag{1.12}$$

To fit the GPR model to the observed data, the following loss function is minimised with respect to the coefficients, $\mathbf{c} = (c_1, \ldots, c_M)$,

$$\mathcal{L} = \sum_{n=1}^{N} \frac{[y_n - \tilde{y}(\mathbf{x}_n)]^2}{\sigma_n^2} + \sum_{m,m'}^{M} c_m k(\mathbf{x}_m, \mathbf{x}_{m'}) c_{m'}. \tag{1.13}$$

The second term in **Eqn. 1.13** is the Tikhonov regularisation used to ensure a regular (i.e., smooth) function.[85] The loss function can be written in matrix form and solved analytically

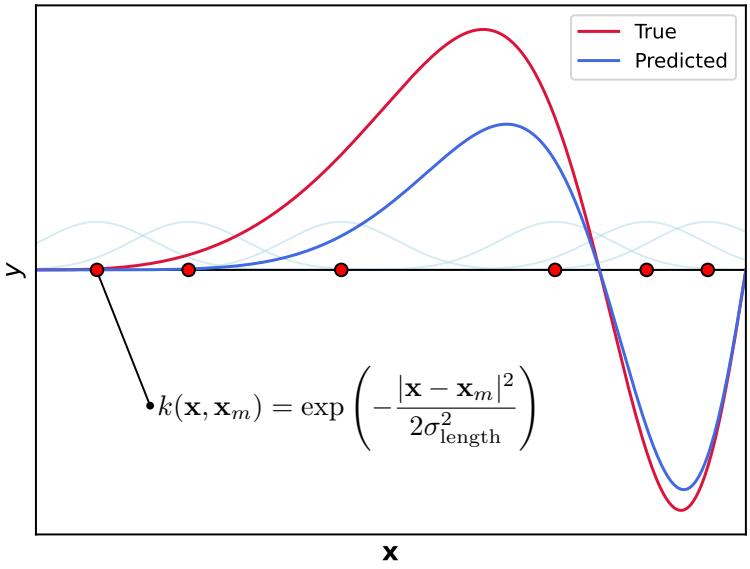


Figure 1.6: GPR for a function fitted using Gaussian kernel basis functions, $k(\mathbf{x}, \mathbf{x}_m)$ placed at locations $\mathbf{x}_m$.

for $\nabla_{\mathbf{c}^T}\mathcal{L} = 0$. The estimation for the function $\tilde{y}(\mathbf{x})$ becomes

$$\tilde{y}(\mathbf{x}) = \mathbf{c}^T\mathbf{k}(\mathbf{x}), \tag{1.14}$$

where $\mathbf{k}(\mathbf{x})$ is the vector of kernel values at $\mathbf{x}$.

1.2.3.2 Atomic cluster expansion

In 2019, the linear scaling ACE potential was introduced by Drautz.[70] ACE is a body-ordered MLP based on a linear combination of invariant atomic basis functions whose coefficients are tunable parameters. The atomic permutationally invariant polynomial (aPIP) MLP is another example of a body-ordered MLP.[86] Introduced only recently, ACE has gained significant popularity due to its accuracy and efficiency compared with existing MLPs.[52] For this reason, ACE is used for molecular reactivity training in **Chapter 3**, and the theory of ACE is detailed as follows.

Traditional FF methods define the energetic contributions for each atom i, as a sum of body-ordered terms, i.e.,

$$E_i = V^{(1)}(\mathbf{r_i}) + \frac{1}{2!}\sum_j V^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \frac{1}{3!}\sum_{jk} V^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \cdots . \tag{1.15}$$

The first term in **Eqn. 1.15** represents the atomic energy of atom i in the absence of interacting neighbours. The second term represents the pair-wise interactions between atoms i and j, and so on. Typically, this sum is truncated after a few terms since most of the chemical information is contained within these terms. An example of this is found in standard FFs, where the bonds in a system are described in terms of harmonic potential, $\sum_{\text{bonds}} k_r(r - r_{\text{eq}})^2$.[52] While this formulation offers a methodical approach to constructing the energy, it suffers from an exponentially growing computational cost as the number of neighbours increase.

ACE begins by defining the energy in a similar way, where body-ordered terms are summed to give the atomic energy for atom, $E_i(\boldsymbol{\sigma}) = E_i(\mathbf{r}_{1i}, \mathbf{r}_{2i}, \ldots, \mathbf{r}_{Ni})$, **Eqn. 1.16**.

$$E_i(\boldsymbol{\sigma}) = \sum_j \sum_v c_v^{(1)} \phi_v(\mathbf{r}_{ji}) + \frac{1}{2!}\sum_{j_1 j_2}\sum_{v_1 v_2} c_{v_1 v_2}^{(2)} \phi_{v_1}(\mathbf{r}_{j_1 i})\phi_{v_2}(\mathbf{r}_{j_2 i}) + \cdots . \tag{1.16}$$

Each term sums over the neighbours j in a cluster containing K bonds and over v single-bond basis functions, $\phi_v(\mathbf{r}_{ji})$. The basis functions are weighted by the expansion coefficients, $c_v^{(K)}$, which are optimised during training. However, this formulation results in N_c^K scaling, where there are N neighbours within cutoff c for each body-ordered term, K, **Fig. 1.7a**. To overcome this exponential scaling, a new atomic basis, A_{iv}, is introduced,

$$A_{iv} = \langle \rho_i | \phi_v \rangle = \sum_j \phi_v(\mathbf{r}_{ji}), \tag{1.17}$$

defined as the projection of the atomic density, $\rho_i = \sum_j \delta(\mathbf{r} - \mathbf{r}_{ji})$, onto the single-bond basis functions, $\phi_v(\mathbf{r}_{ji})$. This sum contains N terms, resulting in an evaluation cost that is linear with the number of atoms. Redefining the energy in terms of this new basis, we get

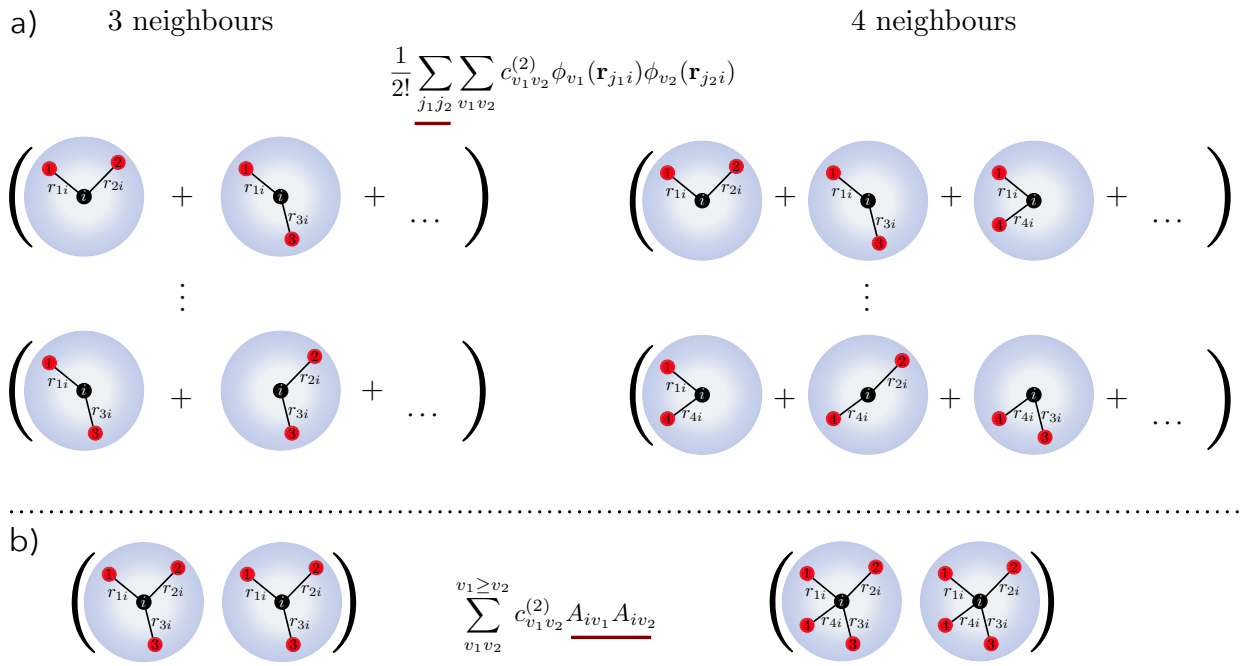


Figure 1.7: a) Second body-ordered term in ACE for three and four neighbours, requiring all combinations of pairs to be computed separately, scaling as N_c^K, **Eqn. 1.16**. b) Second body-ordered term after atomic basis functions are expressed as density projections on the single-bond basis functions, the number of terms required scales as N_c, **Eqn. 1.18**.

$$E_i(\boldsymbol{\sigma}) = \sum_v c^{(1)}_v A_{iv} + \sum_{v_1 v_2}^{v_1 \geq v_2} c^{(2)}_{v_1 v_2} A_{i v_1} A_{i v_2} + \cdots . \tag{1.18}$$

Calculating the atomic bases first and then multiplying them together, instead of explicitly computing all the possible combinations of single-bond basis functions, results in linear scaling in the number of atoms, **Fig. 1.7b**. Note this is for an elemental material, but can easily be extended to multielemental systems through the inclusion of the index μ_i, the atomic species.[70]

The atomic basis, A_{iv}, allows for linear scaling with the number of atoms, but does not imply any particular choice for the form of the functions. To maintain a physical basis to the representation of the atomic energies, the single-bond basis functions comprising radial and angular components are used,

$$\phi_v(\mathbf{r}) = \sqrt{4\pi} R_{nl}(r) Y_l^m(\hat{\mathbf{r}}). \tag{1.19}$$

Here, $v = (nlm)$, analogous to the quantum numbers comprising the atomic basis functions. The radial functions, R_{nl}, depend only on the distance between two atoms.

For ACE, the radial basis is build on the complete set of orthogonal Chebyshev polynomials of the first kind T_k containing a cutoff function to ensure interactions vanish outside r_c, **Eqn. 1.20**.

$$g_k(r) = \frac{1}{4}[1 - T_{k-1}(x)][1 + \cos(\pi r/r_c)] \tag{1.20}$$

These functions, $g_k(r)$, are combined as a linear combination with expansion coefficients, c_{nlk}, to construct the radial basis functions, R_{nl} (except $R_{00}(r) = 1$ required for completeness and $R_{0l}(r) = 0$ for $l > 0$ since this depends on bond angles and not distance).

To transform the spherical harmonics, Y_l^m, to a rotationally invariant basis, an approach similar to the coupling of angular momenta is used.[70] Overall, ACE can be written as a body-ordered expansion of terms invariant to permutation, translation and rotation, $\boldsymbol{B}_{\boldsymbol{inl}}^{(K)}$, **Eqn. 1.21**.

$$E_i(\boldsymbol{\sigma}) = \sum_n c_n^{(1)} B_{in}^{(1)} + {\sum_{n_1 n_2 l}}' c_{n_1 n_2 l}^{(2)} B_{i n_1 n_2 l}^{(2)} + {\sum_{\substack{n_1 n_2 n_2 \\ l_1 l_2 l_3}}}' c_{\substack{n_1 n_2 n_3 \\ l_1 l_2 l_3}}^{(3)} B_{\substack{i n_1 n_2 n_3 (3) \\ l_1 l_2 l_3}} + \dots \ , \tag{1.21}$$

$$= \sum_{K\boldsymbol{nl}} \boldsymbol{c}_{\boldsymbol{nl}}^{(K)} \boldsymbol{B}_{\boldsymbol{inl}}^{(K)}. \tag{1.22}$$

The prime over the summations indicate that only ordered sets of indices are considered.

Since the sum comprises an infinite expansion in **Eqn. 1.21**, a decision on where to truncate the basis set must be made. Truncation can occur in the both the body order (i.e., the number of terms in the sum over K) and the maximum polynomial degree for each term in the K sum. In practice, the truncation is a balance between the desired accuracy and computational expense, which is dependent on the system. ACE training scales non-linearly

with the number of elements, and so typically a five body potential is used for three or fewer elements, while a four body potential is used for four or more elements.[2, 52]. As for the maximum polynomial degree, these are adjusted against fit and accuracy, and typically lie between 8 and 20 depending on the system and body order term.[52]

The coefficients in **Eqn. 1.22**, are optimised by minimising the loss function, $\mathcal{L}(\mathbf{c})$, with respective to both energy and force, **Eqn. 1.23**.

$$\mathcal{L}(\mathbf{c}) = \sum_n w_n^E (E_n(\mathbf{c}) - E_n^{\text{ref}})^2 + \sum_n w_n^F (F_n(\mathbf{c}) - F_n^{\text{ref}})^2 \tag{1.23}$$

The weights, w_n, can be tuned towards either energy or force. The loss function can be regularised, typically using Tychonov regularisation, ensuring a smooth potential.[52] Once an ACE potential has been trained, dynamics can be performed by calculating the forces from $\boldsymbol{F}_k = -\nabla_k E$.

1.2.4 Comparative analysis of PhysNet, GAP and ACE

While there are many interatomic potentials available to date, this work focuses on PhysNet, GAP and ACE. PhysNet has been applied to US of an S_N2 reaction along the $r = d_{C-Br} - d_{C-Cl}$ reaction coordinate. While an accuracy of <0.1 kcal mol^{-1} was obtained, 300,000 MP2 single points were required to train the potential. GAP has traditionally been applied to materials, for example a lithium and carbon nanotube system, where atoms were randomly placed in distorted graphite, graphene and amorphous carbon structures, achieving MAEs of $\approx$ 3 kcal mol^{-1}.[61] ACE has been used on molecular systems, such as the rMD17 dataset, where the energies and forces of 10 small organic molecules were predicted with an average MAE of 0.32 meV/atom energy and 17.4 meV/Å force.[52]

PhysNet, GAP and ACE have been compared on the same training data set.[52] Energy and force predictions on the MD17 small organic molecule DFT set suggested that ACE carries the best overall average error in both force and energy while requiring the fewest configurations to train the potential, **Fig. 1.8**.

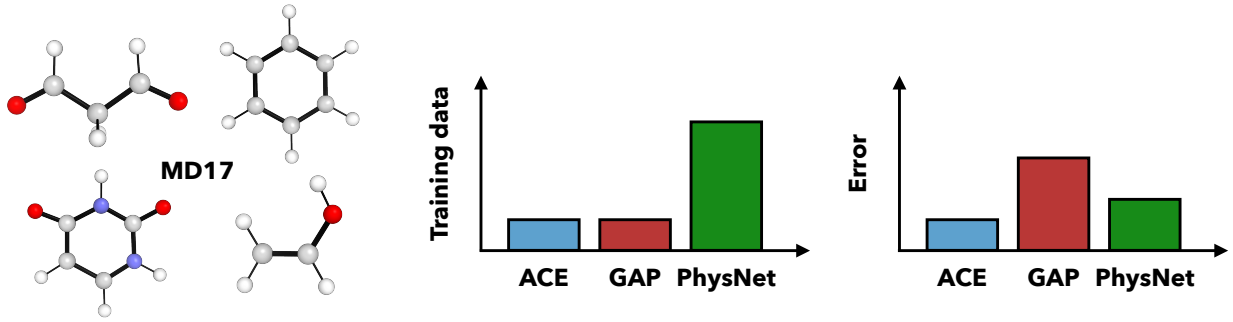


Figure 1.8: Qualitative comparisons of ACE, GAP and PhysNet MLPs based on the MD17 analysis carried out by Csányi et al.[52] Training data represents the number of configurations required to train the potentials and error represents the MAE of energy and force predictions by the potentials. The MD17 training set comprises configurations of 10 small organic molecules (four shown) sampled using DFT AIMD in a vacuum.

1.2.5 Machine learning potential software

Along with the development of the mathematical architecture for these MLPs, software plays a significant role in their use within the wider community. Today, several software exist for training MLPs, with requirements and usability varying between architectures. For example, GAP is available in GitHub and requires the use of QUIP (interfaced to the quippy wrapper) codes for training and the atomic simulation environment (ASE) for MD.[87, 88] A Python script, accompanied by Python libraries, is available from the original publication of PhysNet.[54] Again, dynamics can be performed by interfacing PhysNet to ASE. Finally, ACE is implemented in the ACEsuit, written in Julia[89] and is also available on GitHub.[90]

Other packages available for MLP training include the DeepMD-kit package, which combines a deep NNP interfaced to LAMMPS and i-PI, enabling the training, validation and application of the potential to MD.[91] MLatom, a Fortran-based package, implements a nonparametric kernel-based method supporting training, validation and sampling.[92] PES-Learn is an open source MLP training package that can be used to generate potentials trained on either neural network or Gaussian process models.[93] The sGDML Python package includes an implementation of the sGDML model that is interfaced to ASE, allowing dynamics to be performed.[94] PiNN is another Python package that implements both PiNet and the Behler-Parrinello NNP, enabling the training and running of MLP dynamics through ASE.[95] Finally, TorchANI provides an implementation of the corresponding ANI NNP for

training and simulation through ASE.[96] In **Chapter 2**, we present software for the automation of training and running US for the GAP and ACE potentials using active learning.

1.3 Reactions of terpenes and related molecules

One of the aims of this thesis is to introduce efficient frameworks to study complex chemical reactions of (bio)chemical relevance. One of the most complex organic reactions performed by enzymes is the synthesis of terpenes, where relatively simple acyclic precursors undergo complex transformations that give rise to construction of thousands of different natural products.[97] Many of these systems have been only partially studied by either experimental or computational techniques, and questions remain about their mechanism, relative stability and selectivity.

Terpenes are a class of naturally occurring secondary metabolites, comprising multiple C_5H_8 isoprene units. Terpenes derived from two isoprene units ($C_{10}H_{16}$) belong to the monoterpene family; common examples include menthol and limonene.[98] Sesquiterpenes are a class of terpenes that are built from three isoprene units ($C_{15}H_{24}$) and are used by some plants as antimicrobial and anti-insecticidal agents.[99, 100] Their biosysnthesis by enzymes is complex, involving multiple competing reaction pathways.[101] These pathways include cyclications and alkyl or hydride shifts.[102] Therefore, expanding our understanding of their underlying reaction mechanisms can help provide insight into how enzymes synthesise these natural products.

Terpenes have been previously studied computationally to explore the feasibility of proposed reaction sequences, such as the formation of amorphadiene and amorphene sesquiterpenes.[101] Generally, these studies have often been performed at the DFT level and focused on predicting energies of reaction, either electronic or static free energy. These studies have provided insight into the feasibility of particular pathways and existence of competing pathways.[103] However, there are still unanswered questions, including the existence of post transition state bifurcations, product ratios and free energies, which require the inclusion of

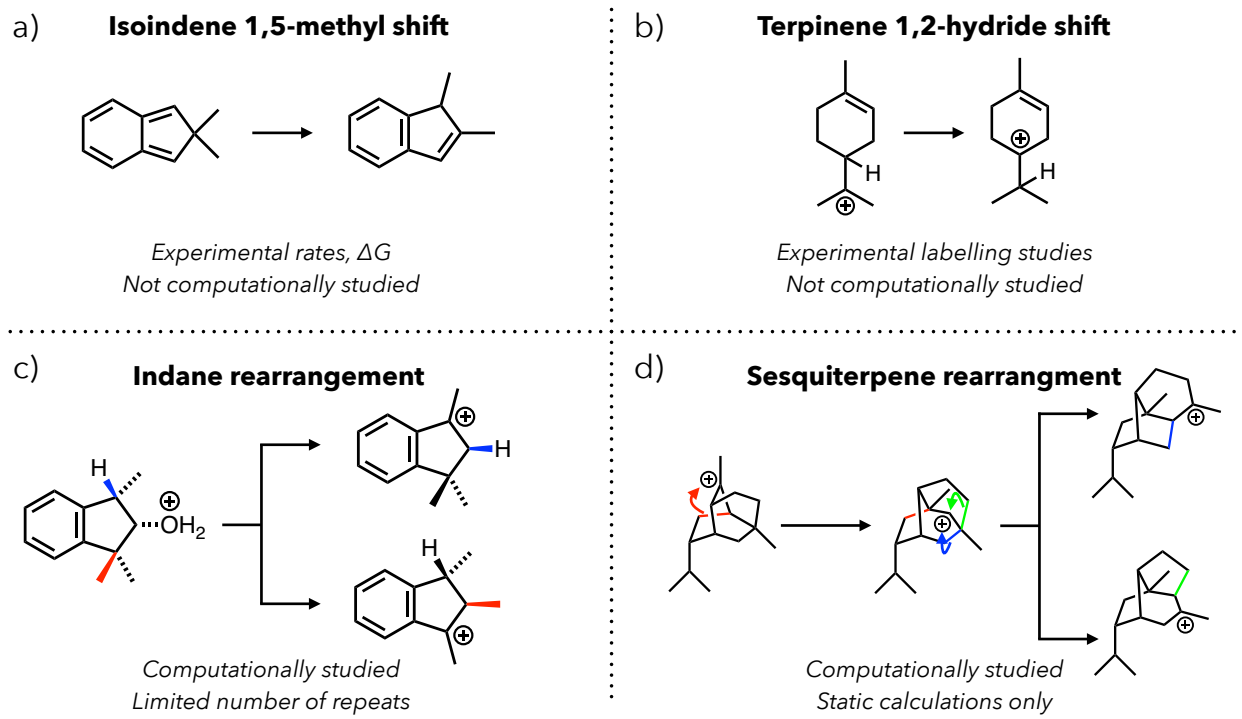


Figure 1.9: Organic reactions studied in this work, highlighting reactants, their rearrangement, and given products. The scope of previous experimental and computational studies carried out on each reaction is described.

dynamic effects.[104]

Recent studies have included such effects by employing AIMD methods; however, they are expensive, which limits such studies in their extent of sampling.[105] In this thesis, we combined US (described in **Section 1.4**) with MLPs to provide a more accurate and quantitative understanding of the competing reaction pathways in terpenes, which would otherwise be prohibitively expensive with conventional techniques. Below, we briefly describe the four terpene systems studied throughout this work.

1.3.1 Isoindenes

The first system studied in this work is the thermally activated 1,5-methyl shift of the isoindene, 2,2-dimethylisoindene, to 1,2-dimethylindene, **Fig. 1.9a**.[106, 107] Rolison et al. experimentally measured the relative migratory aptitudes of various R groups versus methyl for the 1,5-alkyl shift of isoindene. The question as to whether the reaction proceeded via a concerted pericyclic mechanism or a fragmented dissociative radical mechanism was ad-

dressed. From their results, they concluded that the mechanism had some radical character but was largely concerted. This was suggested from the minimal amounts of competing radical products observed as well as the slower than expected rate of migration for benzyl, which has strong radical stabilising capacity. For the 1,5-alkyl shift where R=Me, they calculated the enthalpies of formation for the reactant and radical intermediate. They predicted an activation energy of 41 kcal mol^{-1}, over 10 kcal mol^{-1} higher than the experimentally observed barrier, suggesting a concerted pathway.

In our work, we study the 1,5-methyl migration with the aim of providing further evidence on the mechanism. As well as providing insight on this mechanism, the system is ideal to study using MLPs since there is reaction rate data at various temperatures for validation of our potential.[106] The reaction is thermally activated and was carried out experimentally in pentane, thus making its study computationally more straight forward due to the single reactant and non-polar solvent. Moreover, to our knowledge, no computational work has been carried out on this 1,5-methyl migration reaction to confirm the concerted reaction mechanism.

1.3.2 Terpinenes

The second system investigated is part of the synthesis of a γ-terpinene, which is suggested to be catalysed by the enzyme γ-terpinene synthase.[108–110] γ-terpinene is a monoterpene found in plant thyme and is synthesised from a geranyl pyrophosphate. It has been suggested that this reaction can occur via a 1,2- or 1,3-hydride shift, leading to the same final product after protonation, **Fig. 1.9b**.[111, 112] Experimental studies involving tritium labelling have suggested a 1,2-hydride shift, which requires the formation of a tertiary carbocation, rather than the less stable secondary carbocation required by a 1,3 shift. However, this mechanism has not been studied computationally, which leaves questions on the feasibility of this route energetically and whether any nuclear quantum effect may affect the hydride shift.

1.3.3 Indanes

A model of indane, originally studied by Tantillo and Feng using AIMD is considered.[105] This system can undergo two competing shifts depending on the migratory aptitudes of competing R groups to form a carbocation, **Fig. 1.9c**. The authors initiated the reaction by placing additional momentum into the C−O bond involved in the first step leading to the removal of OH_2, i.e., uphill dynamics. This study confirmed the importance of introducing nonstatistical dynamic effects when modelling the reaction, as static approaches failed to reproduce the dynamic product ratios.[103, 113, 114] However, due to the computational cost of AIMD simulations, questions on whether the ratios are fully converged remain.

1.3.4 Sesquiterpenes

Finally, we study a proposed post transition state bifurcation (PTSB) reaction of a sesquiterpene.[104, 115] The biosynthesis of two sesquiterpene natural products has been proposed to occur via a PTSB involving a pupukeanane secondary carbocation that can undergo two different 1,2-alkyl shifts, leading to different products, **Fig. 1.9d**. Through a dyotropic rearrangement, the two products can interconvert. In the DFT study by Tantillo and McCulley, intrinsic reaction coordination (IRC) calculations were performed to explore the two proposed PTSB pathways.[104] Previous work suggested that linking multiple products through IRCs at different levels of theory is an indication of a PTSB.[116, 117] Tantillo and McCulley linked multiple products starting from similar TSs through IRC calculations, thus suggesting the existence of a PTSB. However, the authors note that reactive MD could be used to more thoroughly test this hypothesis.

1.4 Umbrella sampling

For chemical reactions at room temperature with a barrier of $\mathcal{O}(10\,\text{kcal mol}^{-1})$, reactions typically occur with a half life between μs and hours.[106] Even in the fastest reactions occurring in a μs, current AIMD or MLP-MD methods cannot reach more than a ns. This disconnect

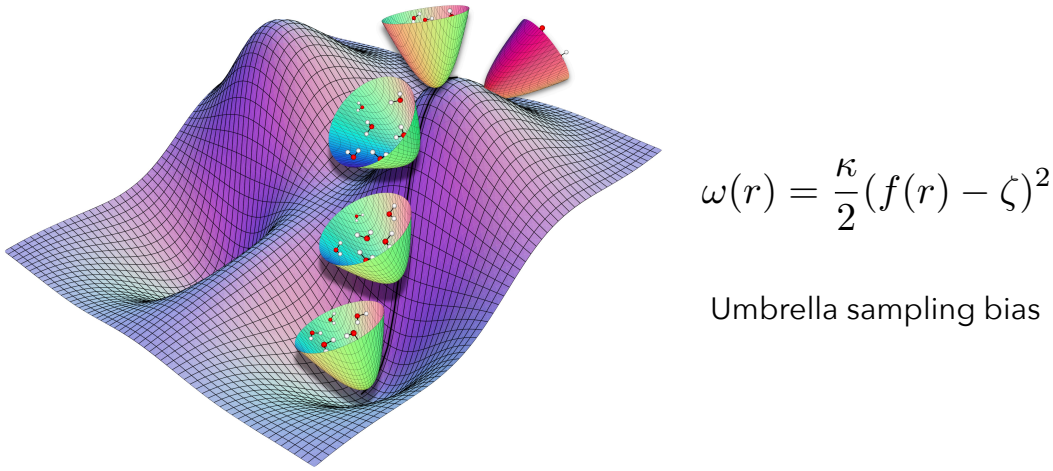


Figure 1.10: Graphical representation of an US simulation where the potential is restrained by an arbitrary harmonic function, $\omega(r)$, along the reaction coordinate, $f(r)$.

in reaction rate versus computational sampling capacity calls for enhanced sampling methods. These methods accelerate the sampling on the PES in the region of interest.[118] While there are many enhanced sampling methods, such as metadynamics and replica-exchange, we primarily focus on the umbrella sampling (US) methodology in this work, with other popular methods discussed in **Appendix A.2**. In this work, we utilise US extensively; first as a method for calculating MLP-driven free energies (**Chapters 2** and **3**); and second, as a basis for developing metrics assessing the influence of overlap on free energy (**Chapter 4**) and developing an automated package for adaptive US (**Chapter 5**).

In US, a bias function is applied along a reaction coordinate to constrain the sampling in the region(s) of interest, **Fig. 1.10** (**Section 1.4.1.1**). Techniques to unbias the resulting distribution, such as the Weighted Histogram Analysis Method (WHAM)[119] (**Section 1.4.1.3**) and Umbrella Integration (UI)[120] (**Appendix A.2.4**), enable free energy profiles to be calculated along a defined reaction coordinate.

US has been widely successful in the MD community.[121–125] US is typically used in large biochemical systems, such as proteins. For instance, US has also been used to investigate the stability of amyloid protofibrils proteins, present in Alzheimer's.[126] An Aβ42 protofilament peptide was pulled away from the remaining peptides under a harmonic bias to generate a binding free energy of the wild type and mutant Aβ protofibrils. We utilise this system in

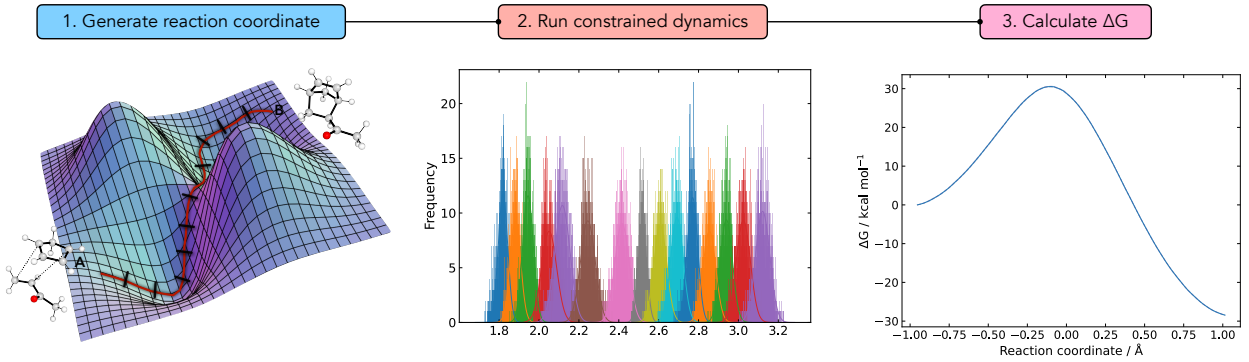


Figure 1.11: Outline of an US simulation in three parts. First a reaction coordinate is defined and a pathway between initial and final points created. Next, MD are performed along the reaction coordinate, constraining each window using a biased potential. Finally, the biased distribution is unbiased using e.g., WHAM or UI.

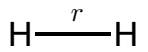
Chapter 4.

As well as large protein systems, smaller molecular systems have also used US for free energy calculations.[127] Since US is independent of the method to compute the energy and force, AIMD can be used in combination with US, which is of particular interest to this work, as we perform reactive US simulations using MLPs. Other work has combined MLPs and US to calculate the ring polymer molecular dynamics (RPMD) rate coefficients for $H + H_2O_2 \longrightarrow OH + H_2O$.[128] The free energy component of the rate coefficient was calculated using UI by running US over the reaction coordinate.

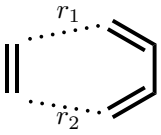
1.4.1 Theoretical basis

The theoretical basis of US is outlined as follows and can broadly be split into three parts, **Fig. 1.11**, as:

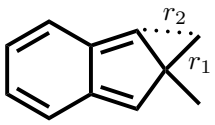
1. Defining and creating a trajectory along a reaction coordinate.
2. Sampling windows along the reaction coordinate.
3. Unbiasing the sampled distribution using an unbiasing method e.g., WHAM.



$f(r) = r$ $[1\ \text{Å}, 3\ \text{Å}]$



$f(r) = \frac{r_1 + r_2}{2}$ $[2\ \text{Å}, 4\ \text{Å}]$



$f(r) = r_1 - r_2$ $[-1\ \text{Å}, 1\ \text{Å}]$

Figure 1.12: Examples of reaction coordinates and their corresponding functions and ranges. The variable r represents a Cartesian distance between two atoms measured in Angstroms, Å.

1.4.1.1 The reaction coordinate

The reaction coordinate *function*, $f(r)$, is user defined and involves a mathematical combination of some or all of the atoms in the system. The following section highlights examples of reaction coordinates and their corresponding ranges, **Fig. 1.12**.

Reactions coordinates may be defined from a 'pulling' simulation, where a system is pulled along a coordinate under a biased potential.[8] For smaller systems, where the electronic energy can be computed and the transition state (TS) is known, reaction coordinates can be generated through an intrinsic reaction coordinate (IRC).[129] This thesis utilises the TS and IRC algorithms available in ORCA.[6] The optTS function in ORCA uses the eigenvector-following (EF) algorithm to find a TS structure.[130] This method uses the second derivative of the PES to walk uphill until the TS is found (confirmed by the presence of one imaginary frequency). The IRC algorithm, by Morokuma et al.,[129] connects a TS with two nearest downhill minima. Initially, the TS is displaced along the eigenmodes of the Hessian to generate a non-zero gradient. The nuclear coordinates then follow a steepest descent path until convergence is reached.

Another approach to generating a reaction coordinate is the nudged elastic band (NEB) method, which only requires the structures of a reactant and product.[131] In NEB, a number

of intermediate images are generated between reactant and product structures. A constrained geometry minimisation is then performed by placing springs between the images and the force orthogonal to the reaction coordinate are followed to move the images closer to the minimum energy pathway. The springs ensure the images are kept at an equal spacing, resulting in a set of evenly spaced images along the minimum energy pathway. A variation on the NEB method is the climbing image nudged elastic band (CINEB) method. This method is identical to NEB with the modification that the forces on the highest energy image are inverted so that the structure is pushed up in energy until a saddle point (i.e., the TS) is reached.

Once a trajectory along the reaction pathway has been created, US dynamics can be performed along this coordinate.

1.4.1.2 Umbrella sampling dynamics

To ensure efficient sampling along a reaction coordinate, a bias is placed on the system during dynamics. This has the effect of constraining the dynamics to sample in the regions corresponding to the reaction coordinate. It is also particularly useful for reactions whose energetic barriers are too high to naturally observe in unbiased dynamics. The form of the bias typically used is harmonic, **Eqn. 1.24**, although in principle it can take any form.[11]

$$\omega(r) = \frac{\kappa}{2}(f(r) - \zeta)^2 \tag{1.24}$$

Here, $\omega(r)$ is the bias (units of energy), κ is the spring constant, which controls the strength of the bias (units of energy per reference units squared), $f(r)$ is the reaction coordinate function (units of e.g., length) and ζ is the reference value for a particular window (units equal to those of $f(r)$). The total potential energy of the biased system is therefore given as

$$E^b(r) = E^u(r) + \omega(r) = E^u(r) + \frac{\kappa}{2}(f(r) - \zeta)^2. \tag{1.25}$$

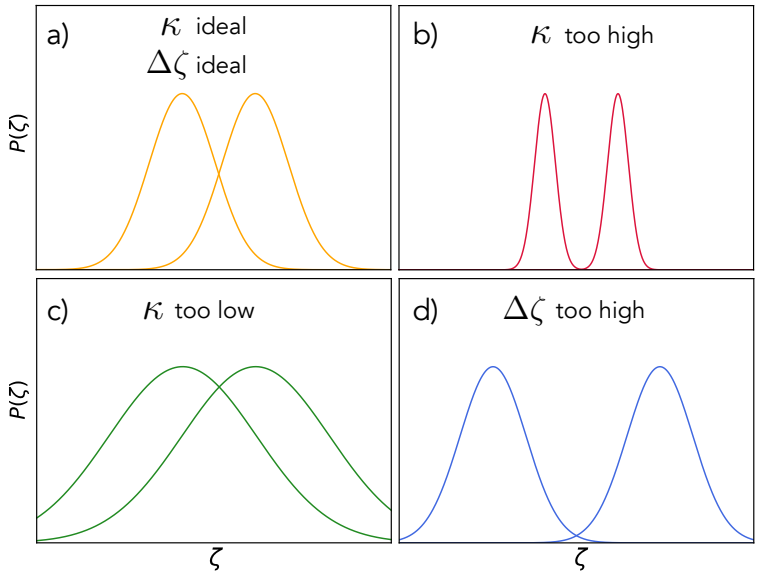


Figure 1.13: The effect on window distributions in a US simulation for different values of κ and ζ.

Here, the b and u superscripts denote the biased and unbiased quantities, respectively. From **Eqn. 1.25**, we can see that the reaction coordinate, $f(r)$, can take any smooth and continuous function of r that is differentiable, as the forces on the atoms are calculated using the negative derivative.

The final step in defining the bias functions is to specify a value for κ and ζ for each window. The choice of κ and ζ can affect the sampling efficiency and overlap between windows, with various cases shown in **Fig. 1.13**. Optimal choice of κ and ζ varies between systems and a method for the automatic selection of these parameters is presented in **Chapter 5**. The distribution resulting from sampling in each window, k, will be roughly Gaussian (with increasing κ), since our bias is harmonic. Our probability distribution therefore goes as $\exp[-\beta\frac{\kappa}{2}(f(r)-\zeta^{(k)})^2]$.

Finally, the sampling length for each window is chosen. This choice encompasses the same considerations as normal MD simulations; long enough to ensure sufficient sampling, but short enough to be computable in a reasonable time frame.

1.4.1.3 The weighted histogram analysis method

Once sampling along the reaction coordinate is complete, we are left with an unphysical biased distribution. Therefore, we require a method to unbias the dynamics, giving us a probability distribution relating to the 'real' system. The weighted histogram analysis method (WHAM) method can be used to unbias our biased distribution and is implemented in *mlp-train*, detailed in **Chapter 2**.

Before discussing WHAM, we will first address the reason behind generating an unbiased distribution along a given reaction coordinate. If we are interested in calculating a non-mechanical property, such as the free energy, we are required to calculate the partition function corresponding to a particular ensemble, e.g., the canonical ensemble (N, V, T). The partition function represents the total number of accessible microscopic states. In the case of the canonical ensemble, the Hamiltonian is not conserved and instead obeys a Boltzmann distribution due to exchange of energy with the surroundings. From the canonical partition function, the Helmholtz free energy, A, can be derived as

$$A(N, V, T) = -kT \ln Q(N, V, T) = -\frac{1}{\beta} \ln Q(N, V, \beta), \tag{1.26}$$

where k is the Boltzmann constant, T is the temperature and β is the inverse temperature.[118] The canonical partition function is defined from statistical mechanics as

$$Q(N, V, T) = \frac{1}{N! h^{3N}} \int \mathrm{dx}\ \mathrm{e}^{-\beta H(\mathrm{x})}. \tag{1.27}$$

Here, N denotes the number of indistinguishable particles, h is Planck's constant, $H(\mathrm{x})$ is the system Hamiltonian and $\mathbf{x}$ represents the phase space coordinates. The phase space distribution function, $P(\mathrm{x})$, corresponding to the canonical ensemble is given as

$$P(\mathrm{x}) = \frac{\mathrm{e}^{-\beta H(\mathrm{x})}}{\int \mathrm{dx}\ \mathrm{e}^{-\beta H(\mathrm{x})}}. \tag{1.28}$$

In US, we wish to express thermodynamic quantities in terms of a reaction coordinate. We can use **Eqn. 1.28** to express the phase space distribution function as an ensemble average in terms of the reaction coordinate values, s_i, **Eqn. 1.29**.

$$P(s_1, ..., s_n) = \frac{(N!h^{3N})^{-1}}{Q(N,V,T)} \int \mathrm{d}^N\mathbf{p}\, \mathrm{d}^N\mathbf{r}\ \mathrm{e}^{-\beta H(\mathbf{p},\mathbf{r})} \prod_{i=1}^{n} \delta(f_i(\mathbf{r}) - s_i) \tag{1.29}$$

Here, we use δ-functions to denote whether our system is at value s_i. For a reaction profile, s should span the whole reaction coordinate and be sufficiently dense to give a smooth probability distribution. The free energy is transformed simply as

$$A(s_1, ..., s_n) = -kT \ln P(s_1, \ldots, s_n). \tag{1.30}$$

For a real system with many interacting particles, calculation of an analytical Hamiltonian, and therefore the partition function, quickly becomes intractable. Therefore, numerical methods to approximate the Hamiltonian are utilised in molecular simulation. A central idea in MD is that we can replace the ensemble average of a property in, e.g., the canonical ensemble,

$$\langle a \rangle = \frac{(N!\lambda^{3N})^{-1}}{Q(N,V,T)} \int_{D(V)} \mathrm{d}\mathbf{r}_1 \cdots \mathrm{d}\mathbf{r}_N\ a(\mathbf{r}_1, \ldots, \mathbf{r}_N) \mathrm{e}^{-\beta U(\mathbf{r}_1,\ldots,\mathbf{r}_N)}, \tag{1.31}$$

with a time average,

$$\langle a \rangle = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau \mathrm{d}t\ a(\mathbf{r}_1, \ldots, \mathbf{r}_N) \equiv \bar{a}. \tag{1.32}$$

Note that in **Eqn. 1.31**, we have integrated out the momentum, which is universally $\mathbf{p}_i^2/2m$ in the Hamiltonian. This leaves us with $U(\mathbf{r}_1, \ldots, \mathbf{r}_N)$ as the energy term and a factor of λ, the thermal wavelength. The integral is over the D-dimensional volume, $D(V)$. The two assumptions we must make in computing the time average of a property in the place of direct

evaluation of the canonical partition function are as follows:

- **Postulate 1** – The time average of a mechanical variable is equal to the ensemble average in the limit $N \to \infty$.
- **Postulate 2** – Given an infinite amount of time, all microstates on the constant energy hypersurface are equally probable, i.e., the system is ergodic.

When generating a distribution through dynamics, we can replace our ensemble average with a time average,

$$P(f(\mathbf{r})) = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau \mathrm{d}t\ \rho[f(\mathbf{r}(t))]. \tag{1.33}$$

MD will, in the ergodic limit, generate the desired distribution in **Eqn. 1.30**. However, this is computationally impractical in almost all cases, and so we resort to using our biased potential from **Eqn. 1.25**. Upon generation of our biased distribution, we must unbias the distribution if we wish to determine the free energy along our reaction coordinate. Next, we present weighted histogram analysis method (WHAM).[119]

⁂

In WHAM, we start by defining our biased probability distribution, $\tilde{P}$, for a particular window, k, and reference, s, for each observed reaction coordinate, q, **Eqn. 1.34**. This probability is defined in a synonymous manner to both the phase space distribution function in **Eqn. 1.29** and the ensemble average, **Eqn. 1.31**.

$$\tilde{P}(q, s^{(k)}) = \frac{\int \mathrm{d}^N\mathbf{r}\ \mathrm{e}^{-\beta U(\mathbf{r})}\mathrm{e}^{-\beta\omega(f(\mathbf{r}),s^{(k)})}\delta(f(\mathbf{r})-q)}{\int \mathrm{d}^N\mathbf{r}\ \mathrm{e}^{-\beta U(\mathbf{r})}\mathrm{e}^{-\beta\omega(f(\mathbf{r}),s^{(k)})}} \tag{1.34}$$

Here, the property that will be calculated from a time average are the instances at which $\delta(f(\mathbf{r}) - q) \neq 0$, under the biased potential, $U(\mathbf{r}) + \omega(f(\mathbf{r}), s^{(k)})$. The denominator in

Eqn. 1.34 is the distribution associated with the biased potential and is denoted herein as $\exp(-\beta A_k)$, where A_k is the free energy for window k.

We define an unbiased distribution for window k, $P_k(q)$ as

$$P_k(q) = \frac{\mathrm{e}^{-\beta U_q}}{\int \mathrm{d}^N\mathbf{r}\ \mathrm{e}^{-\beta U(\mathbf{r})}} = \mathrm{e}^{\beta A_0}\mathrm{e}^{-\beta U_q}, \tag{1.35}$$

where A_0 is the free energy associated with the unbiased potential. We connect U_q to the quantity $\mathbf{r}^*$, a subset of $\mathbf{r}$, through $f(\mathbf{r}^*) = q$. Next we introduce the biased potential and biased free energy through

$$P_k(q) = \mathrm{e}^{\beta A_0}\mathrm{e}^{-\beta U(\mathbf{r}^*)}\mathrm{e}^{-\beta\omega(q,s^{(k)})}\mathrm{e}^{\beta\omega(q,s^{(k)})}\mathrm{e}^{-\beta A_k}\mathrm{e}^{\beta A_k}. \tag{1.36}$$

Our biased distribution contains $f(\mathbf{r})$, so we introduce a Dirac Delta function to **Eqn. 1.36**, which picks out the configurations where reaction coordinate equals q.

$$P_k(q) = \mathrm{e}^{-\beta(A_k-A_0)}\mathrm{e}^{\beta\omega(q,s^{(k)})}\mathrm{e}^{\beta A_k}\int \mathrm{d}^N\mathbf{r}\ \mathrm{e}^{-\beta U(\mathbf{r})}\mathrm{e}^{-\beta\omega(f(\mathbf{r}),s^{(k)})}\delta(f(\mathbf{r})-q) \tag{1.37}$$

We now note that the $\exp(\beta A_k)$ and integral in **Eqn. 1.37** constitute the biased distribution for window k, $\tilde{P}(q, s^{(k)})$ and so

$$P_k(q) = \mathrm{e}^{-\beta(A_k-A_0)}\mathrm{e}^{\beta\omega(q,s^{(k)})}\tilde{P}(q, s^{(k)}). \tag{1.38}$$

We now have an expression for the unbiased distribution of a specific window, including the unbiasing factor. However, we wish to combine multiple windows spanning a range of k. To do this, we create a linear combination of each window distribution,

$$P(q) = \sum_{k=1}^{n} C_k(q) P_k(q), \tag{1.39}$$

$$P(q) = \sum_{k=1}^{n} C_k(q) \mathrm{e}^{-\beta(A_k - A_0)} \mathrm{e}^{\beta\omega(q,s^{(k)})} \tilde{P}(q, s^{(k)}) \tag{1.40}$$

where $C_k(q)$ are the coefficients to be optimised under the constraint $\sum_{k=1}^{n} C_k(q) = 1$. The derivation of these constants can be found in ref [118]; however, we note two assumptions made. The first is that the sampling quality is equal for all windows. The second is that the biased histogram in each window can be approximated by applying the biasing factor to the true distribution, $P(q)$. The second assumption is valid if there is sufficient sampling across all windows. The coefficients are given as

$$C_k = \frac{n_k \mathrm{e}^{\beta A_k} \mathrm{e}^{-\beta\omega(q,s^{(k)})}}{\sum_{j=1}^{n} n_j \mathrm{e}^{\beta A_j} \mathrm{e}^{-\beta\omega(q,s^{(j)})}}. \tag{1.41}$$

Inserting the expression for C_k into the expression for $P(q)$ gives

$$P(q) = \frac{\sum_{k=1}^{n} n_k \mathrm{e}^{\beta A_k} \mathrm{e}^{-\beta\omega(q,s^{(k)})} \mathrm{e}^{-\beta(A_k - A_0)} \mathrm{e}^{\beta\omega(q,s^{(k)})} \tilde{P}(q, s^{(k)})}{\sum_{j=1}^{n} n_j \mathrm{e}^{\beta A_j} \mathrm{e}^{-\beta\omega(q,s^{(j)})}}, \tag{1.42}$$

$$P(q) = \frac{\sum_{k=1}^{n} n_k \tilde{P}(q, s^{(k)})}{\sum_{k=1}^{n} n_k \mathrm{e}^{\beta(A_k - A_0)} \mathrm{e}^{-\beta\omega(q,s^{(k)})}} \tag{1.43}$$

The $\exp(\beta(A_k - A_0))$ in **Eqn. 1.43** contains $P(q)$ implicitly as

$$\mathrm{e}^{\beta(A_k - A_0)} = \int \mathrm{d}q \; P(q) \mathrm{e}^{-\beta\omega(q,s^{(k)})}, \tag{1.44}$$

and therefore **Eqns. 1.43** and **1.44** need to be solved iteratively to yield the unbiased distribution. Finally, the free energy along the reaction coordinate can be calculated using

$$A(q) = -kT \ln P(q). \tag{1.45}$$

Note the free energy generated is the Helmholtz (canonical) energy and it is only by assumption (in an incompressible system) that $A(q) = G(q)$ in the isothermal-isobaric ensemble. An alternative method to WHAM is umbrella integration, developed by Kästner and Thiel and detailed in **Appendix A.2.4**.[120]

1.4.2 Convergence testing

In US, as in all sampling techniques, the accuracy of the results depends on both the accuracy of the method used to calculate the energy and the quality of sampling. Previous efforts to measure the quality of sampling in MD have included investigating the overlap in distributions between simulations starting from different initial conformations[132] and using metrics to compute uncertainty from a simulation, e.g., bootstrapping.[133] We explore convergence and overlap metrics in US in **Chapter 4**.

Studies relating to convergence testing specific to US, as well as sampling quality in terms of windows, are limited. In their paper on adaptive US, Im and Park used the Kullback-Leibler divergence (KL-divergence, detailed in **Chapter 4**) to measure the sampling quality of a particular adaptive US methodology to a reference.[134] The shallower the gradient of KL-divergence with respect to time, the closer to the maximum convergence of the distribution. Additionally, the smaller the KL-divergence, the closer to absolute convergence the distribution becomes. Roitberg and Dashti used the KL-divergence to assess the quality of the resulting potential of mean force (PMF) to a reference distribution.[135] Convergence within a particular window has not been studied to the best of our knowledge, and is addressed in **Chapter 4**.

Measuring sampling coverage across the reaction coordinate often involves inspecting the overlap between adjacent windows. Previous metrics of overlap have used the average acceptance probability, originally derived from replica exchange methods.[134, 136] This metric

calculates the overlap between two windows based on the metropolis criterion for the exchange between two windows, which incorporates the PMF along the reaction coordinate. An optimal overlap, using this approach, was found to be $\approx 39\%$, although relation of this overlap to the quality of the free energy was not investigated. One disadvantage of this approach is prior knowledge of the PMF is required to estimate the overlap between windows. Another study by Roux and Meng used a Gaussian overlap metric to assess the quality of the resulting PMF as a function of overlap.[137] They found an overlap of 20% was sufficient to achieve an accurate PMF for the folding of the pentapeptide Met-enkephalin. However, this was based on Gaussian basis functions with equal widths and the overlap metric was defined assuming the heights for each Gaussian were unity, which does not apply to typical US simulations.[126]

Other aspects of US convergence that have not been previously explored include assessing the convergence of a window with respect to expected Gaussian distribution (in the limit of large κ) and assessing the influence of overlap on the quality of the resulting free energy profile, addressed in **Chapter 4**.

1.4.3 Umbrella sampling software

Computational research often relies on automation for speed and accessibility. Development of theoretical models can enable accurate predictions to be made, but their implementation is key if they are to be used by the larger scientific community. There are many examples of automation paving the way for further research in computational science, including the development of electronic structure packages (ORCA[6], Gaussian[7], CASTEP[138]) and MD packages (GROMACS[8], AMBER[9], OpenMM[139]).

More recently, combination of a variety of packages have enabled different methods to be used together with little prior coding knowledge. The Python package, *autodE*[140], combines a range of electronic structure packages (e.g., ORCA, XTB[141]) with TS finding algorithms (e.g., NEB[142]). Moreover, the open source package can be installed from Conda and a TS can be found in a few lines of python. We present two MLP Python packages in **Chapter**

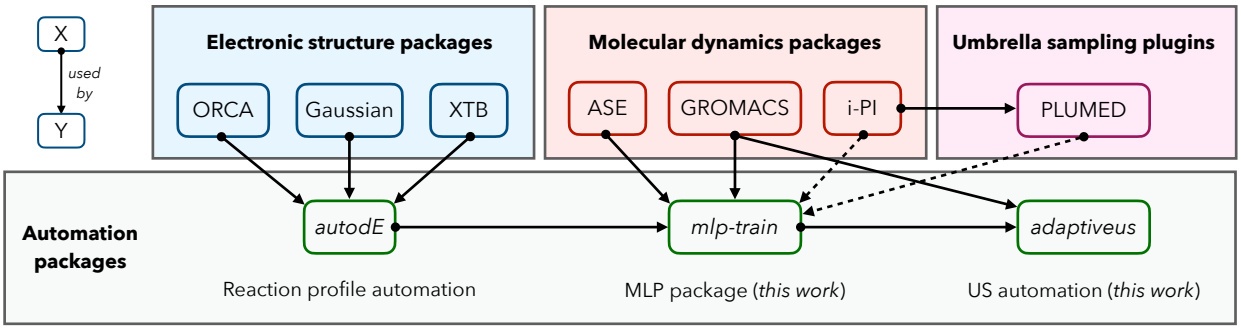


Figure 1.14: An overview of how relevant packages in this work relate to one another, including the *mlp-train* and *adaptiveus* packages from this work. A solid arrow indicates a package is used by another while a dashed arrow means the implementation is in progress.

2, which combine the training of GAP[69] (*gap-train*[1]) and ACE[70] (*mlp-train*[2]) with the MD driver ASE.[88]. *mlp-train* can also perform US simulations in a fully automated fashion, with functionality currently being extended to metadynamics.

Packages to run automated US simulations are widely available, e.g., GROMACS[8], i-PI + PLUMED[143, 144], Grossfield WHAM[145], and we do not wish to create another package doing this. However, where we feel we can contribute in automating the convergence testing and overlap calculations through our package *adaptiveus*[146] described in **Chapters 4** and **5**. A schematic of relevant packages used in this work and how they link to one another is given in **Fig. 1.14**.

1.4.4 Adaptive umbrella sampling

Another important feature of US is the choice of κ and the window reference values. When US uses a harmonic bias as a constraint, the bias energy is expressed as $\omega(r) = \frac{\kappa}{2}(f(r) - \zeta)^2$. The κ parameter and the set of ζ reference values are user chosen and their choice can affect the efficiency and accuracy of a US simulation.[11] Often, κ is chosen either from experience of a system or from running short US simulations to test different suitable values. Likewise, the reference values (sometimes chosen by proxy from the number of windows) must be supplied. This leads to difficulties since the spacing depends on the width of the resulting distribution, which in turn depends on the choice of κ. Previous attempts at adaptively chosen values for κ and ζ have been reported and are explored in **Chapter 5**.

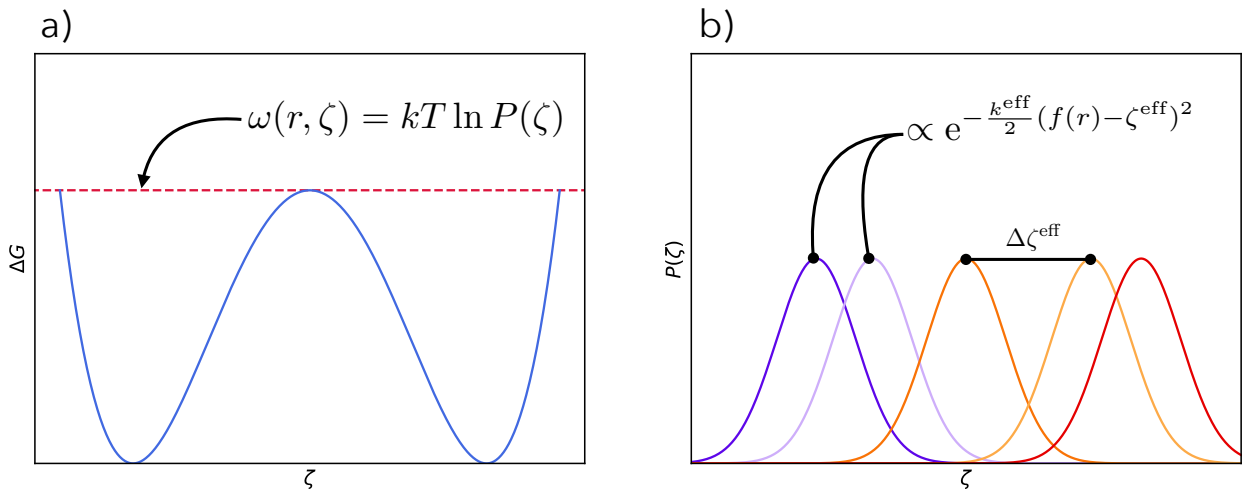


Figure 1.15: Representations of adaptive US methods. a) Mezei adaptive scheme where the bias function is estimated from short US simulation to be the negative of the free energy, thus in theory producing a flat energy surface.[147] b) Im and Park's approach where the spring constant and reference values are chosen based on the concept of 'thermodynamic length'.[134]

1.4.4.1 Adaptive US with a barrierless potential

One of the earliest approaches in adaptive US focused on iteratively modifying the bias energy function to more closely match the ideal bias, $\omega(r,\zeta) = kT \ln P(\zeta)$, the negative of the free energy, **Fig. 1.15**.[147] If $P(\zeta)$ is known, the sampling is trivial along the barrierless surface. This method works by running a short simulation to generate a $P(\zeta)$, which was in turn used to update $\omega(r,\zeta)$. In the paper, the method was tested on calculation of the solvent contribution to the free energy difference between two conformations of the Alanine dipeptide in water using a harmonic bias.[147] The authors found reasonable agreement with conventional US with the reduction of human intervention. Moreover, they state that with relatively simple modifications (including, for instance, determining the optimal iteration length), significant computational gains can be obtained. The downside of this method is it requires an efficient and accurate estimation of distribution. For simple systems, this may be sufficient but for more complex free energy surfaces, the time required to find the optimal $\omega(r,\zeta)$ may be prohibitive.

Further developments to this adaptive scheme were reported. In the work by Bartels and Karplus, the adaptive scheme was based upon a modification of the Hamiltonian and the

umbrella potential for the i^{th} simulation is determined based on the data from all previous simulations, enabling the umbrella potential to be updated rapidly.[148] The approach to adaptive US was applied to a range of systems, including calculation of potential of mean force for the central dihedral angle in glycol[149], to the alanine and threonine dipeptides[150], the folding of the β-hairpin peptide BH8[151] and on the proton transfer reaction between ammonium ion and ammonia in water in CHARMM.[152] Another adaptation to this approach used experimental single molecule pulling data to approximate the PMF. The bias was then modified according to the PMF which found improvements in the efficiency of sampling.[153].

1.4.4.2 Adaptive US with ideal overlap of windows

Alternative methods of adaptive US have also been presented. In the US replica exchange method, systematic control of window positioning was achieved by utilisation of the average acceptance probability, P_a. P_a denotes the probability of exchange between two windows and thus placing windows such that P_a was uniform enabled consistent overlap between US windows.[135] Here, κ was kept constant.

This replica exchange method US method was built on by Im and Park where they developed an adaptive strategy for choosing both the window reference values and κ values, **Fig. 1.15**.[134] In this work, they used the concept of the thermodynamic length (which is defined as the length travelled along the free energy coordinate) to choose ζ and κ. By placing windows uniformly along the thermodynamic length, instead of directly along the reaction coordinate, uniform sampling can be obtained. The estimated PMF is required to derive the thermodynamic length for a particular system. Coupling this with the P_a overlap target of 40%, windows can be arranged to optimise sampling. Woolf and Grossfield presented a similar automatic reference value method whereby data was histogrammed and the half-maximum shoulder is used to place the new window at a reference that ensures the windows "overlap significantly".[154]

An automated κ selection method was also presented in the paper by Im and Park.[134]

They begin by assuming that a window distribution can be approximated by a Gaussian distribution, parameterised by κ^{eff} and ζ^{eff}, the effective bias force constant and shifted centre from the original window centre. The effective PMF is approximated through a Taylor expansion and expressions for the optimal value of κ across the reaction coordinate, based on the estimated PMF, are given. The strategy was tested on a toy model and the assembly of the GpA-TM in CHARMM, finding improvements in the sampling efficiency. One limitation of this method is the reliance on an estimation of the PMF for choosing the reference and κ values. Another drawback is the lack of available code and implementation into existing programmes. In **Chapter 5**, we present a Python package, *adaptiveus*, for the automation of US with convergence testing and adaptive US.

1.5 Methods

In addition to machine learning potentials, **Section 1.2** and US, **Section 1.4**, electronic structure theory (EST), in particular DFT, and MD are utilised in this work. Furthermore, details of the computational methods for each chapter can be found in **Appendix B**. The following sections detail EST and MD.

1.5.1 Electronic structure theory

1.5.1.1 Preamble

Predicting the energies and forces of chemical systems is a central goal in computational chemistry. Relative energies enable prediction of the stabilities of different conformers, guide the design of novel molecules and elucidate reaction mechanisms. Forces allow dynamics of a system to be performed, enabling ensemble properties to be calculated. These properties can be accessed via the non-relativistic time-independent Schrödinger equation, **Eqn. 1.46**, and thus great effort has been made to find (approximate) solutions to it.

$$\hat{H}\Psi = E\Psi. \tag{1.46}$$

Here, $\hat{H}$ is the Hamiltonian operator, containing the kinetic and potential energy terms required to compute the energy eigenvalue, E, from the wavefunction, Ψ, which contains the complete information of the system. A simplification central to quantum chemistry is the Born-Oppenheimer (BO) approximation whereby due to the relatively slow movement of the nuclei, the electrons are said to be moving in a field of stationary nuclei.[155] Therefore, we can set the nuclear kinetic energy term and nuclei-nuclei interaction terms of $\hat{H}$ to a constant and thus zero. The BO Hamiltonian, in atomic units, then becomes

$$\hat{H} = -\sum_{i=1}^{N} \frac{1}{2}\nabla_i^2 - \sum_{i=1}^{N}\sum_{a=1}^{M} \frac{Z_A}{r_{iA}} + \sum_{i>j}^{N} \frac{1}{r_{ij}}. \tag{1.47}$$

The first term corresponds to the kinetic energy of the electrons. The second term involves the Coulomb interaction of electrons with nuclei, where $r_{iA} = |\mathbf{r}_i - \mathbf{r}_A|$. The final term, involving the interactions between electrons, leads to the insolubility of the Schrödinger equation for almost every system of interest. This term is replaced by a functional of the density in Kohn–Sham theory to account for the exchange and correlation of the electrons. The total wavefunction after the BO approximation becomes a product of electronic and nuclei components, **Eqn. 1.48**.[155]

$$\Psi(\{\mathbf{r}_i\};\{\mathbf{R}_A\}) = \Psi_{\text{el}}(\{\mathbf{r}_i\};\{\mathbf{R}_A\})\Psi_{\text{nuc}}(\{\mathbf{R}_A\}). \tag{1.48}$$

Exact solutions to the Schrödinger equation are only possible for simple systems such as the hydrogen atom, and much of the field of quantum chemistry is dedicated to finding accurate and efficient approximations.

The variational method can be used to find the ground state energy from the electronic Hamiltonian.[55] Starting from a 'trial' wavefunction, $\tilde{\Psi}(\alpha)$, the true ground state wavefunction, and thus energy, E_0, can be approached from above by systematically varying α.

$$E(\alpha) = \frac{\langle\tilde{\Psi}(\alpha)|\hat{H}|\tilde{\Psi}(\alpha)\rangle}{\langle\tilde{\Psi}(\alpha)|\tilde{\Psi}(\alpha)\rangle} \geq E_0 \tag{1.49}$$

Here, α represents an adjustable parameter of the wavefunction, $\tilde{\Psi}$. Typically, in quantum chemistry, these wavefunctions are represented by a Gaussian basis set, and α represents the set of coefficients for this basis set. In this work, we utilise both Pople basis sets[156] and Karlsruhe basis sets.[157] Pople basis sets have the form X-YZg, where X is the number of primitive Gaussian functions for each core atomic orbital basis function, while Y and Z are the Gaussian functions for the valence orbitals. For example, the 6-31+G(d,p) basis set indicates a set of diffuse s and p functions have been added to heavy atoms. The (d,p) means a set of polarisation d-functions have been added to heavy atoms and polarisation p-functions have been added to hydrogen. The Karlsruhe def2 basis set have split valence polarisation, e.g., def2-SVP has the polarisation functions on all atoms.

The variational principle is a powerful method of approaching the true ground state energy and is used in common electronic structure theory methods such as HF (**Section 1.5.1.2**) and DFT (**Section 1.5.1.3**).[55, 155]

1.5.1.2 Hartree-Fock theory

HF theory is an application of MO theory to the electronic Schrödinger equation. The many-body wavefunction is constructed from one-electron wavefunctions, where each electron exists in a mean-field of the other electrons.[155] In HF, n spin-orbitals, $\psi_i(\mathbf{x}_1)$, are combined as products for an n-electron system,

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \sqrt{n!}\hat{A}\psi_1(\mathbf{x}_1)\psi_2(\mathbf{x}_2)\cdots\psi_n(\mathbf{x}_n). \tag{1.50}$$

Here, we assume electrons are uncorrelated and can be separated into single electron basis functions. The antisymmetric projection operator, $\hat{A}$, ensures Fermi antisymmetry is followed upon exchange of any pair of electrons. This product is equivalent to the Slater determinant of spin orbital basis functions,

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \cdots & \psi_n(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \cdots & \psi_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x}_n) & \psi_2(\mathbf{x}_n) & \cdots & \psi_n(\mathbf{x}_n) \end{vmatrix}. \tag{1.51}$$

The HF Hamiltonian comprises a one-electron operator, $\hat{h}$, and two-electron operator, $\hat{g}$. The energy can be expressed in terms of these operators and the one-electron wavefunctions, **Eqn. 1.52**.

$$E_{\mathrm{HF}} = \sum_i \langle \psi_i | \hat{h}_1 | \psi_i \rangle + \frac{1}{2} \sum_{ij} \langle \psi_i \psi_j | r_{12}^{-1} | \psi_i \psi_j \rangle - \frac{1}{2} \sum_{ij} \langle \psi_i \psi_j | r_{12}^{-1} | \psi_j \psi_i \rangle \tag{1.52}$$

The first term is the kinetic energy of electron i, the two-electron terms are the classical 'Coulomb' energy and the 'exchange' energy, arising from the overlap of charge distribution between electron 1 and 2 due to the antisymmetric nature of the wavefunction. One advantage of HF theory is the cancellation of the self-interacting Coulomb terms in $\langle \Psi | \hat{g} | \Psi \rangle$, thus treating the exchange energy exactly.[55]

To find the set of optimal HF orbitals, we first minimise the HF Lagrangian with respect to the orbitals, subject to orthonormality,

$$\mathcal{L}[\psi] = E_{\mathrm{HF}}[\psi] - \sum_{ij} \varepsilon_{ij} (\langle \psi_i | \psi_j \rangle - \delta_{ij}). \tag{1.53}$$

This results in an eigenequation of the energies and orbitals,

$$\hat{f}_1 \psi(\mathbf{x}_1) = \varepsilon_k \psi(\mathbf{x}_1), \tag{1.54}$$

where $\hat{f}_1$ is the Fock operator. Practically, the HF equations are solved self-consistently via the Roothaan–Hall equations, where the orbitals are expressed as a linear combination of atomic basis functions, $|\psi_i\rangle = \sum_\mu |\eta_\mu\rangle C_{\mu i}$.

While HF is a simple and fast method for calculating the quantum electronic energy of a system, and estimations for geometries are generally reasonable, it gives poor results for thermochemistry and large changes in bonding. Since the Slater determinant is only an approximation to the true wavefunction and HF is a variational method, the HF energy is always less negative than the exact ground state energy. This difference in energy, called the correlation energy, arises because HF treats the electrons in a static, mean field manner. DFT offers an improvement over HF in terms of correlation, where dynamic electron correlation effects are included.[155] HF is often combined with DFT through hybrid functionals, where the imperfect cancellation of the self-interaction energy causes inaccuracies.[55]

1.5.1.3 Density functional theory

While wavefunction based methods, such as HF, solve the Schrödinger equation, **Eqn. 1.46**, via the wavefunction, DFT uses the electron density of the system to calculate the ground state energy.[158] As with HF, the DFT Hamiltonian contains kinetic energy terms relating to the electrons, as well interaction terms between electrons and nuclei. DFT is built on two theorems, developed by Hohenberg and Kohn; both are formally exact.[34]

- The first states that the wavefunction and energy can be expressed as functionals of the density: $\Psi[\rho], E[\rho]$.
- The second states that we can search over trial densities variationally to calculate Ψ and E.

Using the variational principle, the energy functional can be cast in terms of the universal density functional, $F[\rho]$ and the electron-nuclei interaction, **Eqn. 1.55**.

$$E[\rho] = F[\rho] + \int \mathrm{d}\mathbf{r}\ v(\mathbf{r})\rho(\mathbf{r}) \tag{1.55}$$

Minimising this energy gives the exact ground state energy of the system without reference to the wavefunction. However we do not know the exact form of $F[\rho]$, so we use Kohn–Sham theory to solve for the energy. Here, we introduce a determinant for n non-interacting

electrons, $|\Phi_s\rangle = |\phi_1 \dots \phi_n\rangle$.[35] This allows us to extract the non-interacting kinetic energy, $T_s[\rho]$ as

$$F[\rho] = T_s[\rho] + E_{\text{Hxc}}, \tag{1.56}$$

where $T_s[\rho] = \min_{\Phi_s} \langle \Phi_s | \hat{T} | \Phi_s \rangle$ and E_{Hxc} contains the classical electron-electron repulsion as well as the corrections for exchange and correlation effects. E_{Hxc} can be split into the exchange-correlation functional, $E_{\text{xc}}[\rho]$, **Eqn. 1.57**, and the Hartree energy functional, $E_{\text{H}}[\rho]$, **Eqn. 1.58**, containing the unphysical self-interaction, which many developments in DFT aim to counteract.

$$E_{\text{Hxc}}[\rho] = E_{\text{H}}[\rho] + E_{\text{xc}}[\rho] \tag{1.57}$$

$$E_{\text{H}}[\rho] = \frac{1}{2} \int \mathrm{d}\mathbf{r}_1 \int \mathrm{d}\mathbf{r}_2 \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}} \tag{1.58}$$

Rewriting **Eqn. 1.55** in terms of $T_s[\rho]$ and E_{Hxc} and minimising with respect to the density, we obtain the DFT ground state energy,

$$E_0 = \min_{\Phi_s} \left\{ \langle \Phi_s | \hat{T} + \hat{V} | \Phi_s \rangle + E_{\text{Hxc}}[\rho_\Phi] \right\}, \tag{1.59}$$

where $\hat{V}$ is the electron-nuclei interaction. This energy can be solved in a similar way to HF, where a Lagrangian is written, subject to orthonormal spin-orbitals. Upon minimising, the Kohn–Sham equations are obtained, **Eqn. 1.60**, and can be solved self-consistently to give the orbitals and energies.[55]

$$\left(\hat{h} + v_{\text{hxc}}(\mathbf{r}) \right) \phi_i(\mathbf{r}) = \varepsilon_i \phi_i(\mathbf{r}) \tag{1.60}$$

Here, $\hat{h}$ is the one-electron Hamiltonian for the non-interacting electrons that is modified by the Hartree-exchange-correlation potential, $v_{\text{hxc}}(\mathbf{r})$, to ensure the density is equal to the

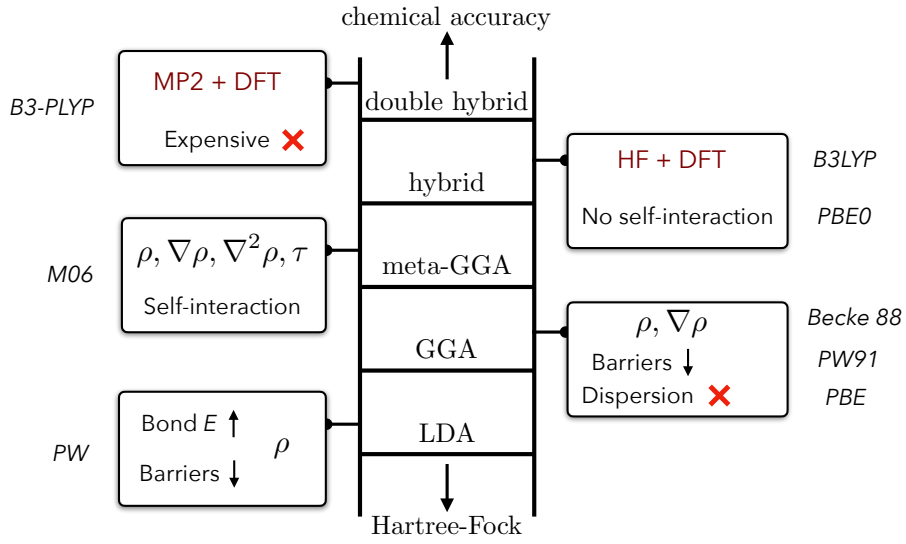


Figure 1.16: Jacob's ladder of DFT functionals, showing their dependence on the density, ρ, or their composition in the case of hybrid functionals. Main limitations or improvements are shown, with arrows indicating over or under-prediction.

exact density for a system of n interacting electrons.

⁂

While DFT is a formally exact theory for the ground state energy, the functional form of the exchange-correlation term is unknown. There have been countless developments in finding fast and accurate approximations to the exchange-correlation functionals, often referred to simply as 'functionals'.[159, 160] Through the years, a hierarchical ladder has been constructed, named Jacob's ladder of DFT functionals, **Fig. 1.16**.

The first rung contains functionals dependent on the electron density, called local density approximation (LDA) functionals, e.g., PW[161]. While the LDA works well for systems with a roughly uniform electron density, it falls short for molecular systems, where electron density is often non-uniform.[159] An improvement to LDAs is to base the energy not only on the density but on its gradient as well, referred to as the generalised gradient approximation (GGA), e.g., B88[162], PW91[163] and PBE.[164]

The third rung, meta-GGA functionals, includes two additional variables, the second deriva-

D3(zero)	D3(BJ)
$\lim_{R_{AB}\to 0} f(R_{AB}) = 0$	$\lim_{R_{AB}\to 0} f(R_{AB}) = C$

Table 1.1: Damping function, $f(R_{AB})$, behaviour of D3(zero) and D3(BJ) DFT dispersion corrections.

tive of the density and $\tau(\mathbf{r})$, the non-interacting positive kinetic energy density, e.g., M06.[165]. While these better describe the exchange hole than GGAs, they still suffer from the self-interaction error. The fourth rung functionals are hybrid functionals, where the HF exchange energy is mixed with the DFT exchange-correlation energy. Hybrid functionals are typically more accurate than pure functionals for thermochemistry, e.g., B3LYP and PBE0, the latter used throughout this work, **Eqn. 1.61**[166, 167]

$$E_{xc}^{\text{PBE0}} = 0.25E_{x}^{\text{HF}} + 0.75E_{x}^{\text{PBE}} + E_{c}^{\text{PBE}} \tag{1.61}$$

Hybrid functionals reduce the self-interaction error through the inclusion of the HF exchange energy and are often more accurate than wavefunction based methods such as MP2 where dispersion is not significant.[55] The final rung of the ladder contains the double-hybrid functionals, where an MP2-like correlation correction is added.[168] While these functionals can be accurate, they are also expensive.[169]

A further correction that can be added onto many functionals is a dispersion correction. Dispersion energy results from the long range interaction of coupled dipoles in a system. The semi-empirical dispersion correction from Grimme et al. introduces a van der Waals correction through empirical coefficients over a distance term, **Eqn. 1.62**.[170]

$$E_{\text{disp}} = -\frac{1}{2}\sum_{A\neq B}\sum_{n=6,8,10,\ldots} s_n \frac{C_n^{AB}}{R_{AB}^n} f_{\text{damp}}(R_{AB}) \tag{1.62}$$

Here, s_n is a scaling parameter and $f_{\text{damp}}(R_{AB})$ is a damping function that tends to unity at large R_{AB} and zero at small R_{AB}. The two dispersion methods used in this work are the D3(zero) and D3(BJ) methods, **Table 1.1**.[171]

The recent $D4$ correction, which generates the C_n^{AB} coefficients through Casimir-Polder integration, where density information is calculated from partial charges, shows lower errors in benchmark studies.[172]

1.5.2 Molecular Dynamics

1.5.2.1 Ensemble theory

Molecular simulation can be used to compute the physical properties of a system, e.g., E, V, N, directly, or T, S, G indirectly.[118] For example, energy is transformed into the Helmholtz free energy under a Legendre transform,

$$F = E(N,V,S) - \left(\frac{\partial E}{\partial S}\right)_{N,V} S \tag{1.63}$$

$$= E(N,V,S) - TS. \tag{1.64}$$

These properties can be accessed via ensembles; a collection of a large number of systems that replicates the macroscopic properties of an actual system. There are different types of ensembles, depending on which properties are fixed. The simplest ensemble is the *micro-canonical* ensemble, which represents an isolated system, where N, V, E are fixed. Allowing E to vary, while fixing T, gives the closed, isothermal *canonical* ensemble. Varying N, while fixing the chemical potential μ, gives the open, isothermal *grand canonical* ensemble. We can also vary the V in the canonical ensemble, while fixing P, to give the isothermal-isobaric ensemble.

In the microcanonical ensemble, the ensemble average of any mechanical variable, $\langle a \rangle$, can be computed as a weighted sum of the variable normalised by the microcanonical partition function, **Eqn. 1.65**.[118]

$$\langle a \rangle = \frac{\int \mathrm{dx}\ a(\mathrm{x})\delta(\mathcal{H}(\mathrm{x}) - E)}{\int \mathrm{dx}\ \delta(\mathcal{H}(\mathrm{x}) - E)} \tag{1.65}$$

Direct computation of the partition function for anything but simple systems is not feasible. Therefore, we must use a different approach to generate the ensemble averages. Since all states in the microcanonical ensemble lie on the constant energy hypersurface $\mathcal{H}(\mathrm{x}) = E$, propagating the equations of motion that conserve the total energy will enable the microcanonical ensemble to be sampled. Hamilton's equations allow a system to evolve on this constant energy hypersurface,

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} \tag{1.66}$$

and therefore provide a suitable set of equations to use to calculate properties in the microcanonical ensemble. Provided the system is able to access all the configurations on the hypersurface after infinite time, i.e., it is *ergodic*, the microcanonical ensemble will be generated exactly. We can therefore replace our ensemble average in terms of the partition function, **Eqn. 1.66**, with a time average over the trajectory, **Eqn. 1.67**.

$$\langle a \rangle = \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^\tau \mathrm{d}t \; a(\mathrm{x}_t) \approx \frac{1}{M} \sum_{n=1}^{M} a(\mathrm{x}_{n\Delta t}) \tag{1.67}$$

In **Eqn. 1.67**, we have included a discrete numerical approximation to the integral, which is required in practice.

We can show that Hamilton's equations of motion are equivalent to Newton's second law of motion, which we use to propagate MD. The Hamiltonian for a set of N particles in Cartesian space is given by

$$\mathcal{H}(\mathbf{r}_1, ..., \mathbf{r}_N, \mathbf{p}_1, ..., \mathbf{p}_N) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}_1, ..., \mathbf{r}_N). \tag{1.68}$$

We first take the derivative of the Hamiltonian with respect to momentum, which is equal to the time derivative of position. Then we take the derivative with respect to the position, which is equal to the time derivative of momentum, giving

$$\dot{\mathbf{r}}_i = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}_i}{m_i}, \tag{1.69}$$

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} = -\frac{\partial U}{\partial \mathbf{r}_i} = \mathbf{F}_i(\mathbf{r}). \tag{1.70}$$

Taking the time derivative of **Eqn. 1.69** and substituting it into **Eqn. 1.70** gives

$$\ddot{\mathbf{r}}_i = \frac{\dot{\mathbf{p}}_i}{m_i}, \tag{1.71}$$

$$\dot{\mathbf{p}}_i = m_i \ddot{\mathbf{r}}_i = \mathbf{F}_i(\mathbf{r}), \tag{1.72}$$

which is the familiar second law of motion. Therefore, evolving our system using Newton's equations is equivalent to sampling from the microcanonical ensemble.[118]

1.5.2.2 Molecular dynamics integrators

Practically, to evolve the equations of motion, **Eqn. 1.66**, numerical integrators with finite time steps are used. One of the earliest integrators was the Verlet algorithm used by Verlet in 1967 for MD simulations under a Lennard-Jones potential, derived in **Appendix A.1.1**.[173] While this algorithm allows us to generate velocities at any point, it does not explicitly evolve the velocities.

A modification to the Verlet algorithm, named the velocity Verlet algorithm, allows for the reconstruction of the full phase space consisting of positions and momenta.[174] One advantage of the velocity Verlet algorithm is that it is symplectic; the system's momenta and positions are stable in phase space since as there exists a Hamiltonian along the trajectory that remains close to the true Hamiltonian.[118]

The velocity Verlet equations for position and velocity are derived in **Appendix A.1.2** and given as

$$\begin{aligned}\mathbf{r}_i(t+\Delta t) &\approx \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t),\\ \mathbf{v}_i(t+\Delta t) &\approx \mathbf{v}_i(t) + \frac{\Delta t}{2m_i}\left[\mathbf{F}_i(t) + \mathbf{F}_i(t+\Delta t)\right].\end{aligned} \tag{1.73}$$

A more computationally efficient variation of velocity Verlet is the leapfrog algorithm, which computes position and velocity at differences of half a time-step, i.e., $\mathbf{r}_i(t+\Delta t)$ and $\mathbf{v}_i(t+\frac{1}{2}\Delta t)$. While velocity Verlet is marginally more accurate in reproducing the correct pressure ensemble, leapfrog is quicker to compute requiring only half as many communication calls, and is used in this work.[118]

1.5.2.3 Thermostats

While using Newton's equation to sample the phase space in the constant energy space generates reproduces the microcanonical ensemble, NVE, to sample in the canonical ensemble, NVT, a thermostat is required. Thermostats act as a thermal reservoir, ensuring that the average kinetic energy remains roughly constant. Common thermostating methods include Berendsen temperature coupling[175], **Appendix A.1.3**, velocity-rescaling temperature coupling[176], **Appendix A.1.4** and Langevin dynamics.[177] The Langevin thermostat is used throughout this work and we briefly outline its form, **Eqn. 1.74**, with further details in **Appendix A.1.5**.[118]

$$\mu\ddot{q} = -\frac{\mathrm{d}W}{\mathrm{d}q} - \dot{q}(t)\zeta_0 + R(t). \tag{1.74}$$

The Langevin equation propagates the dynamics for a system of mass μ at coordinate q. The derivative of the potential of mean force, W, represents the force acting on the system. The system is damped by a friction coefficient, ζ_0. Finally, $R(t)$ is the random force acting on the particles due to friction, sampled from a Gaussian distribution. Integrated equations of motion can be derived in terms of position and velocity, and these reduce to the velocity Verlet equations when the friction coefficient is set to zero. Initial velocities for the canonical ensemble can be generated according to the Maxwell-Boltzmann distribution, **Eqn. 1.75**,

expressed in terms of velocity, v.

$$f(v) = \left(\frac{2m^3}{\pi k_B^3 T^3}\right)^{\frac{1}{2}} v^2 \mathrm{e}^{-mv^2/2k_B T} \tag{1.75}$$

The Box-Muller sampling method allows for the generation of Gaussian random numbers using random numbers taken from a uniform distribution.[118]

2

Machine Learning Potentials Method Development

Contents

2.1 Summary 55
2.2 MLP training strategies for molecular systems 56
2.2.1 Validation of MLPs 57
2.2.2 Training a water system using the PhysNet potential 58
2.2.3 Training a GAP using classical MD 63
2.2.4 Active learning training strategy 66
2.3 Development of automated MLP training 68
2.3.1 *gap-train* 69
2.3.2 *mlp-train* 70
2.4 Conclusions 73

The following chapter comprises work completed in conjunction with Dr Tom Young. Testing MMMD training for GAP was carried out by the author, while all other training was performed by Dr Young. The τ_{acc} error metric and active learning (AL) strategy were developed by Dr Young and described here. The MMMD code in gap-train was written by the author, with the remaining code written by Dr Young. The umbrella sampling implementation in mlp-train was written by the author, with the remaining code written by Dr Young. The MVK/CPD potential and figures in ***Fig. 2.9*** *were produced by Dr Young using the code written by the author. Both the author and Dr Young contributed to code review. All other work in this chapter (and thesis) was completed by the author. The contents are based on two publications.[1, 2]*

2.1 Summary

This chapter introduces a novel approach to training reactive potentials by combining established MLP architectures, AL and software development. It aims to identify efficient approaches for data selection, training and validation in order to extend the wider use of MLPs for molecular systems.

Section 2.2 provides a comparison of different training strategies using bulk water as a representative system. Several issues regarding training and validation of the generated potentials (i.e., on splitting the training and validation data) are discussed and solutions suggested. The reported error metric, τ_{acc}, is used to evaluate the stability of the trained MLPs and compare different training strategies.

This chapter also introduces two open source Python packages, *gap-train* and *mlp-train*, which aim to automate training and dynamics of MLPs, **Section 2.3**. *mlp-train* is an upgraded package of *gap-train*, containing more functionality for both training and dynamics, as well as enhanced sampling methods.

Finally, a discussion of the implications of the development of freely available and automated

tools for developing MLPs for molecular systems in presented, **Section 2.4**. Details of the computational methods used in this chapter can be found in **Appendix B.1**.

2.2 MLP training strategies for molecular systems

The aim of this work has been to extend the use of established MLP architectures, such as GAP (detailed in **Section 1.2.3.1**) and PhysNet (detailed in **Section 1.2.2.1**) to molecular systems and identify efficient approaches for training and data selection. Training an MLP requires performing N ground truth evaluations, for which the computational cost can become quickly intractable when dealing with large data sets or requiring high levels of theory.

Training data configurations can be generated in a number of ways, such as randomly placing atoms in a box, subject to physical constraints, or running classical MD (referred herein as MMMD) or AIMD simulations. Generating random configurations is faster than running MD, but it has the disadvantage of more easily generating high-energy unphysical structures that are not relevant to the region of interest. On the other hand, MMMD, while more efficient than AIMD, can produce configurations significantly different to those on the quantum PES.

How these configurations are selected for training is key for the overall performance of the potential. For example, configurations can be selected from the initial data set (e.g., from an MD trajectory) either randomly or based on a given criteria. CUR selection is a common method where the square kernel matrix between SOAP descriptors (see **Section 1.2.1.2**) is used to generate a diverse subset of configurations.[178] Another method is to use the absolute error in ground truth and predicted energies, which we use in our AL strategy (*vide infra*).

2.2.1 Validation of MLPs

Testing the accuracy of a potential using a parity plot true versus predicted energies or forces) on either the training or validation data can lead to misleading accuracy compared to assessing the accuracy from an MLP-MD simulation.[1] This is because the data can be overfitted to the training data and since the validation data is spliced from the training data, the accuracy may be similar. Since our aim to is run dynamics, in **Section 2.2.2** we chose to validate the accuracy of the potential by measuring the error between the prediction and ground truth method from an MLP-MD simulation.

In addition, this work uses a temporal cumulative error metric developed in the group, τ_{acc}, which presents a prospective validation scheme allowing different potentials to be compared, used in **Section 2.3**. Qualitatively, this metric measures the length of time an MLP-MD simulation can proceed until it accumulates an 'unacceptable' amount of error. Quantitatively, it is defined as

$$\tau_{\text{acc}} = \text{time} \ : \ E_T < \sum_{i \in \text{frames}} \max(|E_i^0 - E_i^{\text{MLP}}| - E_l, 0). \tag{2.1}$$

Here, τ_{acc} is defined as the time taken for the sum of the absolute energy error to reach a threshold, E_T. Note that a baseline threshold is subtracted for each frame error to account for the residual error due to the finite radial cut-off of the model. The error in each frame is also non-negative due to the 'max' function. Typically, E_T is set to 10 times E_l, but these values can be arbitrarily adjusted depending on the context. The advantages of this metric include rewarding potentials that are stable over long times and penalising potentials which have frequent or large errors, as these will typically destabilise the dynamics. One drawback is that the number of singe point evaluations required to calculate τ_{acc} can be large, especially if the potential is particularly stable.

2.2.2 Training a water system using the PhysNet potential

To investigate the most efficient MLP training strategy, we trained a water model using PhysNet, a recently developed neural network potential (NNP), that was previously shown to generate accurate energy predictions for an S_N2 reaction.[54] Here, we aimed to test how PhysNet would perform on a water system as well as whether generating training data using a random displacements of atoms method was an effective training strategy.

2.2.2.1 Water monomer potential

Before building up to a box of water using PhysNet, we developed the training strategy for a water monomer. To train the PhysNet potential, we generated N^3 configurations over an $N \times N \times N$ grid of r_1 $[0.8, 1.5]$ Å, r_2 $[0.8, 1.5]$ Å and θ $[1.0, 2.5]$ radians for water, **Fig. 2.1c**. A PhysNet potential was then trained on the configurations using a ground truth of DFTB(3ob). We found that the optimal number of training points was 512 (with a 90/10 training/validation split) and 2000 epochs in the loss function optimisation, **Fig. 2.1a**. Using this potential, a 1 ps MLP-MD simulation was performed and the absolute energy error was calculated, **Fig. 2.1b**. Finally, the PhysNet energy was compared to the DFTB(3ob) energy along the stretch and bend of the water monomer, **Fig. 2.1c/d**.

Our results demonstrate that PhysNet can be used to generate accurate dynamics and PES surfaces for a water monomer using the grid data generation method, with energies lying close to 1 kcal mol^{-1}. Moreover, dynamics are stable for the duration of the 1 ps simulation, showing a small oscillation in accuracy. This oscillating error could originate from the O−H vibrational mode, where there is an higher error along the extension or compression. The PES of the bending mode shows that the potential under-predicts energies around 1 radian. However, the overall accuracy of the potential during dynamics is unaffected by this since the error occurs in a region of high energy and low probability.

While this potential only required 512 training points, we were able to choose configurations which we knew *a priori* would cover the PES of the water monomer. However, for systems with greater complexity, such as a water cluster, a more general approach to generating

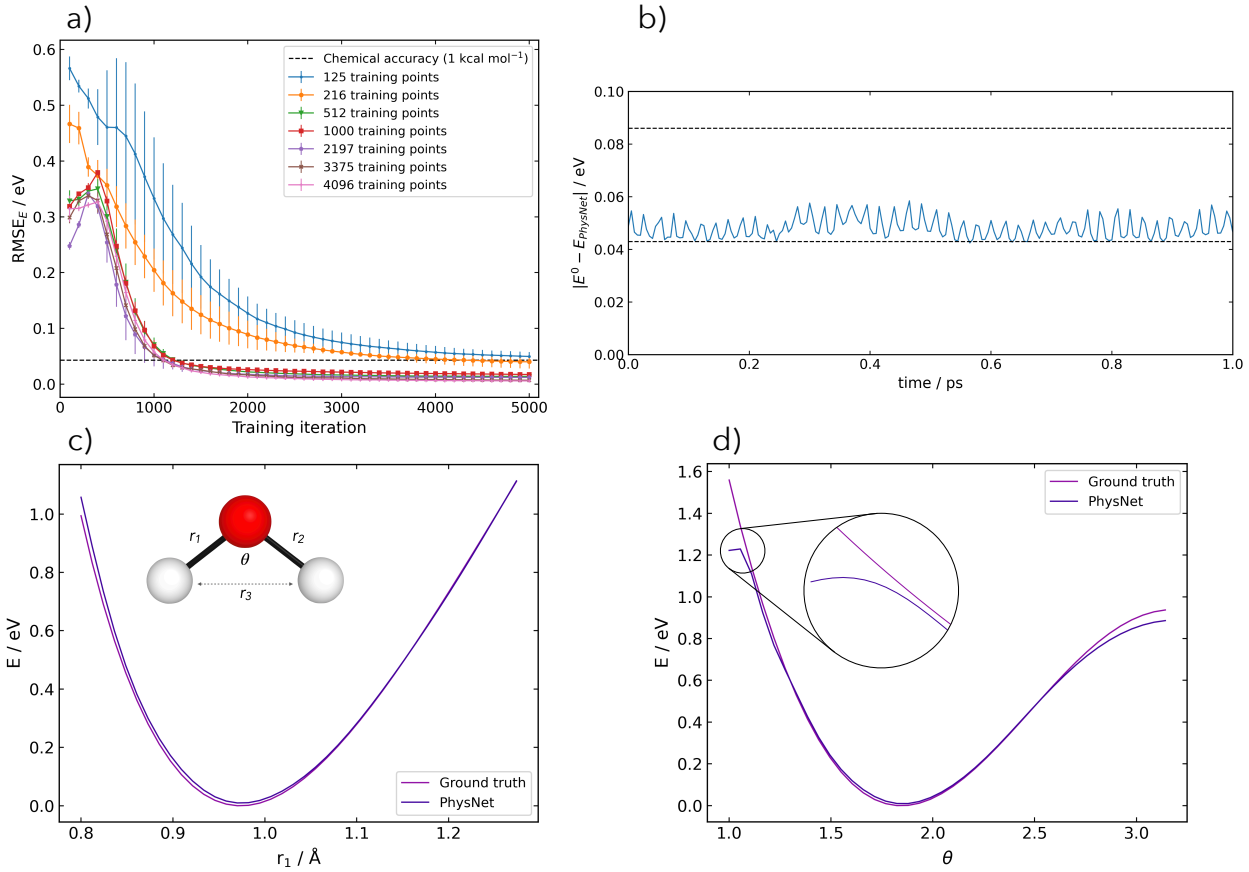


Figure 2.1: PhysNet MLP training and dynamics for a water monomer. PhysNet trained on 512 configurations of water from a uniform grid. Training run for 2,000 epochs. a) RMSE of energy on the validation data for PhysNet trained on various numbers of configurations of a water monomer from a uniform grid over r_1, r_2 and θ. Error bars are given as the standard deviation over three repeats. b) Absolute energy error from a 1 ps 300 K MLP-MD simulation of the PhysNet trained water monomer relative to DFTB(3ob) ground truth. Dashed lines indicate 1 and 2 kcal mol^{-1}. c) PES for the OH stretch, and d) bend generated by the PhysNet potential (with finer grid around 1 radian) and DFTB(3ob) ground truth.

training data is needed.

2.2.2.2 Water cluster potential

Next, we attempted to train a 25 molecule water cluster at 1 g dm^{-3} using PhysNet. For this system, we investigated the accuracy of the potential using configurations generated from random displacements. To do so, atoms within each molecule were perturbed under a normal distribution with $\sigma = 0.05$ Å and a maximum distance of 0.2 Å, with a minimum distance between water molecules of 1.7 Å.

Initial attempts at training a PhysNet potential using random configurations resulted in inaccurate dynamics, often above 10 kcal mol^{-1} in error, as well as inaccurate PES bend and stretch scans. The minima in the PESs was often predicted well, but as the PESs deviated from the minima, the potential under-predicted the energy. By adding 800 additional configurations lying along the PES bend and stretch, the accuracy of the potential greatly improved, enabling us to obtain stable dynamics, with the total error $\approx$ 3 kcal mol^{-1} and error per molecule below 1 kcal mol^{-1}, **Fig. 2.2**.

One consequence of using random configurations for training is that the energy range they lie in is large (50 eV, $>$ 100 kcal mol^{-1}). Many of these configurations will correspond to rare dynamical, or high temperature, configurations, and so are irrelevant to the processes under study, which are often explored at 300 K.

To further evaluate the accuracy of this potential, we performed 10 ps MLP-MD for a box of water containing 64 molecules. This resulted in a stable potential with a total error of $\approx$ 5 kcal mol^{-1} and an error per molecule of $\approx$ 0.2 kcal mol^{-1}, **Fig. 2.3**. In addition, the radial distribution functions (RDFs) for H_2O were calculated and showed good agreement with the DFTB RDFs, **Fig. 2.3**.

The RDFs generally predict the intramolecular OH and HH distances correctly, which is reflected in the accuracy of the PES bend and stretch predictions. The intermolecular OH and HH distances are also reasonably well predicted. Since the r_{OO} distance only comprises

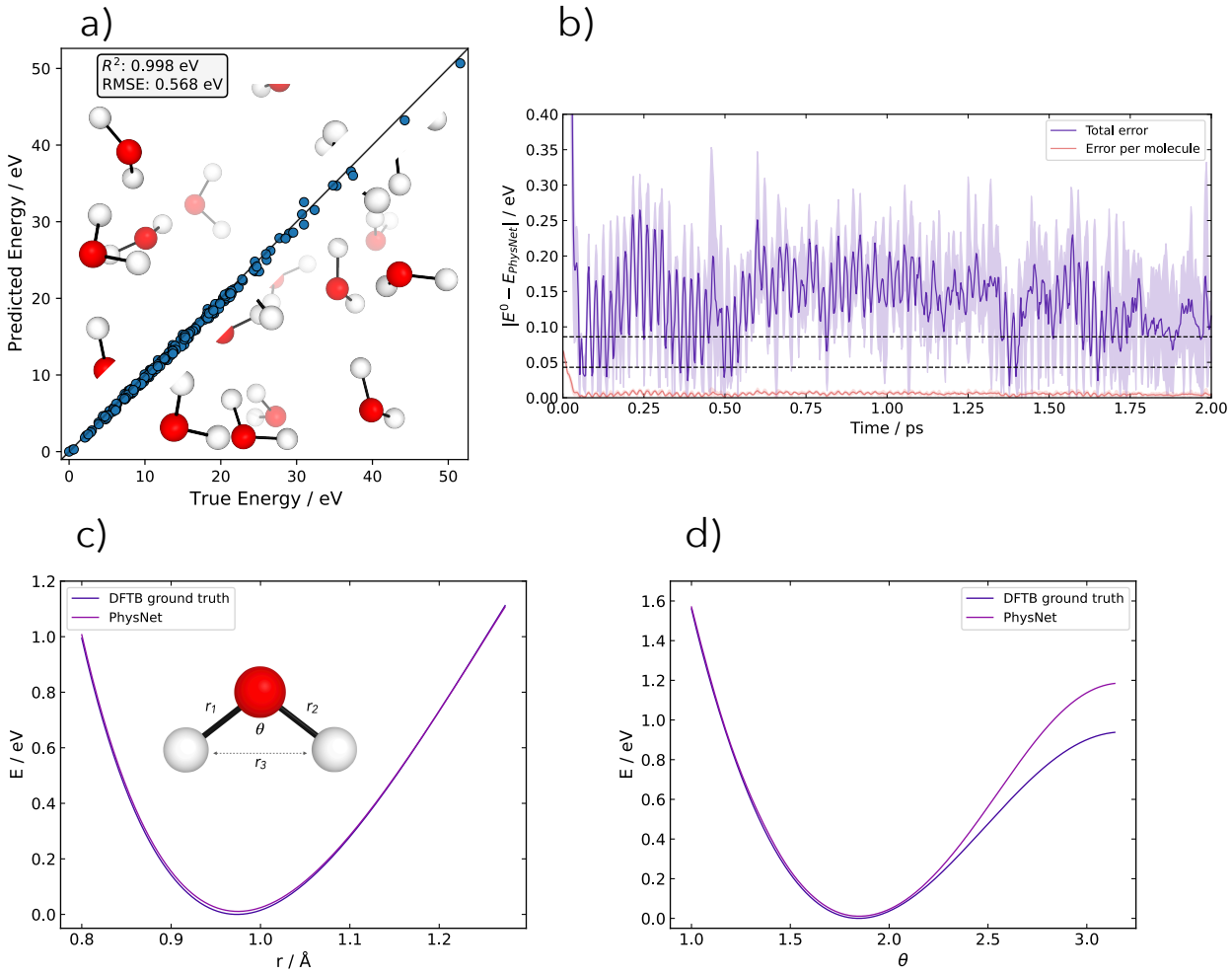


Figure 2.2: PhysNet MLP training and dynamics for 25 water molecules. PhysNet trained on 1000 random configurations with an additional 400 bend and 400 stretch points generated over a grid. Training run for 20,000 epochs. a) Parity plot of PhysNet predicted energies and DFTB(3ob) ground truth training configurations. Snapshot of initial 25 waters shown on the graph. b) Absolute energy error from a 2 ps 300 K MLP-MD simulation of the PhysNet trained water box relative to DFTB(3ob) ground truth. Dashed lines indicate 1 and 2 kcal mol^{-1}. Total error and error normalised by N_{waters} shown. c) PES for the OH stretch, and d) bend generated by PhysNet potential and DFTB(3ob).

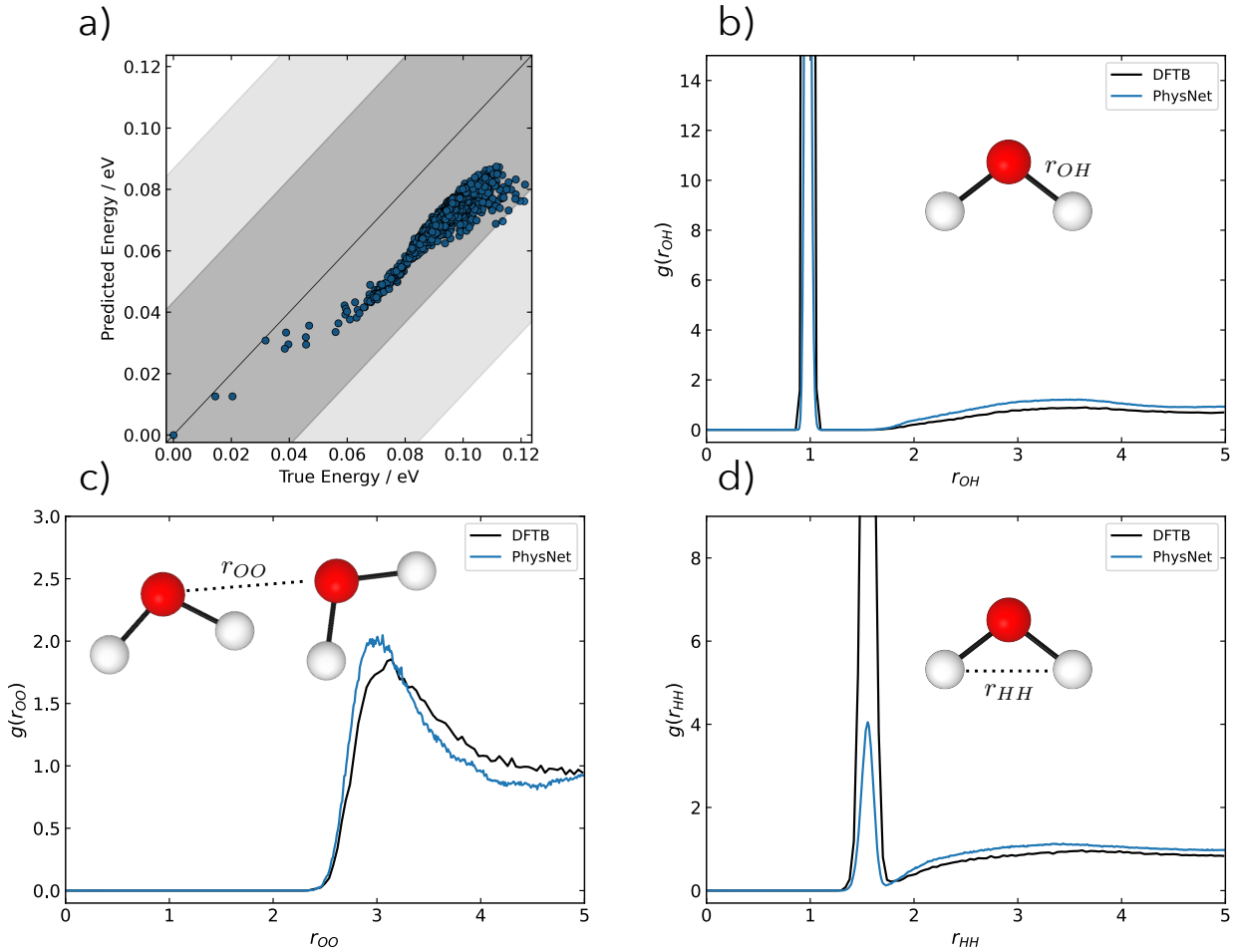


Figure 2.3: PhysNet MLP dynamics for 64 water molecules using potential detailed in **Fig. 2.2**. a) Parity plot of PhysNet predicted energies and DFTB(3ob) ground truth for selected configurations from 10 ps MLP-MD. Energies normalised by N_{waters}. 1 and 2 kcal mol^{-1} regions shown in dark and lighter grey, respectively. RDFs for water from a 10 ps 300 K MLP-MD simulation of 64 waters using PhysNet and DFTB(3ob) for b) r_{OH}, c) r_{OO} and d) r_{HH}.

intermolecular distances, we can gain a better insight into the accuracy of the potential for the intermolecular interactions. The potential predicts the OO distance to be shorter than the DFTB(3ob) prediction, resulting in the molecules sitting closer together than expected during MLP dynamics. The consequence of this is that density will be higher than expected and may be explained by the potential under-predicting the energy in all the configurations. Overall, these results demonstrate the transferability of the potential, originally trained on 25 water molecules, to this larger system of 64 water molecules.

While the strategy of generating random configurations produces reasonably accurate and physically correct dynamics for the water system under study, it still presents several limitations, both in terms of accuracy and efficiency. First, this training strategy under-predicted the energy of configurations, **Fig. 2.2** and **Fig. 2.3**. Second, we relied on the simple intramolecular nature of water, allowing us to manually select configurations to cover the PES bend and stretch. When considering systems where the PES is unknown, many more configurations may be required or the potential may fail completely.

2.2.3 Training a GAP using classical MD

One limitation of PhysNet, and NNPs generally, is the large data requirement for training compared with nonparametric based MLPs.[52, 54] The Gaussian approximation potential (GAP) has been demonstrated on a number of systems, with highly accurate results obtained from the trained potentials.[61, 69, 179] For comparing training strategies, we transitioned to GAP due to the fewer configurations required to train molecular systems compared with PhysNet.

A limitation of using the random displacement training method is that many of the configurations generated are high in energy and therefore not relevant in an MD simulation. Since the phase space region of interest is dynamics at 300 K, generating training data using MD could provide relevant configurations for the potential. To test this training strategy, we used GAP to train a box of 52 water molecules with DFTB(3ob) as the ground truth method, **Fig. 2.4**. The training data was generated by running 100 ps MD (using a classical force

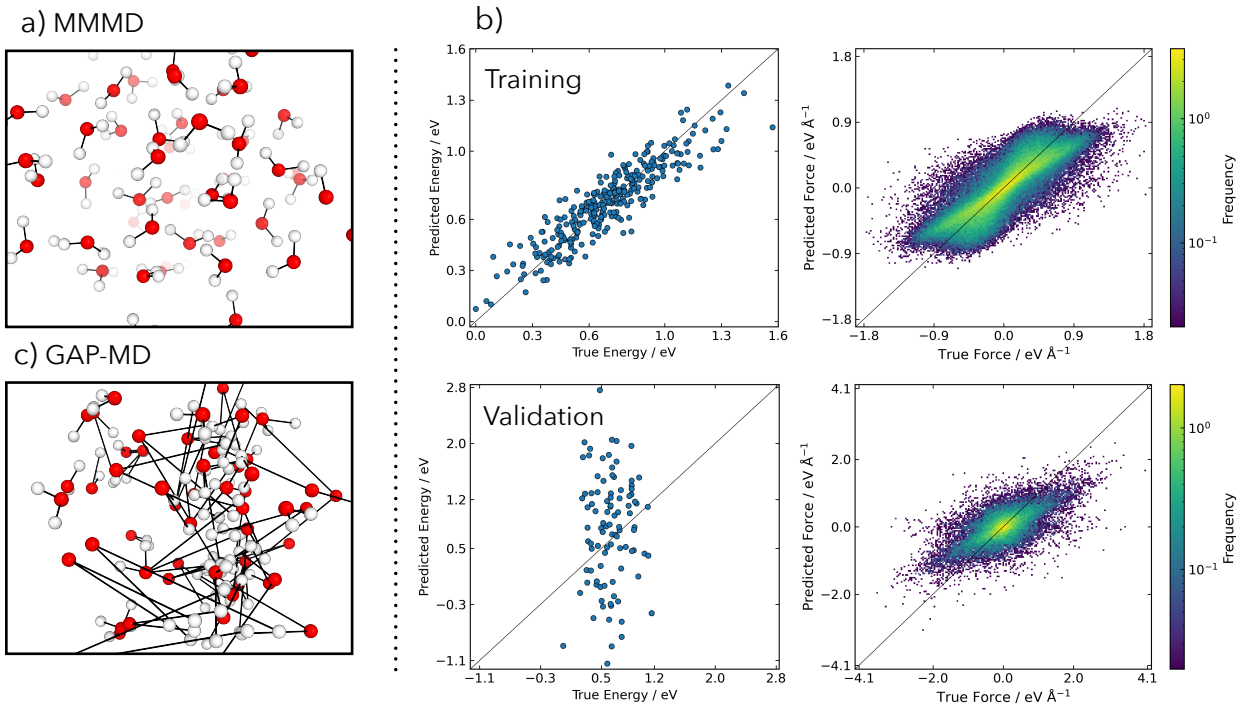


Figure 2.4: GAP MLP training and validation on 52 water molecules in 12 Å^3 box. GAP trained from 300 CUR selected points from an MMMD trajectory with DFTB(3ob) as the ground truth method. a) Snapshot of the NVT GROMACS dynamics at 300 K, 100 ps, 1 fs timestep. b) Top: parity plots of the 300 configurations used in training to DFTB(3ob) ground truth for energy and force. Bottom: Party plots against validation DFTB(3ob) data not seen during training. c) Snapshot of dynamics propagated using the trained GAP.

field) at 300 K using the TIP3P flexible water model (to ensure intramolecular energies were sampled) in GROMACS.[180] From this trajectory, we used CUR selection between SOAP descriptors to truncate the data to the 300 most diverse configurations from 1000 evenly spaced frames.[178]

Overall, the GAP obtained from this training strategy reproduces ground truth energies within a few kcal mol^{-1} for most configurations. Likewise, the majority of force predictions are within a few kcal mol^{-1} Å^{-1}. However, the MLP is unable to extrapolate on unseen data, resulting in errors $\geq$ 10 kcal mol^{-1} in energies. The dynamics propagated using this potential result in unphysical configurations where the atoms in the water molecules quickly dissociate.

We further investigated classical MD training using the prospective error metric,τ_{acc}, **Eqn. 2.1**, in training a GAP for bulk water at various temperatures, **Fig. 2.5**.[1] At low tem-

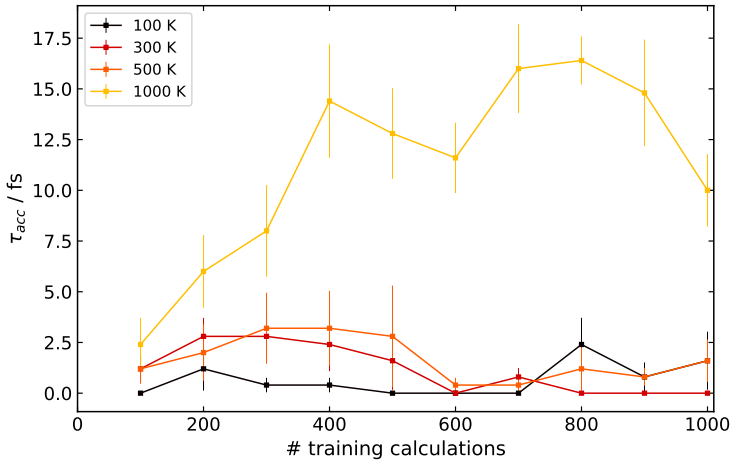


Figure 2.5: Learning curves for a bulk water (30 waters) GAP trained on MMMD configurations at different temperatures. Initial random configuration minimised then equilibrated for 1 ns, TIP3P parameters, flexible water. Configurations taken uniformly from a total of 10 ns of simulation time. τ_{acc} calculated with a 10 fs interval, $E_l = 0.1$ eV, $E_t = 1$ eV averaged over 5 initial random configurations. Error bars are standard error in the mean over 5 independent iterations.

peratures (≤ 500 K), the configurations generated only enable the GAP to obtain a τ_{acc} of < 5 fs, i.e., a few steps in a simulation. This value is not improved upon even when more data is added, up to 1000 configurations. When the temperature is increased to 1000 K, τ_{acc} improves up to ≈ 15 fs, peaking with 800 configurations. Interestingly, this value drops off when 1000 configurations are used. This could be due to the new configurations not accurately representing configurations on the true quantum PES and thus worsening the overall potential. However, these values are still close within their errors so may also be an artefact of incomplete convergence. Even with high temperature sampling, the resulting potential is unsuitable for MLP-MD simulations longer than a few steps ($\tau_{\text{acc}} = 0$ ps).

These results demonstrate that while using MD to generate training data is efficient, this strategy results in inaccurate potentials. This is possibly caused by the MD configurations not matching the DFTB(3ob) PES closely enough. A new approach of generating training data is required if we wish to train a potential efficiently without comprising accuracy. In the following sections, we describe our solution to this challenge, which combines the use of the τ_{acc} metric, AL and software development. These methods provide a means to systematically improve MLPs while also enabling the automation of their training, validation and practical

use.

2.2.4 Active learning training strategy

Additional training strategies were employed for GAP including using DFTB(3ob) AIMD directly to generate the training data. We generated energy minimised random configurations of water in a box by applying a random rotation and translation to each water molecule, subject to a minimum distance between waters of < 1.5 Å. We found that random performed slightly better than MMMD, and AIMD better than random, but both lead to insufficiently accurate dynamics. Minimised random: τ_{acc} (random) $= 0.0008 \pm 0.0007$ ps, requiring 7490 $\pm$ 20 training structures, AIMD: 0.011 ± 0.003 ps, requiring 2570 ± 10 training structures.[1] Depending on the properties we wish to calculate, dynamics may need to be run for many ps, and these potentials are unable to reach 1 ps of stability. Instead, work was proposed in the group for a training strategy based on MLP-MD and AL, where configurations are generated using MLP-MD and those with a large error are added back to the training iteratively.[1] Once the potential can produce dynamics that are stable for a chosen time, the training is finished. The AL training strategy is outlined in **Fig. 2.6**.

The AL strategy requires only a single configuration to begin training, from which 10 further configurations are generated by randomly displacing the atoms according to a Gaussian distribution. An MLP is trained from these 10 configurations. Next dynamics are run using the trained MLP for $2 + n^3$ fs, where n represents the iteration number of cycles reached within the current AL loop. This value is cubed to induce exploration of the dynamics slowly at first, before speeding up as n increases. After each short MLP-MD simulation, the final frame is analysed according to the chosen selector (e.g., *diff* or max atomic environment difference (MAED), see **Section 3.2.4**) and if it is within the threshold, it is returned and the potential is retrained with this new configuration. If, after the dynamics, the final frame is outside the selector threshold, the trajectory is backtracked until a suitable configuration is found for retraining. Otherwise, the first frame in the dynamics is returned for retraining. Finally, if the dynamics produce configurations all within the threshold, the AL loop breaks

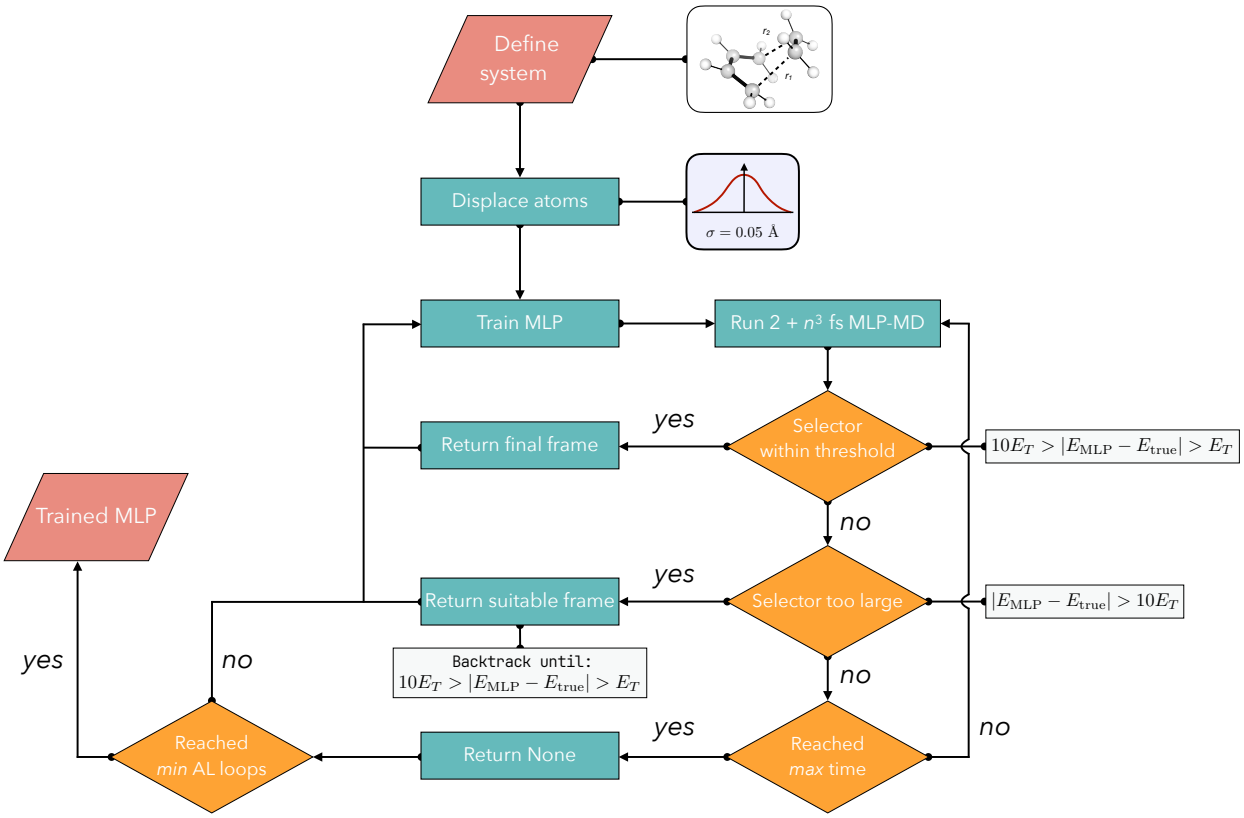


Figure 2.6: Diagrammatic workflow of the AL training strategy used in *gap-train* (*vide infra*). MLP-MD representing dynamics propagated using the MLP trained up to that point, with n corresponding to the current number of loops reached within each AL cycle. Parameters for the energy threshold, min and max AL iterations and MD conditions can be chosen.

and no configurations are returned for retraining. The AL cycle begins again and repeats until no new configurations are found (i.e., the potential has been sufficiently trained such that is stable for a given length of time), or the maximum number of AL cycles is reached. A final potential is returned based on the AL training.

Using this approach, a GAP was trained on bulk water using DFTB(3ob) and the resulting potential reached a τ_{acc} of 31 ± 7 ps with 1160 ± 90 configurations. Note that the 1160 configurations were required to reach the maximum possible τ_{acc}, but fewer configurations could be used to generate a stable potential. The advantage of this method is that training occurs exactly in the region of interest, i.e., dynamics in the canonical ensemble, resulting in a stable potential when MLP-MD is propagated. Moreover, only configurations for which the potential has a large error or are diverse from the others are added to the training, enabling efficient training of the potential. This AL strategy is used herein for all training. One drawback of this method is the uncertainty in the training time. With a fixed number of configurations, one can estimate how long training will take, whereas the AL approach is partly stochastic. Nonetheless, we have found that training typically takes between a few hours and a few days, depending on the complexity of the system.

2.3 Development of automated MLP training

Improving the efficiency of the training strategy for an MLP, i.e., fewer training points, equates to fewer ground truth evaluations needed, saving time and resources. However, an additional cost associated with the generation of MLPs is the user input, as it requires expert knowledge to generate training data and validate the MLP. Typically, each step requires the use of a specific programme and custom input scripts. For example, generating training data (e.g., via random, MMMD, AIMD etc.) requires the appropriate programme to be employed separately and the resulting data is fed into the GAP and QUIP code to be trained.[87]

a)

```python
"""Train a GAP for a gas phase benzene molecule with a DFTB ground truth"""
import gaptrain as gt
gt.GTConfig.n_cores = 10

# Initialise a cubic box 10x10x10 Å containing a single benzene molecule
benzene = gt.System(box_size=[10, 10, 10])
benzene.add_molecules(gt.Molecule('benzene.xyz'))

# Train a GAP at 1000 K using DFTB as the ground truth method
data, gap = gt.active.train(benzene,
                            method_name='dftb',
                            validate=False,
                            temp=1000)

# Run 1 ps molecular dynamics using the GAP at 300 K using a 0.5 fs time-step.
# The initial configuration is methane located at a random position in the box
traj = gt.md.run_gapmd(configuration=benzene.random(),
                       gap=gap,
                       temp=300,          # Kelvin
                       dt=0.5,            # fs
                       interval=5,        # frames
                       ps=1,
                       n_cores=4)

# and save the trajectory for visualisation in Avogadro, VMD, PyMol etc.
traj.save(filename='traj.xyz')
```

b)

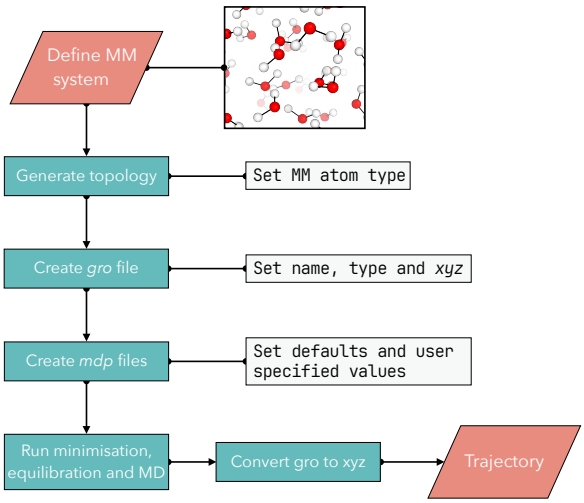


Figure 2.7: a) Example script used to train a GAP for benzene with DFTB(3ob) in *gap-train*. The GAP is trained using AL and a 1 ps trajectory is generated. b) Schematic for MMMD (GROMACS) implementation in *gap-train*. The topology, gro and mdp are generated automatically followed by minimisation, equilibration and production MD. The GROMACS trajectory is then converted into an *xyz* trajectory that *gap-train* can process.

2.3.1 *gap-train*

To facilitate the routine training of MLPs, including the AL strategy from **Fig. 2.6**, and their use, we developed *gap-train*.[1] This open source package enables a GAP to be trained using provided configurations (e.g., from MMMD) or using the AL strategy. It can also generate configurations using random displacements, MMMD (via GROMACS) or AIMD (via DFTB+). These methods can be called from a single function, requiring minimal human input. In addition, GAP-MD can be run using the potential in one line of code, enabling dynamic properties such as RDFs or bifurcating pathways to be generated readily. An example script for training a GAP and running GAP-MD is given, along with the author's contribution to the package with the MMMD interface, **Fig. 2.7**.

Using *gap-train*, training a potential and running MLP-MD is reduced to a few lines of Python code. The ease of training, combined with the AL strategy enabled a range of molecular systems to be studied. These are detailed in the *gap-train* reference, and include water, aqueous Zn(II), metallocage dynamics, gas and solution phase bimolecular nucleophilic substitution and post-TS bifurcating pathway in a Diels–Alder reaction.[1]

	Feature	*gap-train*	*mlp-train*
Training	Random, MMMD, AIMD, AL	All	Random and AL
	MLPs	GAP	GAP, ACE, NequIP
	Validation	τ_{acc}	τ_{acc}, E and F parity plots
Dynamics	MLP-MD	GAP	GAP, ACE, NequIP
	Bias	Momenta	Momenta and bias
	Free energy	None	US and metadynamics

Table 2.1: Highlighted differences between the *gap-train* and *mlp-train* Python packages.

2.3.2 *mlp-train*

During the development of this software and investigation of systems using GAP, ACE (detailed in **Section 1.2.3.2**) was gaining popularity. In addition, an interest in computing the free energy using MLPs became a stronger interest in the group. These factors, combined with a view to restructure and improve the *gap-train* code, led to a rewriting of the code, presented in a new package named *mlp-train*.[2] The major differences and improvements between *gap-train* and *mlp-train* are highlighted in **Table 2.1**.

The syntax for training and running dynamics is largely unchanged between the two software. Training can still be performed in one line, as can dynamics. The major differences are the implementation of the ACE and NequIP MLPs alongside GAP. Through comparisons of GAP, ACE and NequIP for training time, number of configurations required for AL and the MAD (relative to AIMD) on the Diels-Alder reaction of ethene and butadiene, we concluded that ACE was the most suitable potential for studying this type of molecular systems.[2] Therefore, the remaining MLP calculations in this thesis were conducted using ACE.

While the τ_{acc} metric provides a good comparison on the stability of a potential, particularly when comparing between training strategies, it is time-consuming to calculate. Therefore, since AL was shown to be the most effective in training a molecular system, dynamics and parity plots of energy and force are used in the place of τ_{acc}. These plots, generated automatically by *mlp-train*, show the predicted and ground truth energies and forces of a MLP-MD simulation. These are less computationally expensive and serve as a sufficient

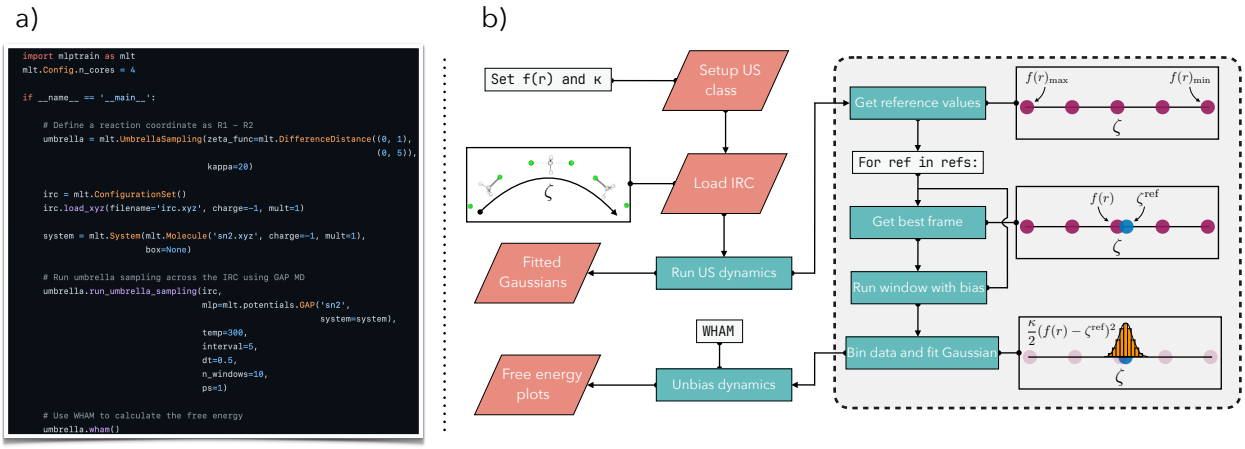


Figure 2.8: a) Example script used to run MLP-US for an S_N2 reaction in *mlp-train*. After defining a reaction coordinate and κ, a previously generated IRC is loaded in. US is then performed with a user chosen number of windows and time length. Finally the free energy is calculated using WHAM. b) Schematic for the US implementation in *mlp-train*. Reference values are chosen automatically and the closest frame from the IRC is chosen as the initial frame for the biased dynamics. Plots for the biased distribution and free energy are generated automatically.

measure when considering the stability and accuracy of an MLP-MD simulation.

The other major difference between the software is the inclusion of enhanced sampling methods, such as US (both self-contained and interfaced to plumed) and metadynamics (interfaced to plumed).[144] This addition enables free energy to be calculated along a reaction coordinate using the trained MLP. Initially, US was implemented directly in *mlp-train* for flexibility of automation. Reference values could be automatically chosen from a provided *xyz* IRC. The weighted histogram analysis method (WHAM) can be invoked automatically, generating a free energy plot. Later versions of *mlp-train* include the implementation for plumed, allowing for a greater diversity of reaction coordinates as well as metadynamics. An example of how to perform MLP-US along with the code design for the implementation of US is shown in **Fig. 2.8**.

After unit testing, *mlp-train* was used to train a range of Diels-Alder systems.[2] The US implementation was applied to the reaction of MVK and CPD to calculate the activation free energy using an ACE potential trained at PBE0-D3BJ/def2-SVP, **Fig. 2.9**.

One uncertainty with the trained ACE potentials using AL was whether they would remain

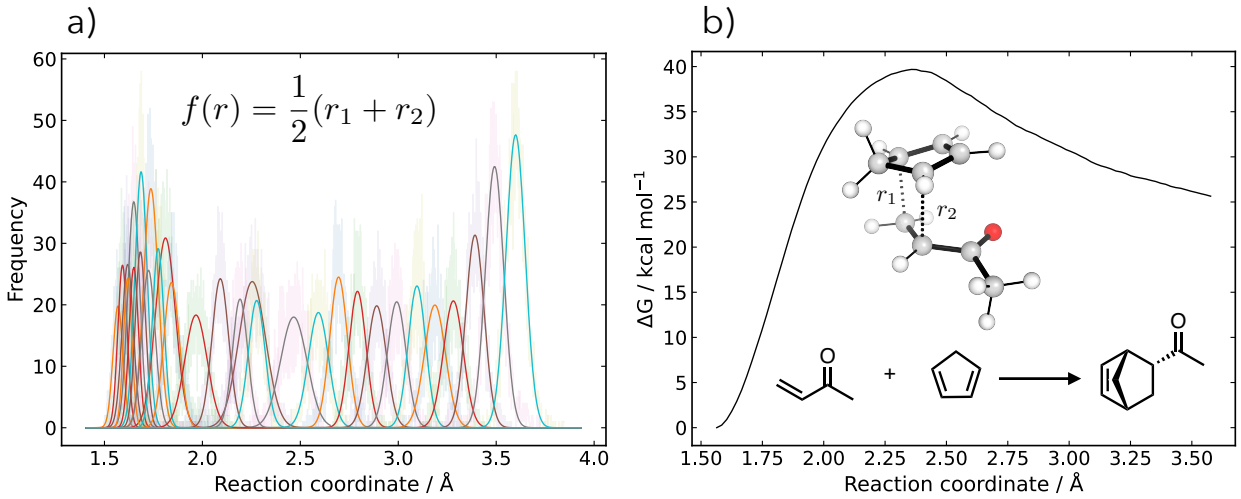


Figure 2.9: Umbrella sampling simulation of MVK and CPD reaction using an ACE potential trained at PBE0-D3BJ/def2-SVP from the TS. US run at 300 K, 0.5 fs time step, 10 ps per window, 20 windows equally spaced in [1.564, 3.5765] Å with $\kappa = 10$ eV Å^{-2} and 10 equally spaced over [1.7, 2.3] Å with $\kappa = 20$ eV Å^{-2}. a) Histogrammed data and fitted Gaussian from the US simulation. b) Free energy profile generated using WHAM to unbias the US simulation.

stable when constrained along an IRC, e.g., under a US bias. The dynamics for the MVK and CPD US simulation were stable for each 10 ps window, generating a smooth free energy profile. Due to the difference in free energy gradient on the reactant side versus the product side, additional strongly constrained windows were included between r [1.7, 2.3] Å. Further validation of the US implementation and trained ACE potentials was conducted for other Diels-Alder reactions and documented in ref [2]. Using *mlp-train* for training and US, other molecular systems can be studied with limited human input; this is the focus in **Chapter 3**.

2.4 Conclusions

In this chapter, we have investigated different methods for training and validating machine learnt potentials (MLPs), using PhysNet and GAP. We then developed *gap-train* and *mlp-train* to automate training, validation and free energy computation, with the inclusion of ACE in *mlp-train* due to its efficiency and accuracy for molecular systems.

First, we used PhysNet to train potentials for both a water monomer and a cluster. Training the monomer using configurations along the bend and stretch PES resulted in a stable and accurate potential. To account for intermolecular interactions in bulk water, we utilised the random displacement method for generating training configurations. While this was found to be ineffective by itself, combining this data with the intramolecular PES configurations, a stable potential was generated, albeit with the intermolecular interactions being under-prediced by PhysNet. However, the main drawbacks of this approach were the high data requirement (1800 configurations) and specific knowledge of the intramolecular PES of the molecule. Therefore, we sought a more general and efficient approach to training a potential.

With the success of GAP for a range of chemical systems and the low data requirement, we focused our efforts on developing a stable water potential using molecular mechanics molecular dynamics (MMMD) simulations as training data. While training produced moderate accuracy, the potential failed more drastically on the validation data. In addition, using this potential for GAP-MD resulted in unstable dynamics after a few steps. Using the newly developed prospective error metric, τ_{acc}, we found that validation in the region of interest, i.e., dynamics, led to more accurate potentials.

It was found that using random, MMMD or AIMD data to train a GAP was inefficient and led to potentials stable only for a few fs. The AL strategy outlined in this chapter enabled stable potentials to be trained for bulk water and other systems. The AL strategy was combined with a Python package we developed, *gap-train*, resulting in automated training, validation and dynamics for a system using GAP. This package was superseded by *mlp-train* to include the ACE and NequIP MLPs, as well as umbrella sampling, allowing free energy

calculations from MLPs to be run automatically. This implementation was taken forward to test a range of Diels-Alder systems.

The aim of this chapter was to develop a robust and automated training strategy for MLPs, particularly for GAP and ACE. If we wish MLPs to be routinely used by computational scientists in reaction analysis, minimal user effort is required in both the training and running dynamics. We believe that *mlp-train*, combined with the AL strategy, can provide a method for achieving this flexibility and ease. To this end, the following chapter comprises a range of organic systems trained using *mlp-train* and AL, which are used to compute free energies and simulate reaction dynamics.

3

Reaction Dynamics from MLPs

Contents

3.1 Summary 76
3.2 Optimisation of MLP training 77
3.2.1 Training an ACE potential from reactants 78
3.2.2 Training an ACE potential from the TS 79
3.2.3 Automation of training along a reaction coordinate 84
3.2.4 Choice of MLP selector 87
3.3 Methyl shifts in isoindene 87
3.3.1 Training an isoindene ACE potential 90
3.3.2 Free energy of the isoindene reaction 91
3.3.3 Path integral molecular dynamics of isoindene 96
3.4 Hydride shifts in terpinenes 98
3.4.1 Training a terpinene ACE potential 98
3.4.2 Free energy of the terpinene reaction 99
3.4.3 Path integral molecular dynamics of terpinene 100
3.5 Product ratios in indanes 103

3.5.1 Retraining an ACE potential of indane . . . 103
3.5.2 Uphill dynamics of indane . . . 103
3.5.3 Path integral molecular dynamics of indane . . . 106
3.6 Bifurcating surfaces in sesquiterpenes . . . 108
3.6.1 Training an ACE potential on sesquiterpene . . . 109
3.6.2 Product ratios of sesquiterpene . . . 111
3.7 Conclusions . . . 112

3.1 Summary

In the previous chapter, we introduced an active learning (AL) training strategy to develop MLPs for molecular systems, with a particular focus on organic reactions. In this chapter, we apply this AL training strategy to generate robust ACE MLPs for modelling H-transfer and rearrangement reactions in terpene systems (detailed in **Section 1.3**), which are relevant in enzymatic biochemistry.[181, 182] We discuss the challenges associated with the training of MLPs for chemical reactions, including the need to efficiently generate training configurations far from the minima, along with potential solutions.

The chapter begins in **Section 3.2** by applying the AL strategy to model a methyl shift reaction of an indane. We discuss different strategies for generating initial training configurations as well as different selectors, evaluating their accuracy and efficiency. We also present an algorithm for systematically improving the accuracy of the potential along the reaction coordinate.

In **Section 3.3**, we train an ACE potential and perform umbrella sampling (US) to generate a free energy profile for a methyl shift of an isoindene, comparing reaction rate constants with experiment as well as static DFT calculations. Furthermore, we explore the nuclear quantum effects (NQEs) on a modified system by introducing path integral molecular dynamics (PIMD).

Section 3.4 extends the ACE potentials to the study of a hydride shift in a terpinene carbocation. A free energy profile is generated using MLP-MD and a comparison of the DFT

TS to the free energy surface TS is made. PIMD is then utilised to predict the difference in free energy between classical and quantum dynamics.

The product ratio from the indane-derived carbocation is predicted using MLP-MD and uphill dynamics in **Section 3.5**. The results are compared to a previous computational study and using the fast evaluation power of the MLP, additional simulations are run to ensure convergence. PIMD is combined with the potential to investigate whether NQEs affect the product ratio.

Finally, in **Section 3.6**, we explore the proposed bifurcation of a sesquiterpene system, where the ACE potential is trained at three different levels of theory. Dynamics are initiated from the corresponding TSs and the ratio of products is measured.

We end the chapter with a discussion of the results and a look to the future of MLPs in modelling reactive systems in **Section 3.7**. Details of the computational methods used in this chapter can be found in **Appendix B.2**.

3.2 Optimisation of MLP training

Previous work has focused on bimolecular reactions, such as Diels-Alder and S_N2 reactions using GAP and ACE.[1, 2] Here, we extend the training protocol to study more complex chemically and biologically relevant reactions, namely unimolecular reactions where one or more atoms migrate from one part of the molecule to another. We utilise the AL strategy, outlined in **Section 2.2.4** and implemented in *mlp-train*.

We begin by investigating the optimal AL training strategy for a model indane system, **Fig. 3.1**, developed by Tantillo and Feng[105], described in **Section 1.3.3**. We chose this system due to the availability of computational data as well as representing the class of terpene reactions we later study.

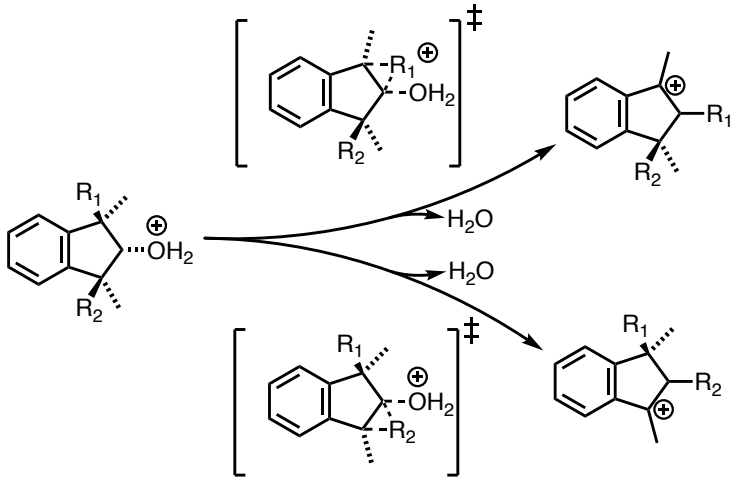


Figure 3.1: Indane-derived reaction where water is removed resulting in a bifurcation of R_1 and R_2. After migration of either R_1 or R_2, a carbocation is formed on either side of the 5-membered ring.

3.2.1 Training an ACE potential from reactants

Initially, the reactant was minimised using the molecular mechanics universal force field (UFF)[183] and used to train an ACE potential at PBE/def2-SVP with AL. The maximum propagation time in the AL loop was set to 500 fs, meaning once the potential was stable for 500 fs, the AL loop would stop. However, this approach led to a moderately high mean absolute deviation (MAD) energy error in MLP dynamics of 0.07 eV (1.6 kcal mol^{-1}).

We switched to PBE0 due to its improvements in energies and geometries over PBE, while maintaining a relatively low computational cost.[1, 167] In addition, we optimised the starting structure for AL at PBE0/def2-SVP. We also increased the maximum propagation time in the AL loop to 1 ps to ensure greater stability over a long time. Using the PBE0 optimised reactant configuration as the starting point in AL, an ACE potential was trained, **Fig. 3.2**.

Validation of this potential from 500 fs MLP-MD showed over a two-fold accuracy improvement (MAD of 0.026 eV (0.6 kcal mol^{-1})) compared to the original potential trained at PBE/def2-SVP with the UFF minimised starting structure. The dynamics remain stable for the full 500 fs, with a MAD in force below 0.1 eV $Å^{-1}$.

While generating a stable potential for this system required only 81 single point config-

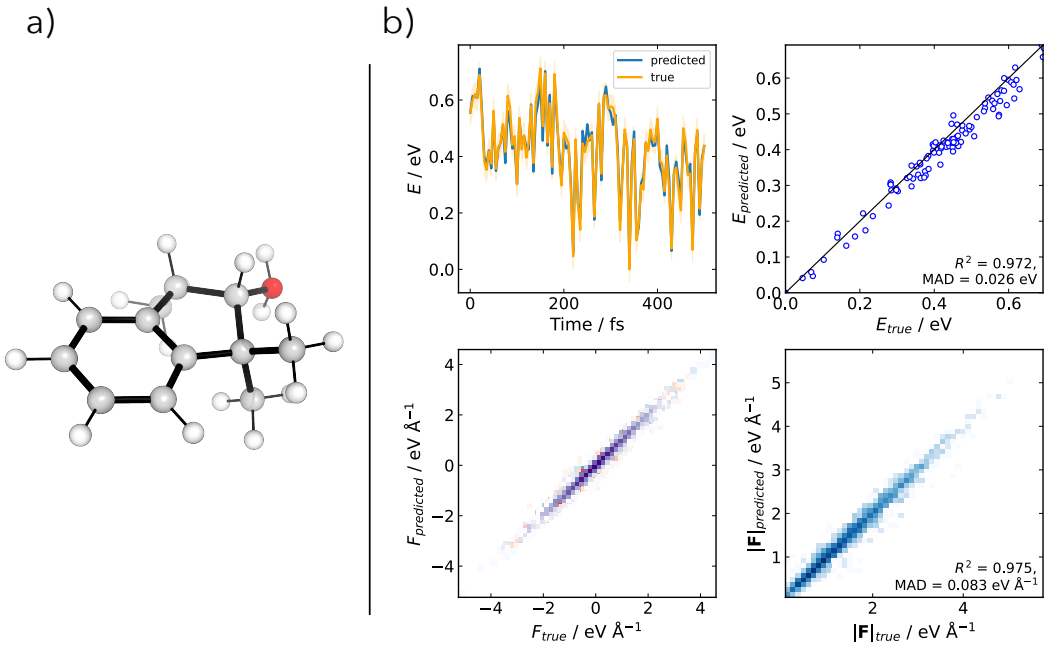


Figure 3.2: MLP dynamics of indane-derived reactant (R_1=Me and R_2=H). ACE potential trained at PBE0/def-SVP using the standard *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 1 ps. a) Reactant structure minimised using PBE0/def2-SVP. b) Ground truth and ACE energies over a 500 fs MLP-MD simulation at 300 K. Parity plot between true and predicted energies, force components and force magnitude.

urations in the AL, the potential is only suitable for dynamics run around the reactant minimum. To extend the ability of the potential to accurately predict the energy across the whole reaction coordinate, configurations must be generated that correctly describe the system along it.

3.2.2 Training an ACE potential from the TS

We previously found that training a potential along a reaction coordinate using AL is most efficiently done when the initial configuration is fixed as close to the TS as possible.[1] This is because each subsequent MD simulation that occurs in the AL cycle will begin from the TS with $\approx 50\%$ probability of leading to either reactants or products, ensuring the generated configurations lie on both sides of the TS. While TS coordinates have been published for this reaction[105], we sought to develop a general approach to generate IRCs and TS structures suitable for MLP training.

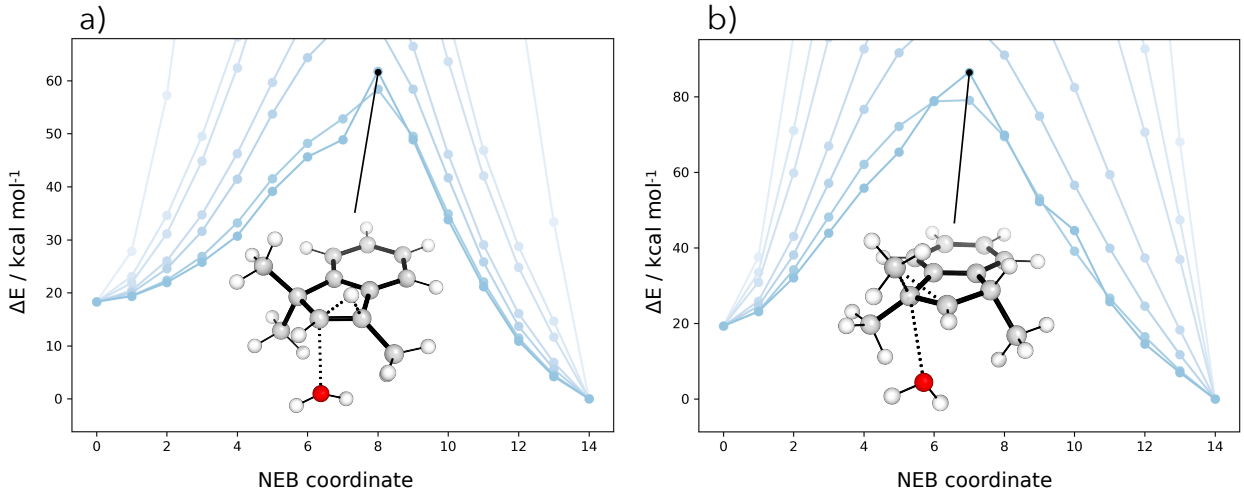


Figure 3.3: CINEB energies and highest energy structure along the NEB coordinate using 15 images with reactants and products optimised using PBE0/def2-SVP for a) H migration and b) Me migration of the indane derived reaction. Each curve represents one NEB cycle, with the deepest colour representing the final converged pathway.

3.2.2.1 AL via a relaxed PES scan

First, a PBE0/def2-SVP relaxed PES scan was performed over the H–C distance in the H-migration reaction of indane, from where the highest energy structure (≈ TS) was taken as the initial configuration to train an ACE MLP. Validation MLP-MD was run from the initial training structure and an MAD error of 0.041 eV (0.9 kcal mol^{-1}) was achieved. Despite the low error of the potential, the configurations generated by the AL loop were unable to cover the reactant region, with most configurations being similar to either the TS or products.

3.2.2.2 AL via climbing image nudged elastic band

We then turned to the climbing image nudged elastic band (CINEB) method to generate a reaction path and identify a TS configuration. CINEB only requires reactant and product structures and generates a minimum energy pathway between these two configurations. CINEB, as opposed to NEB, drives the highest energy image to a saddle point, thus seeking a TS. A CINEB calculation for both the H migration and Me migration was performed, **Fig. 3.3**.

The pathway generated using CINEB for the H/Me migrating reaction shows a concerted ring flip of the 5-membered ring as the H/Me group migrates. The energy cost associated

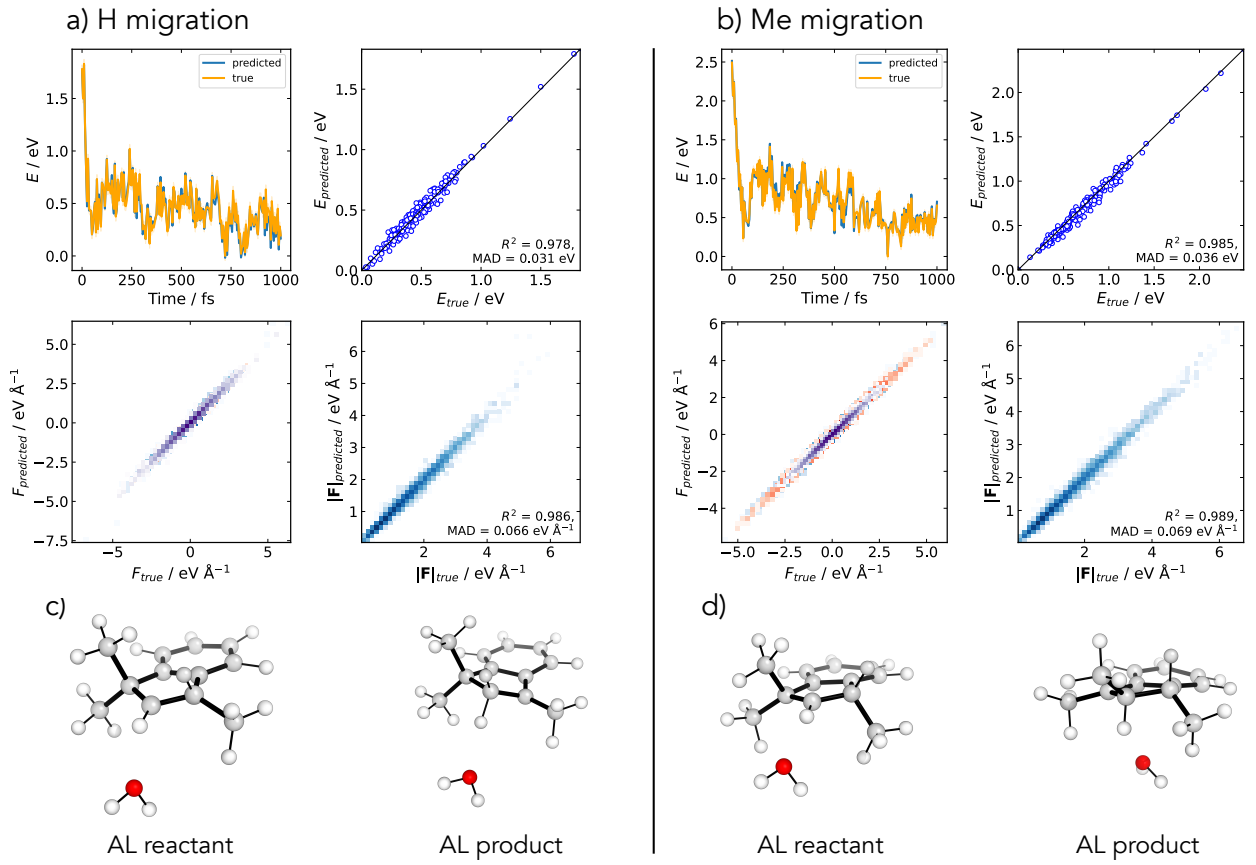


Figure 3.4: MLP dynamics of indane-derived migration reactions (R_1=Me and R_2=H). ACE potential trained at PBE0/def-SVP using the standard *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 1 ps. Training initialised from CINEB TS structure. Ground truth and ACE energies over a 1 ps MLP-MD simulation at 300 K. Parity plot between true and predicted energies, force components and force magnitude. a) Energy and force error analysis trained on H migration only. b) Energy and force error analysis trained on H and Me migration. Selected structures from AL cycles for c) H migration and d) Me migration.

with this process may in part contribute to the high observed energy barrier. Using the peak configuration from the H migration CINEB path, an ACE potential was trained using AL, followed by 1 ps of MLP-MD for validation, **Fig. 3.4a**. A second potential was trained by combining the AL configurations from the H migration with an AL cycle initiated from the highest Me migration CINEB structure, **Fig. 3.4b**.

For the combined potential, 570 configurations were generated from the AL loop. This is significantly more than what was required for describing the reactant minimum (81 configurations). Both reactants and products were seen for each migration reaction in the AL, suggesting the full reaction coordinate space was learned by the ACE potential.

While training required 570 configurations, the energy error of the potential for both migration reactions was below 1 kcal mol^{-1}. Moreover, the stability of the potential demonstrates additional configurations can be added to the potential without losing accuracy. This is particularly useful for training reactions with more than one reaction pathway. Interesting, for three runs of each reaction from the TS, the products were quickly formed. This suggests that the transition state from CINEB may be slightly biased towards the product side.

3.2.2.3 AL via CINEB + TS optimisation

To investigate and potentially address this TS issue, an ACE potential was trained on both the H and Me migrations using AL initiated from a PBE0/def2-SVP optimised TS, which itself was obtained from the CINEB TS. The CINEB TS and QM optimised TS structures were also compared, **Fig. 3.5**.

Interestingly, this approach only required 263 configurations in the AL loop with QM TS optimisation, compared to 570 configurations generated without TS optimisation. Comparing the two TS states from each migration, we can see that the QM optimised TS has a significantly different geometry compared to the one obtained via CINEB, with migrating groups and water differing in their positions relative to the indane ring. These results suggest that starting AL from a closer-to-true TS is likely to result in better coverage of the reaction coordinate and thus fewer training configurations. However, a quantitative measure of where training configurations lie along the IRC would provide better insight into AL initiated from the TS (*vide infra*).

The comparative analysis of these three AL strategies (PES scan, CINEB and CINEB + TS optimisation) demonstrate that training is highly sensitive to the choice of the initial configuration used for AL. Since dynamics are used in the AL loop, selecting a configuration that leads to an even spread of configurations on either side of the TS is key. An accurate TS, for example one obtained via the Eigenvector-Following algorithm, leads to fewer configurations in the AL loop than those providing estimated TSs.[6] However, questions remain about the stability of the potential for longer timescales as well as under-explored PES regions.

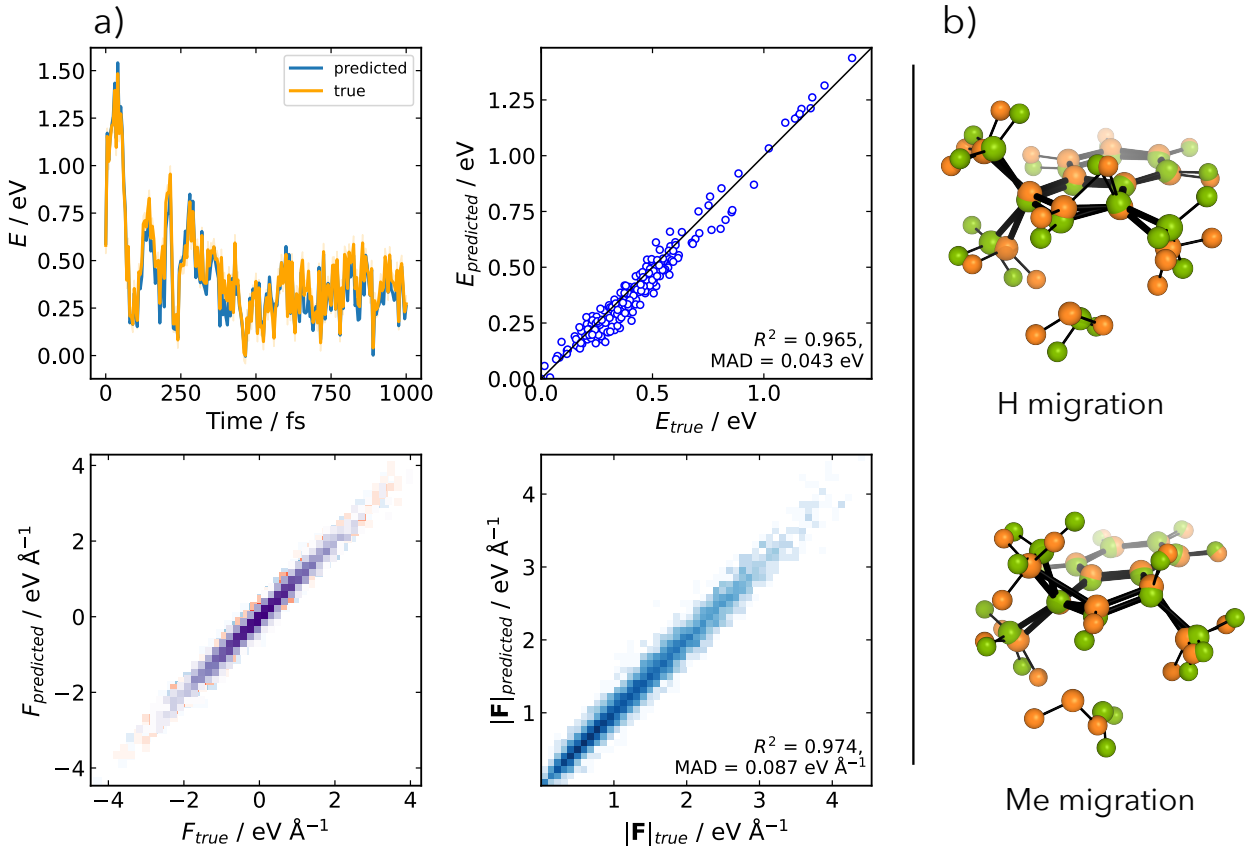


Figure 3.5: MLP dynamics of indane-derived Me migration reaction (R_1=Me and R_2=H). ACE potential trained at PBE0/def-SVP using the standard *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 1 ps. Training initialised from PBE0/def2-SVP TS structure. a) Ground truth and ACE energies over a 1 ps MLP-MD simulation at 300 K for the potential trained on H and Me migration. Parity plot between true and predicted energies, force components and force magnitude. b) CINEB TS (orange) aligned with PBE0/def2-SVP TS (green) for Me-TS (top, RMSD: 0.29 Å) and H-TS (bottom, RMSD: 0.44 Å).

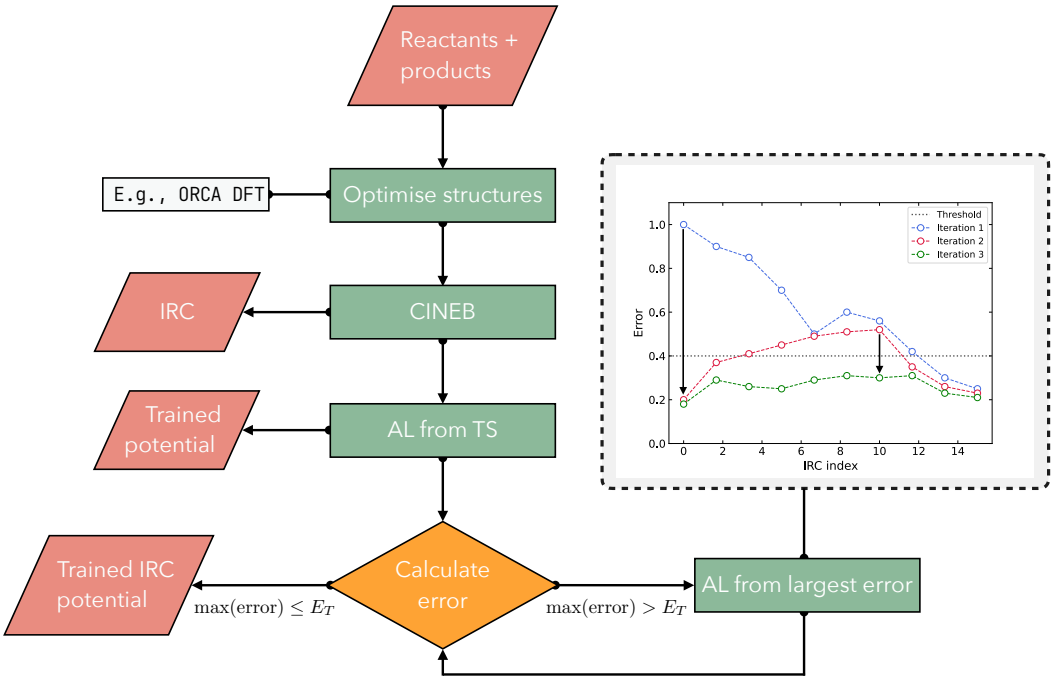


Figure 3.6: Diagrammatic flowchart outlining automated MLP training along a CINEB coordinate. An AL cycle is initiated from configurations with the highest energy (MLP prediction versus ground truth). This is repeated until all configurations along the CINEB coordinate have an error $\leq E_T$, a chosen error threshold. Graph shown is an imagined example of iteratively improving the accuracy of the potential along the IRC using the described method.

3.2.3 Automation of training along a reaction coordinate

Using the ACE potential for indane trained from the CINEB + QM optimised TS, the stability of the potential over long time simulations (10 ps), dynamics along a constrained reaction coordinate (umbrella sampling) and uphill dynamics (excess momentum placed in a breaking bond) were tested. Long time dynamics initiated from both the reactant and product state were visually stable. However, when 10 × 20 ps of US simulations were performed across the CINEB coordinate, the system was seen to 'explode' around the TS, i.e., all the atoms dissociated from one another. This suggests that the potential is poorly trained in certain regions along the reaction coordinate.

3.2.3.1 Method development

To overcome this problem, we test an automated method to train a potential along a reaction coordinate, outlined in **Fig. 3.6**.

In this method, reactant and product structures are optimised using a ground truth method, e.g., DFT. An IRC is then generated by employing CINEB, using the highest energy point as the starting structure for the AL (where this TS can be further optimised using QM). The energy obtained from the trained MLP is compared to the ground truth energy for each point along the IRC. If the maximum error is greater than a user-defined threshold, E_T, AL is initiated from the configuration with the greater error, retaining the previous configurations in the retraining of the MLP. This cycle is repeated until all the configurations have an error $\leq E_T$.

The RMSD between each configuration in the AL and the IRC is calculated using the Kabsch algorithm in *autodE* and the AL configurations are binned according to which IRC configuration they are closest.[184] This gives an indication of where the potential has learnt the region along the IRC. The value of the RMSD between the closest configuration in the AL and the IRC is also analysed, showing how far away a configuration is from the IRC.

3.2.3.2 Application to indane

We applied this strategy to the H migration in the indane system, generating the following MLPs, **Fig. 3.7**. Using the maximum energy error as a guide to choosing the configuration to initiate the next AL training from, we see a systematic decrease in error after three AL iterations. After the reactant was introduced into the AL sampling, the energy error across the IRC was below 0.1 eV for all configurations. This method shows that the error can be systematically improved by introducing configurations in poorly sampled areas.

Other improvements can be made to this algorithm. For example, AL may be be unnecessary to reduce the error at a particular point. Instead, adding a single configuration, i.e., the configuration with the largest energy error, may be sufficient. Since AL dynamics will naturally fall towards an energy minimum, applying a harmonic bias to a bond will constrain the sampling in the region of interest.

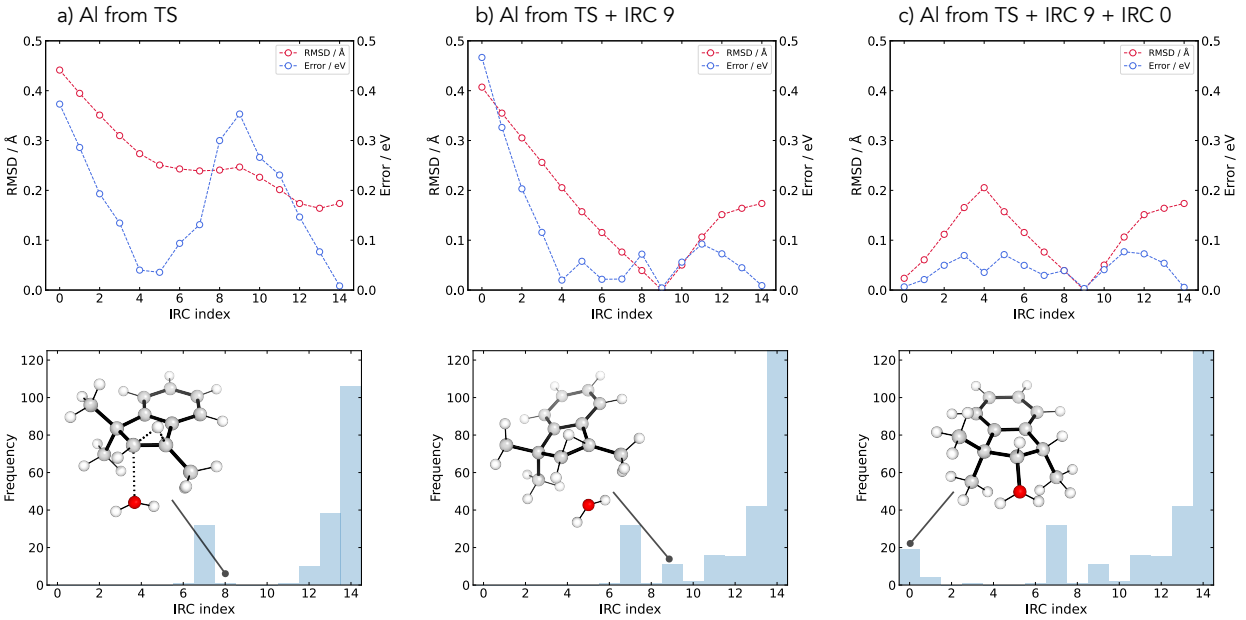


Figure 3.7: Top row: absolute energy error and RMSD along the IRC for the H migration reaction from an ACE potential trained using AL at PBE0/def2-SVP. RMSD is calculated from the IRC configuration to the closest point in the AL configurations. Bottom row: number of configurations in the AL closest to a point in the IRC (water removed from RMSD calculation). a) ACE trained using AL initiated from the TS. b) ACE trained on TS AL configurations plus AL initiated from IRC index 9. c) ACE trained on TS AL, IRC index 9 and AL initiated from IRC index 0.

3.2.4 Choice of MLP selector

Choosing which configurations to add to the AL cycle is key to ensuring efficiency. So far, we have used the energy selector, referred to simply as *diff*, which is based on the absolute difference between the true and predicted energy (see **Fig. 2.6**). An alternative selector is the *max atomic environment difference* (MAED) selector. The MAED selector is based on the maximum distance between any of the training set configuration and a new configuration using the similarity SOAP kernel vector, k^* (see **Section 1.2.1.2**).[2]

Here, we evaluated the impact of the *diff* and MAED selector on the efficiency of the training strategy and accuracy of an ACE potential for the H migration reaction in indane (**Fig. 3.1**) trained using PBE0/def2-SVP. A 1 ps MLP-MD simulation was performed for both selectors, **Fig. 3.8**. For the MAED selector, different thresholds of k^* were tested.

Our results show that the training time is roughly the same for both selectors. However, even after hyperparameter tuning for MAED, the *diff* selector offers the same accuracy in the potential for approximately half as many configurations. Moreover, tuning the k^* parameter to obtain a suitable value for a particular system takes time, and the optimal value varies depending on the system. Therefore, for flexibility and the low number of configurations required, the *diff* strategy was taken forward for training.

Having identified a suitable selector and training strategy, we turned our attention to applying MLPs to a range of chemically and biologically interesting systems.

3.3 Methyl shifts in isoindene

One advantage of MLPs is that validation to the ground truth can be easily performed by comparing the MLP prediction to a single point energy or force. However, MLPs are often used to run dynamics, generating many thousands of configurations, and in turn, energies and forces over a simulation. Therefore, validation for each step is self-defeating and so turning to validation of ensemble properties is more efficient.

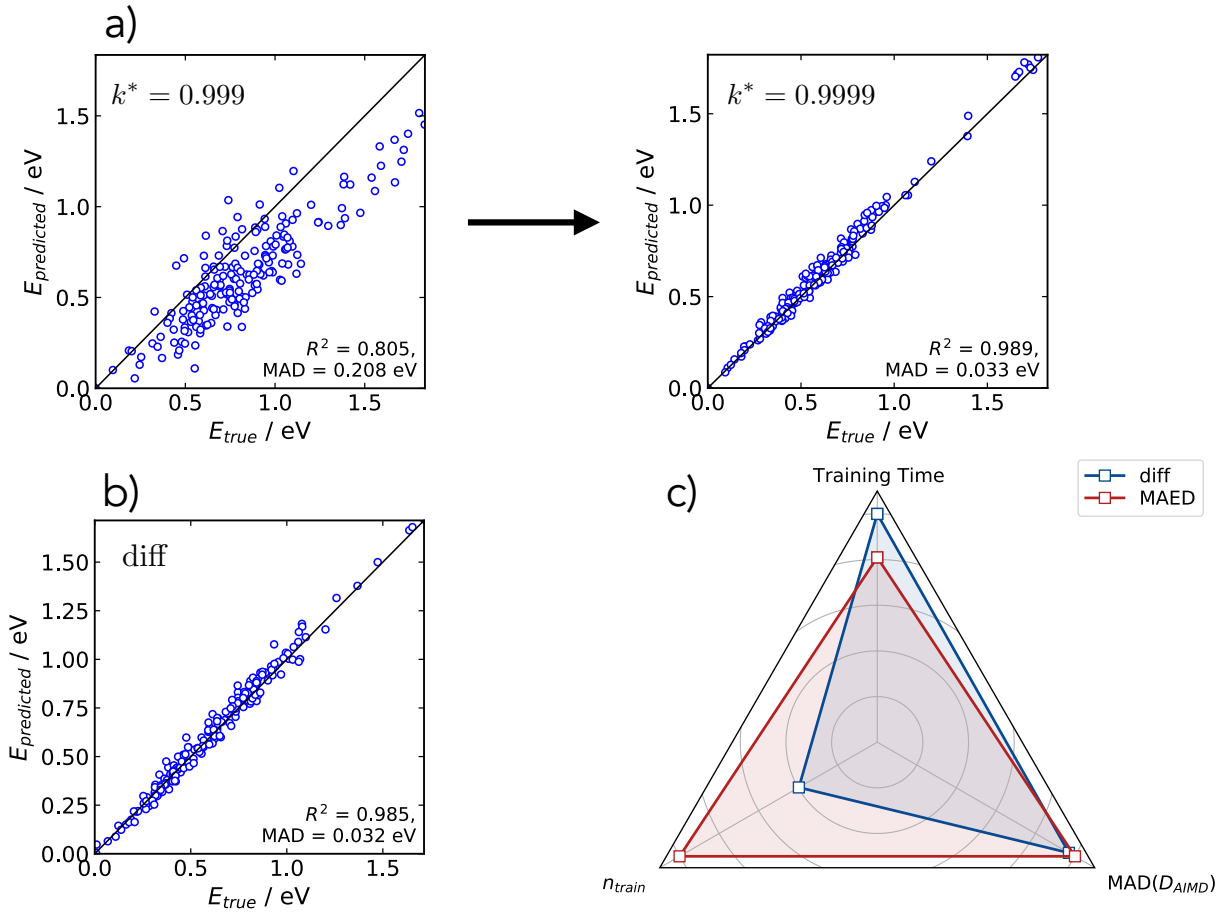


Figure 3.8: Comparison of *diff* and *max atomic environment difference* (MAED) selectors. An ACE potential was trained using each selector with AL and PBE0/def2-SVP level of theory on the H migration of the indane derived reaction before a 1 ps MD trajectory was run. Errors are taken between the predicted and ground truth from this trajectory. a) Parity plot using MAED selector with $k^* = 0.999$ showing change to when $k^* = 0.9999$. b) Parity plot for the *diff* method. c) Comparison of MAED ($k^* = 0.9999$) and *diff* for MAE, training time and number of training configurations required.

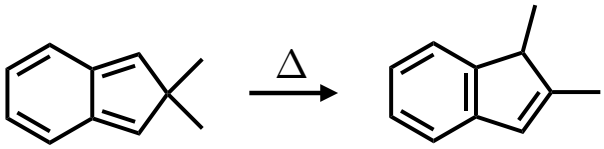


Figure 3.9: The thermally activated reaction of 2,2-dimethylisoindene to 1,2-dimethylindene via a 1,5-methyl shift.

Ideally, validation is made to experimental observables, such as reaction rate constants. For classically behaved systems, we can relate the rate constant, k, at a given temperature, T, to the activation free energy, $\Delta G^{\ddagger}$, using the Eyring equation, **Eqn. 3.1**.

$$k = \frac{\kappa k_{\mathrm{B}} T}{h} \mathrm{e}^{-\frac{\Delta G^{\ddagger}}{RT}} \tag{3.1}$$

Here, κ is the transmission coefficient, assumed to be unity for classical reactions, k_{B} is the Boltzmann constant, h is the Planck constant and R is the gas constant.

As a starting point, we studied the methyl migration reaction of 2,2-dimethylisoindene (detailed in **Section 1.3.1** and shown in **Fig. 3.9**), for which experimental data exists.[106] Experimentally, it was found that 2,2-dimethylisoindene would convert thermally to 1,2-dimethylindene in pentane at 90° C, while our potential was trained in the gas phase. However, considering the apolar non-protic nature of this solvent, we made the assumption that solvent effects would be negligible in this case.

The product, 1,2-dimethylindene, was found to be thermally stable to a hydrogen shift rearrangement. This makes the reaction ideal to study computationally, as the reaction only involves the methyl migrating group with no competing or subsequent pathways. Moreover, the original work proposed the rearrangement could occur via two pathways: a concerted mechanism or a dissociative radical mechanism.[106] They argued, on the basis of $\Delta G^{\ddagger}$, that it likely occurs via the concerted route. Our work further investigates this hypothesis by calculating $\Delta G^{\ddagger}$ along the concerted route and comparing to the experimental value.

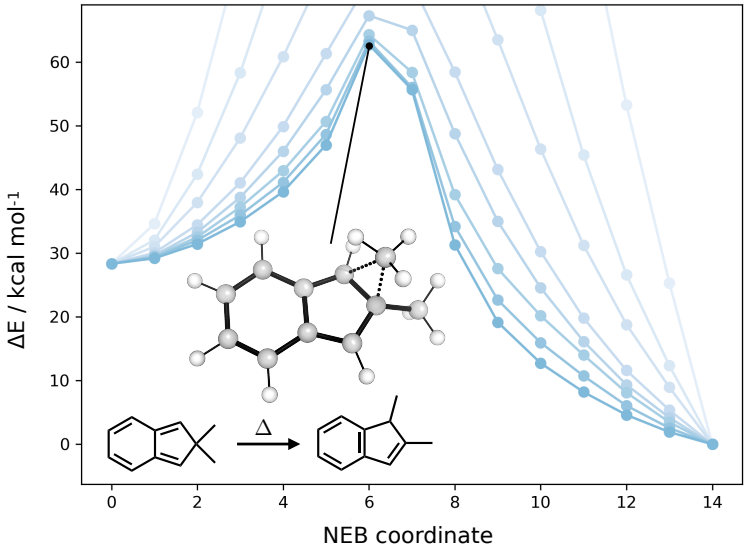


Figure 3.10: CINEB energies and highest energy structure along the NEB coordinate using 15 images with reactants and products optimised using PBE0-D3BJ/def2-SVP for the reaction of 2,2-dimethylisoindene to 1,2-dimethylindene. Each curve represents one NEB cycle, with the deepest colour representing the final converged pathway. The structure corresponding to the CINEB TS is shown.

3.3.1 Training an isoindene ACE potential

Training an ACE potential was initiated by first performing a CINEB calculation to generate a reaction coordinate, using the reactant and product structures optimised at PBE0-D3BJ/def2-SVP. From this CINEB pathway, an approximate TS was identified, **Fig. 3.10**.

The CINEB method generates a relatively smooth coordinate connecting the reactant and product, reflecting the change in *sp* character of the reacting carbons. The potential energy barrier, $\Delta E^{\ddagger}$, for this reaction was predicted to be 34.4 kcal mol^{-1}, higher than the experimental free energy barrier by over 5 kcal mol^{-1} (c.f. 29.2 kcal mol^{-1}).[106] We expected this prediction to change upon QM optimisation of the TS and inclusion of entropy via dynamics. Indeed, when this configuration was optimised to a TS, the barrier decreases by 6.4 kcal mol^{-1} to $\Delta E^{\ddagger} = 28.0$ kcal mol^{-1}, **Fig. 3.11a**. An IRC calculation was performed to confirm the TS linked the reactants to products, **Fig. 3.11b**.

This result highlights the limitation of using CINEB, which even though provides a good first guess, still requires a full TS optimisation to obtain a suitable TS structure. Using the optimised TS as an initial structure for AL, an ACE potential was trained at PBE0-

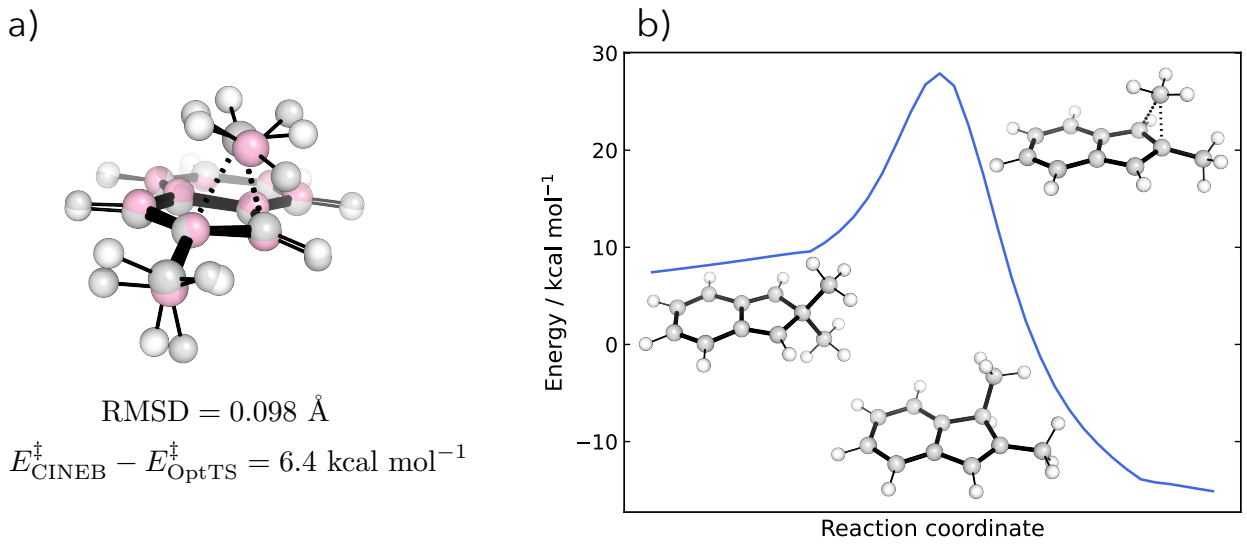


Figure 3.11: a) RMSD and absolute energy difference between CINEB TS and QM optimised TS (PBE0-D3BJ/def2-SVP). QM optimisation was initiated from the CINEB and confirmed with one imaginary frequency. Pink structure represents the CINEB TS, grey QM TS. b) PBE0-D3BJ/def2-SVP IRC for methyl shift in isoindene. The figure is truncated at 20 points either side of the TS due to the slow convergence (1000 points in IRC total). Energy minimised to zero on the reactant side. Potential energy of products is −56.1 kcal mol^{-1} below the TS.

D3BJ/def2-SVP for this reaction, **Fig. 3.12**.

The AL cycle required 131 configurations to reach 5 ps of stability. This is a fairly small training set compared with the more expensive training required for the H/Me migration reaction of the indane system, **Section 3.2.2.3** (263 configurations required for 1 ps of stability). This efficiency for the isoindene system could be due to the limited flexibility over the reaction and intramolecular nature with no water leaving groups. The error of the trained ACE potential is within 1 kcal mol^{-1} error (0.6 kcal mol^{-1}, 0.025 eV) on energy and shows good stability throughout the 2 ps simulation.

3.3.2 Free energy of the isoindene reaction

To validate the accuracy of the potential against experimental rates, we used the Eyring equation, **Eqn. 3.1**, to convert an activation free energy, $\Delta G^{\ddagger}$, to rate at a given temperature. $\Delta G^{\ddagger}$ is computed using US (detailed in **Section 1.4**), which is implemented in *mlp-train*.

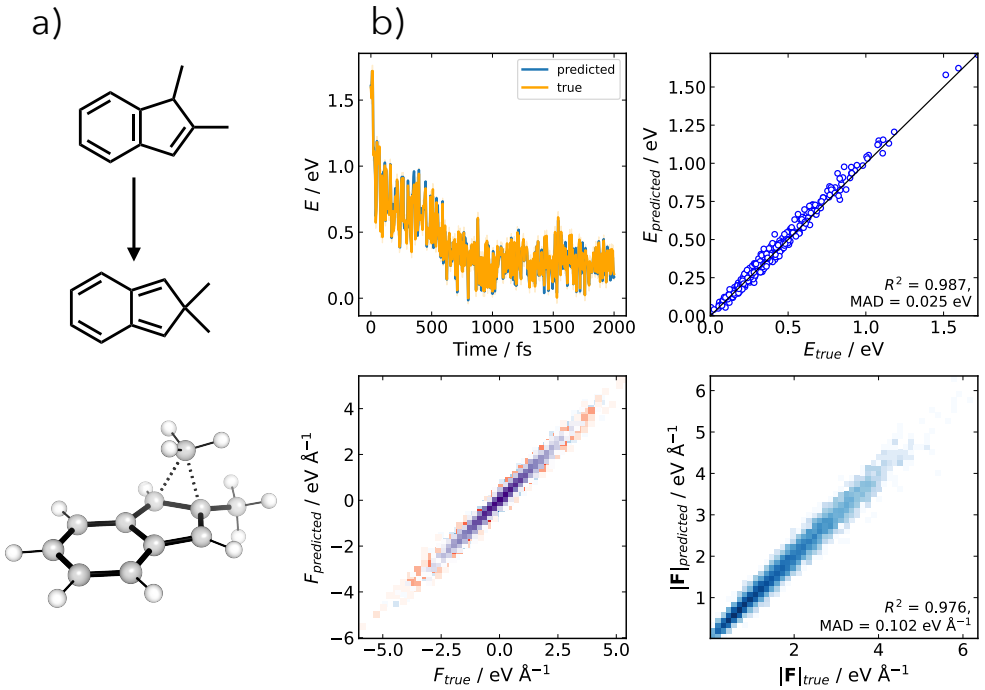


Figure 3.12: MLP dynamics of the 2,2-dimethylisoindene to 1,2-dimethylindene reaction. ACE potential trained at PBE0-D3BJ/def-SVP using the standard *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 5 ps. Training initialised from PBE0-D3BJ/def2-SVP TS structure. a) Schematic of the reaction with the QM TS structure below. b) Ground truth and ACE energies over a 2 ps MLP-MD simulation at 300 K. Parity plot between true and predicted energies, force components and force magnitude.

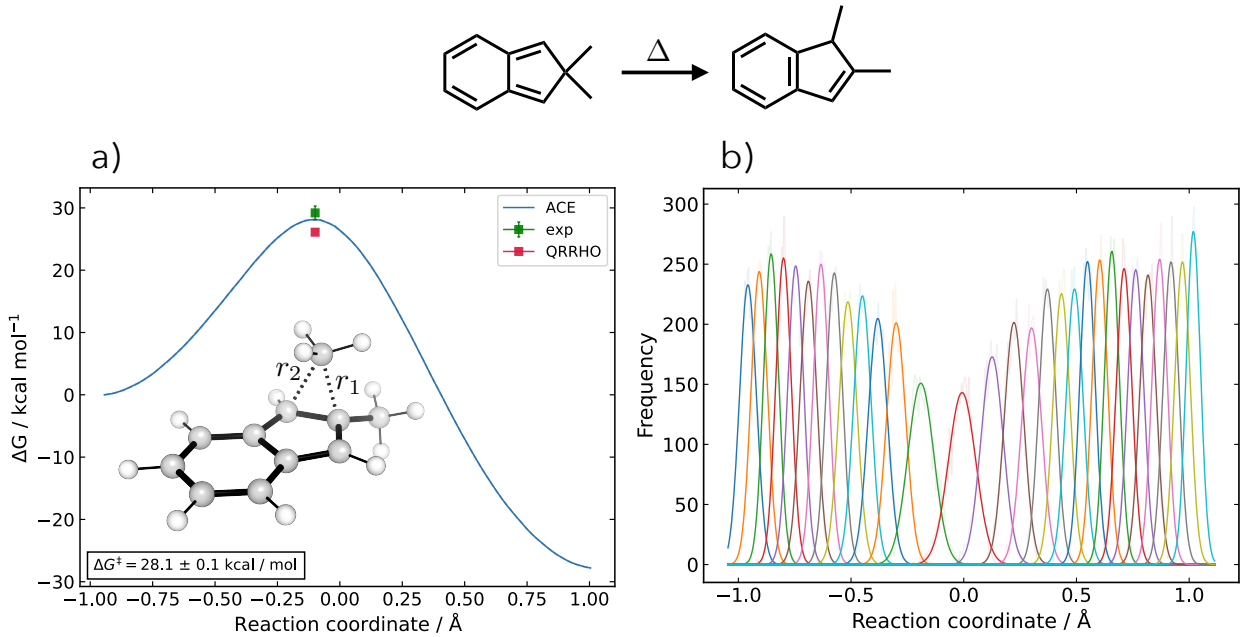


Figure 3.13: Free energy of the 2,2-dimethylisoindene to 1,2-dimethylindene reaction at 365.6 K. a) Umbrella sampling free energy profile for the $r_1 - r_2$ reaction coordinate using a PBE0-D3BJ/def-SVP trained ACE potential compared with experiment and static PBE0-D3BJ/def-SVP quasi rigid rotor harmonic oscillator (QRRHO) approach by Grimme.[185] US run for 30 windows, 10 ps each, 3 repeats, $\kappa = 20$ eV Å^{-2}. b) US windows with fitted Gaussians over the reaction coordinate.

The reaction coordinate chosen for the US was $f(r) = r_1 - r_2$, where r_1 and r_2 are the distances between the breaking and forming carbon bonds, **Fig. 3.13**. A US simulation at 365.6 K (the temperature that the experimental $\Delta G^{\ddagger}$ was reported[106]) was performed and compared to both experiment and static free energy, **Fig. 3.13**.

Visual inspection the dynamics across the US pathway confirmed that the configurations were stable, including around the TS. Moreover, the free energy profile we obtained was smooth, with very small variation over the repeats ($\pm$ 0.1 kcal mol^{-1}). This suggests the trained potential can accurately predict the energies and forces along the reaction coordinate.

The calculated $\Delta G^{\ddagger}$ from US is 28.1 $\pm$ 0.1 kcal mol^{-1}, which is in excellent agreement with the experimental activation free energy (29.2 $\pm$ 1.1 kcal mol^{-1}). This result also supports the experimental hypothesis that the rearrangement was concerted, as the activation energy predicted from ΔH_f° values for a stepwise radical reaction was 41 kcal mol^{-1}.[106]

Using the quasi rigid rotor harmonic oscillator (QRRHO) approach to calculating free energy

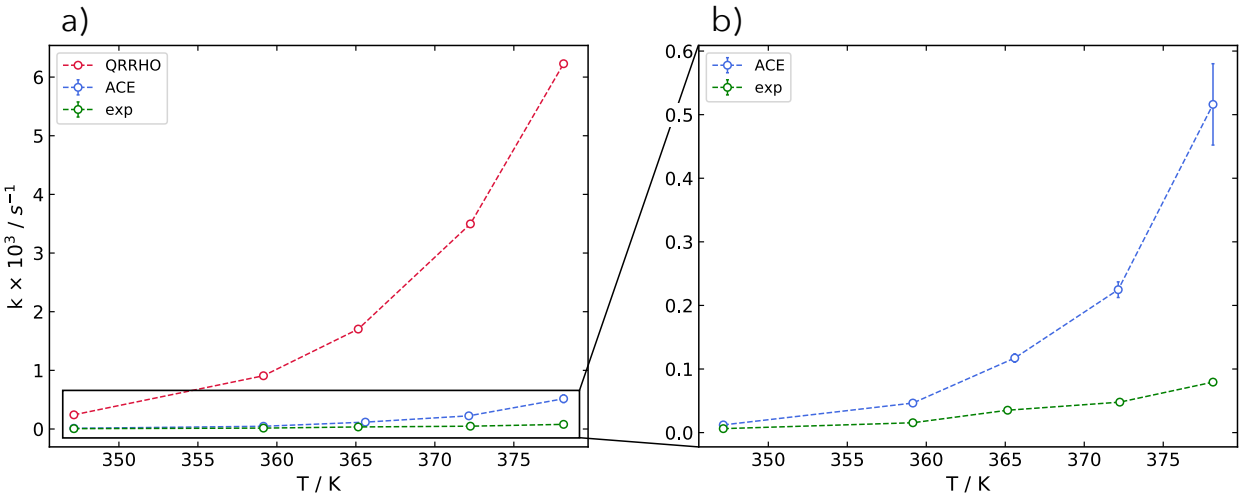


Figure 3.14: Rate constant predictions for the 2,2-dimethylisoindene to 1,2-dimethylindene rearrangement reaction at various temperatures. QRRHO represents the quasi rigid rotor harmonic oscillator approximation to compute the activation free energy (PBE0-D3BJ/def-SVP). ACE represents the activation free energy obtained from MLP-MD (PBE0-D3BJ/def-SVP), **Fig. 3.13**. The activation energies were converted to rates using the Eyring equation. The experimental rate was measured directly in pentane.[106] a) All three methods (static, dynamic and experimental) and b) static removed and y-axis rescaled for clarity.

developed by Grimme[185] and implemented in ORCA, we obtain a static activation free energy of $\Delta G^{\ddagger} = 26.1$ kcal mol^{-1}, which is 3 kcal mol^{-1} lower than the experimental value. The static approach makes a number of approximations, including treating rotations at low frequency as vibrations and assuming all vibrations are harmonic, which are not made when using US. Using the Eyring equation, the $\Delta G^{\ddagger}$ values for both static and dynamic free energies were converted to rates and compared to experiment, **Fig. 3.14**.

Rate predictions over the range tested in the experimental study show a significance discrepancy between static and dynamic calculations. Although the discrepancy is not constant with temperature, at most the dynamic prediction overestimates by a factor of 6.5, while the static prediction overestimates by at most a factor of 78. Two assumptions we have made when comparing computational rates to experimental are a) the rate in gas phase is the same as in pentane and b) the Eyring equation offers a valid conversion from activation free energy to rate. While these two assumptions may not be completely valid, we nonetheless show that a dynamical approach to predicting rate offers a more accurate prediction than the static QRRHO method. Moreover, static predictions have no error bars by construction,

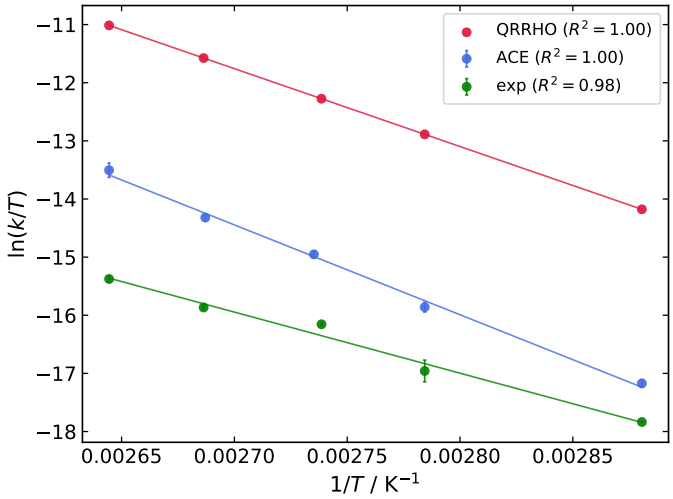


Figure 3.15: Eyring plot for the 2,2-dimethylisoindene to 1,2-dimethylindene rearrangement reaction using three methods (static, dynamic and experiment). QRRHO represents the quasi rigid rotor harmonic oscillator approximation to compute the activation free energy (PBE0-D3BJ/def-SVP). ACE represents the activation free energy obtained from MLP-MD (PBE0-D3BJ/def-SVP), **Fig. 3.13**. The experimental rate was measured directly in pentane.[106] Line fit using scipy linear regression. Errors for experiment and ACE dynamics given as one standard deviation over repeats.

whereas dynamical approaches offer error bars from the sampling variance.

We can also generate an Eyring plot, **Eqn. 3.2**, although the small temperature range of the experimental makes prediction of activation entropy and enthalpy prone to large errors, **Fig. 3.15**.

$$\ln \frac{k}{T} = \frac{-\Delta H^{\ddagger}}{R} \cdot \frac{1}{T} + \ln \frac{\kappa k_B}{h} + \frac{\Delta S^{\ddagger}}{R} \tag{3.2}$$

Within the temperature range of the experiment, the Eyring plot shows a line lying closer to the experiment for the ACE US compared to the static approach. From the Eyring plot, we can predict the activation enthalpy and entropy and compare to two sets of experimental data: the parameters as determined from the rates via the Eyring plot and the parameters as determined from thermochemical estimates, **Table 3.1**.

While the free energy predictions, and corresponding rates, are within chemical accuracy and similar to those obtained experimentally, the computed activation enthalpy and entropy show

Method	$\Delta H^{\ddagger}$ / kcal mol^{-1}	$\Delta S^{\ddagger}$ / cal K^{-1} mol^{-1}
Exp. (Eyring)	20.9 ± 6.4	-22.5 ± 17.5
Exp. (Thermochemical)	25.3 ± 1.1	-10.6 ± 3.0
ACE	30.7 ± 4.6	$+7.0 \pm 12.5$
QRRHO	26.6	+1.3

Table 3.1: Activation enthalpies and entropies from various methods for the 2,2-dimethylisoindene to 1,2-dimethylindene rearrangement reaction calculated using the Eyring plot from rates or, in the case of thermochemical, from $\log A$ and E_a, **Fig. 3.15**. Errors for experimental Eyring and ACE were computed from the scipy.stats.linregress standard errors from **Fig. 3.15**. Errors on thermochemical experiments taken from ref [106]. Note that QRHHO has no error bars by construction.

large deviations. Both $\Delta H^{\ddagger}$ and $\Delta S^{\ddagger}$ are overpredicted by both ACE and QRRHO, with the incorrect sign of $\Delta S^{\ddagger}$. A positive $\Delta S^{\ddagger}$ suggests a partly dissociate mechanism in conflict with the experimental prediction.[106, 186] However, these results should be interpreted with care as small changes in the gradient and intercept of the Eyring equation can lead to significant differences in $\Delta H^{\ddagger}$ and $\Delta S^{\ddagger}$, particularly when only a small temperature range (30 K) is considered. Error in the sign of $\Delta S^{\ddagger}$ may also be arising from the lack of explicit solvent or the PBE0 functional. It is also important to note that the two reported $\Delta H^{\ddagger}$ and $\Delta S^{\ddagger}$ experimental values (one from kinetic data and the other from thermochemical measurements) show significant deviations between them, as reflected in the large standard deviations.

From these results, we can conclude that our ACE potential combined with US provides a reasonable estimate of the free energy change over the reaction coordinate around the experimental temperatures, but overpredicts the activation parameters $\Delta H^{\ddagger}$ and $\Delta S^{\ddagger}$.

3.3.3 Path integral molecular dynamics of isoindene

Nuclear quantum effects (NQEs) can be challenging to study due to the increased computational cost required to run dynamics, however by combining MLPs with path integral molecular dynamics (PIMD), we are able to more quickly generate converged quantum dynamics to investigate whether NQEs affect the free energy of this reaction class. Typically, we expect NQEs to become significant when reactions involve light atoms, such as hydrogen.[2,

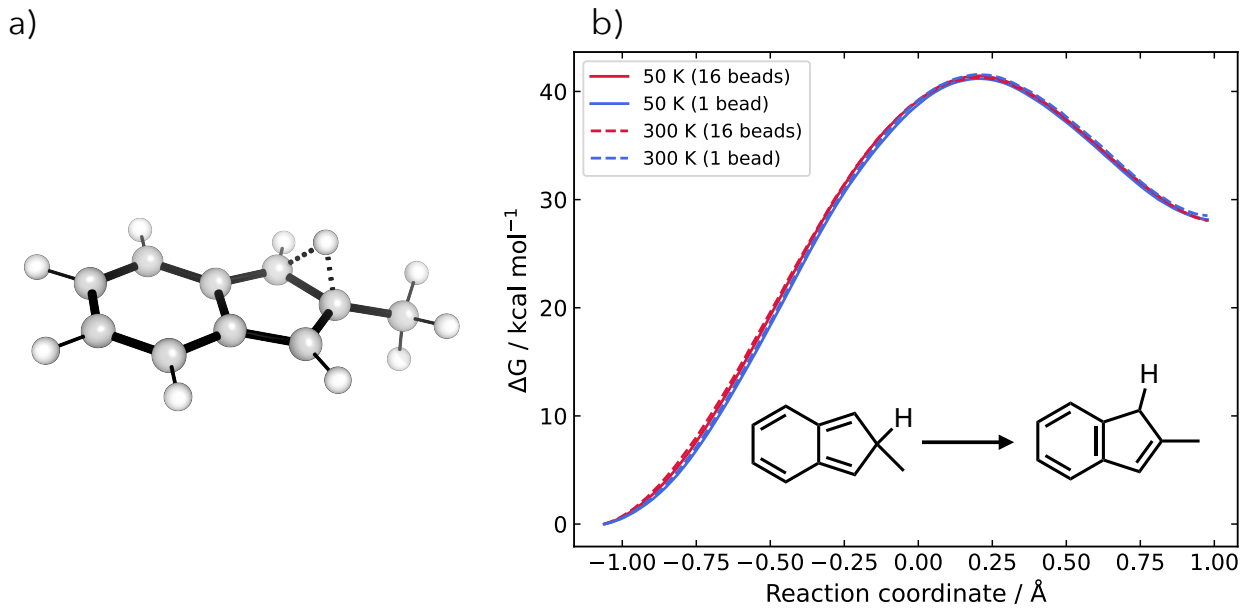


Figure 3.16: a) TS structure of 2-methylisoindene optimised using PBE0-D3BJ/def-SVP. b) Free energy of the hydride shift in 2-methylisoindene to 2-methylindene at 300 K and 50 K using 1 and 16 bead PIMD generated using US. The $r_1 - r_2$ reaction coordinate using a PBE0/def-SVP trained ACE potential was used, with 30 windows, 10 ps each, 3 repeats, $\kappa = 20$ eV Å^{-2}.

187] For the methyl shift reaction of 2,2-dimethylisoindene, such effects are expected to be negligible. Therefore, we substituted the methyl group for a hydrogen atom and calculated the PIMD free energy, **Fig. 3.16**.

When 16 bead PIMD was initially run, nonphysical configurations were observed in the dynamics, for example configurations where the H atom was hovering far above the expected TS distance. To alleviate this issue, we included the problematic configurations and retrained the potential. The resulting MLP predictions were within 1 kcal mol^{-1}. To assess whether there were any NQEs affecting the free energy of this reaction, the potential was used to run both 1 and 16 bead PIMD US interfaced through i-PI, **Fig. 3.16**.

The 1 bead simulation (equivalent to classical mechanics) was also performed with the rest of the setup unchanged and we compared to the results obtained from classical dynamics using *mlp-train*. The free energy profile for 1 bead i-PI dynamics was equivalent to classical dynamics from *mlp-train*. In addition, the dynamics were run using Plumed instead of *mlp-train*, and again gave the same results.[144]

Interestingly, there is no significant difference between classical dynamics and quantum dynamics at both 300 K and 100 K. This suggests that NQEs are negligible for this reaction. Furthermore, the fact that ΔG does not change with temperature suggests that entropic effects are minimal. To assess whether NQEs affect the rate of the reaction, the quantum transition state theory rate would need to be calculated using further PIMD methods, such as ring polymer molecular dynamics (RPMD)[188], to include NQEs that result in a non-unity transmission coefficient.

3.4 Hydride shifts in terpinenes

Next, we examined the experimentally studied synthesis of a monoterpene, γ-terpinene precursor, found in thyme (detailed in **Section 1.3.2**).[108] The formation of this molecule is catalysed by monoterpene synthase enzymes, and the reaction can proceed via either a 1,2- or 1,3-hydride shift reaction. The two pathways lead to different coproducts; α-thujene coproducts of the γ-terpinene synthase from *T. vulgaris*.

Here, we investigated the 1,2 shift reaction in the gas phase using MLPs to elucidate the intrinsic reactivity of this system and potential NQEs. From tritium labelling NMR studies, the 1,2 pathway has been suggested to be the dominant pathway but an activation barrier has not been determined.[108] We also attempted to train an ACE potential on the 1,3 pathway but despite efforts the resulting potential was not stable for US free energy computation. We did, however, generate a CINEB coordinate for the 1,3 pathway and calculated a $\Delta E^{\ddagger} = 77.9$ kcal mol^{-1}, suggesting this is not a feasible pathway.

3.4.1 Training a terpinene ACE potential

For the 1,2 shift reaction, we began our study by performing CINEB calculations from the QM optimised reactants and products at PBE0/def2-SVP, **Fig. 3.17**.

This resulted in an activation barrier of $\Delta E^{\ddagger} = 8.6$ kcal mol^{-1}. Such a low barrier was expected considering the small change in geometry and the intramolecular nature of this

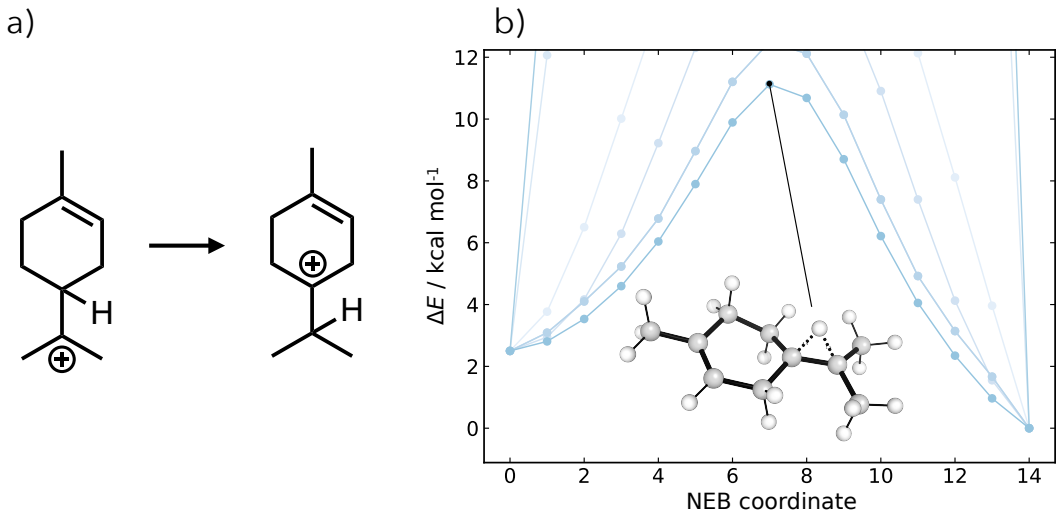


Figure 3.17: a) 1,2-hydride shift of a carbocation γ-terpinene precursor resulting in a carbocation rearrangement. b) CINEB energies along the NEB IRC using 15 frames from PBE0/def2-SVP optimised reactants and products. CINEB highest energy structure shown.

reaction. Taking the highest energy structure as a starting point for the TS optimisation at PBE0/def2-SVP, we observed the migrating H atom move closer to the product side, **Fig. 3.18**. Following this TS optimisation, ACE AL was initiated from the TS configuration, with 1 ps in the AL loop, **Fig. 3.18**. The ACE potential generated accurate dynamics from a 1 ps simulation, MAE = 0.7 kcal mol^{-1}. Moreover, the accuracy of the dynamics did not decrease towards the end of the 1 ps simulation.

3.4.2 Free energy of the terpinene reaction

Using this potential, a free energy profile was generated along the coordinate $f(r) = r_1 - r_2$, the two distances defining both the bond breaking and bond making, **Fig. 3.19**. Averaging over three repeats gives a fairly smooth free energy plot over the reaction coordinate, with $\Delta G^{\ddagger} = 4.7 \pm 0.4$ kcal mol^{-1}. The free energy does not reach a minimum at either end of the reaction coordinate (which is defined as the starting and end points of the reaction from the NEB IRC). Therefore, US dynamics were extended past these points. Within 0.1 Å to the negative side, a minimum was reached ($\approx$ 0.5 kcal mol^{-1}) and 0.02 Å on the positive side ($\approx$ 0.2 kcal mol^{-1}). The free energy difference was taken from the minimum in the PES to the TS, as the $f(r) = r_1 - r_2$ coordinate does not necessarily represent the minimum in the

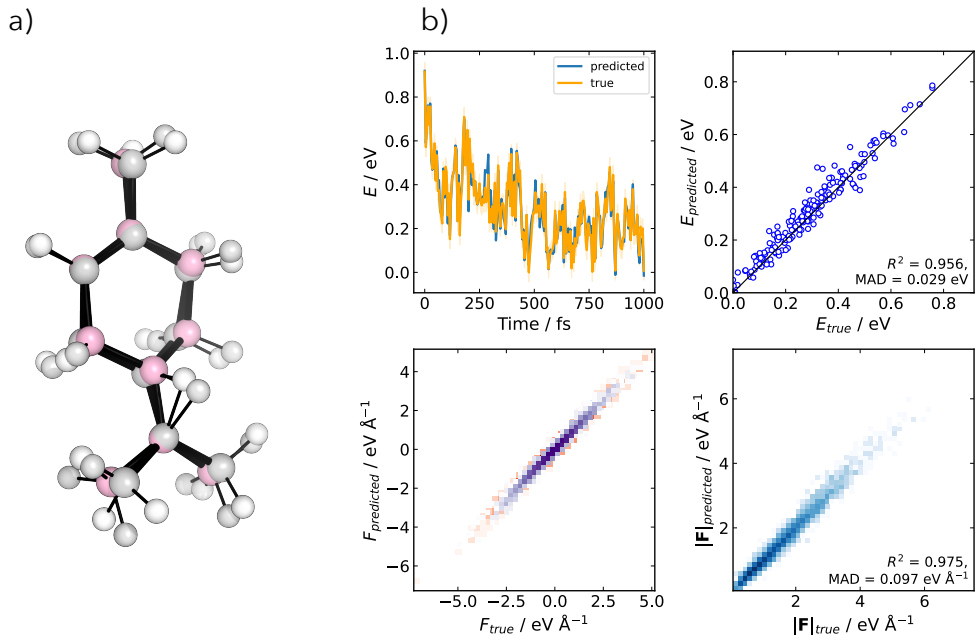


Figure 3.18: a) Aligned structures of γ-terpinene reactant with pink as the CINEB TS and grey as the PBE0/def2-SVP TS. b) MLP dynamics initiated from the γ-terpinene TS. ACE potential trained at PBE0/def-SVP using the standard *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 1 ps. Training initialised from PBE0/def2-SVP ORCA TS structure. Ground truth and ACE energies over a 1 ps MLP-MD simulation at 300 K. Parity plot between true and predicted energies, force components and force magnitude.

whole free energy surface.

The individual values of r_1 and r_2 were plotted for this simulation, **Fig. 3.20**, along with the locations of the PES minima and TSs. The 2D histogram shows the path traversed along the biased reaction coordinate from reactant to product. The $r_1 - r_2$ distance observed at the peak in the free energy surface ($r_1 - r_2 = -0.064$ Å) is similar to the one obtained after optimisation at PBE0/def2-SVP ($r_1 - r_2 = -0.167$ Å). The 2D histogram also shows the spread of r_1 and r_2 values that comprise the free energy surface TS and reaction coordinate values, r, along the IRC. For this reaction, the majority of configurations lie within a r_1 and r_2 range of 0.1 Å for a particular r.

3.4.3 Path integral molecular dynamics of terpinene

Since the 1,2 shift reaction involves a hydrogen atom migration, NQEs are expected to be present. To investigate these effects, PIMD was employed using the trained ACE potential

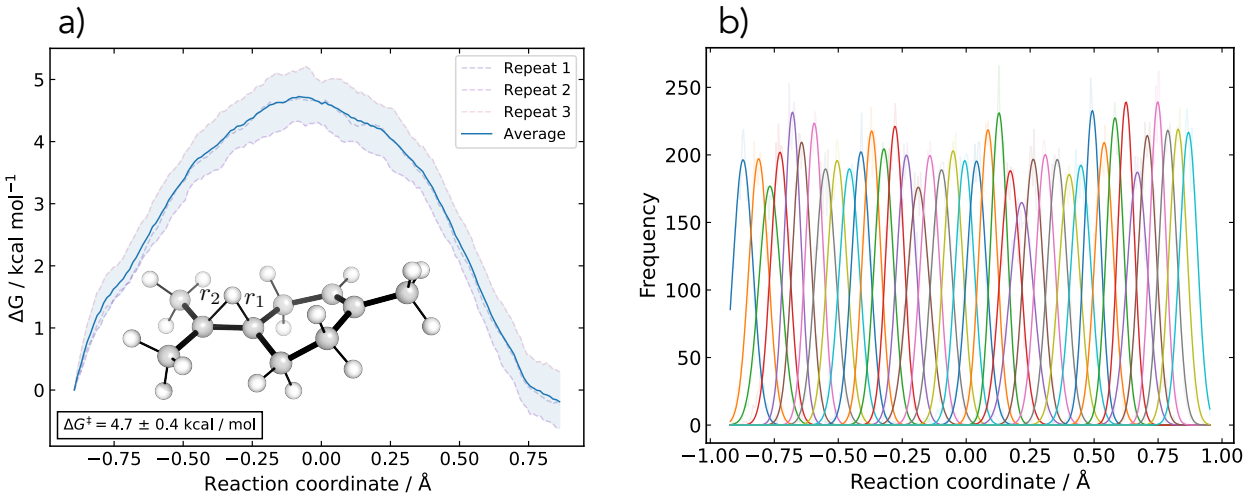


Figure 3.19: Free energy of the 1,2-hydride shift of γ-terpinene reaction at 300 K. a) Umbrella sampling free energy profile for the $r_1 - r_2$ reaction coordinate using a PBE0/def-SVP trained ACE potential. PBE0/def2-SVP TS shown with corresponding r_1 and r_2 distances. US run for 40 windows, 10 ps each, 3 repeats, $\kappa = 20$ eV Å^{-2}. b) US windows with fitted Gaussians over the reaction coordinate.

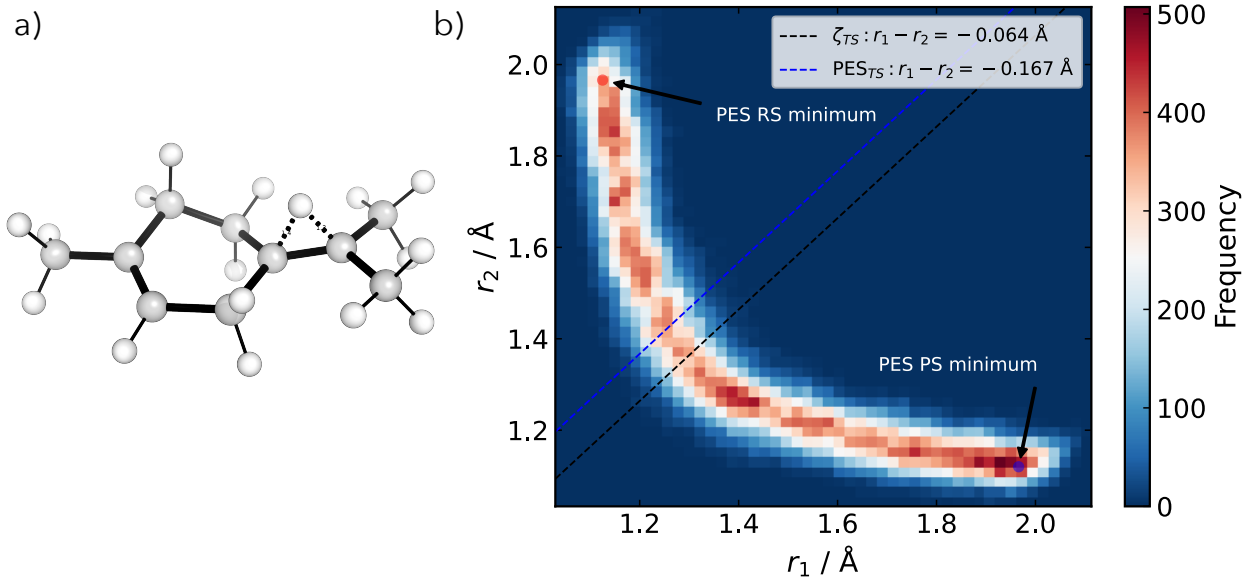


Figure 3.20: 2D histogram of r_1 and r_2 values observed during a US of the 1,2-hydride shift of γ-terpinene reaction at 300 K (detailed in **Fig. 3.19**) (b). The black dashed line represents the combinations of r_1 and r_2 that correspond to the peak in the free energy surface. The TS from PBE0/def2-SVP optimised TS is $r_1 - r_2 = -0.167$ Å (blue dashed line), with the structure shown on the left (a). The PES minima for both reactant and product are also marked on the histogram.

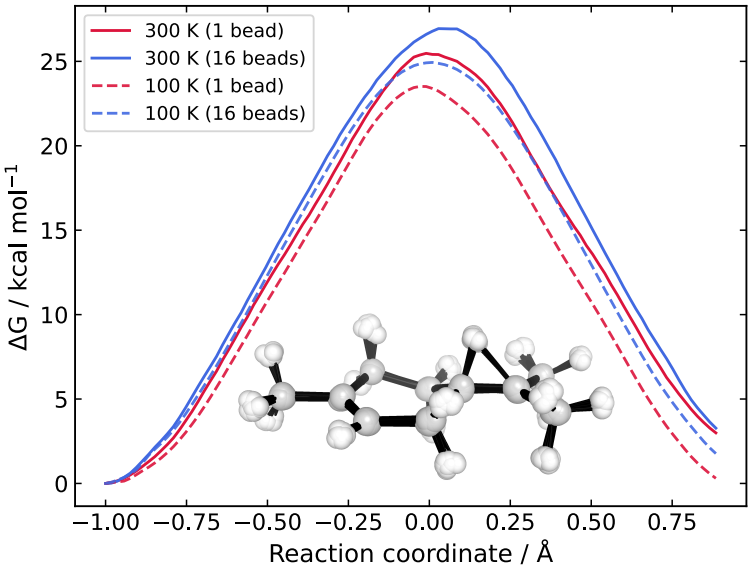


Figure 3.21: Free energy of the 1,2-hydride shift of γ-terpinene reaction at 300 K and 100 K using 1 and 16 bead PIMD generated using US. The $r_1 - r_2$ reaction coordinate using a PBE0/def-SVP trained ACE potential was used, with 30 windows, 10 ps each, 3 repeats, $\kappa = 20$ eV Å^{-2}. A representative frame from the 16 bead PIMD is shown with all 16 beads overlaid.

and US, **Fig. 3.21**. To ensure an exact comparison between classical and quantum dynamics, 1 bead PIMD was performed along with 16 bead PIMD. The temperature was also decreased from 300 K to 100 K. Initially, as with the isoindene system, nonphysical configurations were originally observed in the 16 bead PIMD, whereby the migrating H atom became close to the carbon atoms. This problem was solved by adding this configuration to the training set and retraining the potential.

Typically, a lower $\Delta G^{\ddagger}$ is expected when NQEs are included compared to classical dynamics. However, we see an increase in $\Delta G^{\ddagger}$ when the number of beads is increased from 1 to 16. This effect is also observed at 100 K and 300 K. Since the free energy at the reactant and product states is the same for 1 and 16 beads at 300 K, the difference in $\Delta G^{\ddagger}$ is due to dynamics around the TS. One possible explanation is an increased zero point energy around the TS structure, leading to a higher activation free energy. However, another reason for the unusual observations could be inaccuracies due to the level of theory or ACE potential. While further investigations are required to validate the PIMD free energy, these results demonstrate the ability of ACE to be combined with PIMD to rapidly compute quantum

free energies.

3.5 Product ratios in indanes

Having successfully trained a stable ACE potential for the H/Me migration reactions of indane in **Section 3.2**, we investigated the product ratio for the competing H and Me migrations, originally studied by Tantillo and Feng.[105]

Two modifications made to the training described in **Section 3.2** were the level of theory as well as the starting TS configuration. First, the potential was trained at M06-2X-D3zero/6-31G(d), following the work of Tantillo and Feng. Interesting, we found an RMSD discrepancy between the our TS and Tantillo's of 0.275 Å for the Me migration TS and 0.121 Å for the H migration TS when using the default grid size in ORCA 5 (defgrid2) for M06. The RMSD vanished when upon increasing the grid size to defgrid3.

3.5.1 Retraining an ACE potential of indane

Separate ACE potentials for the H and Me migrations were trained from the optimised TS structures using AL and the configurations generated by the AL cycles were combined to train a final potential, **Fig. 3.22**. Each individual AL cycle initiated from the respective H and Me TSs resulted in an energy MAD of 0.9 kcal mol^{-1} and 0.6 kcal mol^{-1}, both within chemical accuracy. Upon combining these configurations and retraining, an MAD of 0.5 kcal mol^{-1} was obtained.

3.5.2 Uphill dynamics of indane

Uphill dynamics involve placing additional momentum into a bond to promote dissociation. In the study by Tantillo and Feng, they placed 25 kcal mol^{-1} excess energy in the C−O bond and ran 88 M06-2X-D3zero/6-31G(d) AIMD trajectories to generate a product ratio. Using the trained potential, we ran uphill dynamics to obtain a product ratio between the H and Me migrations. In this case, we placed the excess energy in the C−O bond, creating

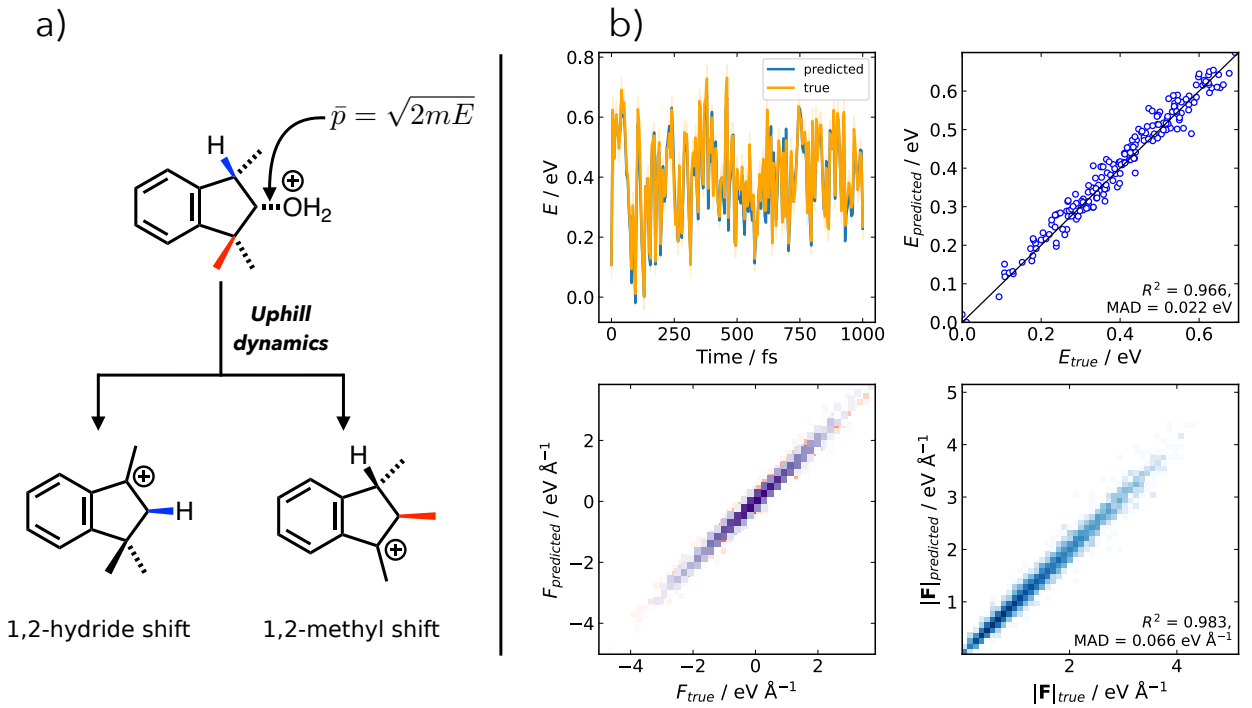


Figure 3.22: a) Reaction scheme for uphill dynamics of the indane-derived molecule, leading to either a hydride or methyl shift. Momentum equivalent to $\sqrt{2mE}$ is placed in the C−O bond to promote dissociation. b) MLP dynamics initiated from the indane H-migration TS. ACE potential trained at M06-2X-D3zero/6-31G(d) with a fine grid (defgrid3) by combining the configurations from the *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 3 ps for H and Me migration TSs. Ground truth and ACE energies over a 1 ps MLP-MD simulation at 300 K. Parity plot between true and predicted energies, force components and force magnitude.

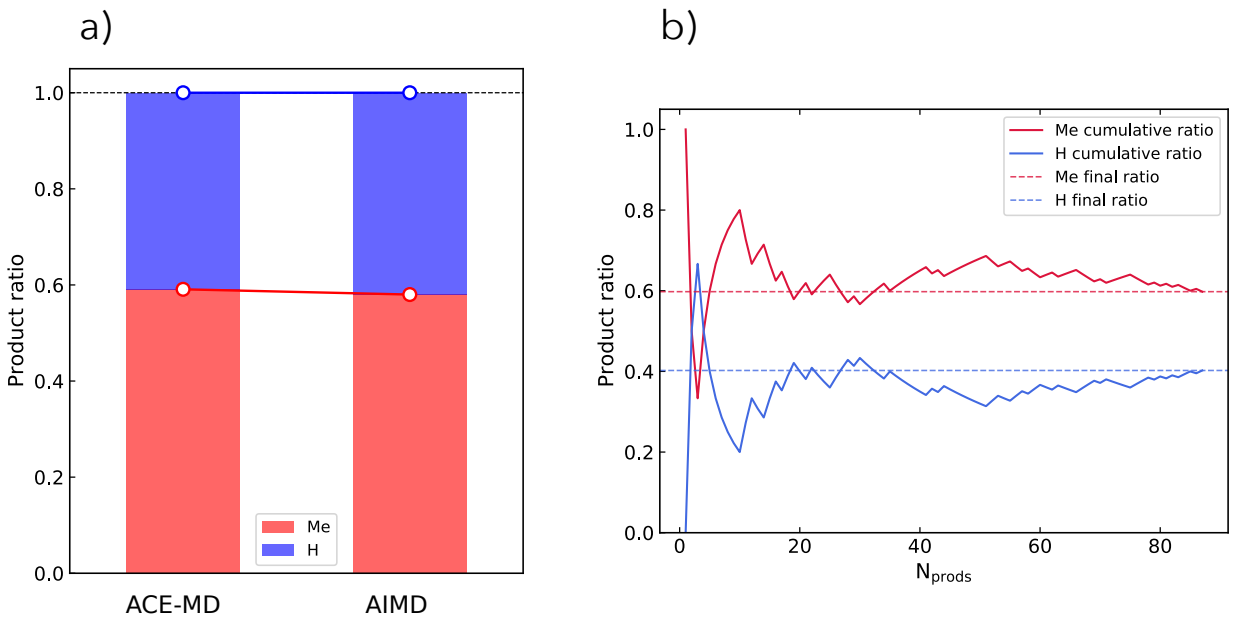


Figure 3.23: a) Product ratios from NVE uphill dynamics of indane-derived 1,2-shift reaction of H and Me using an ACE MLP trained using AL at M06-2X-D3/6-31G(d) (88 trajectories) and AIMD product ratios (88 trajectories) from Tantillo and Feng.[105] Initial velocities sampled from a Maxwell-Boltzmann distribution at 298 K. 25 kcal mol^{-1} of energy was used to break the water bond, with each simulation lasting 1 ps. b) Cumulative product ratio for Me and H shifts for indane-derived uphill dynamics over 88 trajectories using ACE.

an unstable secondary carbocation with OH_2 as the leaving group. From this point, three possible states can form. The water can reattach itself to the carbocation, or the H or Me can migrate, forming a more stable tertiary carbocation.

Uphill dynamics are necessary in this reaction since the rate at which water leaves is beyond achievable timescales of regular MLP-MD simulations. Moreover, we cannot initiate dynamics from a TS since there are two separate TSs. Therefore, uphill dynamics provides a method to generate two products from a single reactant configuration. One drawback of this approach is the artificially high energy placed in the bond, which is assumed not to bias the ratio of products.[105] We attempted to reproduce these product ratio calculations by Tantillo and Feng using the trained ACE potential, **Fig. 3.23**.

The computed ratio of Me and H migration products using the ACE potential (60 Me:40 H) is in good agreement with the one reported by Tantillo and Feng (58 Me:42 H). Inspection of the cumulative ratios shows a large variation in ratio initially, which then converges to

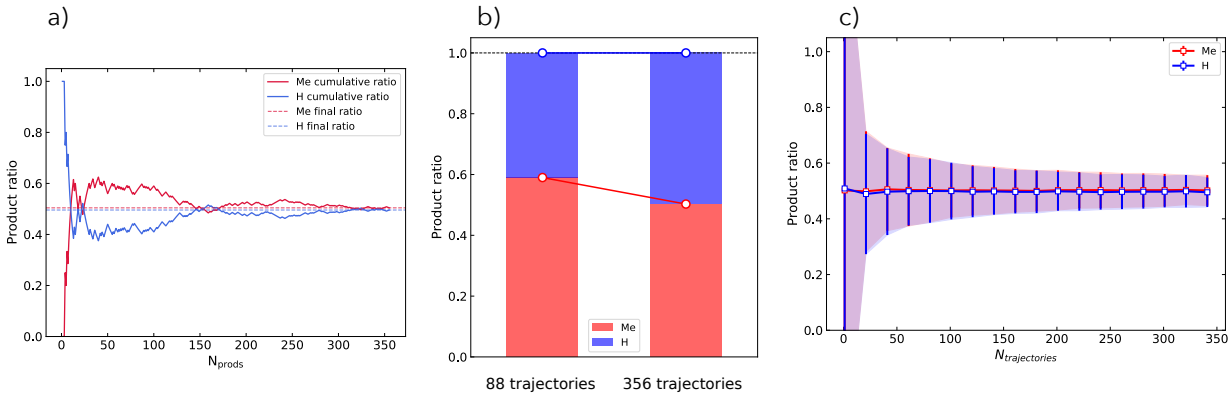


Figure 3.24: a) Cumulative product ratio from NVE uphill dynamics of indane-derived 1,2-shift reaction of H and Me using an ACE MLP trained using AL at M06-2X-D3/6-31G(d) (356 trajectories reaching products from 1000 trajectories). Initial velocities sampled from a Maxwell-Boltzmann distribution at 298 K. 25 kcal mol^{-1} of energy was used to break the water bond, with each simulation lasting 1 ps. b) Final product ratios of indane reaction for 88 trajectories and 356 trajectories. c) Bootstrapping error of product ratios over 356 trajectories.

(60 Me:40 H).

One advantage of MLPs is their lower computational cost compared to AIMD, which enabled us to easily extend the number of trajectories to 1000 to ensure full convergence, **Fig. 3.24**. For example, we calculated that M06-2X-D3zero/6-31G(d) takes about 200 times longer on 8 cores compared to MLP-MD on 1 core for this system. Here, we propagated 1000 trajectories, of which 356 successfully reached a product, while the remaining trajectories saw either no migration or the water reattaching to the carbocation. Using these converged results, we obtained a product ratio of approximately 50:50. Bootstrapping gives an error estimate of about 10%. This is consistent with our convergence plot, where after 150 trajectories, the ratio remains unchanged around 50:50. These calculations demonstrate the importance of ensuring convergence over many repeats and the power of MLPs, as such computation would be intractable using AIMD.

3.5.3 Path integral molecular dynamics of indane

We performed PIMD using the trained ACE potential and analysed the product ratios. We initially used an excess momentum of 25 kcal mol^{-1} , the same as for the classical dynamics,

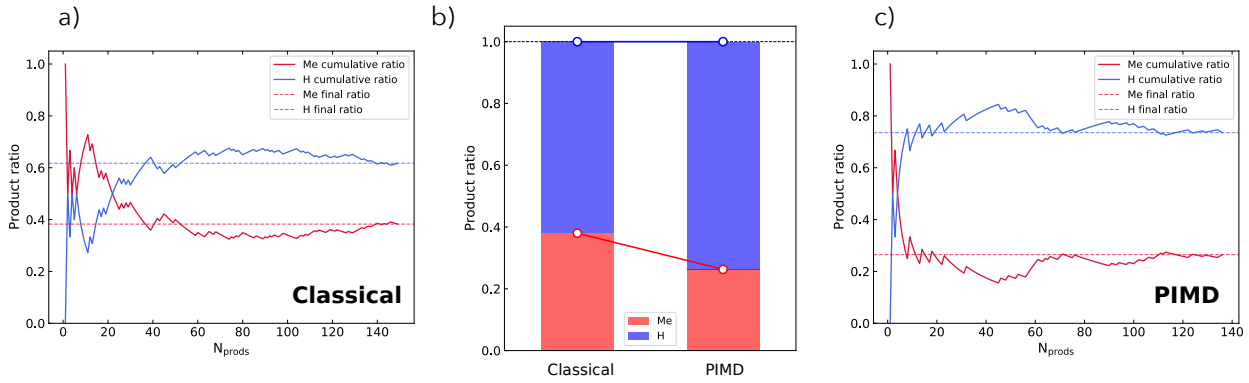


Figure 3.25: a) Cumulative product ratio from NVE uphill classical dynamics of indane-derived 1,2-shift reaction of H and Me using an ACE MLP trained using AL at M06-2X-D3/6-31G(d) (150 successful trajectories). Initial velocities sampled from a Maxwell-Boltzmann distribution at 298 K. 30 kcal mol^{-1} of energy was used to break the water bond, with each simulation lasting 1 ps. b) Final product ratios of indane reaction for 150 successful trajectories for classical and quantum (PIMD) dynamics. c) Cumulative product ratio from NVE uphill quantum (PIMD) dynamics of indane-derived 1,2-shift reaction of H and Me using the same potential.

however this resulted in all trajectories leading back to the reactants. This may be caused by a higher barrier for the water dissociation and migration in PIMD compared to classical MD, however further studies are required to test this hypothesis. To overcome the barrier leading to products, we increased the momentum to 30 kcal mol^{-1} and propagated classical and quantum dynamics, **Fig. 3.25**.

Comparison of product ratios for classical dynamics using 30 kcal mol^{-1} of excess momentum to the original 25 kcal mol^{-1} shows a change in the product ratios from (60 Me:40 H) to (38 Me:62 H). This could be due to the excess of energy being placed in the C−H bond, resulting in an easier breakage. After 150 successful trajectories, we see a notable increase in the number of H products formed in the PIMD (22 Me:78 H), suggesting that the H atom more easily migrates than the Me group when quantum dynamics are run. Although the excess energy of 30 kcal mol^{-1} alters the product ratio compared with 25 kcal mol^{-1}, this analysis suggests that NQEs may play a role in product ratios formed for reactions involving H atoms migrating in indane systems.

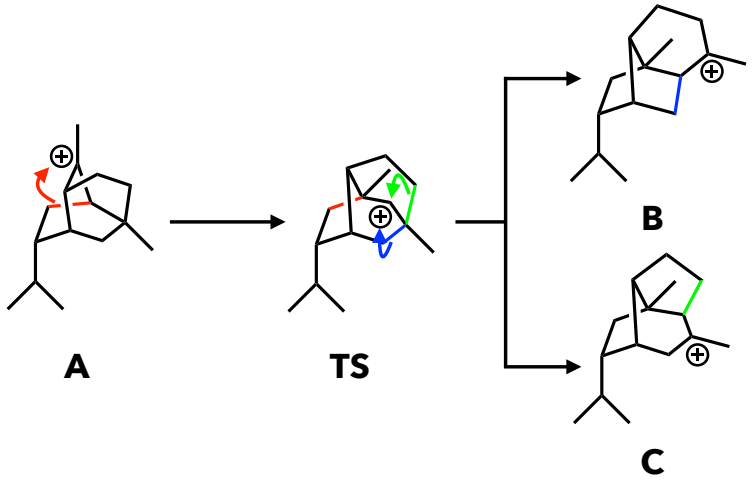


Figure 3.26: Proposed post-transition state bifurcation of pupukeanane (**TS**) into products **B** and **C**. Reactant **A** is also proposed to be connected to the **TS** via a carbocation rearrangement. Bonds that break and reform are shown in different colours.

3.6 Bifurcating surfaces in sesquiterpenes

We have previously seen that due to the assumptions made in static free energy calculations, MD simulations can offer an improvement in accuracy in activation free energy prediction. Another area where dynamics can surpass static calculations is when a system experiences a bifurcating surface. Systems where the PES is relatively flat around the TS can result in multiple products spurring from a single TS. The ratio of these products, and even their existence, can be challenging to predict using static methods that do not account for a post transition state bifurcation (PTSB). One example that has been previously studied using static methods, and for which the authors postulate that a PTSB may occur, is the rearrangement reaction of a sesquiterpene (detailed in **Section 1.3.4**), **Fig. 3.26**.[104]

The previous study used DFT to optimise the sesquiterpene TS at different levels of theory (mPW1PW91/6-31+G(d,p) and B3LYP-D3/6-31+G(d,p)), before running an IRC calculation to connect each TS to a reactant and a product. They found that the mPW1PW91/6-31+G(d,p) TS is connected to **A** and **B**, while the B3LYP-D3/6-31+G(d,p) TS is connected to **B** and **C**. While the static DFT method suggests a link between these products from a common TS, the authors postulate that direct dynamics simulations could provide further evidence. MLPs provide an accurate and fast method for generating dynamics, and were

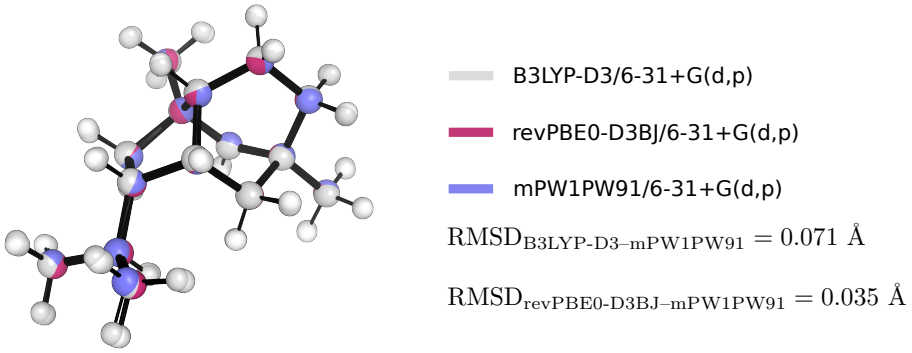


Figure 3.27: Optimised transition state structure for sesquiterpene TS using different levels of theory (B3LYP-D3/6-31+G(d,p), mPW1PW91/6-31+G(d,p) and revPBE0-D3BJ/6-31+G(d,p)). Structures are aligned to the mPW1PW91/6-31+G(d,p) and RMSD relative to this structure is shown.

applied to this system.

3.6.1 Training an ACE potential on sesquiterpene

We trained three ACE potentials at different levels of theory from their corresponding optimised TSs. These were mPW1PW91/6-31+G(d,p) and B3LYP-D3/6-31+G(d,p) (the two levels of theory used in the paper) and also with revPBE0-D3BJ/6-31+G(d,p), which has been shown to provide accurate energy prediction for carbocation systems.[189] The TSs for each level of theory were optimised in ORCA and compared, **Fig. 3.27**.

All three structures are very similar in geometry, with the main noticeable differences being in the two hanging methyl groups on the left hand side of **Fig. 3.27**. Therefore, we expect similar dynamics when initiated from these structures. There may be other differences in the PES as dynamics progress, and any difference will be observed when comparing the ratio of products. First, we trained an ACE potential from each of the TSs at the corresponding level of theory, **Fig. 3.28**.

Training for all three levels of theory resulted in an energy error of less than 1 kcal mol^{-1}. While mPW1PW91/6-31+G(d,p) results in the most accurate dynamics, the difference between all three potentials is only 0.1 kcal mol^{-1}. During the AL training for all three levels of theory, a roughly even split of **B** and **C** was observed, along with the **TS** as expected. In

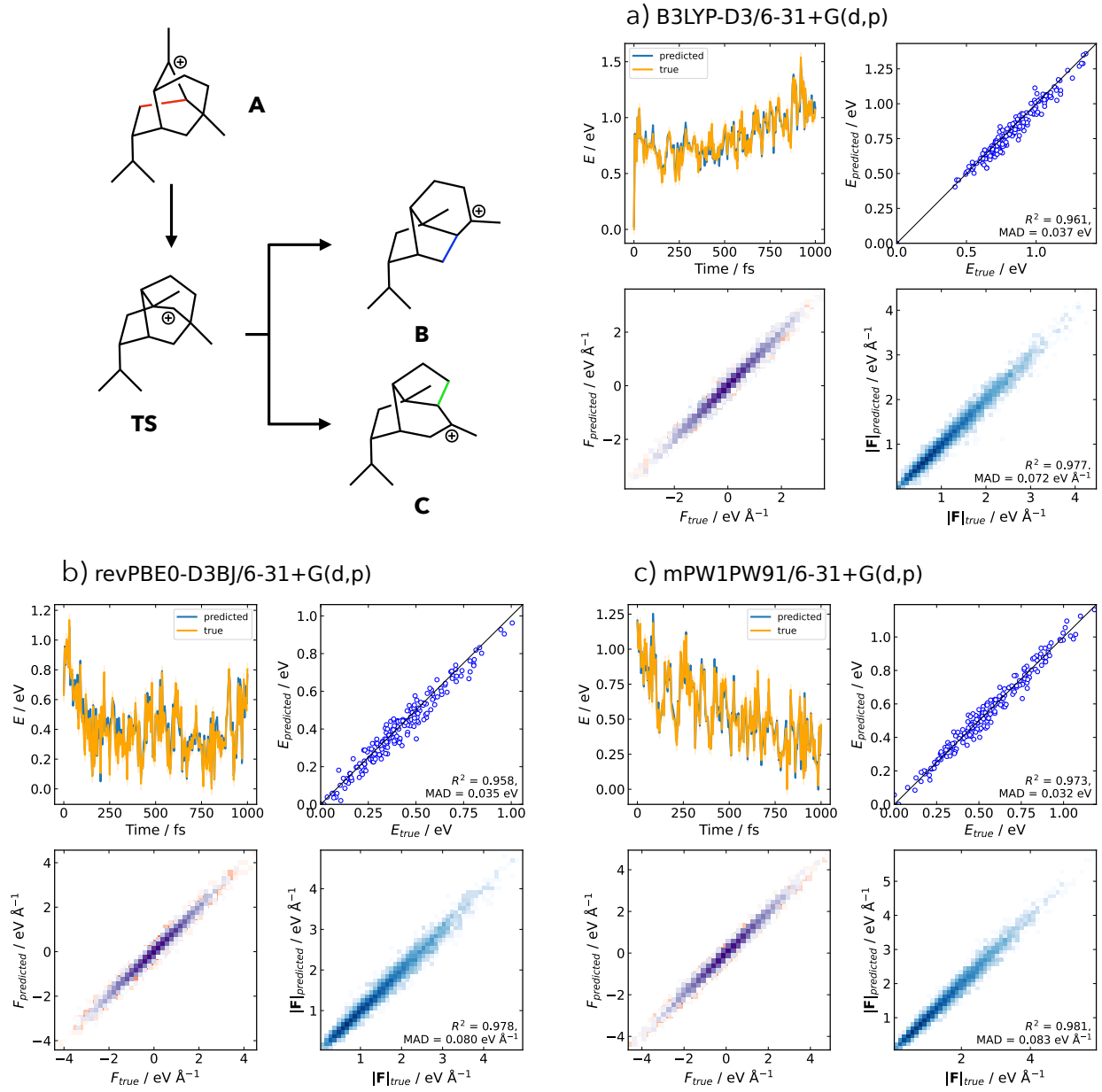


Figure 3.28: MLP dynamics initiated from sesquiterpene TSs. ACE potential trained at various levels of theory (B3LYP-D3/6-31+G(d,p), mPW1PW91/6-31+G(d,p) and revPBE0-D3BJ/6-31+G(d,p)) using the standard *mlp-train* AL strategy at 500 K with maximum propagation time in the AL loop of 3 ps from the corresponding TS. Ground truth and ACE energies over a 1 ps MLP-MD simulation at 300 K. Parity plot between true and predicted energies, force components and force magnitude.

addition, a small number of **A** was observed. This provides confidence that the potential is able to predict the energies and forces for all the possible products if they are observed in MLP dynamics.

3.6.2 Product ratios of sesquiterpene

Using these potentials, we ran MLP-MD initiated from the TSs and measured the number of each product observed over 200 repeats, **Fig. 3.29**, with each 1 ps simulation taking approximately two minutes to finish. Initiating dynamics from the TS affords a majority of products **B** and **C**, providing evidence of the PTSB. The split of **B** versus **C** depends on the level of theory used, with revPBE0-D3BJ giving the most **C**, while B3LYP-D3zero gives the most **B**. Taking a holistic view of the product ratios over all three levels of theory, the data suggests there is a fairly even split of both **B** and **C** when dynamics are initiated from the TS. We also observe a small amount of **A** for mPW1PW91 and revPBE0-D3BJ levels of theory, but not for B3LYP-D3zero. While **A** is connected to the TS, it is likely more uphill than **B** and **C**, making it more difficult to access.

This analysis suggests that **A**, **B** and **C** are all connected via the **TS**, highlighting that the synthesis of these sesquiterpenes may involve complex competing pathways in enzymes. Dynamics could be run in explicit solvent to more closely mimic biological conditions, but ultimately the reaction must be performed in the true enzymatic environment to draw conclusions about pathways. However, this analysis gives weight to the possibility of bifurcating reactions occurring in these biological systems.

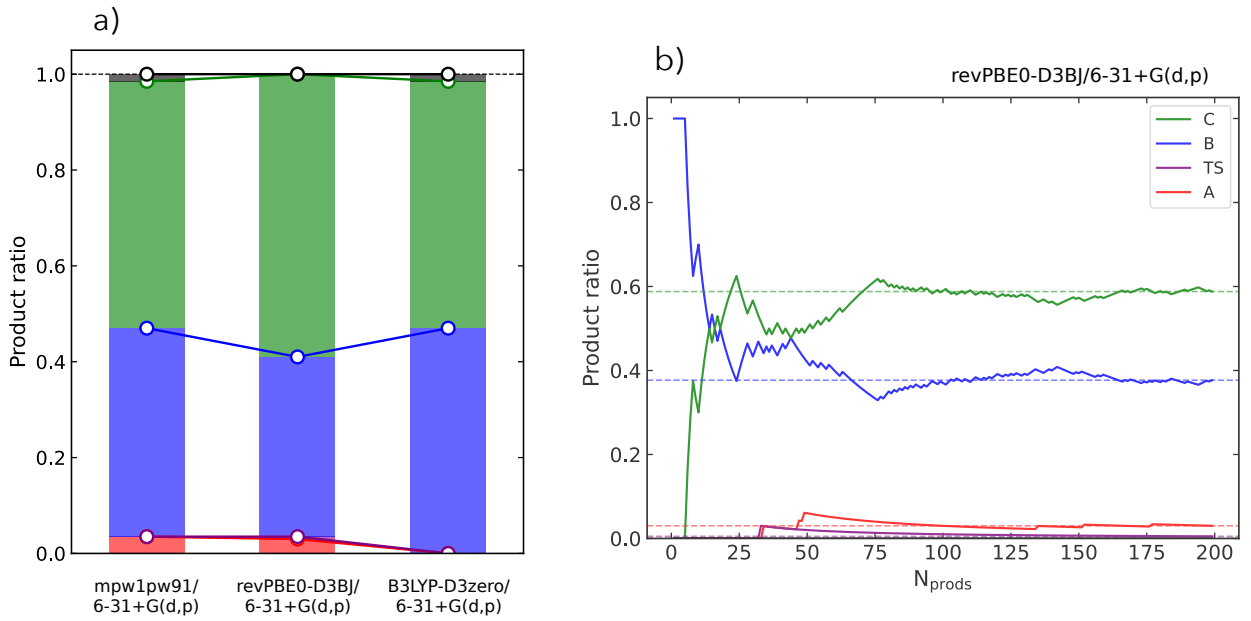


Figure 3.29: Product ratios of sesquiterpene as initiated from three TSs, using an ACE potential trained at different levels of theory (500 K, 3 ps). a) Product ratios from 200 × 1 ps MD simulations using an ACE trained on mpw1pw91/6-31+G(d,p), revPBE0-D3BJ/6-31G(d,p) and B3LYP-D3zero/6-31+G(d,p). The colours correspond to the structures given in (b), with grey denoting structures not categorised. b) Cumulative product ratio over the 200 MD simulations for the revPBE0-D3BJ/6-31G(d,p) ACE potential.

3.7 Conclusions

In this chapter, we have outlined different strategies to train MLPs for reactive systems using active learning (AL). We applied this approach to study a series of organic terpene reactions in the gas phase, including hydrogen and methyl shifts in an indane, a methyl shift in an isoindene, a hydrogen shift in a terpinene and alkyl shifts in a sesquiterpene.

We observed that while training around the energy minimum is less sensitive to the initial configuration and requires few training points, training around the TS region is computationally more demanding, requiring many more configurations. Moreover, the resulting accuracy and efficiency is sensitive to the quality of the TS guess. For example, using estimated TSs from CINEB led to less accurate potentials than when optimising these TSs at the QM level. This is because the CINEB TS is often not a true TS and AL dynamics can cause the training to fall to only one side of the reaction coordinate. This was of particular importance when running US calculations, as these require stable long time dynamics at each point along the

reaction coordinate.

Next, we proposed an automated method to systematically improve an MLP along the IRC. This was done by taking the configuration with the highest energy error as the new starting point in the AL training and repeating this process until all configurations are below a given error threshold. While this strategy was found to improve the potentials' accuracy, it led to inefficient training as a whole new AL cycle needs to be repeated until convergence is reached. Instead, we proposed that only the configuration that had the highest energy error was included in the retraining. This was found to improve the accuracy of the potentials, as illustrated in the PIMD of isoindene, **Section 3.3.3** and terpinene, **Section 3.4.3**.

We then validated an ACE potential on an isoindene methyl shift reaction, for which experimental data is available. We trained the potential from an optimised TS and performed US calculations to obtain the activation free energy. The computed $\Delta G^{\ddagger}$ was found to be consistent with experiment and closer than the corresponding static free energy prediction, supporting the experimental hypothesis that the reaction mechanism is concerted. Moreover, the rates within the studied temperature range were found to be more accurate with the ACE potential than when using static DFT methods. However, although the standard deviation for both experimental rates and ACE were large, both ACE and QRRHO methods failed to reproduce experimental values. This analysis suggests that either the potential needs to be improved, e.g., level of theory, or that other factors, such as explicit solvent, are not sufficiently accounted for. We also explored NQEs on a modified isoindene involving a H migration, employing ACE to perform PIMD simulations. We found NQEs to be negligible at both 100 K and 300 K. Moreover, the free energy at both temperatures was found to be similar, suggesting that entropic effects do not play a significant role.

We then studied a 1,2-hydride shift reaction in a terpinene system present in plant enzymes. The low activation free energy for this pathway, $\Delta G^{\ddagger} = 4.7 \pm 0.4$ kcal mol^{-1}, suggests that this pathway is possible in the synthesis, however testing in solvent may be required to account for polarisation effects. We also attempted to train and run US on the competing 1,3-hydride shift ($\Delta E^{\ddagger} = 77.9$ kcal mol^{-1}), however, the potential was unstable and could

not be used to accurately generate a free energy profile. We performed PIMD on this system and found that the activation free energy was higher when NQEs were included. While it could be possible that there is an increased zero point energy around the TS structure, further validation is required.

The product ratios for the H/Me migration in an indane system were studied. Our results are in line with previously reported computational studies. Utilising the speed of MLPs, we also extended the number of trajectories from 88 to 356 and found a change of 10% when the ratios were converged, demonstrating the importance of ensuring convergence. Inclusion of NQEs using PIMD led to a greater proportion of H migrating compared with Me, suggesting NQEs may play a role in these reactions.

Finally, we trained an ACE potential on the alkyl rearrangement reaction of a sesquiterpene, for which a post transition state bifurcation (PTSB) has been proposed using static DFT. Using MLP-MD, we showed multiple products forming from one TS, supporting the hypothesis that a PTSB exists and that the enzymatic reactions of this sesquiterpene involve competing reaction pathways.

Overall, we have shown that MLPs can be applied to a range of organic rearrangement reactions. These reactions occur frequently in biological synthesis and due to their accuracy and efficiency, MLPs were used to investigate their pathways and mechanisms. While these reactions were studied in the gas phase, we hope that as MLPs develop, solvent can be more easily included and the analysis in this chapter provides a starting point for those wishing to study further organic reaction mechanisms using MLPs.

4

Umbrella Sampling Method Development

Contents

4.1 Summary 116
4.2 Quantitative measures of window sampling 117
4.2.1 US bin metrics 117
4.2.2 Quantitative overlap of windows 119
4.2.3 Effect of κ on sampling 124
4.3 Convergence of window distributions 127
4.3.1 Convergence of fitted Gaussians 127
4.3.2 Convergence of windows to fitted Gaussians 129
4.4 Free energy dependence on overlap 130
4.4.1 Free energy dependence using analytic systems 131
4.4.2 Free energy dependence using MLPs and MD 138
4.5 Conclusions 147

4.1 Summary

In umbrella sampling (US) simulations, dynamics are run generating a set of Gaussian-like windows (see **Section 1.4**).[11] Typically, these windows are visually inspected to ensure a 'sufficient', but quantitatively undefined, overlap is achieved.[122, 190] Some efforts have been made to quantify the overlap and its effect on US but these have been based on either the sampling between windows[134] or have assumed equal width and height windows[191]; these are discussed further in **Section 1.4.2**. This chapter presents a normalised overlap metric not reliant on equal windows and applies this metric to assess the impact of window overlap on the free energy for a range of analytic, MLP and MMMD systems.

Work has been carried out to assess the convergence of sampling within a US window.[191, 192] However, these efforts focus on convergence of the dynamics rather than convergence across the US reaction coordinate or to an ideal Gaussian distribution; both of which are addressed in this chapter. Further discussion of convergence metrics in US is presented in **Section 1.4.2**.

We begin this chapter in **Section 4.2** by proposing two metrics for quantifying the sampling quality in US. The first focuses on examining the extent of sampling in each bin along the reaction coordinate, named the *bin metric*, and the other measures the overlap between windows using the *normalised overlap* metric based on the overlap of fitted Gaussian functions. These metrics enables users to assess the quality of their US simulations and investigate optimal parameters to achieve sufficiently sampled distributions. We also define discrepancy, a quantity measuring the difference between the window reference and observed mean of the window distribution, which allows users to assess the effectiveness of the bias applied for each window.

In **Section 4.3**, we present a number of methods to measure the sampling convergence of a window under a harmonic bias. The convergence of the data to a Gaussian distribution is measured using the Kullback–Leibler (KL) divergence from statistics.[193]

Section 4.4 focuses on determining how free energy depends on overlap, using the nor-

malised overlap and KL-divergence metric. We examine two analytic systems, two simple reactive systems using MLPs and a protein system using classical MD simulations. We propose an approximate ideal minimum overlap to produce a converged free energy profile.

We conclude with a discussion in **Section 4.5** on the implications of our proposed minimum overlap for users of US. Details of the computational methods used in this chapter can be found in **Appendix B.3**.

4.2 Quantitative measures of window sampling

Free energy prediction from US relies on sufficient sampling across the reaction coordinate. However, defining what is 'sufficient' is challenging and often left to the experience of the user. In our work, since we are aiming for a free energy independent of our choice of initial conditions, we can define sufficient sampling as that where the free energy does not change when we increase the sampling time. This must be true in any particular region along the reaction coordinate. Therefore, our aim is to create a quantitative measure of sampling quality along our reaction coordinate. In the context of discrete data, each bin along the reaction coordinate should be a representative sample of the distribution in the ergodic limit. Before we investigate what constitutes sufficient sampling, we first present metrics that enable us to quantitatively measure sampling.

4.2.1 US bin metrics

One metric that partially captures the extent of sampling is to consider the proportion of bins from a histogram along the reaction coordinate that contain at least N samples, named the *bin metric*. If a significant proportion of bins, e.g., 10%, are not sampled or sampled poorly, then it is likely that we will have a poor estimate of the true distribution in these regions and consequently the free energy. One drawback of the bin metric is that our sampling estimate depends on both the number of samples, N, and our bin width. Choosing $N = 0$ may apparently solve the first dependency, but a distribution where many bins have only

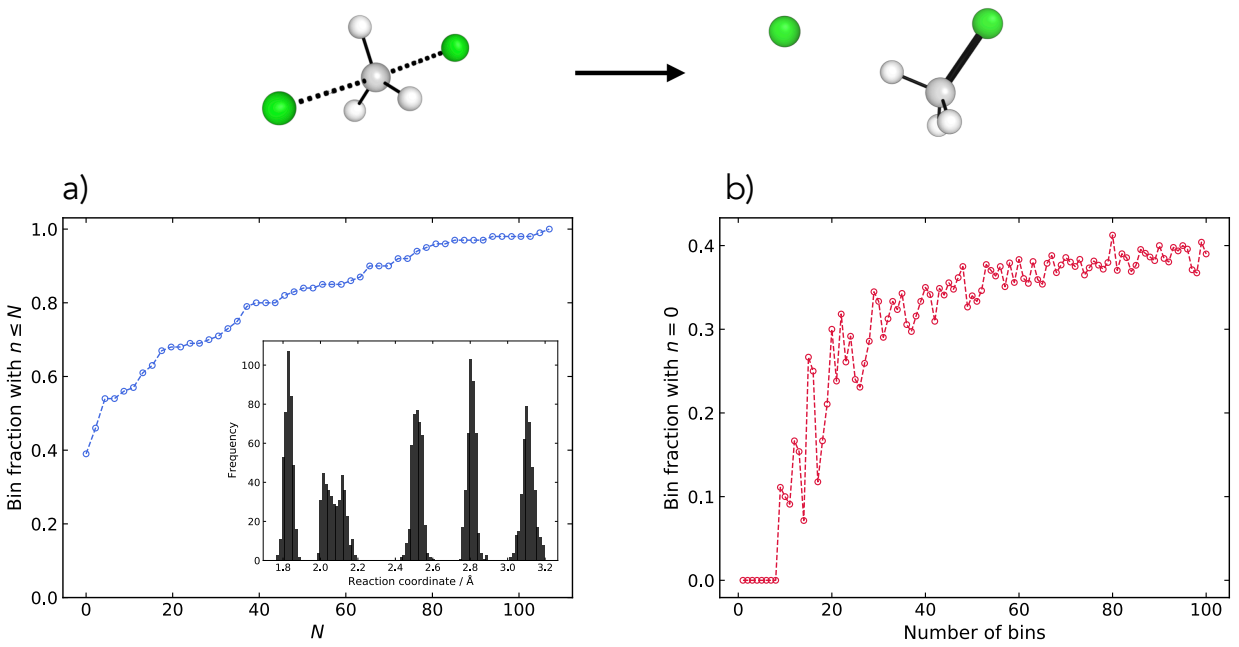


Figure 4.1: Measures of the coverage of sampling from a US simulation ($f(r) = r_{\mathrm{C–Cl}}$, $\kappa = 25$ eV Å^{-2}, $T = 300$ K, $n_{\mathrm{windows}} = 5$, $t = 1$ ps) of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ using a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP. a) The fraction of the number of bins (100 total) with $n \leq N$ samples for the distribution shown. b) The fraction of the number of bins containing $n = 0$ samples versus the number of bins.

one sample is also unlikely to represent the true distribution of our system.

To demonstrate the applicability of the bin metric, a simple US simulation of the $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ S_N2 reaction, **Fig. 4.1**, was performed using a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP.[1] The data was binned into a histogram and the bin fractions with $n \leq N$ samples were calculated as a function of both N and number of bins, **Fig. 4.1**.

Visual inspection of **Fig. 4.1a** shows that about half of the reaction coordinate is sampled at least once. The bin fraction with no samples ($n = 0$) is equal to about 40%, which fits with our visual inspection of **Fig. 4.1a**. If we choose $N = 20$ as a lower bound, about 60% of the bins have fewer than 20 points. This plot shows us how many bins are empty, but also how many bins contain only a few data points.

Given we are aiming for a sufficiently sampled reaction coordinate at every point, we want the bin fraction with zero points sampled to be as high as possible (ideally unity). We also

want to choose a reasonable number of bins such that there are enough to distinguish the data, but not too many so that most bins contain no data. From **Fig. 4.1b**, we can see that the bin fraction plateaus around 40%, suggesting 100 bins provides a reasonable balance of granularity and bin occupancy.

While the bin metric approach provides a quantitative measure of sample coverage across our reaction coordinate, it is sensitive to both the choice of N and the bin width. In addition, interpretation of this data is complex and doesn't provide information on the quality of specific windows. Therefore, we turned our attention to a simpler parameter independent approach.

4.2.2 Quantitative overlap of windows

In US, a system is typically constrained used a harmonic bias,

$$\omega(r) = \frac{\kappa}{2}(f(r) - \zeta^{\text{ref}})^2. \tag{4.1}$$

Here, κ is the spring constant and controls the strength of the bias, while the reference value, $\zeta^{\text{ref}} \equiv \zeta$, centres the bias at $f(r) = \zeta$ (further theoretical details on US can be found in **Section 1.4.1**). Using a harmonic bias, the window distributions in US go as a Gaussian with increasing κ, i.e., $p \sim \exp[-\beta\frac{\kappa}{2}(x - \zeta^{(k)})^2]$, resulting in the familiar Gaussian-like distributions. Discussing US convergence in terms of overlap between two window distributions relies on the assumption of convergence to a Gaussian (*vide infra*) but provides a clear representation of the sampling coverage across a US reaction coordinate.

As discussed in **Section 1.4.2**, previous overlap metrics involve the replica exchange average acceptance probability or assume equal width and height Gaussians. Instead, we present a simple normalised overlap metric based on the US data directly. We start by fitting a Gaussian of the following form to each distribution, i, setting $f(r) = x$, as

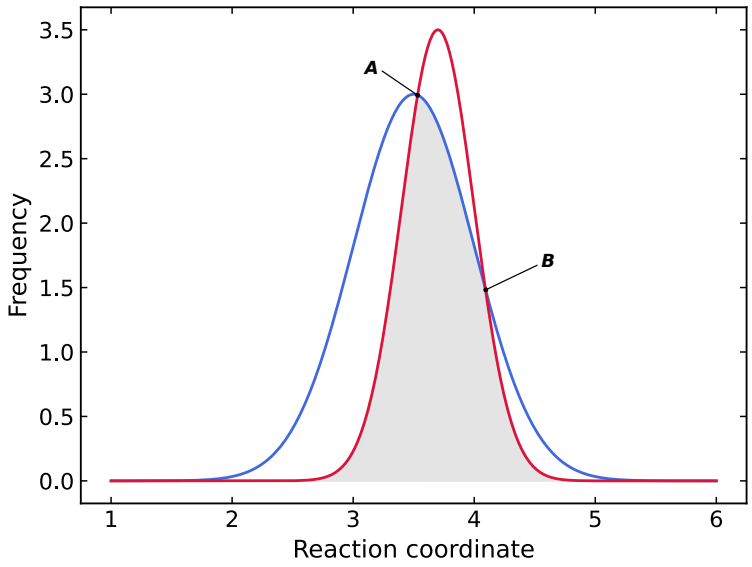


Figure 4.2: Two arbitrary Gaussian functions with the shaded region corresponding to their overlap. The shaded region traces the height of the lower Gaussian at each x-value, with intersections A and B.

$$y_i = a_i \mathrm{e}^{-(x-b_i)^2/2c_i^2}. \tag{4.2}$$

The parameters a_i, b_i and c_i represent the height, mean and standard deviation of the Gaussian for each window i, respectively. The window overlap is measured by computing the Gaussian overlap integral. As this value is relative to the a_i parameter of the specific Gaussians, it is not normalised across different windows. Instead we consider the overlapping area occupied by both distributions and normalise this value by the areas of each Gaussian, **Fig. 4.2**.

For the case where the two Gaussians have some overlap, S, (denoted by the shaded region in **Fig. 4.2**), the boundary of this region is defined as the value of the lesser of the two Gaussians at each x-value and the $y = 0$ line. Since the intersections are not smooth, there is no analytical integral we can solve involving the two Gaussians. The area, I, is instead formed as a sum of integrals, whose limits are defined by the points of intersection.

$$I = \int_{-\infty}^{A} a_1 \mathrm{e}^{-(x-b_1)^2/2c_1^2}\,\mathrm{d}x + \int_{A}^{B} a_2 \mathrm{e}^{-(x-b_2)^2/2c_2^2}\,\mathrm{d}x + \int_{B}^{\infty} a_1 \mathrm{e}^{-(x-b_1)^2/2c_1^2}\,\mathrm{d}x \tag{4.3}$$

Here, the parameters a_1, b_1, c_1 refer to the Gaussian which takes lower x values up to A and after B, and the a_2, b_2, c_2 parameters refer to the Gaussian describing the region between A and B. The limits A and B are the points of intersections between the Gaussians and can be computed using the quadratic equation by equating the two Gaussians, **Eqn. 4.4**.

$$B, A = \frac{\frac{c_1^2}{c_2^2}b_2 - b_1 \pm \sqrt{\left(\frac{c_1^2}{c_2^2}b_2 - b_1\right) - \left(\frac{c_1^2}{c_2^2} - 1\right)\left(2c_1^2 \ln \frac{a_1}{a_2} + \frac{c_1^2}{c_2^2}b_2^2 - b_1^2\right)}}{\frac{c_1^2}{c_2^2} - 1} \tag{4.4}$$

Practically, the simplest way to compute this integral is to calculate the integral of the minimum of two Gaussians between $-\infty$ and ∞.

$$S = \int_{-\infty}^{\infty} \min\left(a_1 \mathrm{e}^{-(x-b_1)^2/2c_1^2}, a_2 \mathrm{e}^{-(x-b_2)^2/2c_2^2}\right) \, \mathrm{d}x \tag{4.5}$$

We can then normalise this overlap by the area of either Gaussian ($\sqrt{2}\pi a_i c_i$), giving us a value of normalised overlap, S_i, between 0 and 1.

$$S_i = \frac{S}{\sqrt{2}\pi a_i c_i} \tag{4.6}$$

Since S_i is the overlap relative to each Gaussian i, we obtain two values of S_i for each window. This rapidly computable value provides a quantitative method of measuring the overlap between two US windows. One limitation of this method is that it assumes the data is Gaussian; this is addressed in **Section 4.3**.

The derivative of the overlap as one Gaussian is shifted through another Gaussian tells us how sensitive the overlap is with respect to $\Delta\zeta$. To determine the functional form of the derivative of the overlap, i.e., the sensitivity, we first express the overlap in closed form, **Eqn. 4.7**.

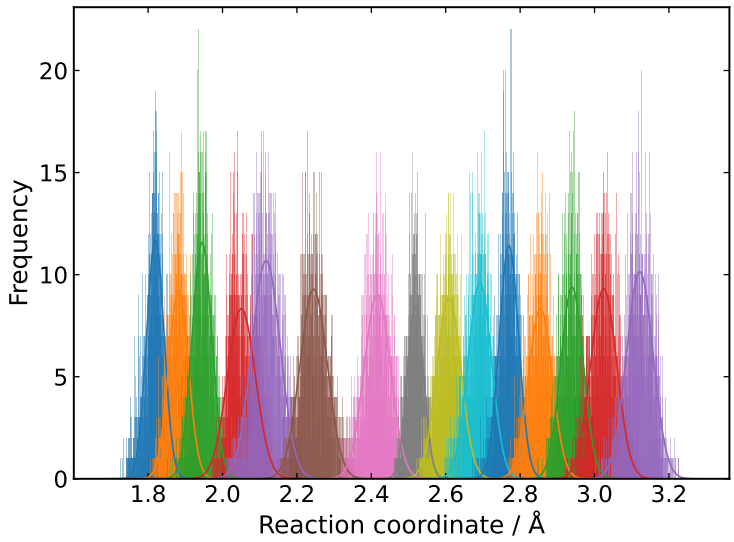


Figure 4.3: Window distributions from a US simulation ($f(r) = r_{\mathrm{C–Cl}}$, $\kappa = 25$ eV Å^{-2}, $T = 300$ K, $n_{\mathrm{windows}} = 15$, $t = 1$ ps) of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ using a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP. Gaussian functions fitted to each window histogram.

$$S = \frac{1}{2a_i c_i}\left\{ a_1 c_1 \left[2 - \mathrm{erf}\left(\frac{b_1 - B}{c_1\sqrt{2}}\right) + \mathrm{erf}\left(\frac{b_1 - A}{c_1\sqrt{2}}\right)\right] + a_2 c_2 \left[\mathrm{erf}\left(\frac{b_2 - B}{c_2\sqrt{2}}\right) - \mathrm{erf}\left(\frac{b_2 - A}{c_2\sqrt{2}}\right)\right]\right\}. \tag{4.7}$$

Taking the derivative of this expression with respect to x gives the sensitivity, which changes similarly to $\tanh x$ for two identical Gaussians (the explicit form has been suppressed for the sake of brevity). The sensitivity is mostly zero while they do not overlap, but rises exponentially as they approach one another before reaching a maximum as their x centres become equal. The exponentially changing behaviour of the overlap means that a small shift in the position of the Gaussian can have a large impact on the sampling overlap. Therefore, the effect on free energy is likely to also be highly sensitive to the overlap.

We can test this measure of window overlap to our simple test system, $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ using a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP. A US simulation was performed along the reaction coordinate of this system, with Gaussians fitted to the window distributions, **Fig. 4.3**.

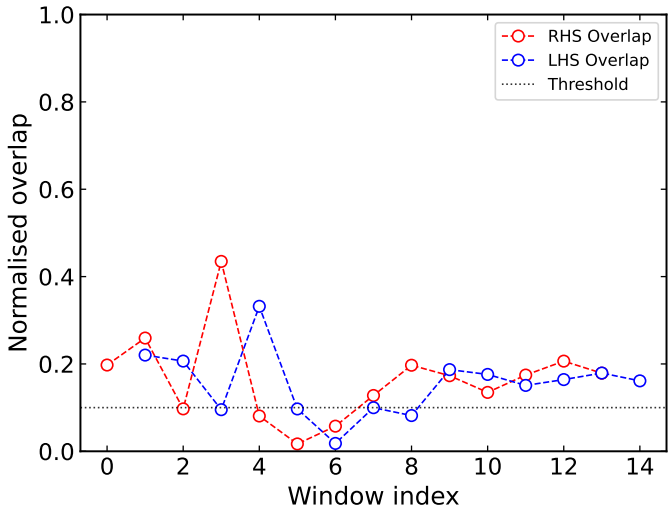


Figure 4.4: Normalised Gaussian overlap for each window pair of the reaction $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ from a US simulation (detailed in **Fig. 4.3**). Overlap is calculated between a window with the window to its left (LHS) and right (RHS). A threshold dotted line is drawn on to show a *potential* cutoff between an acceptable and insufficient overlap.

Window reference values, ζ, were chosen uniformly between the reactants and products, as determined from the IRC. For a flat underlying potential energy surface (PES), in the ergotic limit, the distributions would be perfectly centred around ζ, with each bin height equalling the value of the fitted Gaussian at that point. However, the steep free energy profile around the TS in the S_N2 reaction provides an external force acting against the bias. The resulting window distributions are unevenly distributed along the reaction coordinate, particularly around 2.3 Å, the C–Cl distance corresponding to the transition state.

We can calculate the normalised overlap between each pair of windows using **Eqn. 4.6**, shown in **Fig. 4.4**. The corresponding overlap for each window shows an increase in S_i in the minimum energy regions while around the TS, S_i is lower. Additional windows around low overlap regions solve the issue of sufficient sampling along the reaction coordinate, but can result in a considerably higher computational cost.

The quantitative method of measuring overlap between two windows enables users to assess the potential quality of their US simulations. In addition, we can use this metric to investigate which values of normalised overlap, S_i, lead to sufficiently sampled distributions.

4.2.3 Effect of κ on sampling

Within each window in US, the reaction coordinate is controlled by the underlying unbiased PES and the added bias, **Eqn. 4.1**, which is parameterised by the spring constant, κ. Generally, the higher the value of κ, the smaller the standard deviation in each Gaussian. When $\kappa = 0$, the system is completely controlled by the underlying energy surface. If the forces along the reaction coordinate are low, then a small κ may be sufficient. But along a reaction coordinate there will be a tendency towards local minima, resulting in poor sampling around the TS. Increasing κ may improve overlap up to a point as a result of a more uniform distribution along the reaction coordinate. However, as κ tends to infinity, the overlap will tend to zero, as all the sampled points will be equal to ζ.

To investigate the effect of κ on the sampling, we can use the normalised overlap metric, **Eqn. 4.6** and another quantity, the discrepancy. We define the discrepancy as the absolute difference between the reference value, ζ, and the mean of the fitted Gaussian, in units of the reaction coordinate, **Eqn. 4.8**. This provides an indication of whether the bias is strong enough to overcome the forces due to the underlying PES.

$$D = |\zeta^{\text{ref}}_{\text{actual}} - \zeta^{\text{ref}}_{\text{target}}| \tag{4.8}$$

From the US simulation of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$, we can calculate the discrepancy for each window, **Fig. 4.5**.

From **Fig. 4.5**, it can be seen that the discrepancy follows an inverse trend to the overlap, with windows of lower overlap corresponding to windows of high discrepancies. For the S_N2 system under study, the highest discrepancy is observed in windows 2–4, which correspond to the TS region.

We can now investigate the effect of varying κ on the average overlap and discrepancy by running the US with the same parameters but increasing κ, **Fig. 4.6**. In total, 13 US simulations were run, with κ ranging from 1 eV Å^{-2} to 1000 eV Å^{-2}. The histograms and

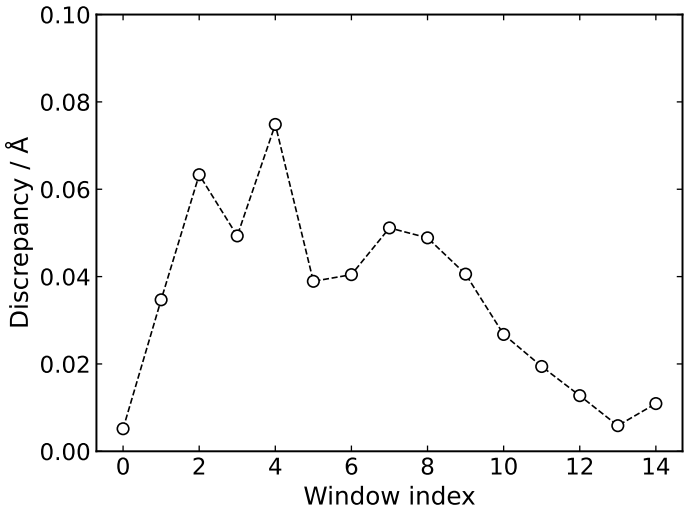


Figure 4.5: The discrepancy, D, **Eqn. 4.8**, between the target reference value and the observed mean of the fitted Gaussian for the reaction $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ from a US simulation (detailed in **Fig. 4.3**).

fitted Gaussians for four of these simulations are shown in **Fig. 4.6**.

For low values of κ (≈ 1 eV Å^{-2}), the distributions are controlled by the underlying PES, forcing the system towards the minima. As κ increases, the distributions become more evenly spaced, becoming less affected by the PES. Finally, as κ approaches excessively large values (≈ 1 eV Å^{-2}), the windows are evenly spaced with small standard deviations. We can quantify the effect of κ on both the overlap and discrepancy of our simulations, **Fig. 4.7**.

The overlap and discrepancy in **Fig. 4.7** quantitatively reflect what we see from **Fig. 4.6**. Small values of κ result in large overlap and discrepancy as the system is pulled towards the minima. Large values of κ cause tight distributions centred on the target reference but will low overlap. In US, finding the optimal κ becomes a constrained minimisation problem subject to a sufficient overlap and sufficient discrepancy allowance. While this is a known problem, the overlap metric presented in this work enables users to quantitatively define suitable κ and ζ values based on the overlap. This problem also assumes equal values of κ for all windows, as well as uniformly chosen reference values, which is the subject of **Chapter 5**, where these are varied.

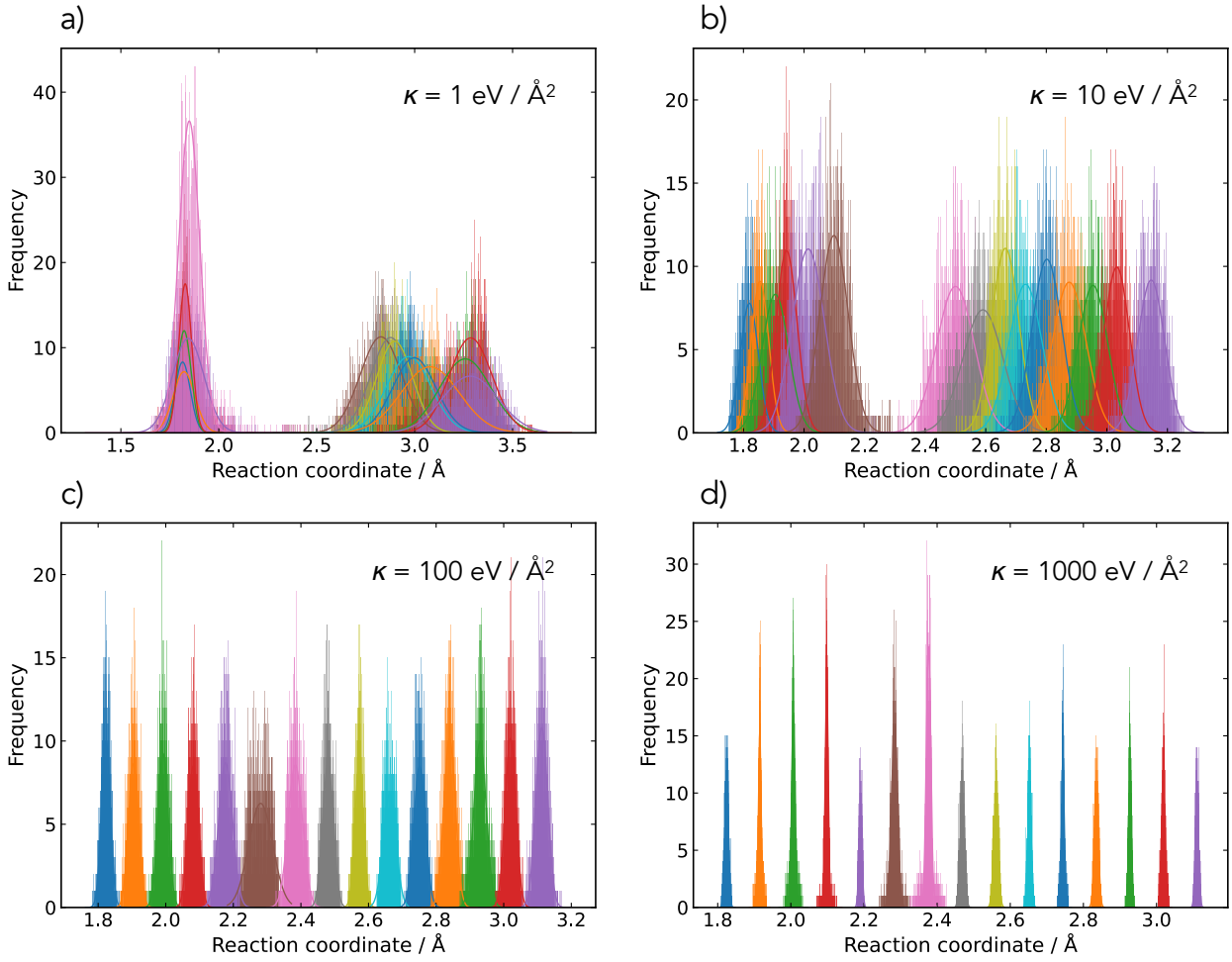


Figure 4.6: Reaction coordinate distribution of the reaction $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ from a US simulation (detailed in **Fig. 4.3**) for four evenly logarithmically spaced values of κ.

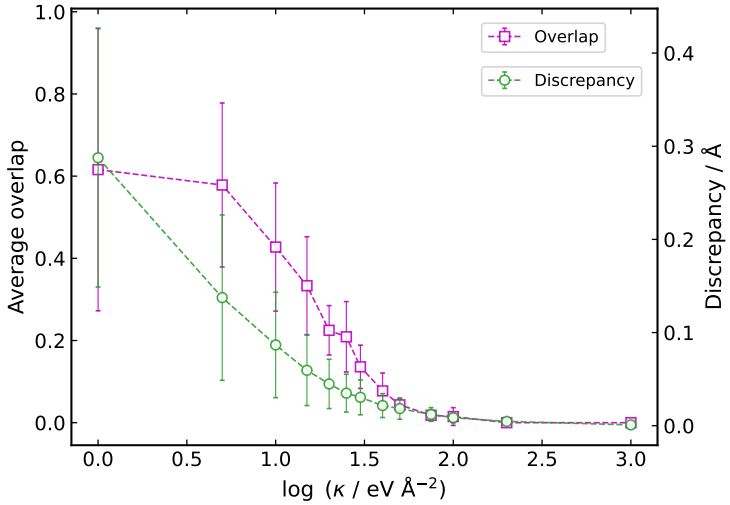


Figure 4.7: The average overlap and discrepancy as a function of κ of the reaction $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ from a US simulation (detailed in **Fig. 4.3**). Errors are given as one standard deviation.

4.3 Convergence of window distributions

Obtaining sufficient overlap and converged sampling is crucial to producing a complete free energy profile along a reaction coordinate. In addition to this is the condition that the sampling within each window has converged. We can measure the sampling convergence of the data from two perspectives: the convergence of the fitted Gaussians with time along a trajectory, **Section 4.3.1,** and the convergence of the binned data to the fitted Gaussians, where we utilise Kullback-Leibler (KL) divergence, **Section 4.3.2**.

4.3.1 Convergence of fitted Gaussians

Taking a single window sample, the simulation data can be divided into 10% increments and we can fit a Gaussian to each cumulative increment up to the final fit, which includes 100% of the data, **Fig. 4.8**. From this analysis, we can visually inspect whether the mean and standard deviation changes. This provides a qualitative insight on whether the window length is long enough to ensure convergence of the Gaussian.

For example, for the S_N2 reaction under study, **Fig. 4.8**, we can see that the mean of the Gaussians do not change significantly from 1 ps of sampling to 10 ps. However, the

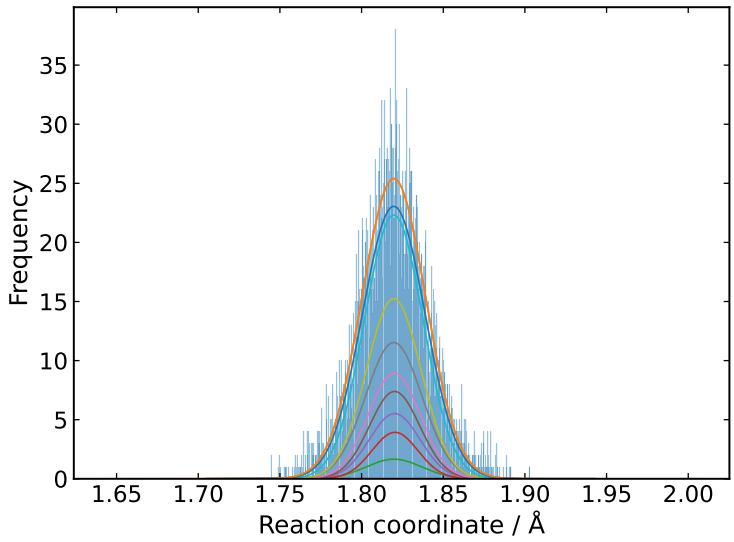


Figure 4.8: Fitted Gaussians to 10% increments of one US window ($Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ (detailed in **Fig. 4.3**), $t = 10$ ps), showing the underlying histogram for the complete data set. Heights of the Gaussians increase as more data is added, showing the convergence more clearly.

mean is a somewhat crude measure of convergence as the underlying data may lead to error cancellation since sampling is roughly equally on either side mean of the window reference. The standard deviation is more difficult to discern from this plot due to variance in height of the Gaussians (which is related the number of data points and changes as more data is included in the fit). However, from visual inspection this also does not vary significantly. The difference between the underlying bin heights and the fitted Gaussians shows the data is not fully Gaussian. As κ increases, the data will become more Gaussian and as the sampling time increases, these bins will fit more tightly to their 'true' distribution, which may or may not be Gaussian. We can improve upon this plot by calculating the numerical values of the means and standard deviations as we increase the sampling, **Fig. 4.9**.

In **Fig. 4.9**, due to the scale of the y-axes, it appears that convergence is slow for the standard deviation and mean. However, both experience a variation of only 0.025 Å and 0.0012 Å, respectively, showing that both parameters are quickly converged. This highlights a limitation of this analysis, since we do not know *a priori* what an acceptable variation of mean and standard deviation will be. Plotting these values relative to the window width may prove more insightful, however, the importance of the variation is somewhat dependent

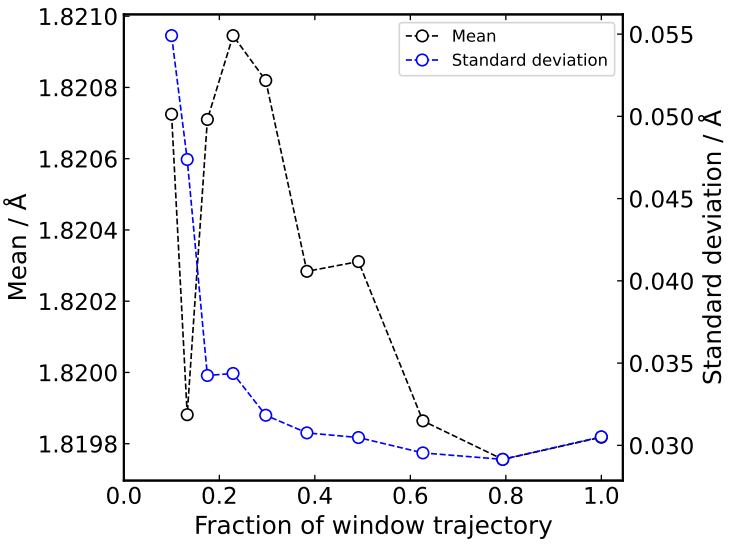


Figure 4.9: Variation of the mean and standard deviation in 10% increments of one US window ($Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ (detailed in **Fig. 4.3**), $t = 10$ ps).

on the system. Nonetheless, they provide a quantitative description of how our window distributions vary with our dataset size.

4.3.2 Convergence of windows to fitted Gaussians

One assumption that we have been making up to this point, particularly in the analysis of overlap, is that our data is Gaussian. Since we only sample relatively short time dynamics ($\mathcal{O}$(ps)), this is often not the case. We can plot the difference between the bin heights and the fitted Gaussian to measure the convergence of the data to the Gaussian. In addition to plotting this difference visually, we can measure how Gaussian the data is by using the KL-divergence.

KL-divergence, also called the relative entropy, is a statistical distance that measures the difference between two distributions.[193] It has been previously used with US to measure the convergence of distributions to a reference set (see **Section 1.4.2**), but not convergence within individual windows.[134, 135] For two identical distributions, the KL-divergence is zero, and increases in magnitude as the distributions diverge. It has a form similar to Shannon entropy, as shown in **Eqn. 4.9**.

$$D_{\mathrm{KL}}(P||Q) = -\sum_{x\in\chi} P(x)\log\left(\frac{Q(x)}{P(x)}\right) \tag{4.9}$$

KL-divergence, in the context of a US window, is the expectation of logarithmic difference between the distributions Q and P, where Q is our fitted Gaussian and P is our observed data. When a bin in our distribution is empty, i.e., $P(x) = 0$, then its contribution to our KL-divergence is also zero since,

$$\lim_{x\to 0^+} x\log(x) = 0. \tag{4.10}$$

While a KL-divergence of zero indicates perfectly Gaussian data, interpreting non-zero values becomes difficult since the KL-divergence depends on the number of bins in the set. Therefore, we can define an intrinsic KL-divergence, S, a KL-divergence per bin, N,

$$S = -\frac{1}{N}\sum_{x\in\chi} P(x)\log\left(\frac{Q(x)}{P(x)}\right). \tag{4.11}$$

Here, N refers to the number of bins in our distribution. This value can then be used to measure how the data converges to our fitted Gaussian as we increase the sampling. Applying this measure to one of our US windows, we can produce a description of the convergence of the data to the fitted Gaussian, **Fig. 4.10**.

4.4 Free energy dependence on overlap

Now that we have an range of tools to measure the quality of the sampling in terms of overlap, discrepancy and convergence of data, we can vary our input parameters to assess how the free energy depends on the overlap of windows.

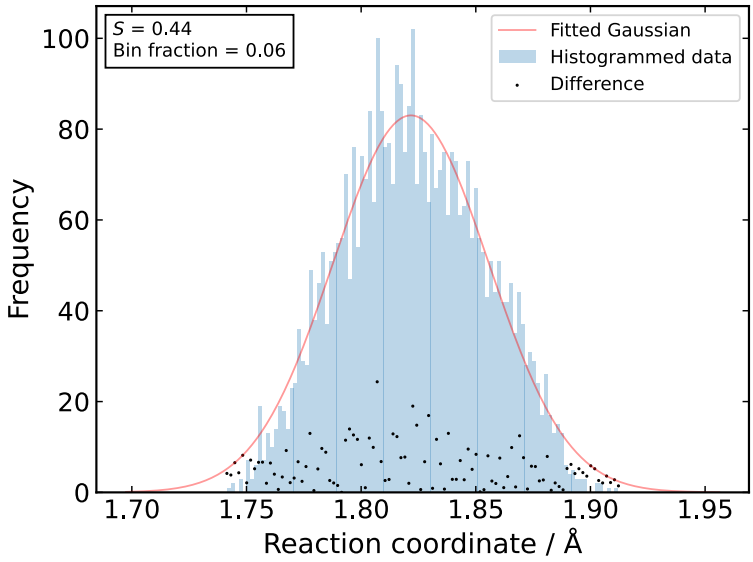


Figure 4.10: Fitted Gaussian, underlying histogrammed data and difference between the two. KL-divergence, S and bin fraction of the data with no samples annotated. One US window with 100 bins ($Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ (detailed in **Fig. 4.3**), $t = 10$ ps).

4.4.1 Free energy dependence using analytic systems

To reduce the influence of external factors on assessing the free energy quality, such as the underlying potential and sampling, we start with simple 1D systems, which we can solve analytically. We focus on two 1D models, $U(r) = r^2$, **Section 4.4.1.1**, and $U(r) = r^4 - 4r^2 + 4$, **Section 4.4.1.2**.

4.4.1.1 The $U(r) = r^2$ model

We first focus on the potential $U(r) = r^2$. The probability of observing our system at $q = r$ from the Boltzmann distribution is given by **Eqn. 4.12**.

$$P(q) = \mathrm{e}^{\beta A_0} \int_{-\infty}^{\infty} \mathrm{d}r \; \mathrm{e}^{-\beta r^2} \delta(r - q) \tag{4.12}$$

Here, $\mathrm{e}^{-\beta A_0}$ is the configurational partition function, which is given as

$$\mathrm{e}^{-\beta A_0} = \int_{-\infty}^{\infty} \mathrm{d}r \; \mathrm{e}^{-\beta r^2} = \sqrt{\frac{\pi}{\beta}}. \tag{4.13}$$

Therefore, the probability distribution function, $P(q)$, is simply

$$P(q) = \sqrt{\frac{\beta}{\pi}} \mathrm{e}^{-\beta q^2}. \tag{4.14}$$

Since free energy, $A(q)$, is related to the probability by $A(q) = -(1/\beta) \ln P(q)$, the free energy can be calculated analytically for a harmonic potential.

$$A(q) = -\frac{1}{\beta} \ln \left(\sqrt{\frac{\beta}{\pi}} \mathrm{e}^{-\beta q^2} \right) \tag{4.15}$$

$$= q^2 - \frac{1}{2\beta} \ln \frac{\beta}{\pi} \tag{4.16}$$

Therefore, the free energy for a 1D harmonic oscillator is equal to its potential energy, up to an additive factor which depends on the temperature. When $T = 0$, the entropy correction to the potential energy is zero, since $\lim_{x \to 0} x \ln x = 0$. If we consider a bias of the form $\omega(r) = \frac{1}{2}\kappa(r - \zeta)^2$, our total biased potential energy takes the following form.

$$E(r) = r^2 + \frac{1}{2}\kappa(r - \zeta)^2 \tag{4.17}$$

The corresponding biased distribution, which we will sample from, for a given reference ζ in window k at reaction coordinate q is

$$\tilde{P}(q, \zeta^{(k)}) = \mathrm{e}^{\beta A_k} \int_{-\infty}^{\infty} \mathrm{d}r \ \mathrm{e}^{-\beta r^2} \mathrm{e}^{-\beta \frac{\kappa}{2}(r - \zeta^{(k)})^2} \delta(r - q). \tag{4.18}$$

Here, A_k is the free energy associated with the biased potential

$$\mathrm{e}^{-\beta A_k} = \int_{-\infty}^{\infty} \mathrm{d}r \ \mathrm{e}^{-\beta r^2} \mathrm{e}^{-\beta \frac{\kappa}{2}(r - \zeta^{(k)})^2} = \sqrt{\frac{2\pi}{\beta(\kappa + 2)}} \mathrm{e}^{-\frac{\beta \kappa \zeta^{(k)2}}{\kappa + 2}} \neq f(q). \tag{4.19}$$

Since the free window free energy is not a function of the reaction coordinate q, this factor

will simply scale the probability distribution, thus not affecting each individual probability. Our biased probability distribution therefore becomes

$$\tilde{P}(q,\zeta^{(k)}) = A\mathrm{e}^{-\beta q^2}\mathrm{e}^{-\beta\frac{\kappa}{2}(q-\zeta^{(k)})^2} \tag{4.20}$$

$$= B\mathrm{e}^{-\beta\left(q^2+\frac{\kappa}{2}q^2-\kappa q\zeta\right)}, \tag{4.21}$$

where we have collected terms not depending on q in A and then B, and dropped the superscript k. Completing the square on the exponent and rearranging to the form of a standard Gaussian, we get

$$\tilde{P}(q,\zeta^{(k)}) = C\exp\left(-\frac{\beta\left(q-\frac{\kappa\zeta}{\kappa+2}\right)^2}{2(\kappa+2)^{-1}}\right). \tag{4.22}$$

To sample from a biased window, we sample from a Gaussian distribution with mean $\frac{\kappa\zeta}{\kappa+2}$ and standard deviation $[\beta(\kappa+2)]^{-\frac{1}{2}}$. Here, we see that increasing κ will shift the Gaussian towards ζ and tighten up the distribution. Varying ζ will change the mean but it is not equal to the mean except in the limit of $\kappa \to \infty$. Finally, increasing the temperature will loosen up the distribution, as expected physically. Using this simple model, we can now now assess the impact of sampling on our free energy and convergence of free energy. As an example of this system, we first run US over 10 windows each containing 10,000 steps, **Fig. 4.11**.

From this simple system, we can directly assess the accuracy of the WHAM method (see **Section 1.4.1.3**), including the extent of sampling to the exact free energy. For 10 windows with 10,000 steps, we produce a free energy and corresponding probability distribution close to the true answer. While we know the exact free energy, including the additive constant, A_0, WHAM predicts a free energy only up to an additive constant depending on the WHAM input parameters such as the value of κ. Therefore, the WHAM free energy in **Fig. 4.11** has been shifted by the appropriate constant to give the exact free energy. We can measure the 'closeness' of the true and WHAM-predicted probability distributions quantitatively using

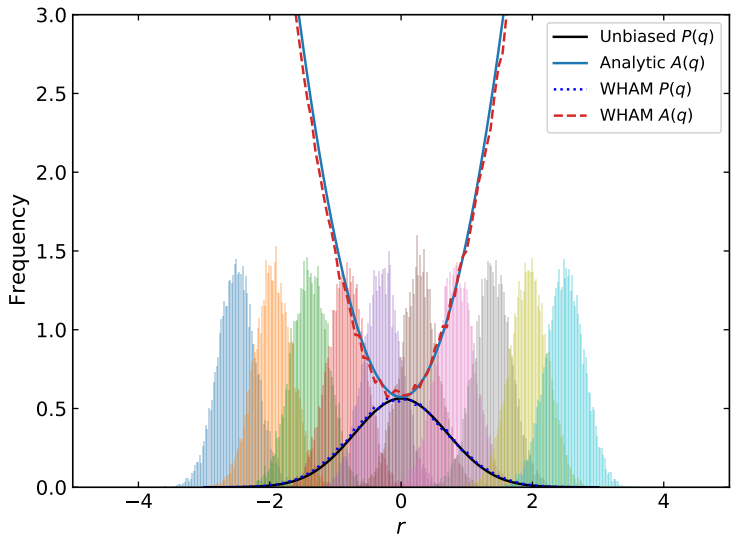


Figure 4.11: US of a 1D harmonic oscillator under a harmonic bias. Underlying window distributions are shown along with the exact and WHAM predicted free energy and probability distribution. Sampling run for 10,000 steps in each window, totalling 10 windows.

the KL-divergence, **Eqn. 4.23**.

$$S = \frac{1}{N} \sum_{q} \mathrm{e}^{-\beta(q^2 + S_{\mathrm{corr}})} \ln \frac{\mathrm{e}^{-\beta(q^2 + S_{\mathrm{corr}})}}{Q(q)} \tag{4.23}$$

Here, N is the number of bins in our distribution, $Q(q)$ is our WHAM-predicted probability distribution and S_{corr} is the entropy correction we must make to account for the shifted values of the WHAM free energies. For the example above in **Fig. 4.11**, the KL-divergence is calculated to be -0.0044, where a value under 0.005 produces a smooth free energy surface.

We can monitor the KL-divergence as a function of the extent of sampling and overlap. The extent of sampling, i.e., the number of points sampled per window, is a measure of the data convergence. Varying the number of samples, we can relate the KL-divergence to the quality of the free energy. First, we will assess the effect of ergoticity by varying the number of steps in each window, **Fig. 4.12**.

Increasing the number of steps per window reduces the KL-divergence between the exact and simulated free energies exponentially up to 1000 steps per window. At this point, the inherent difference between the exact free energy and the WHAM-predicted free energy is

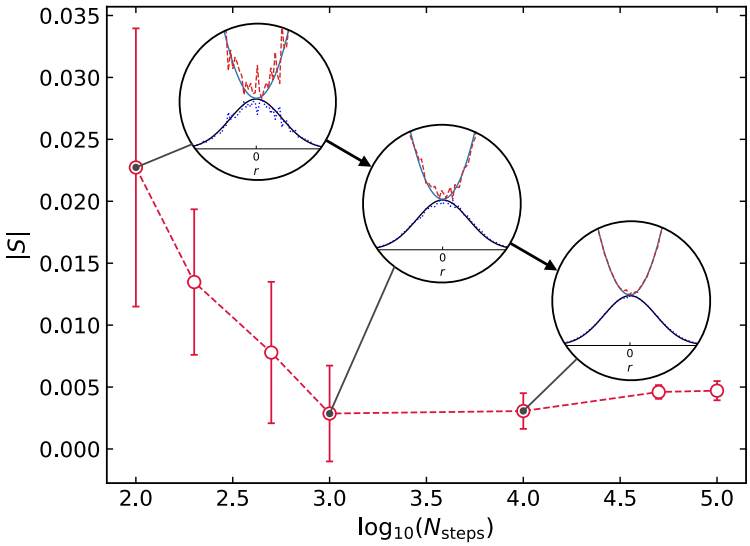


Figure 4.12: Absolute value of KL-divergence for a true and WHAM-predicted distributions of a 1D harmonic oscillator under a harmonic bias. Sampling run for varying number of steps, with each simulation containing 50 windows. Error shown as standard deviation over 20 repeats. True and predicted free energy and probability distributions shown for selected windows for one run of US.

reached. A further increase of steps per window reduces the standard deviation, with the free energy for 10,000 steps being almost identical to the exact value. Importantly, we must translate the values of KL-divergence to a meaningful description. Naturally, this will depend on our desired accuracy of free energy. However, we can give an approximate guide in that a KL-divergence of below 0.005 provides a smooth free energy surface at each point along the reaction coordinate.

Since we do not know the exact value of the free energy in real systems, we may be best to choose a number of steps in which the standard deviation of the results is small, to ensure an ergodic sampling. We can look at the effect of overlap on the quality of the free energy surface and use the results above to choose a sufficiently long window time to ensure ergodicity, **Fig. 4.13**.

Surprisingly, to achieve an accurate free energy (KL-divergence of 0.005), an overlap of only 15% is sufficient. Boosting the overlap significantly above 15% may therefore result in a greater computational cost without a significant gain in accuracy.

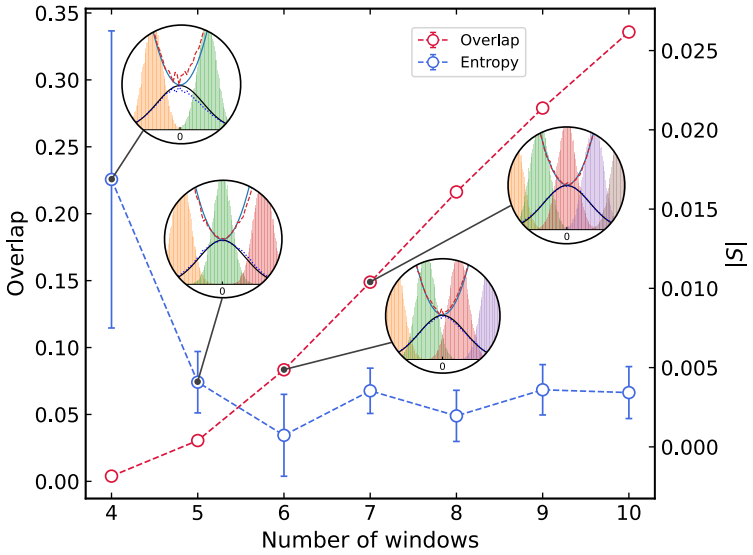


Figure 4.13: Absolute value of KL-divergence and average overlap for a true and WHAM-predicted distributions of a 1D harmonic oscillator under a harmonic bias. Sampling run for 50,000 steps, with each simulation containing varying number of windows. Error shown as standard deviation over 20 repeats. True and predicted free energy and probability distributions shown for selected windows for one run of US.

4.4.1.2 The $U(r) = r^4 - 4r^2 + 4$ model

With a view to extending this analysis to more complex 1D potentials, we can generate numerically exact free energies for comparison. For any 1D potential, our piece-wise free energy is given as

$$A(q) = -\frac{1}{\beta} \ln \frac{\int_{-\infty}^{\infty} \mathrm{d}r \ \mathrm{e}^{-\beta U(r)} \delta(r - q)}{\int_{-\infty}^{\infty} \mathrm{d}r \ \mathrm{e}^{-\beta U(r)}}. \tag{4.24}$$

The integral in the numerator is simply the Boltzmann factor of $U(r)$, and rearranging the arguments of the logarithm, we get

$$A(q) = U(q) + \frac{1}{\beta} \ln \int_{-\infty}^{\infty} \mathrm{d}r \ \mathrm{e}^{-\beta U(r)}. \tag{4.25}$$

Provided we can solve this integral numerically, we can obtain the exact free energy to an arbitrary accuracy. Moreover, for 1D potentials, the free energy is just the potential energy shifted by a constant. As for the sampling, we must sample from the following biased

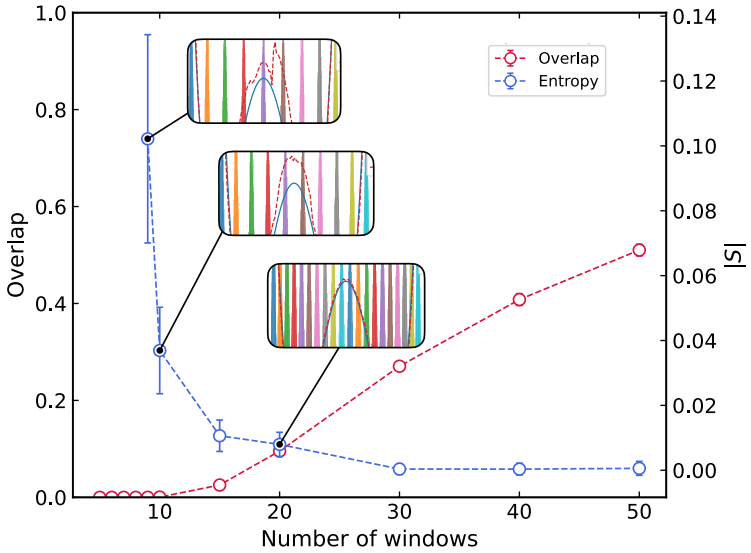


Figure 4.14: Absolute value of KL-divergence and average overlap for a true and WHAM-predicted distributions of a 1D $r^4 - 4r^2 + 4$ under a harmonic bias. Sampling run for 100,000 steps, with each simulation containing varying number of windows. Error shown as standard deviation over 20 repeats. True and predicted free energy and probability distributions shown for selected windows for one run of US.

distribution, **Eqn. 4.26**.

$$\tilde{P}(q, \zeta^{(k)}) = \frac{\mathrm{e}^{-\beta U(r)} \mathrm{e}^{-\beta \frac{\kappa}{2}(r-\zeta^{(k)})^2}}{\int_{-\infty}^{\infty} \mathrm{d}r \; \mathrm{e}^{-\beta U(r)} \mathrm{e}^{-\beta \frac{\kappa}{2}(r-\zeta^{(k)})^2}} \tag{4.26}$$

The integral in the denominator can be calculated numerically, which provides the normalisation constant. Meanwhile, points in the numerator can be sampled using the `numpy.random.choice` method. A potential more akin to a chemical reaction profile could be of the form $U(r) = r^4 - 4r^2 + 4$, giving a quartic function at the extrema while containing a barrier around $r = 0$. We can test the influence of overlap on the quality and convergence of the free energy for this potential.

Similarly for the harmonic potential, we see that almost as soon as some overlap is seen ($\lesssim 15\%$), the free energy converges to its respective KL-divergence value (0.001 in this case). The value of κ was set to 200 to ensure that the overlap across the reaction coordinate was approximately constant. The largest standard deviation in the overlap across the reaction coordinate using this κ was ~ 0.025; well within the variation of average overlaps between

different window numbers.

From the results of the 1D models, we have found that an overlap of $\approx 15\%$ is sufficient to obtain a smooth free energy profile. We have used qualitative analysis of the free energy compared to the exact analytical or numerical curves as well as quantitative analysis using KL-divergence to define a 'sufficient' overlap. Our analysis of KL-divergence also suggests that values of below 0.005 result in converged distributions.

4.4.2 Free energy dependence using MLPs and MD

The analytical systems in the previous section enable predictions of suitable values of overlap but do not necessarily translate to 'real' systems, such as those modelled using MLPs. We therefore turned to the study of three systems, an S_N2 reaction, **Section 4.4.2.1**, an isoindene intramolecular reaction, **Section 4.4.2.2** and the unbinding of Aβ42 protofibril, **Section 4.4.2.3**.

4.4.2.1 The S_N2 test system

The first is the $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ S_N2 reaction, where a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP was used to calculate the energies.[1] We started by utilising our US Python Package, *adaptiveus*, **Chapter 5**, to automate dynamics, overlap calculations, convergence criteria and free energy generation, **Fig. 4.15**. We also moved to a new reaction coordinate, $f(r) = r_{C\text{–}Cl} - r_{C\text{–}Cl'}$, where both chlorine atoms are considered in the reaction coordinate. This ensures a symmetric distribution about the TS.

The chosen parameters result in evenly spaced windows with a consistent overlap. The simulation of window 0 also shows reasonable convergence to the fitted Gaussian. Using this data, WHAM was used to calculate the ΔG over three repeats, **Fig. 4.16**.

The variation in ΔG for 10 ps windows is small across each side of the TS, but increases around the TS. For measuring the activation free energy barrier, $\Delta G^{\ddagger}$, this is taken as the difference between the minimum on the right-hand side to the maximum in the TS. Before

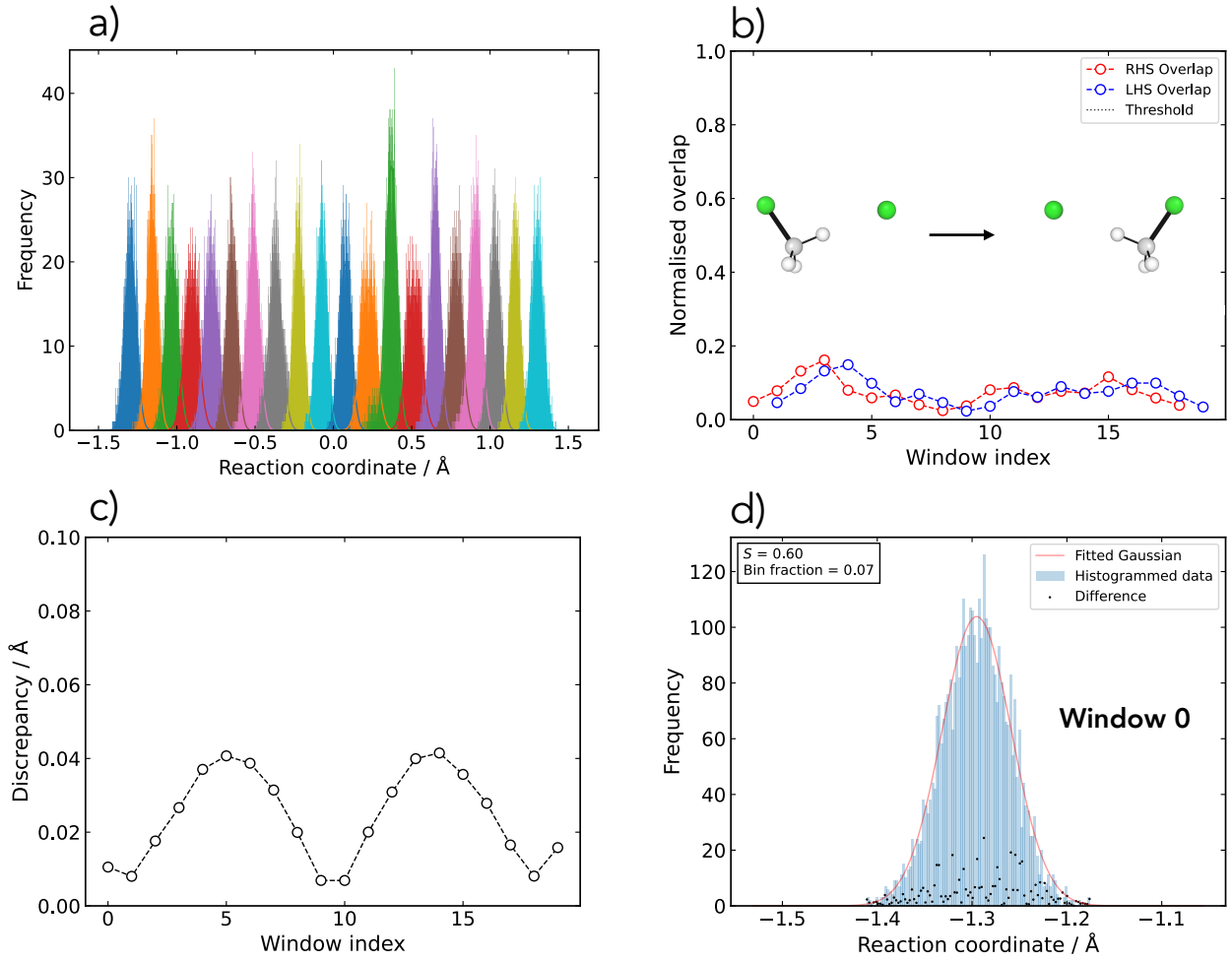


Figure 4.15: US simulation of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ ($f(r) = r_{C\text{–}Cl} - r_{C\text{–}Cl'}$, $\kappa = 20$ eV Å^{-2}, $T = 300$ K, $n_{\text{windows}} = 20$, $t = 10$ ps). a) Window histograms and fitted Gaussians. b) Overlap on LHS and RHS of each window. c) Discrepancy between the mean and target reference value. d) Fitted Gaussian for window 0 showing difference in height between the histograms and Gaussian. The KL-divergence and fraction of bins with no samples is also annotated.

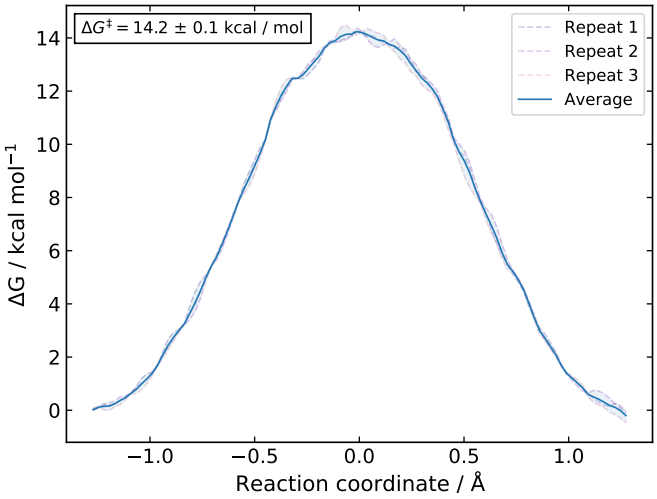


Figure 4.16: Free energy profile of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ (detailed in **Fig. 4.15**) Three repeats for each umbrella simulation are shown along with the average. Reaction free energy using the average is annotated.

investigating the influence of the overlap on the free energy, we assessed the most appropriate window trajectory length, such that the dynamics are converged, **Fig. 4.17**.

Plotting the activation free energy, $\Delta G^{\ddagger}$, shows that the spread of $\Delta G^{\ddagger}$ converges at around 10 ps per window. Increasing the sampling time to 30 ps does not change $\Delta G^{\ddagger}$ more than half a kcal mol^{-1}, but does decrease the standard deviation over the repeats. Using this setup, and choosing 30 ps per window to ensure sufficient sampling, we can now assess the effect of overlap on the quality of free energy for this S_N2 reaction.

Controlling for all variables except the overlap is not possible, since if the number of windows is kept constant and the range of ζ varied, then comparing the reference free energies is not justified. Likewise, keeping the range of the reaction coordinate the same but changing the number of windows changes the sampling across the reaction coordinate. Varying κ will also change the overlap, but it will often lead to a large variation in overlap across the reaction coordinates for lower values of κ. We opted to vary the number of windows, since it keeps the reaction coordinate range constant and is most likely to reflect a real case scenario, where more windows are added to increase the overlap (alongside varying κ).

Unlike the analytic systems, we have no exact reference value to measure in the KL-divergence.

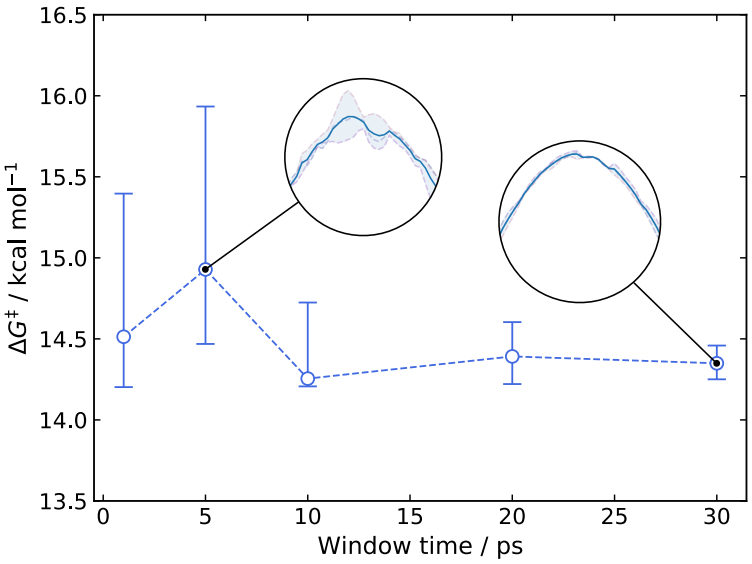


Figure 4.17: Activation free energy of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ (detailed in **Fig. 4.15**) with varying simulation times for each window. Free energy is averaged from three repeats and error bars are given as the maximum and minimum from these repeats. The free energy around the TS and products for the 5 and 30 ps simulations, respectively, are shown in the circles.

Instead, we choose a sufficiently converged free energy profile as our reference. In this case, $\max(n_{\text{windows}}) \equiv \mathcal{N} = 14$. Therefore, our KL-divergence calculation becomes

$$S = \frac{1}{N} \sum_{q \in \zeta} P_{n \neq \mathcal{N}}(q) \log \left(\frac{P_{n \neq \mathcal{N}}(q)}{Q_{\mathcal{N}}(q)} \right). \tag{4.27}$$

This comparison is naturally imperfect but serves as an approximate guide to compare overlap values to free energy quality. Moreover, we can visually assess the free energy for a given overlap to further justify when our sampling becomes sufficient. These quantities are calculated for the S_N2 reaction in **Fig. 4.18**.

For this system, the overlap steadily increases with the number of windows. The standard deviation also increases as a proportion of overlap. This is due to the discrepancy increasing around the steep energy barriers. While increasing κ would decrease the discrepancy, this was found to lead to a rougher free energy surface and is also unrepresentative of a typical κ used in a simulation. Nevertheless, we can correlate the overlap with the absolute value of KL-divergence. For $n_{\text{windows}} < 6$, $|S|$ could not be calculated due to missing regions in the free

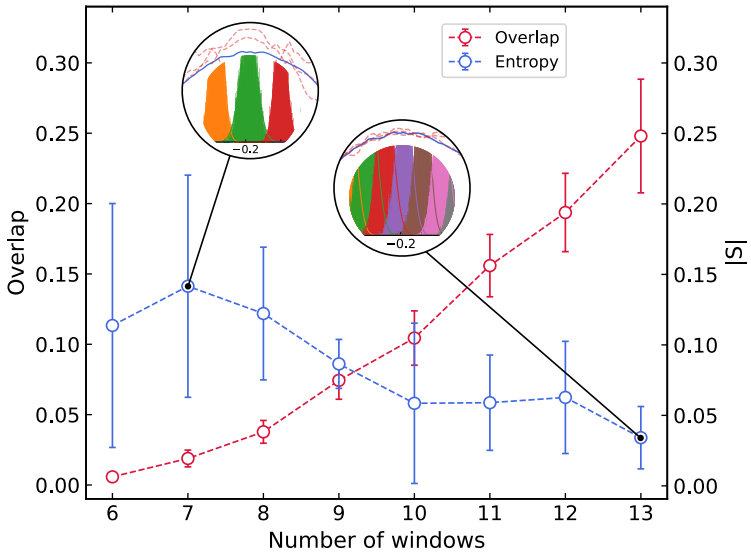


Figure 4.18: Overlap and absolute KL-divergence, $|S|$, for $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ (detailed in **Fig. 4.15**), with $t = 30$ ps and range $\zeta = [-0.5, 0.5]$), as a function of number of windows. Overlap and error for each n_{windows} is calculated as the average across all windows and then averaged over three repeats, with the standard deviation used as the error. Absolute KL-divergence is calculated by using averaged free energy from $n_{\text{windows}} = 14$ as the reference, Q, and each of the three repeats in $n_{\text{windows}} \neq 14$ as P. The $|S|$ values are averaged across the repeats and error is given as the standard deviation. For $n_{\text{windows}} = 7$ and 13, the free energy is shown (dashed red) with reference (solid blue), along with a sample of the histogrammed windows indicating visual overlap.

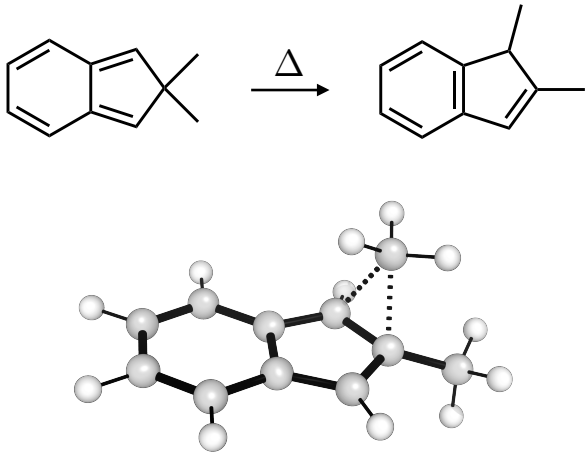


Figure 4.19: Rearrangement of isoindene molecular via heat activation and the corresponding transition state.

energy surface and can be assumed to have infinite KL-divergence. For $n_{\text{windows}} = 6$, where we have near zero overlap, we see a large $|S|$ with a large standard deviation. ΔG differs from the reference by ≈ 1 kcal mol^{-1}. As the overlap increases, $|S|$ decreases, settling around $|S| = 0.05$. We see some further improvement in $|S|$ as the overlap further increases, but this is well below 1 kcal mol^{-1}, and so only represents incremental improvements. Therefore, we can conclude that an overlap of between 10% and 20% is sufficient to provide free energy convergence within 1 kcal mol^{-1}.

4.4.2.2 The isoindene test system

We then moved to a more complex chemical system, an intramolecular methyl migration in an isoindene, previously studied in **Section 3.3** and detailed in **Section 1.3.1**, to assess the effect of overlap on free energy, **Fig. 4.19**.

First, US was performed along the reaction coordinate and the following quantities were generated, **Fig. 4.20**. The US produced a smooth free energy profile, with a minimal deviation from the mean over the repeats. The overlap across windows is fairly constant and high, except around the transition state where it dips to $\approx 10\%$. However, this only results in a small variance in free energy in the reaction profile. The data itself is close to Gaussian, with a representative example from the first window shown in **Fig. 4.20c**.

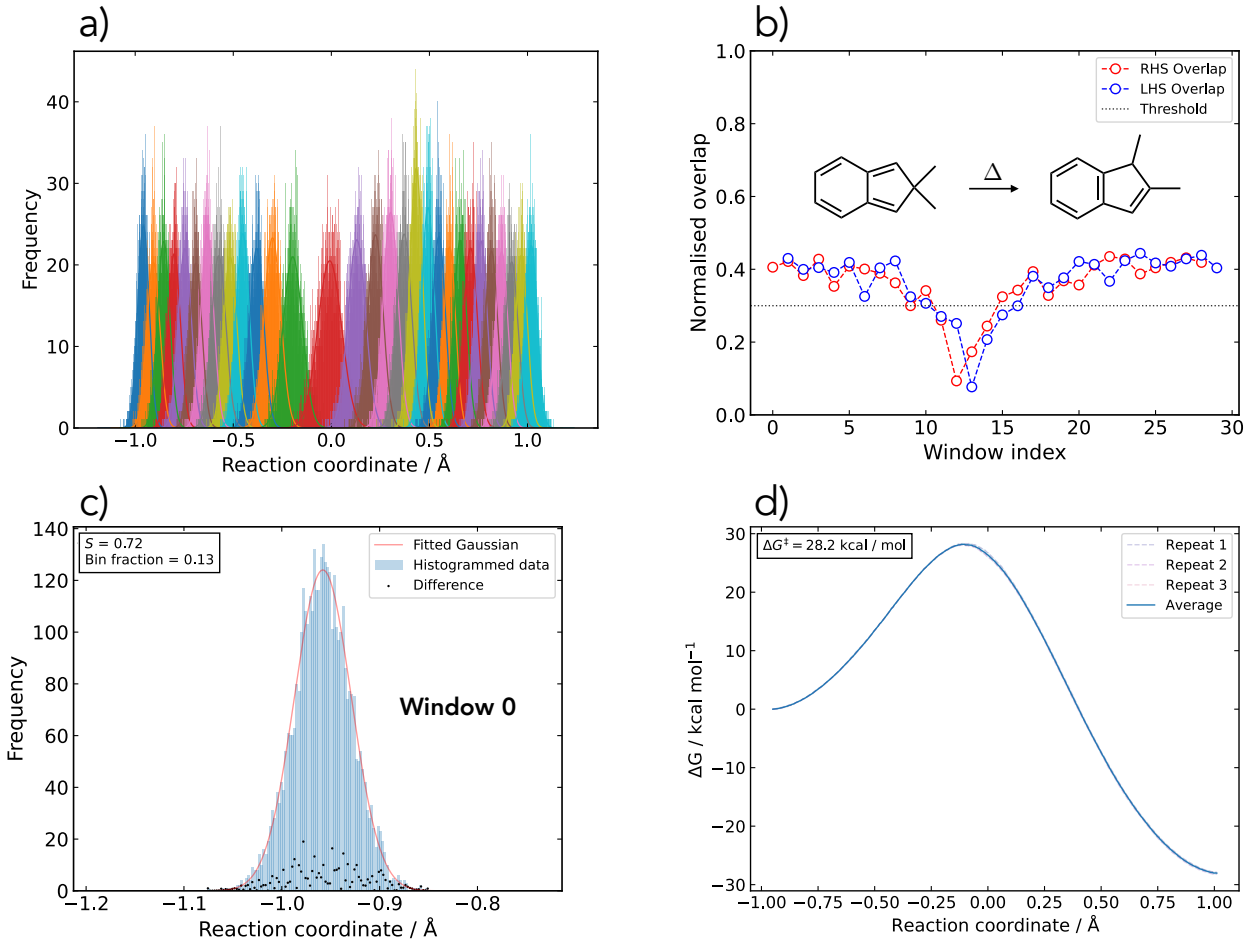


Figure 4.20: US simulation of isoindene methyl migration ($f(r) = r_{C_1–C_3} - r_{C_2–C_3}$, $\kappa = 20$ eV Å^{-2}, $T = 300$ K, $n_{\text{windows}} = 30$, $t = 10$ ps) using an ACE potential trained at PBE0-D3BJ/def2-SVP. a) Window histograms and fitted Gaussians. b) Overlap on LHS and RHS of each window, including an arbitrary threshold line and the reaction scheme. c) Discrepancy between the mean and target reference value. d) Fitted Gaussian for window 0 showing difference in height between the histograms and Gaussian. The KL-divergence and fraction of bins with no samples is also annotated.

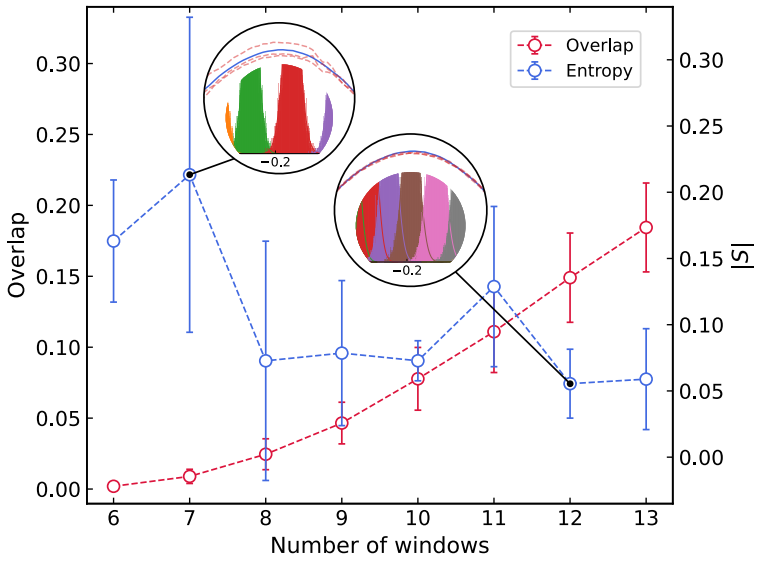


Figure 4.21: Overlap and absolute KL-divergence, $|S|$, for isoindene methyl migration ($f(r) = r_{C_1\text{–}C_3} - r_{C_2\text{–}C_3}$, $\kappa = 40$ eV Å^{-2}, $T = 300$ K, $n_{\text{windows}} = 30$, $t = 30$ ps and range $\zeta = [-0.5, 0.25]$ Å), as a function of number of windows. Overlap and error for each n_{windows} is calculated as the average across all windows and then averaged over three repeats, with the standard deviation used as the error. Absolute KL-divergence is calculated by using averaged free energy from $n_{\text{windows}} = 14$ as the reference, Q, and each of the three repeats in $n_{\text{windows}} \neq 14$ as P. The $|S|$ values are averaged across the repeats and error is given as the standard deviation. For $n_{\text{windows}} = 7$ and 12, the free energy is shown (dashed red) with reference (solid blue), along with a sample of the histogrammed windows indicating visual overlap.

To ensure a more consistent overlap across windows, κ was set to 40 Å eV^{-2}. The window sampling time was also increased to 30 ps to remain consistent with previous figures and ensure fully converged sampling. We then varied the number of windows to vary the overlap between windows and measured the resulting effect on the free energy, **Fig. 4.21**.

As with the overlap from the S_N2 system, with 6 windows the overlap is close to zero, and the mean and standard deviation rise as the number of windows is increased. The KL-divergence, referenced to US with 14 windows, remains large for 7 windows, with the deviation between repeats causing a large standard deviation. Here, ΔG differs from the reference by ≈ 1 kcal mol^{-1}. As the overlap increases, the KL-divergence and thus the free energy, closely approaches the reference free energy. At an overlap of about 15%, ΔG lies within 1 kcal mol^{-1}. This result is similar to the S_N2 system, where an overlap of roughly $15 - 20\%$ resulted in a converged free energy.

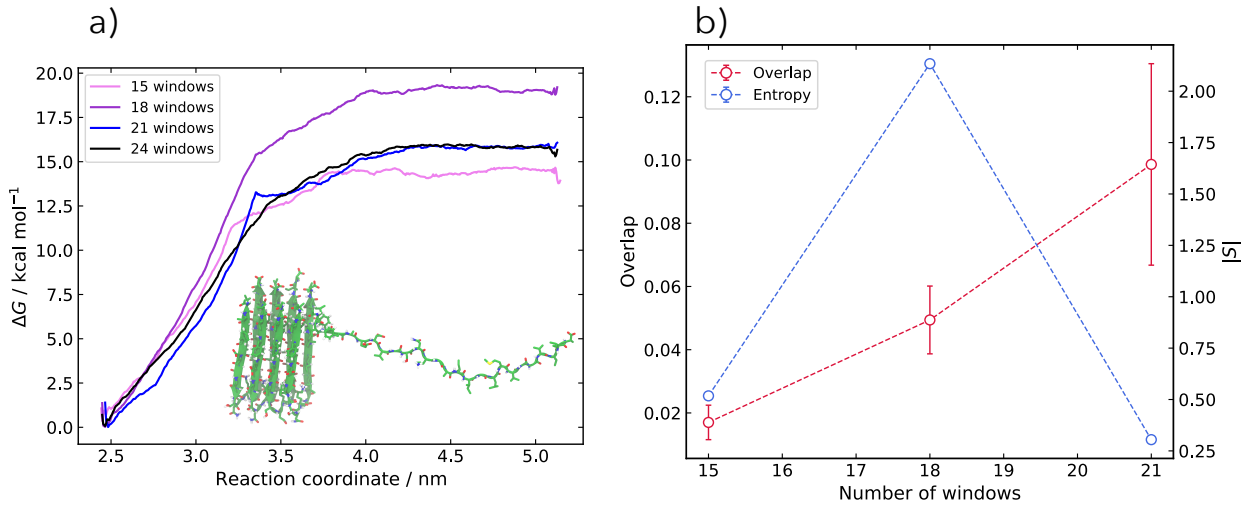


Figure 4.22: a) Free energy of unbinding of Aβ42 protofibril (2BEG PBD code) from ref [126] with varying numbers of US windows ($f(r) = \text{COM}_{AB}$, $\kappa = 2000$ KJ mol^{-1}, $T = 300$ K, $t = 5$ ns with 100 ps equilibration, $\zeta = [2.6, 5.0]$ nm). b) Overlap and absolute KL-divergence, $|S|$, as a function of number of windows. Overlap and error for each n_{windows} is calculated as the average across all windows with the standard deviation used as the error. Absolute KL-divergence is calculated by using averaged free energy from $n_{\text{windows}} = 24$ as the reference, Q, and each of the $n_{\text{windows}} \neq 24$ as P.

4.4.2.3 The Aβ42 protofibril test system

The final test system we investigated was a classical MD US simulation of a fibril unbinding of Aβ42 protofibril (2BEG PBD code), previously studied by Lemkul and Bevan.[126] This commonly used example of US provides a simple protein system to assess the influence of window overlap on free energy. After a pulling simulation to generate the reaction coordinate of one of the fibrils unbinding from the remaining four, a series of US simulations were performed using the GROMOS96 53A6 force field, **Fig. 4.22**.

In the original work, the authors calculated an unbinding $\Delta G^{\ddagger} \approx 16$ kcal mol^{-1} for the dissociating fibril between 2.5 nm and 5 nm. In our work, we also calculated a $\Delta G^{\ddagger} \approx 16$ kcal mol^{-1}. Using 24 windows, corresponding to an overlap of 14% $\pm$ 5%, as a reference, we compared simulations with lower overlaps. Using 15 windows, we obtained an average overlap of under 2% and see $\Delta G^{\ddagger}$ under-predicted by ≈ 2 kcal mol^{-1}. While this is reasonably close, this could be due to error cancellation, as using 18 windows (an overlap of 5%) leads to an over-prediction of $\Delta G^{\ddagger} \approx 3$ kcal mol^{-1}. Finally, when we used 21 windows, we obtained

an overlap 10% ± 3%, and a $\Delta G^{\ddagger} \approx 16$ kcal mol^{-1}. These results further support the proposition that an overlap of ≈ 15% is sufficient for sampling across windows in US.

4.5 Conclusions

In this chapter, we have explored how the quality of sampling affects the accuracy of free energy prediction in umbrella sampling (US). To aid our investigation, we developed a range of metrics to assess the sampling quality.

In **Section 4.2**, we developed the *bin metric*, a metric to assess the extent of sampling along the reaction coordinate, but found that it was highly parameter dependent and complex to interpret. We also developed the *normalised overlap* metric, based on the area of two Gaussians, which provides a quantitative tool to analyse the extent of sampling between two windows. One advantage of this overlap metric is its computational efficiency; one only needs to fit two Gaussians to the data and compute two fast numerical integrals. Another advantage is it only depends on the observed reaction coordinate data, rather than on a prediction of the free energy as is required in other methods.[134] We also introduced the discrepancy, a simple but useful measure of how far a particular sampled distribution is from the target reference value.

In **Section 4.3**, we introduced a set of metrics measuring the variance of fitted Gaussians along the trajectory, indicating how converged the system is within each US window. Kullback-Leibler (KL) divergence was then used to calculate how converged the data is within each window to a Gaussian. The KL-divergence provides an indication of whether the distributions are Gaussian, which is important when considering the overlap between windows. As with the overlap calculation, the computational effort to compute the convergence quantities is effectively zero and can be run alongside normal US simulations at next to no extra cost.

In **Section 4.4**, we applied the normalised overlap and KL-divergence metrics to investigate how the overlap affects the free energy profile for a range of test systems. We began our

investigation on simple 1D model systems, for which exact analytic or numerical solutions exist. This allowed us to predict an ideal overlap between windows that produced a sufficiently accurate free energy profile. For both models, this was found to be $\approx 15\%$ overlap. One disadvantage of the toy models is the lack of energy units. Generally, we are aiming for an accuracy in our free energy of 1 kcal mol^{-1}. Turning to simulating real systems, including two MLP systems (S_N2 and isoindene) and a classical MD system (Aβ42 protofibril), we found a similar relation between the overlap and free energy, where an overlap of $\approx 15\%$ produced a free energy within 1 kcal mol^{-1} of the ideal reference.

These results provide a value on the overlap ($\approx 15\%$) that results in a smooth and converged free energy, provided the dynamics within each window have converged. This may help US users in deciding when their US windows have sufficient overlap thus saving computational resource or preventing inaccurate free energy profiles due to poorly overlapping windows. Our overlap value of $\approx 15\%$ is broadly consistent with the previous study who suggested a value of 20% was sufficient.[191]

We hope that the combination of the development of the overlap and convergence metrics with the analysis of how overlap and κ affect the free energy may be useful for those routinely performing US. However, we also note that presenting theoretical models without implementation can be restrictive in utilising the available methods. The aim is that these methods can be easily implemented alongside existing software, such as GROMACS or *mlp-train*, enabling users to automatically generate these analyses whilst they run US. To this end, we have written a Python package, *adaptiveus*, which automates the computation and plotting of all the quantities outlined in this chapter. The next chapter outlines the *adaptiveus* code as well as explores some adaptive US methods.

5

adaptiveus Python Module and Adaptive US

Contents

5.1 Summary 149
5.2 *adaptiveus* Python module 150
5.3 Adaptive US scheme 153
5.3.1 Automatic reference value selection 155
5.3.2 Automatic κ value selection 161
5.4 Conclusions 166

5.1 Summary

Traditionally in umbrella sampling (US), the spring constant, κ, and reference, ζ in the bias, $\omega(r) = \frac{\kappa}{2}(f(r) - \zeta^{\text{ref}})^2$, are chosen manually, with κ often kept constant across US

windows.[11]. Several efforts have been developed towards an adaptive US scheme, whereby the optimal bias, or bias parameters κ and ζ, is automatically determined. Details of these studies are given in **Section 1.4.4**.

In the present work, we combine the ideas of adaptively choosing a suitable reference value based on window overlap, using the overlap metric developed in **Chapter 4**, to provide an energy-independent adaptive reference algorithm. In addition, we provide an alternative method of calculating the optimal κ, based on the discrepancy measure presented and the potential energy of the underlying surface. This is implemented in the *adaptiveus* Python package and tested on a variety of systems.

We start the chapter in **Section 5.2** by outlining the functionality of *adaptiveus*. The required inputs are outlined and example plots that are generated by the package are presented. A closer inspection of the workflow for setting up and running *adaptiveus* is presented. This is then complimented by a overview of the computations that *adaptiveus* does to generate the various plots.

In **Section 5.3**, a mathematical scheme for automatically choosing window reference values based on the predicted overlap is given. This scheme is then tested on two 1D models and two MLP systems. Comparisons to reference values chosen uniformly then highlight the different outcomes of the methods. The same procedure is followed for the automatic κ selection method. Finally, potential future ideas for adaptive US are highlighted.

We finish the chapter with a discussion of the *adaptiveus* code and the future of adaptive US in **Section 5.4**. Details of the computational methods used in this chapter can be found in **Appendix B.4**.

5.2 *adaptiveus* Python module

In **Chapter 4**, we presented metrics for quantifying overlap between windows and measuring convergence in terms of KL-divergence. However, these calculations would take time to implement manually for users and require experience of mathematical programming. We

believe that automation of methodology is as important as the proposal of the methods if researchers are to use it in their simulations. To this end, we developed the *adaptiveus* Python module. Broadly, the aspects that this packages address are as follows:

1. Setting up and running US using GROMACS and our machine learnt potential package, *mlp-train*.[2]
2. Testing the convergence of US window(s), with plotting, using the KL-divergence measure.
3. Fitting Gaussians to each window and calculating the overlap and discrepancy of the windows.
4. Attempting to adaptively choose the window reference and κ for optimal sampling.

The architecture of *adaptiveus* is split into two main methods, `run_non_adaptive_sampling()` and `run_adaptive_sampling()`. Users define the relevant inputs for US (type of dynamics, bias, temperature etc.) as well as providing the input files (e.g., a structure file covering points along the reaction coordinate). Then the number of windows and length of each window is specified and dynamics are initiated. All quantities and graphs are calculated and generated automatically. An overview of the architecture is shown in **Fig. 5.1**.

The figures relating to US in **Chapter 4** and this chapter have been generated by *adaptiveus*, and the architecture from relevant sections of the package are explained herein. First, dynamics are setup and run to obtain the biased distribution, **Fig. 5.2**. Reference values are chosen automatically using evenly spaced points between the chosen or natural end points of the reference trajectory. Then a suitable frame is chosen from the reference trajectory. The precise method of doing this depends on whether *mlptrain* or *gmx* is being used, but in both cases, the frame with the reaction coordinate, ζ, close to the desired reference value is chosen. A single biased window simulation is run (with an optional equilibrium run for *gmx*) and the convergence properties of this window are calculated. The observed ζ values are collected and a new window with the next reference value is run. This repeats until all

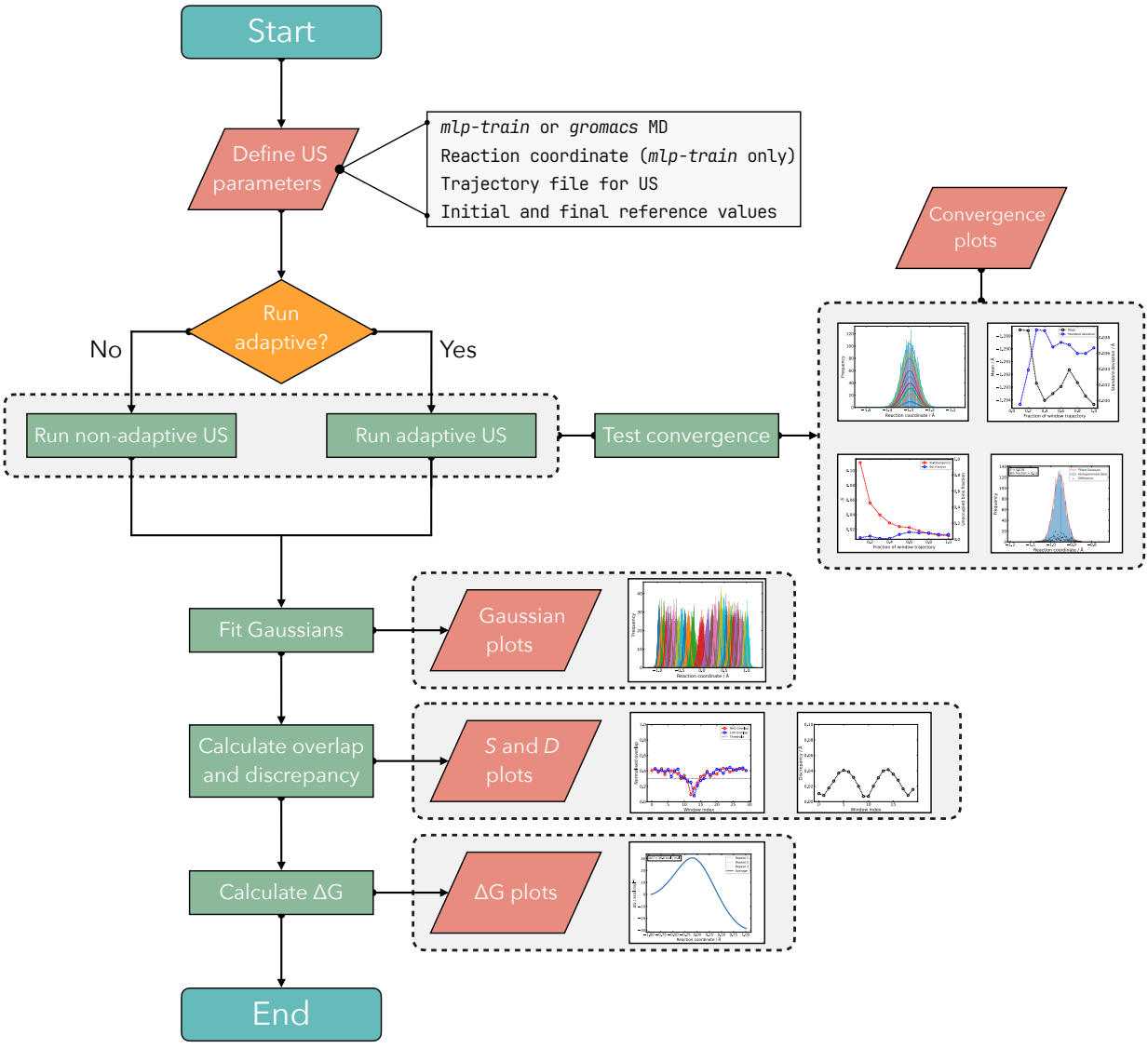


Figure 5.1: Diagrammatic workflow of the *adaptiveus* Python package for running US. Example output plots are shown for convergence, Gaussians, overlap (S), discrepancy (D) and free energy (ΔG).

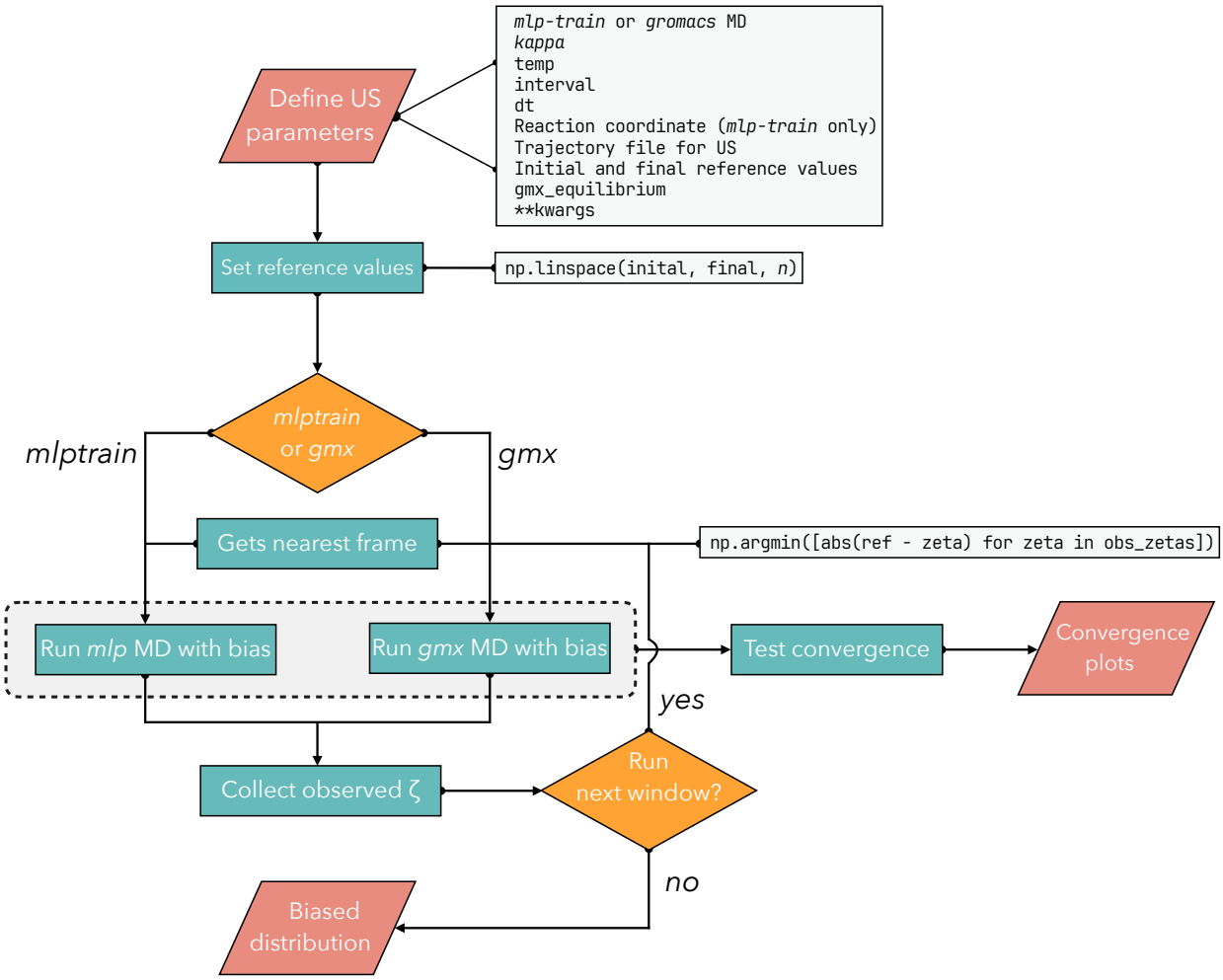


Figure 5.2: Diagrammatic workflow of the setup and running of US using *adaptiveus*. Code snippets of specific actions are shown, as well as the required inputs for the programme. The keyword arguments relate to the *gmx* files required to run an MD simulation.

window simulations are finished, resulting in an overall biased distribution.

In the next section of the code, the data is processed to generate the plots shown in **Fig. 5.1**, including the free energy, **Fig. 5.3**.

5.3 Adaptive US scheme

The final aspect of the code is the adaptive sampling scheme. Here, the reference and κ values can be automatically chosen to optimise overlap between windows.

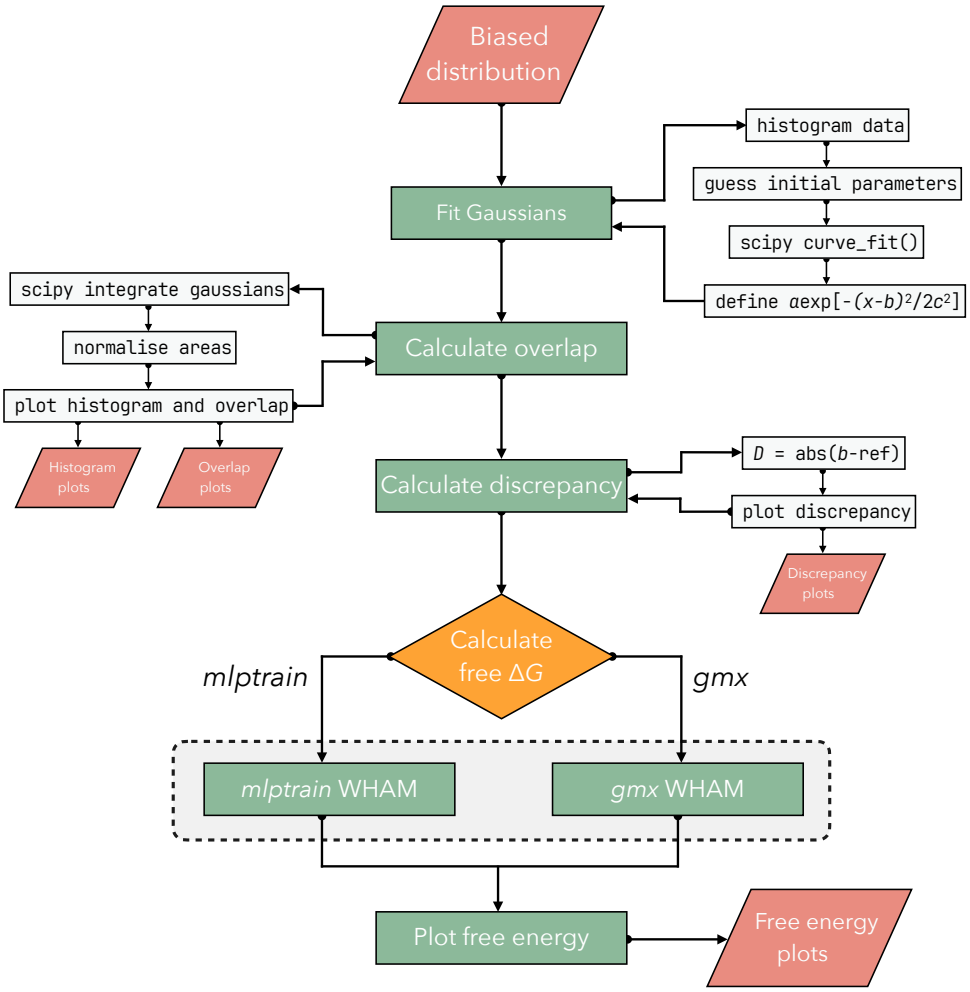


Figure 5.3: Diagrammatic workflow of the calculation and plotting of various quantities using *adaptiveus*.

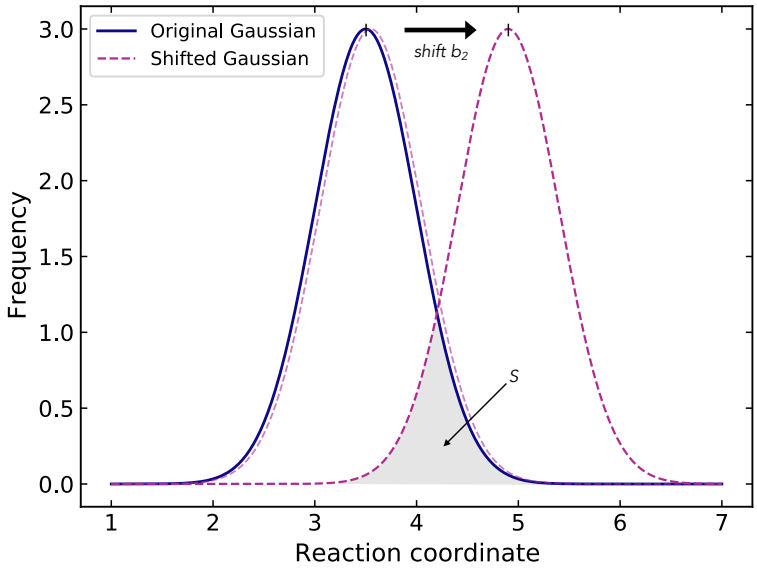


Figure 5.4: Fitted Gaussian from observed data with an identical Gaussian shifted to the right, thus altering the overlap, S.

5.3.1 Automatic reference value selection

Ordinarily, the reference values, ζ, are chosen manually by the user or placed uniformly along the reaction coordinate. For flat potential energy surfaces (PESs), perfectly Gaussian distributions with known standard deviations and mean can be calculated. Appropriate reference values can then be spaced apart evenly to give the desired overlap. However in real systems, the underlying surface is finite and variable. This results in unpredictable Gaussian distributions, making the overlap unknowable *a priori*. Therefore, choosing a reference value based on the previous window Gaussian to aim for specific overlap would alleviate the need for human intervention to optimise the overlap. We do this by first fitting a Gaussian to the window we have just run and then assuming that a window run close to the previous window will have a similar form.

The dashed Gaussian in **Fig. 5.4** that is almost identical to the original Gaussian will lead to an overlap of almost unity. As we shift the dashed Gaussian further to the right, the overlap will decrease at a rate according to **Eqn. 5.2**. We can determine the overlap numerically from the sum of the two Gaussians, **Eqn. 5.1**.

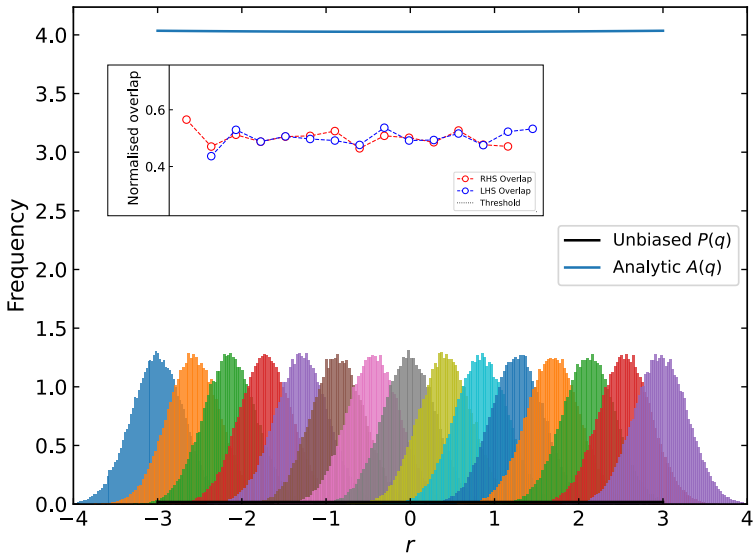


Figure 5.5: US of an effective free particle (ϵx^2, $\epsilon = 0.001$) with windows centred based on target overlap of 0.5. Normalised overlap for each side of the Guassians is shown in the box. Each US simulation has 20 windows, each with 100,000 steps. Numerically exact and WHAM free energies (arbitrary units) and probability distributions shown, along with the underlying histogrammed data.

$$S = \frac{1}{\sqrt{2\pi}ac}\left(\int_{-\infty}^{A} a\mathrm{e}^{-(x-b_2)^2/2c^2}\mathrm{d}x + \int_{A}^{\infty} a\mathrm{e}^{-(x-b_1)^2/2c^2}\mathrm{d}x\right), \qquad A = \frac{b_2^2 - b_1^2}{2(b_2 - b_1)} \tag{5.1}$$

Above, we have assumed $a_1 = a_2$ and $c_1 = c_2$, leading to only one point of intersection and hence the overlap is the sum of just two integrals. The point where these Gaussians intersect is found by equating the two Gaussians. We can solve this expression numerically by finding the roots of the error form of the integrals, **Eqn. 5.2**.

$$S = 1 + \frac{1}{2}\,\mathrm{erf}\left(\frac{b_1 - A}{c\sqrt{2}}\right) - \frac{1}{2}\,\mathrm{erf}\left(\frac{b_2 - A}{c\sqrt{2}}\right), \qquad b_2 > b_1 \tag{5.2}$$

After we choose some target value of S, we can vary b_2 such that the target value of S is obtained. This value is then used as the reference value for the next window. We first test this on a flattened harmonic potential to simulate a free particle, i.e., ϵx^2, where ϵ is vanishingly small, **Fig. 5.5**.

Under a uniform free energy, windows can be successfully placed at intervals giving overlap

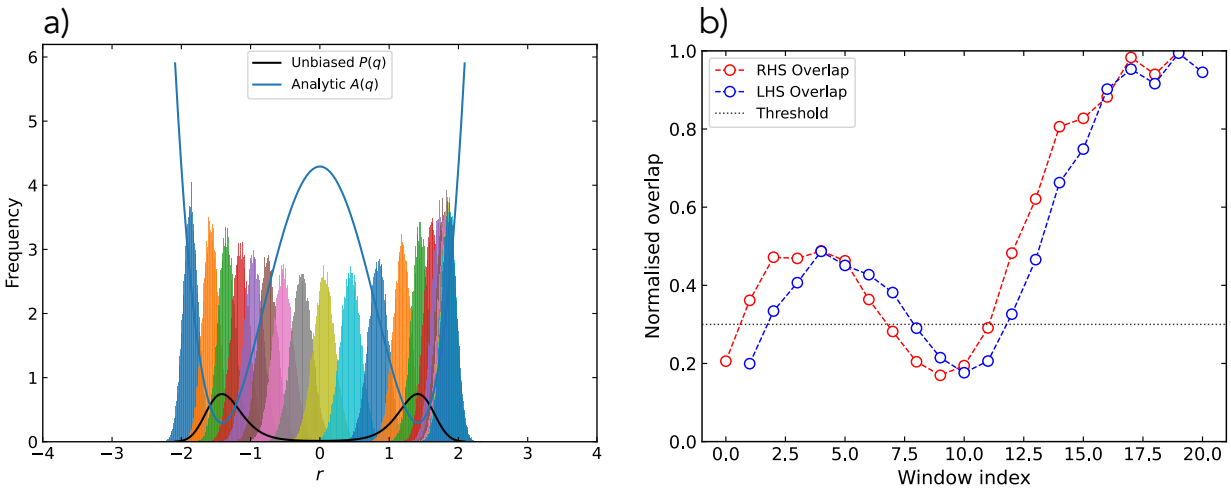


Figure 5.6: a) US over the $r^4 - 4r^2 + 4$ 1D potential using $\kappa = 50$ and 100,000 steps per window and an adaptive reference method for placing windows. The overlap target was set to 0.3. b) Resulting normalised overlaps from the adaptive reference method with the horizontal dashed line showing the target overlap of 0.3.

values close to the target, 0.5. This method therefore provides a way to generate optimal overlaps, provided the free energy surface does not significantly affect the window placement. To extend this method to a model that more appropriately represents a real system, i.e., one with a steep barrier, we will test this method on the quartic potential, $r^4 - 4r^2 + 4$, **Fig. 5.6**.

When we move to a more typical PES, i.e., a steep well with a barrier, the automatic reference method performs worse than a uniformly chosen set of window references. However, it is worth noting that with a uniform selection we cannot know *a priori* what the overlap will be and have to rely on intuition or previous US simulation data. The automatic reference will attempt to fit to a specific overlap but needs more information to accurately assign a reference value for the sequential window.

The reason for the poor performance of this method lies not in the assumption that the next window will have the same Gaussian parameters as the current, but in the discrepancy between the target reference and the mean of the window data. Consider the overlap values in **Fig. 5.6b** and the corresponding potential in **Fig. 5.6a**. When calculating the next reference value, the total energy is assumed to be constant as we shift the second Gaussian along. On the left hand side of the quartic potential, the gradient is steeply negative. A

reference will be calculated to give a particular overlap but the system will be pulled more strongly towards the minima. This leads to an increase in the overlap until we reach the barrier. Here, the windows are more spread out than expected, with windows falling to either side of the barrier. As the potential rises steeply to infinity, the windows bunch up and eventually become stuck. A new reference will be chosen but the system will fall back into the previous position, thus beginning an infinite recursive loop.

We can test this automatic reference selection method on our two 'real' systems, the S_N2 reaction (detailed in **Section 4.4.2.1**) and the isoindene methyl rearrangement (detailed in **Section 3.3**). To compare the adaptive scheme, we will compare the automatic reference selection with a simple uniform distribution along the reaction coordinate. First, we look at the S_N2 reaction, **Fig. 5.7**.

The non-adaptive US with $\kappa = 20$ eV Å^{-2} results in a uniform overlap across the reaction coordinate. The resulting free energy is smooth since there is sufficient overlap between each window. Note we have previously refined the choice of the number of windows to optimise the overlap, so this represents the ideal situation. When we invoke our adaptive reference method, we see the overlap is not uniform across the distribution. As the potential energy gradient changes, the prediction of the overlap will become further from the actual overlap for subsequent reference values. This results in an increase and then decrease of overlap around the TS. However, we do obtain an identical $\Delta G^{\ddagger}$ due to the still sufficient overlap between each window. One advantage in this case is that we only need 23 windows, saving sampling time. Next, moving onto the isoindene system, we can run the same comparison, **Fig. 5.8**.

The steeper energy gradient along the reaction coordinate results in a non-uniform overlap around the TS due to the increased discrepancy. However, the free energy profile is smooth as overlap is sufficient. When the adaptive scheme is used, we see many of the windows bunch up as the energy increases. While this was seen in the S_N2 reaction, due to the steeper gradient, this results in inefficient sampling on the reactant side of the reaction. As the energy drops after the TS, the windows become significantly spread out to the point

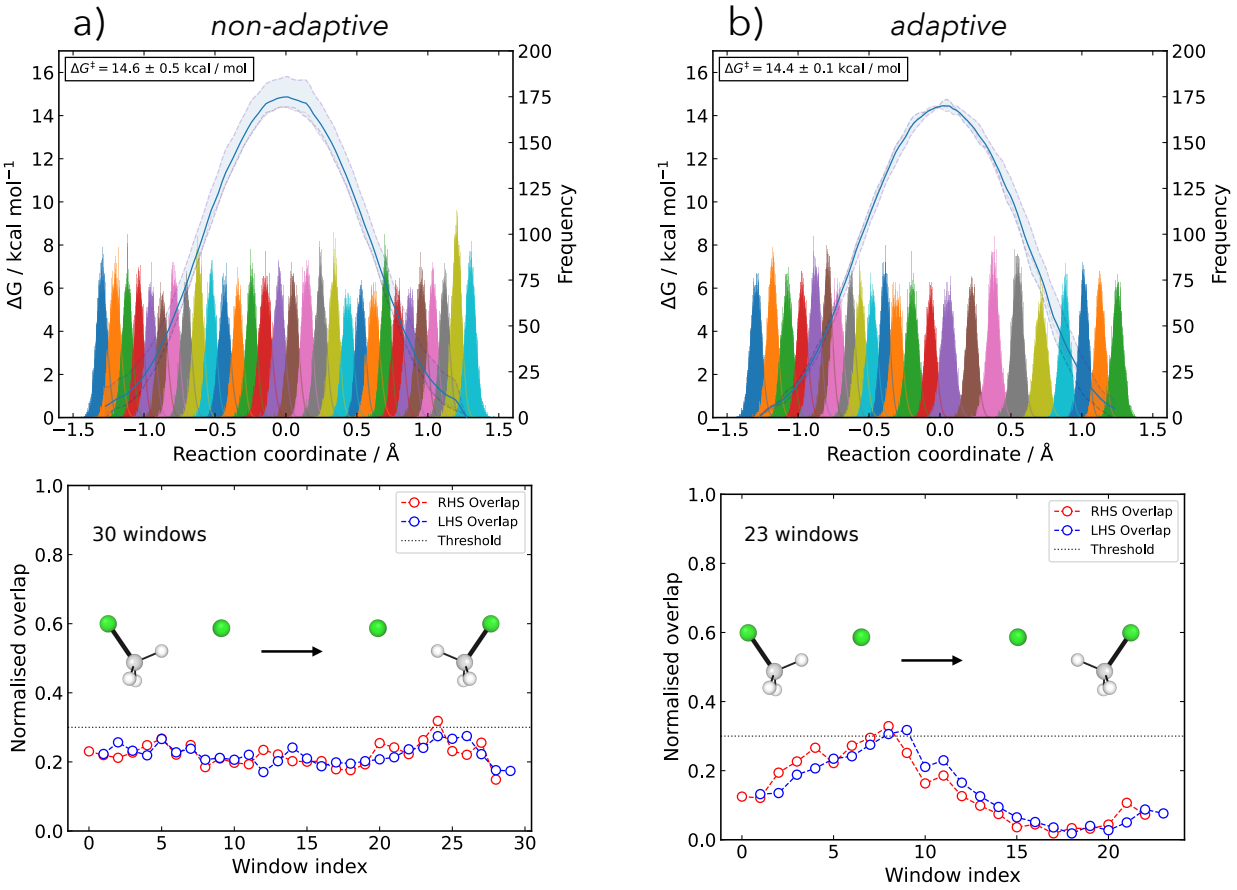


Figure 5.7: US of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ ($f(r) = r_{C–Cl} - r_{C–Cl'}$, $\kappa = 20$ eV Å^{-2}, $T = 300$ K, $n_{windows} = 30$ or adaptive, $t = 30$ ps) using a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP. The free energy over three repeats is shown over the window histograms, along with the overlaps. a) Non-adaptive US using $n_{windows} = 30$. b) Adaptive US using references chosen automatically resulting in 23 windows in this case.

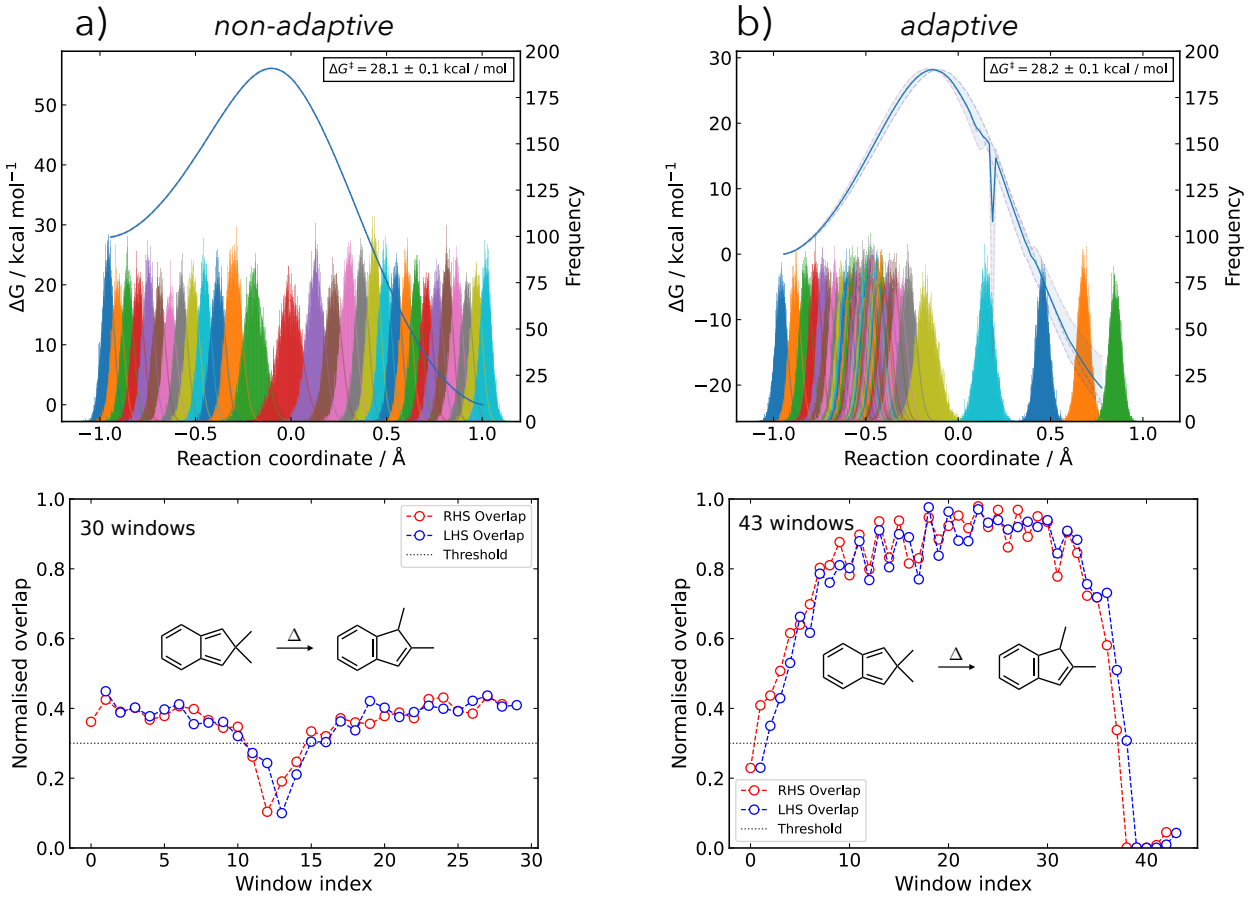


Figure 5.8: US of isoindene methyl migration ($f(r) = r_{C_1-C_3} - r_{C_2-C_3}$, $\kappa = 20$ eV Å^{-2}, $T = 300$ K, $n_{\text{windows}} = 30$ or adaptive, $t = 10$ ps) using an ACE potential trained at PBE0/def2-SVP. The free energy over three repeats is shown over the window histograms, along with the overlaps. a) Non-adaptive US using $n_{\text{windows}} = 30$. b) Adaptive US using references chosen automatically resulting in 43 windows in this case.

where overlap is too small to provide a smooth free energy profile. Moreover, due to the inefficient window placing, 43 windows are required to sample the reaction coordinate.

These results show that for reactions where the energy changes are gradual, the automatic reference selection can improve the efficiency of US. However, due to the assumption that the PES is flat for small increments along the reaction coordinate, systems with steeper energy gradients result in windows becoming bunched up.

5.3.2 Automatic κ value selection

The choice of the spring constant, κ, is often selected manually, and kept constant for all windows during a US simulation.[11] However, regions near the energy minimum need a smaller κ than those on a steep slope in the energy, or near a maximum. Previous studies have estimated the optimal κ by running short US simulations and then selecting κ based on the resulting free energy profile.[134] Selecting appropriate values of κ *a priori* based on the underlying energy surface without sampling could bypass prior simulations needing to be run while still leading to more efficient and effective sampling of the reaction coordinate. The total energy along the reaction coordinate when a bias is introduced is

$$E(r,\zeta) = U(r) + \omega(r,\zeta) \tag{5.3}$$

$$= U(r) + \frac{\kappa}{2}(f(r) - \zeta)^2. \tag{5.4}$$

We seek a κ which gives us a Gaussian sampled distribution near our target value ζ. We test a potential solution to this problem based on an initial guess of the underlying energy surface. We start by assuming a sufficiently Gaussian distribution will be obtained when the total energy only has one minimum, i.e., it resembles a quadratic function, **Eqn. 5.5**.

$$\frac{\partial E(r,\zeta)}{\partial f(r)} = 0 \tag{5.5}$$

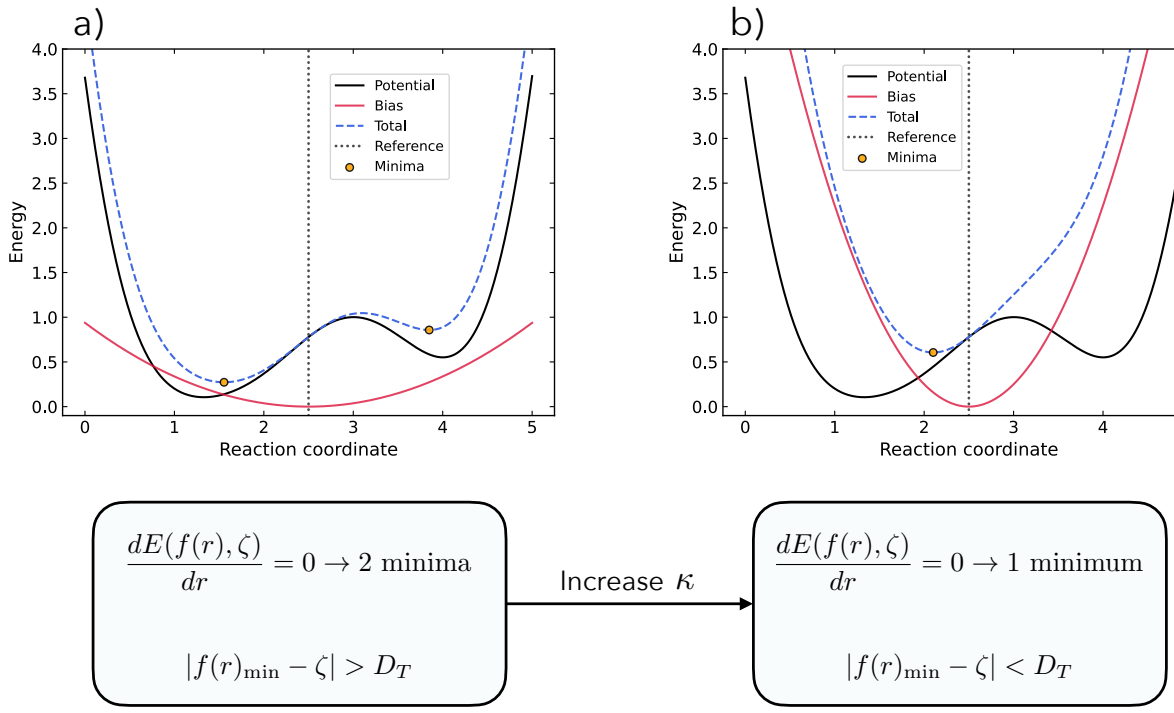


Figure 5.9: Adaptive κ selection strategy for US. a) A weak bias results in the total energy approximately equal to the potential energy. The number of minima may be greater than unity and the discrepancy greater than the discrepancy threshold, D_T. b) κ is increase until only one minimum remains and the discrepancy is below D_T.

This will ensure that the system will most likely be observed around the minimum corresponding to our quadratic function. Then κ is increased until only one minimum is found, and increased further such that $|r_{\min} - \zeta| < D_T$ (the minimum we find from our total energy function is sufficiently close to our reference, within a user-defined threshold, D_T). This method is illustrated in **Fig. 5.9**.

From **Fig. 5.9**, we can see that we increase κ from an unsuitably low value (zero) until the *singular* minimum in the *total* PES is within our discrepancy threshold, D_T. Note, this methodology assumes the PES qualitatively matches the free energy surface, which may not be suitable for more complex systems. We can now test this method on the two MLP systems previously studied, first applying it to the S_N2 reaction, **Fig. 5.10**.

In shallower regions of the free energy surface, κ is given a small value, resulting in a wide distribution without compromising on discrepancy. As the free energy increases steeply, κ is increased, constraining the dynamics near the reference value. Around the TS, κ drops

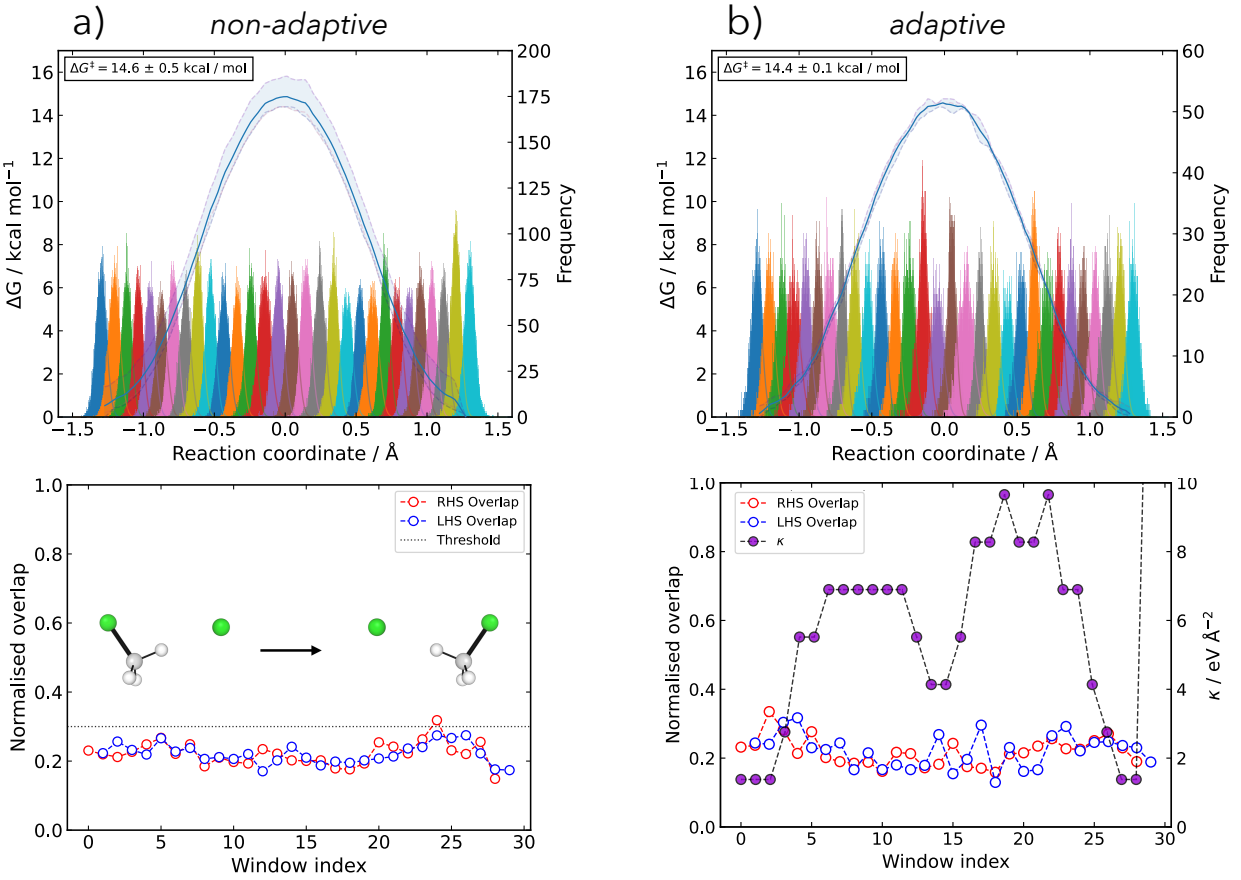


Figure 5.10: US of $Cl^- + CH_3Cl \longrightarrow ClCH_3 + Cl^-$ ($f(r) = r_{C\text{–}Cl} - r_{C\text{–}Cl'}$, adaptive κ, $T = 300$ K, $n_{\text{windows}} = 30$, $t = 10$ ps) using a GAP trained at DLPNO-CCSD(T)/ma-def2-TZVPP. The free energy over three repeats is shown over the window histograms, along with the overlaps. a) Non-adaptive US using $\kappa = 20$ eV Å^{-2}. b) Adaptive US using κ values chosen automatically.

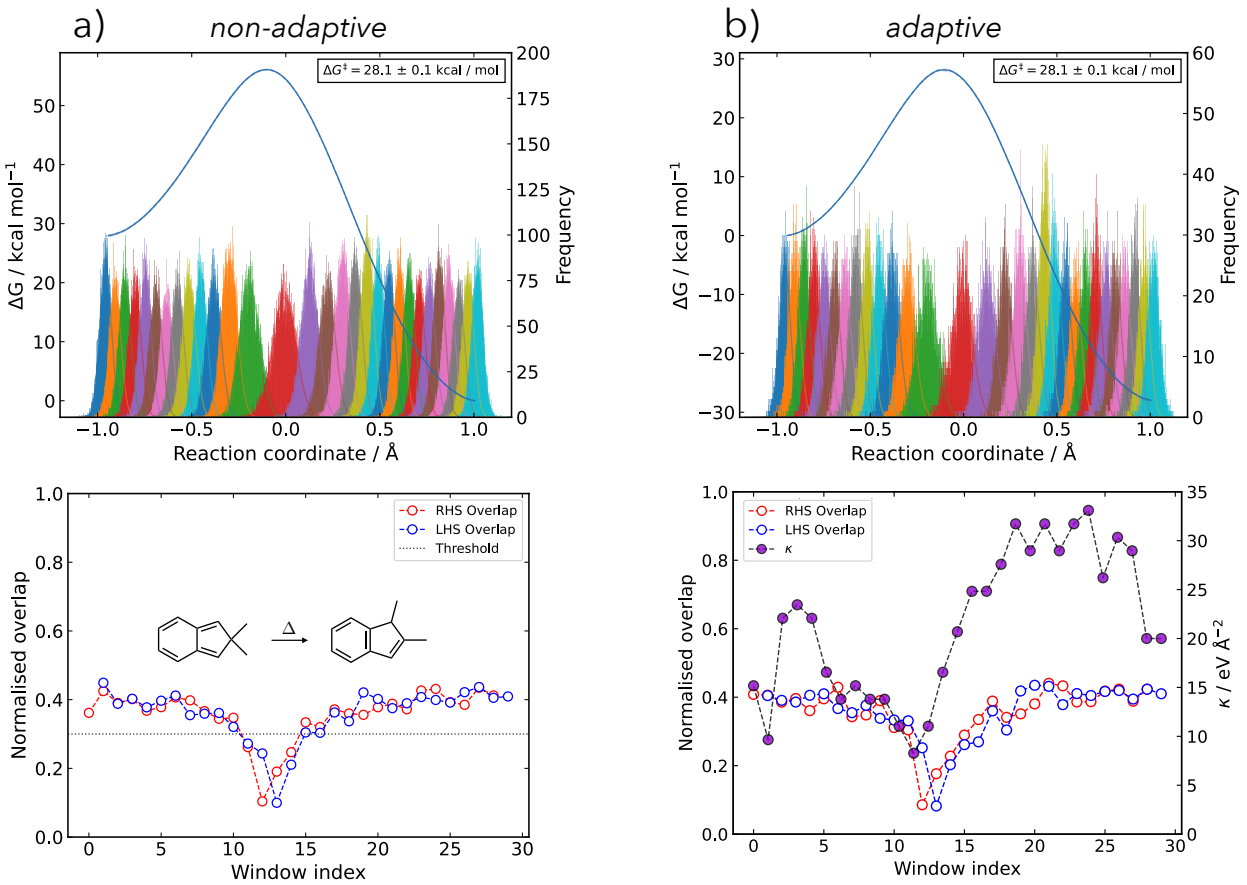


Figure 5.11: US of isoindene methyl migration ($f(r) = r_{C_1–C_3} - r_{C_2–C_3}$, adaptive κ, $T = 300$ K, $n_{\text{windows}} = 30$, $t = 10$ ps) using an ACE potential trained at PBE0/def2-SVP. The free energy over three repeats is shown over the window histograms, along with the overlaps. a) Non-adaptive US using $\kappa = 20$ eV Å^{-2}. b) Adaptive US using κ values chosen automatically.

slightly, reflecting the flattening of the free energy surface, before rising again as the energy drops. Overall, this results in a fairly uniform overlap, minimising κ without having to choose a value *a priori*. Next, we apply this algorithm to the isoindene system, **Fig. 5.11**.

Interestingly, we see a similar overlap trend for both the non-adaptive and the adaptive κ selection. This suggests that the approximation that the free energy surface ≈ PES is less valid in this case. However, we still achieve the same free energy profile using this method without the need for choosing a value of κ before the simulation.

⁂

Another method of adaptive κ selection could be based on the KL-divergence, where we increase κ such that the KL-divergence difference between our total energy and the Gaussian arising from our harmonic bias is equal to some target value. For a continuous set of function, the KL-divergence is defined as

$$S = \int_{-\infty}^{\infty} P(x) \ln \frac{P(x)}{Q(x)} \mathrm{d}x. \tag{5.6}$$

This aims to generate a biased distribution that is close to the Gaussian arising from a purely harmonic potential. Using our expression for total energy from **Eqn. 5.4**, our KL-divergence becomes a combination of the total energy distribution, $P(x)$, and the bias energy distribution, $Q(x)$. We must also weight our distribution by the correct normalisation constant, A, which we will leave as a constant for now.

$$S = A \int_{-\infty}^{\infty} \mathrm{e}^{-\beta\left(U(r)+\frac{\kappa}{2}(f(r)-\zeta)^2\right)} \ln \frac{\mathrm{e}^{-\beta\left(U(r)+\frac{\kappa}{2}(f(r)-\zeta)^2\right)}}{\mathrm{e}^{-\beta\left(\frac{\kappa}{2}(f(r)-\zeta)^2\right)}} \mathrm{d}r \tag{5.7}$$

Simplifying this expression, we find that the KL-divergence depends on the unbiased energy, $U(r)$, weighted by the total energy (biased and unbiased). As κ tends to infinity, the KL-divergence tends to zero.

$$S = -A\beta \int_{-\infty}^{\infty} U(r) \mathrm{e}^{-\beta\left(U(r)+\frac{\kappa}{2}(f(r)-\zeta)^2\right)} \mathrm{d}r \tag{5.8}$$

This method has not been implemented *adaptiveus* and could be investigated in future work.

5.4 Conclusions

In this chapter, we presented our automated US package, *adaptiveus*. In this package, we built functionality to setup and run US simulations interfaced to GROMACS and *mlp-train*. US simulations can be run with a few lines of Python after providing a reaction coordinate trajectory file. The free energy is computed automatically using WHAM and plotted. The package also computes and plots a range of convergence metrics, developed in **Chapter 4**. The convergence analysis helps users decide on the time length required for each window, as well as κ (if the distribution is not Gaussian enough). The overlap and discrepancy, detailed in **Chapter 4**, of US windows are calculated and plotted, allowing users to quantitatively assess the overlap between windows. This helps with choosing suitable values for κ and ζ. Finally, there are options to adaptively choose the ζ reference values and spring constants κ. The former is based on the overlap measure and the latter on the position of the total energy minimum.

As with any automation software, the installation, ease-of-use, efficiency speed up and reliability of the tool is critical to its success. *adaptiveus* can be installed from GitHub, requiring the same packages that are required when running standard US computations. While it does require some scripting, we have tried to minimise the Python knowledge required. To this end, the adaptive and non-adaptive US can be run in two lines, which includes the convergence testing, plotting and calculation of free energy.

Given a completely unknown system, choosing suitable κ and ζ can be arduous, requiring many trial and error attempts. Normally, these parameters are chosen manually, and previous efforts rely on prior sampling.[134, 191] We have shown that for relatively simple systems, such as an S_N2 reaction, the adaptive scheme can reproduce converged and well sampled dynamics more efficiently than manual selections, especially when the trial and error attempts are taken into consideration. However, for more complex free energy surfaces, namely those with steeper energy gradients, the adaptive scheme can take longer and potentially create gaps in the free energy profile and so care is needed when applying this to challenging PESs. Nonetheless, we believe *adaptiveus* is a useful tool to check whether the US windows overlap

sufficiently and if they converged to the desired Gaussian distribution. Moreover, while there have been many adaptive US schemes developed, they are not routinely used and this work could provide further motivation and basis for development and implementation of adaptive schemes.

6

Conclusions

6.1 Thesis overview

Properties of chemical reactions, such as mechanisms, feasibility and rate, can be predicted using dynamic simulations. Non-dynamic methods, such as DFT, often fail to account for anharmonic effects and post transition state bifurcations. A range of dynamic methods are available to sample reactivity, including classical molecular mechanics (MM) force field methods and quantum *ab initio* molecular dynamics (AIMD) methods. While MM methods are fast to evaluate, enabling large systems to be simulated for long time periods, they are typically nonreactive and inaccurate outside of the regions they have been parameterised on. On the other hand, AIMD methods account for electrons explicitly and can model bond

breaking and making events. However, solving the Schrödinger equation at each step in an AIMD simulation is expensive, and as a result, AIMD is limited by system size and time scale. Machine learning potentials (MLPs), first developed in 1995, are growing in popularity as an alternative approach to simulating accurate dynamics. MLPs enable quantum mechanical energies and forces to be quickly computed to propagate dynamics. They can also be combined with enhanced sampling methods, such as umbrella sampling (US), to generate free energy profiles, providing insight into mechanisms and reaction rates.

However, even though MLPs have existed for over 20 years, they have not replaced MM force fields or AIMD. MLPs typically require specialist knowledge to curate datasets and validate dynamics, thus limiting the user base that can access them. MLPs also typically require large datasets for training, restricting their use to simple systems or requiring extensive computational resources. Finally, their suitability in modelling reactive molecular systems is still being explored and needs to be further demonstrated to the wider scientific community to enable more widespread usage.

In the first part of this thesis, we set out to address these limitations of MLPs through bespoke data curation and validation, automation of training and dynamics and application to a set of (bio)chemically relevant reactive terpene systems. In **Chapter 2**, we presented methods of generating training data and assessing the accuracy of the PhysNet and GAP MLPs. We found that training using data obtained from random displacements, MM or AIMD did not lead to accurate or efficient potentials. Instead, an active learning (AL) approach that generated configurations on-the-fly from dynamics was found to greatly improved the quality of the potentials while requiring fewer training configurations. Moreover, validating the MLPs to test data spliced from the training data could produce misleadingly accurate potentials. Validating to the region of interest, i.e., dynamics, resulted in potentials that were more stable when used to conduct MLP-MD. We presented two open source Python packages, *gap-train* and *mlp-train*, which automate the AL strategy for training and enable US calculations to be performed in a few lines of code, without expert knowledge. During this analysis, we found the ACE potential to be an improvement over PhysNet and GAP for

molecular systems, and was taken forward to model reactive systems.

Chapter 3 extended the AL training strategy and automation to a range of (bio)chemically relevant reactions, utilising the accuracy and efficiency afforded by MLPs to address questions related to their mechanisms. Using an indane model system, we found that the accuracy and efficiency of a potential trained along a reaction coordinate using AL was sensitive to the quality of the TS guess. We presented an automated method for reaction coordinate training, whereby the error could be systemically improved. ACE was combined with US to accurately predict the activation free energy of a methyl migration of an isoindene, supporting the proposed concerted reaction mechanism. The predictions from ACE dynamics showed a closer fit to experimental data than the corresponding static DFT predictions. However, both ACE and DFT failed to reproduce the activation enthalpy and entropy of this reaction. An ACE potential was trained on a modified isoindene reaction, involving a hydrogen migration. Results from path integral molecular dynamics (PIMD) suggested that nuclear quantum effects (NQEs) did not play a significant role, indicating that entropic contributions were minimal.

Next, ACE was used to generate a free energy profile for a proposed 1,2-hydride shift in a terpinene reaction, supporting the proposed mechanism from experiment. However, training an ACE on an alternative pathway resulted in an unstable potential and could not be used to compare the free energy pathways. Using PIMD, we observed an increase in activation free energy when NQEs were considered, contrary to expectation.

The hydrogen and methyl migration reactions of an indane were simulated using an ACE trained from two TSs. Uphill dynamics of this reaction provided estimations of product ratios, which were in alignment with previous computational work. Using the rapid evaluation of dynamics through MLPs, we extended the number of uphill dynamic trajectories and found that previous product ratios were not fully converged, highlighting the importance of convergence testing. Combining PIMD with uphill dynamics enabled us to investigate the effect of NQEs on the hydrogen versus methyl migration, and we found a greater number of hydrogen migrations in the product ratios when using quantum dynamics.

Finally, we investigated a sesquiterpene system with a proposed post transition state bifurcation (PTSB). While previous work had used IRC calculations to propose this PTSB, they were unable to confirm its existence due to the limitations of static DFT methods. However, using MLPs, we were able to run direct dynamics from the TS and we observed the existence of a PTSB for three levels of theory, obtaining product ratios for each.

In the latter part of the thesis, we presented metrics to address limitations in the quantification of sampling within US, along with developing software that automates US and adaptive US.

Chapter 4 presented the *bin metric* which quantifies the extent of sampling along a reaction coordinate and the *normalised overlap* metric which enables quantification of the overlap between two windows in US. The discrepancy provides a simple measure of how far the target window distribution is from the sampled distribution, which can be related to the strength of the spring constant and the underlying free energy surface. A set of metrics measuring the variance of fitted Gaussians with trajectory time allow the convergence of data within each window to be assessed. The KL-divergence metric indicates how converged sampling is within each window relative to an ideal Gaussian. All of these metrics provide insight into convergence within US, helping users check whether the reaction coordinate has been sufficiently sampled.

These metrics were then used to investigate the influence of overlap on free energy for a) 1D analytic systems, b) S_N2 and isoindene systems, and c) an Aβ42 protofibril. We found that across all of these systems, an overlap of $\approx 15\%$ was sufficient to converge free energy results to their true value.

In **Chapter 5**, we presented an open source Python package, *adaptiveus*, which automates running US (using both *mlp-train* and GROMACS), and convergence, overlap and free energy calculations. Finally, the choice of the spring constant and reference values in the US bias can affect the accuracy and efficiency of an US simulation. Building on previous studies, we developed an automated adaptive scheme for selecting the spring constant and

reference values based on the PES and normalised overlap metric. While we found improvements in efficiency for simple potential energy surfaces, limitations arose in the scheme when considering complex or steep PESs.

6.2 Future outlook

MLPs represent a growing area of development in simulating dynamics and we hope that this thesis provides contributions to how these tools are further developed and used. Work is currently underway to implement the AL training strategy and other features of *mlp-train* into the MD package OpenMM.[139] This will be the first instance of MLPs being available in a large open source MD package with full support. Our hope is that this will enable the wider MD community to train and use potentials for a range of systems, without requiring extensive knowledge of programming or MLPs.

Additional features are being implemented into *mlp-train* including metadynamics, AL driven by metadynamics (thus not requiring a transition state for training a reaction), MACE (the message passing neural network succeeding ACE), and solvent effects for condensed phase reactions. These features may help in extending the modelling of the biochemically relevant reactions presented in this thesis to environments closer to those in nature.

As the ease and functionality of MLP packages improve, we expect to see MLPs being used for dynamics more routinely due to their speed and accuracy. MLPs are still in the relatively early stages of development, however improvements in architecture and software are being rapidly made. This work provides demonstration of their potential and automation of training and dynamics. As these areas improve, we believe that the complexity and accuracy of the systems they are applied to will only increase.

Meanwhile, US continues to be a widely used method for generating free energy profiles, with few parameters required. As the accuracy of simulations improve, more attention must be paid to errors arising from sampling. This includes regular assessment of the sampling convergence within each window, as well as the overlap between windows. The work in

this thesis provides methods for assessing this error and can be taken forward into other programmes.

Finally, adaptive US has been explored multiple times before but is not routinely used. This is in part due to the difficulty of reliably choosing suitable parameters as well as implementation limitations, which are partially addressed in this work. If these methods become more reliable, they may be implemented in future packages, along with convergence and overlap testing, to support US simulations.

Bibliography

(1) Young, T. A.; Johnston-Wood, T.; Deringer, V. L.; Duarte, F. *Chemical Science* **2021**, *12*, 10944–10955.

(2) Young, T. A.; Johnston-Wood, T.; Zhang, H.; Duarte, F. *Physical Chemistry Chemical Physics* **2022**, *24*, 20820–20827.

(3) Shaw, R. A.; Johnston-Wood, T.; Ambrose, B.; Craggs, T. D.; Hill, J. G. *Journal of Chemical Theory and Computation* **2020**, *16*, 7817–7824.

(4) Chan, H. H.; Moesser, M. A.; Walters, R. K.; Malla, T. R.; Twidale, R. M.; John, T.; Deeks, H. M.; Johnston-Wood, T.; Mikhailov, V.; Sessions, R. B., et al. *Chemical Science* **2021**, *12*, 13686–13703.

(5) Pople, J. A. *Reviews of Modern Physics* **1999**, *71*, 1267.

(6) Neese, F. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2022**, *12*, e1606.

(7) Frisch, A et al., *Gaussian 09W Reference*, 2009; Vol. 470.

(8) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. *SoftwareX* **2015**, *1*, 19–25.

(9) Case, D. A.; Aktulga, H. M.; Belfon, K.; Ben-Shalom, I.; Brozell, S. R.; Cerutti, D. S.; Cheatham III, T. E.; Cruzeiro, V. W. D.; Darden, T. A.; Duke, R. E., et al., *Amber 2021*; University of California, San Francisco: 2021.

(10) Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J. *The Journal of Chemical Physics* **1995**, *103*, 4129–4137.

(11) Kästner, J. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 932–942.

(12) Rudolf, K.; Spellmeyer, D. C.; Houk, K. *The Journal of Organic Chemistry* **1987**, *52*, 3708–3710.

(13) Murakami, M.; Miyamoto, Y.; Ito, Y. *Angewandte Chemie International Edition* **2001**, *40*, 189–190.

(14) Murakami, M.; Hasegawa, M.; Igawa, H. *The Journal of Organic Chemistry* **2004**, *69*, 587–590.

(15) Bahmanyar, S; Houk, K. *Journal of the American Chemical Society* **2001**, *123*, 12911–12912.

(16) Mitsumori, S.; Zhang, H.; Ha-Yeon Cheong, P.; Houk, K.; Tanaka, F.; Barbas, C. F. *Journal of the American Chemical Society* **2006**, *128*, 1040–1041.

(17) Wang, Y.; Wang, J.; Su, J.; Huang, F.; Jiao, L.; Liang, Y.; Yang, D.; Zhang, S.; Wender, P. A.; Yu, Z.-X. *Journal of the American Chemical Society* **2007**, *129*, 10060–10061.

(18) Piel, I.; Steinmetz, M.; Hirano, K.; Froehlich, R.; Grimme, S.; Glorius, F. *Angewandte Chemie International Edition* **2011**, *50*, 4983–4987.

(19) Xu, H.; Zuend, S. J.; Woll, M. G.; Tao, Y.; Jacobsen, E. N. *Science* **2010**, *327*, 986–990.

(20) Fleming, E. M.; Quigley, C.; Rozas, I.; Connon, S. J. *The Journal of Organic Chemistry* **2008**, *73*, 948–956.

(21) Chung, L. W.; Wu, Y.-D.; Trost, B. M.; Ball, Z. T. *Journal of the American Chemical Society* **2003**, *125*, 11578–11582.

(22) Rappé, A. K.; Goddard III, W. A. *Nature* **1980**, *285*, 311–312.

(23) Bahmanyar, S; Houk, K. *Journal of the American Chemical Society* **2001**, *123*, 11273–11283.

(24) DelMonte, A. J.; Haller, J.; Houk, K.; Sharpless, K. B.; Singleton, D. A.; Strassner, T.; Thomas, A. A. *Journal of the American Chemical Society* **1997**, *119*, 9907–9908.

(25) Singleton, D. A.; Hang, C.; Szymanski, M. J.; Meyer, M. P.; Leach, A. G.; Kuwata, K. T.; Chen, J. S.; Greer, A.; Foote, C. S.; Houk, K. *Journal of the American Chemical Society* **2003**, *125*, 1319–1328.

(26) Heitler, W.; London, F. *Zeitschrift für Physik* **1927**, *44*, 455–472.

(27) Pauling, L. *The Nature of the Chemical Bond* **1939**.

(28) Coulson, C. A. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **1939**, *169*, 413–428.

(29) Coulson, C. A. *Quarterly Reviews, Chemical Society* **1947**, *1*, 144–178.

(30) Coulson, C. A. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1950; Vol. 46, pp 202–205.

(31) Voller, V.; Porte-Agel, F *Journal of Computational Physics* **2002**, *179*, 698–703.

(32) Hehre, W. J. *Accounts of Chemical Research* **1976**, *9*, 399–406.

(33) Pople, J. A.; Hehre, W. J. *Journal of Computational Physics* **1978**, *27*, 161–168.

(34) Hohenberg, P.; Kohn, W. *Physical Review* **1964**, *136*, B864.

(35) Kohn, W.; Sham, L. J. *Physical Review* **1965**, *140*, A1133.

(36) Westheimer, F.; Shookhoff, M. W. *Journal of the American Chemical Society* **1940**, *62*, 269–275.

(37) Alder, B. J.; Wainwright, T. E. *The Journal of Chemical Physics* **1959**, *31*, 459–466.

(38) Rahman, A. *Physical Review* **1964**, *136*, A405.

(39) Wiberg, K. B. *Journal of the American Chemical Society* **1965**, *87*, 1070–1078.

(40) Karplus, M.; McCammon, J. A. *Nature Structural Biology* **2002**, *9*, 646–652.

(41) Rossky, P. J.; Karplus, M. *Journal of the American Chemical Society* **1979**, *101*, 1913–1937.

(42) Leven, I.; Hao, H.; Tan, S.; Guan, X.; Penrod, K. A.; Akbarian, D.; Evangelisti, B.; Hossain, M. J.; Islam, M. M.; Koski, J. P., et al. *Journal of Chemical Theory and Computation* **2021**, *17*, 3237–3251.

(43) Iftimie, R.; Minary, P.; Tuckerman, M. E. *Proceedings of the National Academy of Sciences* **2005**, *102*, 6654–6659.

(44) Marx, D.; Hutter, J. *Modern Methods and Algorithms of Quantum Chemistry* **2000**, *1*, 141.

(45) Piccini, G.; Lee, M.-S.; Yuk, S. F.; Zhang, D.; Collinge, G.; Kollias, L.; Nguyen, M.-T.; Glezakou, V.-A.; Rousseau, R. *Catalysis Science & Technology* **2022**, *12*, 12–37.

(46) Car, R.; Parrinello, M. *Physical Review Letters* **1985**, *55*, 2471.

(47) Fu, Y.; Bernasconi, L.; Liu, P. *Journal of the American Chemical Society* **2021**, *143*, 1577–1589.

(48) De Angelis, F.; Fantacci, S.; Sgamellotti, A. *Coordination Chemistry Reviews* **2006**, *250*, 1497–1513.

(49) Zheng, Q.; Chu, W.; Zhao, C.; Zhang, L.; Guo, H.; Wang, Y.; Jiang, X.; Zhao, J. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2019**, *9*, e1411.

(50) Hassanali, A. A.; Cuny, J.; Verdolino, V.; Parrinello, M. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2014**, *372*, 20120482.

(51) Brémond, É.; Savarese, M.; Rega, N.; Ciofini, I.; Adamo, C. *Journal of Chemical Theory and Computation* **2022**, *18*, 1501–1511.

(52) Kovács, D. P.; Oord, C. v. d.; Kucera, J.; Allen, A. E.; Cole, D. J.; Ortner, C.; Csányi, G. *Journal of Chemical Theory and Computation* **2021**, *17*, 7696–7711.

(53) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. *Chemical Reviews* **2021**, *121*, 10073–10141.

(54) Unke, O. T.; Meuwly, M. *Journal of Chemical Theory and Computation* **2019**, *15*, 3678–3693.

(55) Koch, W.; Holthausen, M. C., *A chemist's Guide to Density Functional Theory*; John Wiley & Sons: 2015.

(56) Szabó, P. B.; Csóka, J.; Kállay, M.; Nagy, P. R. *Journal of Chemical Theory and Computation* **2021**, *17*, 2886–2905.

(57) Schütz, M.; Manby, F. R. *Physical Chemistry Chemical Physics* **2003**, *5*, 3349–3358.

(58) Brickel, S.; Das, A. K.; Unke, O. T.; Turan, H. T.; Meuwly, M. *Electronic Structure* **2019**, *1*, 024002.

(59) Ang, S. J.; Wang, W.; Schwalbe-Koda, D.; Axelrod, S.; Gómez-Bombarelli, R. *Chem* **2021**, *7*, 738–751.

(60) Devereux, C.; Smith, J. S.; Huddleston, K. K.; Barros, K.; Zubatyuk, R.; Isayev, O.; Roitberg, A. E. *Journal of Chemical Theory and Computation* **2020**, *16*, 4192–4202.

(61) Fujikake, S.; Deringer, V. L.; Lee, T. H.; Krynski, M.; Elliott, S. R.; Csányi, G. *The Journal of Chemical Physics* **2018**, *148*, 241714.

(62) Lahey, S.-L. J.; Rowley, C. N. *Chemical Science* **2020**, *11*, 2362–2368.

(63) Bartók, A. P.; Csányi, G. *International Journal of Quantum Chemistry* **2015**, *115*, 1051–1057.

(64) Ko, T. W.; Finkler, J. A.; Goedecker, S.; Behler, J. *Accounts of Chemical Research* **2021**, *54*, 808–817.

(65) Anstine, D. M.; Isayev, O. *The Journal of Physical Chemistry A* **2023**.

(66) Bartók, A. P.; Kondor, R.; Csányi, G. *Physical Review B* **2013**, *87*, 184115.

(67) Behler, J.; Parrinello, M. *Physical Review Letters* **2007**, *98*, 146401.

(68) Smith, J. S.; Isayev, O.; Roitberg, A. E. *Chemical Science* **2017**, *8*, 3192–3203.

(69) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. *Physical Review Letters* **2010**, *104*, 136403.

(70) Drautz, R. *Physical Review B* **2019**, *99*, 014104.

(71) Web of Science, advanced search, 'machine learning potentials', https://www.webofscience.com/wos/woscc/advanced-search, Accessed: 2023-04-11.

(72) Gassner, H.; Probst, M.; Lauenstein, A.; Hermansson, K. *The Journal of Physical Chemistry A* **1998**, *102*, 4596–4605.

(73) Lorenz, S.; Groß, A.; Scheffler, M. *Chemical Physics Letters* **2004**, *395*, 210–215.

(74) Manzhos, S.; Carrington, T. *The Journal of Chemical Physics* **2007**, *127*.

(75) Jose, K. J.; Artrith, N.; Behler, J. *The Journal of Chemical Physics* **2012**, *136*, 194111.

(76) Jiang, B.; Guo, H. *The Journal of Chemical Physics* **2013**, *139*, 054112.

(77) Kronik, L.; Tkatchenko, A. *Accounts of Chemical Research* **2014**, *47*, 3208–3216.

(78) Yao, K.; Herr, J. E.; Toth, D. W.; Mckintyre, R.; Parkhill, J. *Chemical Science* **2018**, *9*, 2261–2269.

(79) Kocer, E.; Ko, T. W.; Behler, J. *Annual Review of Physical Chemistry* **2022**, *73*, 163–186.

(80) Ghasemi, S. A.; Hofstetter, A.; Saha, S.; Goedecker, S. *Physical Review B* **2015**, *92*, 045131.

(81) Ko, T. W.; Finkler, J. A.; Goedecker, S.; Behler, J. *Nature Communications* **2021**, *12*, 398.

(82) Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. *Advances in Neural Information Processing Systems* **2017**, *30*.

(83) Pinheiro, M.; Ge, F.; Ferré, N.; Dral, P. O.; Barbatti, M. *Chemical Science* **2021**, *12*, 14396–14413.

(84) Dral, P. O. *The Journal of Physical Chemistry Letters* **2020**, *11*, 2336–2347.

(85) Golub, G. H.; Hansen, P. C.; O'Leary, D. P. *SIAM Journal on Matrix Analysis and Applications* **1999**, *21*, 185–194.

(86) Allen, A. E.; Dusson, G.; Ortner, C.; Csányi, G. *Machine Learning: Science and Technology* **2021**, *2*, 025017.

(87) Csányi, G.; Bernstein, N.; Kermode, J. R. libAtoms/QUIP, version 0.9.9, 2021.

(88) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C., et al. *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

(89) Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V. B. *SIAM Review* **2017**, *59*, 65–98.

(90) Dusson, G.; Bachmayr, M.; Csányi, G.; Drautz, R.; Etter, S.; van der Oord, C.; Ortner, C. *Journal of Computational Physics* **2022**, *454*, 110946.

(91) Wang, H.; Zhang, L.; Han, J.; Weinan, E *Computer Physics Communications* **2018**, *228*, 178–184.

(92) Dral, P. O. *Journal of Computational Chemistry* **2019**, *40*, 2339–2347.

(93) Abbott, A. S.; Turney, J. M.; Zhang, B.; Smith, D. G.; Altarawy, D.; Schaefer III, H. F. *Journal of Chemical Theory and Computation* **2019**, *15*, 4386–4398.

(94) Chmiela, S.; Sauceda, H. E.; Poltavsky, I.; Müller, K.-R.; Tkatchenko, A. *Computer Physics Communications* **2019**, *240*, 38–45.

(95) Shao, Y.; Hellström, M.; Mitev, P. D.; Knijff, L.; Zhang, C. *Journal of Chemical Information and Modeling* **2020**, *60*, 1184–1193.

(96) Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J. S.; Roitberg, A. E. *Journal of Chemical Information and Modeling* **2020**, *60*, 3408–3415.

(97) Zebec, Z.; Wilkes, J.; Jervis, A. J.; Scrutton, N. S.; Takano, E.; Breitling, R. *Current Opinion in Chemical Biology* **2016**, *34*, 37–43.

(98) Dambolena, J.; López, A.; Cánepa, M.; Theumer, M.; Zygadlo, J.; Rubinstein, H. *Toxicon* **2008**, *51*, 37–44.

(99) Ishikawa, N. K.; Fukushi, Y.; Yamaji, K.; Tahara, S.; Takahashi, K. *Journal of Natural Products* **2001**, *64*, 932–934.

(100) Corrêa, E. J. A.; Carvalho, F. C.; de Castro Oliveira, J. A.; Bertolucci, S. K. V.; Scotti, M. T.; da Silveira, C. H.; Guedes, F. C.; Melo, J. O. F.; De Melo-Minardi, R. C.; de Lima, L. H. F. **2022**.

(101) Hong, Y. J.; Tantillo, D. J. *Chemical Science* **2010**, *1*, 609–614.

(102) Tantillo, D. J. *Natural Product Reports* **2011**, *28*, 1035–1053.

(103) Tantillo, D. J. *Chemical Society Reviews* **2010**, *39*, 2847–2854.

(104) McCulley, C. H.; Tantillo, D. J. *The Journal of Physical Chemistry A* **2018**, *122*, 8058–8061.

(105) Feng, Z.; Tantillo, D. J. *Journal of the American Chemical Society* **2021**, *143*, 1088–1097.

(106) Dolbier Jr, W. R.; Anapolle, K. E.; McCullagh, L.; Matsui, K.; Riemann, J. M.; Rolison, D. *The Journal of Organic Chemistry* **1979**, *44*, 2845–2849.

(107) Klärner, F.-G. *Topics in Stereochemistry* **1984**, *15*, 1–42.

(108) LaFever, R. E.; Croteau, R. *Archives of Biochemistry and Biophysics* **1993**, *301*, 361–366.

(109) Degenhardt, J.; Köllner, T. G.; Gershenzon, J. *Phytochemistry* **2009**, *70*, 1621–1637.

(110) Dewick, P. M. *Natural Product Reports* **2002**, *19*, 181–222.

(111) Croteau, R., *Monoterpene Biosynthesis: Cyclization of Geranyl Pyrophosphate to (+)-sabinene*; ACS Publications: 1992.

(112) Croteau, R.; Miyazaki, J. H.; Wheeler, C. J. *Archives of Biochemistry and Biophysics* **1989**, *269*, 507–516.

(113) Ess, D. H.; Wheeler, S. E.; Iafe, R. G.; Xu, L.; Celebi-Oelcuem, N.; Houk, K. N. *Angewandte Chemie International Edition* **2008**, *47*, 7592–7601.

(114) Ghigo, G.; Maranzana, A.; Tonachini, G. *The Journal of Organic Chemistry* **2013**, *78*, 9041–9050.

(115) Jaisamut, S.; Prabpai, S.; Tancharoen, C.; Yuenyongsawad, S.; Hannongbua, S.; Kongsaeree, P.; Plubrukarn, A. *Journal of Natural Products* **2013**, *76*, 2158–2161.

(116) Maeda, S.; Harabuchi, Y.; Ono, Y.; Taketsugu, T.; Morokuma, K. *International Journal of Quantum Chemistry* **2015**, *115*, 258–269.

(117) Hare, S. R.; Tantillo, D. J. *Chemical Science* **2017**, *8*, 1442–1449.

(118) Tuckerman, M., *Statistical Mechanics: Theory and Molecular Simulation*; Oxford university press: 2010.

(119) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. *Journal of Computational Chemistry* **1992**, *13*, 1011–1021.

(120) Kästner, J.; Thiel, W. *The Journal of Chemical Physics* **2005**, *123*, 144104.

(121) Govind Kumar, V.; Polasa, A.; Agrawal, S.; Kumar, T. K. S.; Moradi, M. *Nature Computational Science* **2022**, 1–12.

(122) Patel, J. S.; Ytreberg, F. M. *Journal of Chemical Theory and Computation* **2018**, *14*, 991–997.

(123) Meshkin, H.; Zhu, F. *Journal of Chemical Theory and Computation* **2017**, *13*, 2086–2097.

(124) You, W.; Tang, Z.; Chang, C.-E. A. *Journal of Chemical Theory and Computation* **2019**, *15*, 2433–2443.

(125) Tam, N. M.; Nguyen, T. H.; Ngan, V. T.; Tung, N. T.; Ngo, S. T. *Royal Society Open Science* **2022**, *9*, 211480.

(126) Lemkul, J. A.; Bevan, D. R. *The Journal of Physical Chemistry B* **2010**, *114*, 1652–1660.

(127) Mancardi, G.; Terranova, U.; de Leeuw, N. H. *Crystal Growth & Design* **2016**, *16*, 3353–3358.

(128) Lu, X.; Wang, X.; Fu, B.; Zhang, D. *The Journal of Physical Chemistry A* **2019**, *123*, 3969–3976.

(129) Ishida, K.; Morokuma, K.; Komornicki, A. *The Journal of Chemical Physics* **1977**, *66*, 2153–2156.

(130) Cerjan, C. J.; Miller, W. H. *The Journal of chemical physics* **1981**, *75*, 2800–2806.

(131) Henkelman, G.; Uberuaga, B. P.; Jónsson, H. *The Journal of Chemical Physics* **2000**, *113*, 9901–9904.

(132) Nemec, M.; Hoffmann, D. *Journal of Chemical Theory and Computation* **2017**, *13*, 400–414.

(133) Grossfield, A.; Patrone, P. N.; Roe, D. R.; Schultz, A. J.; Siderius, D. W.; Zuckerman, D. M. *Living Journal of Computational Molecular Science* **2018**, *1*.

(134) Park, S.; Im, W. *Journal of Chemical Theory and Computation* **2014**, *10*, 2719–2728.

(135) Sabri Dashti, D.; Roitberg, A. E. *Journal of Chemical Theory and Computation* **2013**, *9*, 4692–4699.

(136) Park, S.; Kim, T.; Im, W. *Physical Review Letters* **2012**, *108*, 108102.

(137) Meng, Y.; Roux, B. *Journal of Chemical Theory and Computation* **2015**, *11*, 3523–3529.

(138) Clark, S. J.; Segall, M. D.; Pickard, C. J.; Hasnip, P. J.; Probert, M. I.; Refson, K.; Payne, M. C. *Zeitschrift für Kristallographie-crystalline Materials* **2005**, *220*, 567–570.

(139) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D., et al. *PLoS Computational Biology* **2017**, *13*, e1005659.

(140) Young, T. A.; Silcock, J. J.; Sterling, A. J.; Duarte, F. *Angewandte Chemie* **2021**, *133*, 4312–4320.

(141) Bannwarth, C.; Ehlert, S.; Grimme, S. *Journal of Chemical Theory and Computation* **2019**, *15*, 1652–1671.

(142) Jónsson, H.; Mills, G.; Jacobsen, K. W. In *Classical and Quantum Dynamics in Condensed Phase Simulations*; World Scientific: 1998, pp 385–404.

(143) Ceriotti, M.; More, J.; Manolopoulos, D. E. *Computer Physics Communications* **2014**, *185*, 1019–1026.

(144) Tribello, G. A.; Bonomi, M.; Branduardi, D.; Camilloni, C.; Bussi, G. *Computer Physics Communications* **2014**, *185*, 604–613.

(145) Grossfield, A. WHAM: the weighted histogram analysis method, http://membrane.urmc.rochester.edu/wordpress/?page_id=126, version 2.0.11, 2023.

(146) Johnston-Wood, T. *adaptiveus*, https://github.com/tristan-j-wood/adaptive-us, version 1.0.0a1, 2022.

(147) Mezei, M. *Journal of Computational Physics* **1987**, *68*, 237–248.

(148) Bartels, C.; Karplus, M. *The Journal of Physical Chemistry B* **1998**, *102*, 865–880.

(149) Hooft, R. W.; van Eijck, B. P.; Kroon, J. *The Journal of Chemical Physics* **1992**, *97*, 6690–6694.

(150) Bartels, C.; Karplus, M. *Journal of Computational Chemistry* **1997**, *18*, 1450–1462.

(151) Bartels, C.; Schaefer, M.; Karplus, M. *Theoretical Chemistry Accounts* **1999**, *101*, 62–66.

(152) Rajamani, R.; Naidoo, K. J.; Gao, J. *Journal of Computational Chemistry* **2003**, *24*, 1775–1781.

(153) Mills, M.; Andricioaei, I. *The Journal of Chemical Physics* **2008**, *129*, 114101.

(154) Grossfield, A.; Woolf, T. B. *Langmuir* **2002**, *18*, 198–210.

(155) Szabo, A.; Ostlund, N. S., *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*; Courier Corporation: 2012.

(156) Krishnan, R.; Binkley, J. S.; Seeger, R.; Pople, J. A. *The Journal of chemical physics* **1980**, *72*, 650–654.

(157) Schäfer, A.; Horn, H.; Ahlrichs, R. *The Journal of chemical physics* **1992**, *97*, 2571–2577.

(158) Thomas, L. H. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1927; Vol. 23, pp 542–548.

(159) Mardirossian, N.; Head-Gordon, M. *Molecular Physics* **2017**, *115*, 2315–2372.

(160) Goerigk, L.; Mehta, N. *Australian Journal of Chemistry* **2019**, *72*, 563–573.

(161) Perdew, J. P.; Ziesche, P; Eschrig, H, *Electronic Structure of Solids*; Akademie Verlag, Berlin: 1991.

(162) Becke, A. D. *Physical Review A* **1988**, *38*, 3098.

(163) Perdew, J. P.; Chevary, J. A.; Vosko, S. H.; Jackson, K. A.; Pederson, M. R.; Singh, D. J.; Fiolhais, C. *Physical Review B* **1992**, *46*, 6671.

(164) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Physical Review Letters* **1996**, *77*, 3865.

(165) Zhao, Y.; Truhlar, D. G. *The Journal of Chemical Physics* **2006**, *125*, 194101.

(166) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. *The Journal of Physical Chemistry* **1994**, *98*, 11623–11627.

(167) Adamo, C.; Barone, V. *The Journal of Chemical Physics* **1999**, *110*, 6158–6170.

(168) Goerigk, L.; Grimme, S. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*, 576–600.

(169) Grimme, S. *The Journal of Chemical Physics* **2006**, *124*, 034108.

(170) Grimme, S.; Ehrlich, S.; Goerigk, L. *Journal of Computational Chemistry* **2011**, *32*, 1456–1465.

(171) Becke, A. D.; Johnson, E. R. *The Journal of Chemical Physics* **2005**, *123*, 154101.

(172) Caldeweyher, E.; Ehlert, S.; Hansen, A.; Neugebauer, H.; Spicher, S.; Bannwarth, C.; Grimme, S. *The Journal of Chemical Physics* **2019**, *150*, 154122.

(173) Verlet, L. *Physical Review* **1967**, *159*, 98.

(174) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. *The Journal of Chemical Physics* **1982**, *76*, 637–649.

(175) Berendsen, H. J.; Postma, J. v.; Van Gunsteren, W. F.; DiNola, A.; Haak, J. R. *The Journal of Chemical Physics* **1984**, *81*, 3684–3690.

(176) Bussi, G.; Donadio, D.; Parrinello, M. *The Journal of Chemical Physics* **2007**, *126*, 014101.

(177) Turq, P.; Lantelme, F.; Friedman, H. L. *The Journal of Chemical Physics* **1977**, *66*, 3039–3044.

(178) Mahoney, M. W.; Drineas, P. *Proceedings of the National Academy of Sciences* **2009**, *106*, 697–702.

(179) Szlachta, W. J.; Bartók, A. P.; Csányi, G. *Physical Review B* **2014**, *90*, 104108.

(180) Berendsen, H.; Postma, J.; Van Gunsteren, W.; Hermans, J, *Intermolecular forces*; Reidel Dordrecht: 1981.

(181) Raz, K.; Levi, S.; Gupta, P. K.; Major, D. T. *Current Opinion in Biotechnology* **2020**, *65*, 248–258.

(182) Tholl, D. *Current Opinion in Plant Biology* **2006**, *9*, 297–304.

(183) Rappé, A. K.; Casewit, C. J.; Colwell, K.; Goddard III, W. A.; Skiff, W. M. *Journal of the American Chemical Society* **1992**, *114*, 10024–10035.

(184) Lawrence, J.; Bernal, J.; Witzgall, C. *Journal of research of the National Institute of Standards and Technology* **2019**, *124*, 1.

(185) Grimme, S. *Chemistry–A European Journal* **2012**, *18*, 9955–9964.

(186) Espenson, J., *Chemical Kinetics and Reaction Mechanisms (2002)*; McGraw-Hill: 1995.

(187) Habershon, S.; Manolopoulos, D. E.; Markland, T. E.; Miller III, T. F. *Annual Review of Physical Chemistry* **2013**, *64*, 387–413.

(188) Craig, I. R.; Manolopoulos, D. E. *The Journal of Chemical Physics* **2005**, *122*, 084106.

(189) Zev, S.; Gupta, P. K.; Pahima, E.; Major, D. T. *Journal of Chemical Theory and Computation* **2021**, *18*, 167–178.

(190) Kokubo, H.; Tanaka, T.; Okamoto, Y. *Journal of Computational Chemistry* **2011**, *32*, 2810–2821.

(191) Stroh, K. S.; Risselada, H. J. *Journal of Chemical Theory and Computation* **2021**, *17*, 5276–5286.

(192) Schiferl, S. K.; Wallace, D. C. *The Journal of Chemical Physics* **1985**, *83*, 5203–5209.

(193) Csiszár, I. *The Annals of Probability* **1975**, 146–158.

(194) Laio, A.; Parrinello, M. *Proceedings of the National Academy of Sciences* **2002**, *99*, 12562–12566.

(195) Hénin, J.; Lelièvre, T.; Shirts, M. R.; Valsson, O.; Delemotte, L. *arXiv preprint arXiv:2202.04164* **2022**.

(196) Laio, A.; Rodriguez-Fortea, A.; Gervasio, F. L.; Ceccarelli, M.; Parrinello, M. *The Journal of Physical Chemistry B* **2005**, *109*, 6714–6721.

(197) Barducci, A.; Bussi, G.; Parrinello, M. *Physical Review Letters* **2008**, *100*, 020603.

(198) Pfaendtner, J.; Bonomi, M. *Journal of Chemical Theory and Computation* **2015**, *11*, 5062–5067.

(199) Tiwary, P.; Parrinello, M. *Physical Review Letters* **2013**, *111*, 230602.

(200) Branduardi, D.; Bussi, G.; Parrinello, M. *Journal of Chemical Theory and Computation* **2012**, *8*, 2247–2254.

(201) Wang, F.; Landau, D. P. *Physical Review Letters* **2001**, *86*, 2050.

(202) Kirkpatrick, S.; Gelatt Jr, C. D.; Vecchi, M. P. *Science* **1983**, *220*, 671–680.

(203) Comer, J.; Gumbart, J. C.; Hénin, J.; Lelièvre, T.; Pohorille, A.; Chipot, C. *The Journal of Physical Chemistry B* **2015**, *119*, 1129–1151.

(204) Hourahine, B.; Aradi, B.; Blum, V.; Bonafé, F; Buccheri, A; Camacho, C.; Cevallos, C.; Deshaye, M.; Dumitrică, T; Dominguez, A, et al. *The Journal of Chemical Physics* **2020**, *152*, 124101.

(205) Gaus, M.; Lu, X.; Elstner, M.; Cui, Q. *Journal of Chemical Theory and Computation* **2014**, *10*, 1518–1537.

(206) Csányi, G.; Winfield, S.; Kermode, J. R.; De Vita, A; Comisso, A.; Bernstein, N.; Payne, M. C. *Expressive Programming for Computational Physics in Fortran 95+*; 2007.

(207) Kermode, J. R. *J. Phys. Condens. Matter* **2020**, 305901.

(208) Harris, C. R. et al. *Nature* **2020**, *585*, 357–362.

(209) Hunter, J. D. *Computing in Science & Engineering* **2007**, *9*, 90–95.

(210) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J., et al. *Nature Methods* **2020**, *17*, 261–272.

A

Appendices

A.1 Molecular dynamics integrators and thermostats

A.1.1 Verlet algorithm

Suppose we wish to know the Cartesian coordinates of N particles in a system at time $t+\Delta t$. This expression can be written as a Taylor series truncated at third order, **Eqn. A.1**.

$$\mathbf{r}_i(t+\Delta t) \approx \mathbf{r}_i(t) + \Delta t\ \dot{\mathbf{r}}_i(t) + \frac{\Delta t^2}{2}\ddot{\mathbf{r}}_i(t) \tag{A.1}$$

Writing time derivative of position as the velocity, $v_i(t)$, and substituting Newton's second

law in as $\ddot{\mathbf{r}}_i = \mathbf{F}_i(\mathbf{r})/m_i$, yields

$$\mathbf{r}_i(t + \Delta t) \approx \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t) \tag{A.2}$$

We now have an expression for computing the positions of a set of particles at time $t + \Delta t$ given their positions, velocities and forces at time t. However, we can't easily calculate the velocities given only the positions and force at one time. But we can obtain a velocity-independent scheme by considering $\mathbf{r}_i(t - \Delta t)$, **Eqn. A.3**.

$$\mathbf{r}_i(t - \Delta t) \approx \mathbf{r}_i(t) - \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t) \tag{A.3}$$

Note only the velocity changes sign as the time in the force term is quadratic. Adding **Eqn. A.2** and **Eqn. A.3**, we obtain **Eqn. A.4**.

$$\mathbf{r}_i(t + \Delta t) + \mathbf{r}_i(t - \Delta t) \approx 2\mathbf{r}_i(t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t) \tag{A.4}$$

And so rearranging for $r_i(t + \Delta t)$, gives **Eqn. A.5**.

$$\mathbf{r}_i(t + \Delta t) \approx 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t) \tag{A.5}$$

Thus, given a set of initial conditions $\{\mathbf{r}_n(0)\}$, $\{\mathbf{v}_n(0)\}$, we can obtain coordinates at $\{\mathbf{r}_n(\Delta t)\}$ and generate a trajectory of arbitrary length.

A.1.2 Velocity Verlet algorithm

Consider again $\mathbf{r}_i(t + \Delta t)$, **Eqn. A.2**. If we instead start from $\mathbf{r}_i(t + \Delta t)$ and $\mathbf{v}_i(t + \Delta t)$ and compute $\mathbf{F}_i(t + \Delta t)$, we can evolve backwards in time to $r_i(t)$ using a time step of $-\Delta t$, **Eqn. A.6**.

$$\mathbf{r}_i(t) \approx \mathbf{r}_i(t+\Delta t) - \Delta t\ \mathbf{v}_i(t+\Delta t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t+\Delta t) \tag{A.6}$$

Substituting **Eqn. A.2** into **Eqn. A.6**, we obtain an expression for the velocities at time $t+\Delta t$, **Eqn. A.7**.

$$\begin{aligned} \mathbf{r}_i(t) &\approx \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t) - \Delta t\ \mathbf{v}_i(t+\Delta t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t+\Delta t) \\ \therefore \quad \mathbf{v}_i(t+\Delta t) &\approx \mathbf{v}_i(t) + \frac{\Delta t}{2m_i}\left[\mathbf{F}_i(t) + \mathbf{F}_i(t+\Delta t)\right] \\ \text{and} \quad \mathbf{r}_i(t+\Delta t) &\approx \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t) \end{aligned} \tag{A.7}$$

These velocity Verlet equations solve the positions and velocities (phase space) simultaneously.

A.1.3 Berendsen temperature coupling

This thermostat controls the temperature of the system by rescaling the velocities of each particle every step by a time–dependent factor λ (usually $0.8 \leq \lambda \leq 1.25$).[175] The deviation from the actual temperature, T, to the target temperature, T_0, follows an exponential decay, meaning the system approaches the target temperature exponentially quickly, as controlled by the time constant τ, **Eqn. A.8**.

$$T - T_0 = Ae^{-t/\tau} \tag{A.8}$$

While this allows the target temperature to be reached quickly, the thermostat suppresses fluctuations in the kinetic energy, meaning the correct NVT ensemble is not obtained. Therefore, in practice this thermostat is not often used as the velocity rescaling thermostat is preferred.

A.1.4 Velocity-rescaling temperature coupling

The velocity–rescaling thermostat is identical to the Berendsen thermostat but includes an additional term, dK, which ensures the correct kinetic energy distribution is obtained.[176] This is a more commonly used thermostat as it retains the speed of Berendsen as well as the correct canonical ensemble.

A.1.5 Langevin dynamics

The Langevin equations provide a method of coupling a system to a heat bath.[177] This approach allows the velocities of the particles to be temperature controlled according to the friction felt between the system and bath. To start, a Hamiltonian defining a set of system coordinates, q, and bath coordinates, x_α, **Eqn. A.9**.[118]

$$\mathcal{H} = \frac{p^2}{2\mu} + V(q) + \sum_{\alpha=1}^{n}\left[\frac{p_\alpha^2}{2m_\alpha} + \frac{1}{2}\sum_{\alpha=1}^{n} m_\alpha \omega_\alpha^2 x_\alpha^2\right] + q\sum_{\alpha=1}^{n} g_\alpha x_\alpha \tag{A.9}$$

The first term on the right-hand side of **Eqn. A.9** denotes the kinetic energy of the system, while the second denotes the potential energy depending on q. The first term in the bracketed quantity is the kinetic energy of the bath particles and the second is interaction between all the bath particles. The final term, involving q, is the interaction between the system and bath. ω_α are the bath frequencies and g_α is shorthand for the matrix coupling the bath coordinates. From this Hamiltonian, the equations of motion can be derived and solved to yield

$$\mu\ddot{q} = -\frac{dW}{dq} - \int_0^t \mathrm{d}\tau\ \dot{q}(\tau)\zeta(t-\tau) + R(t), \tag{A.10}$$

the generalised Langevin equation (GLE). W represents the potential of mean force acting on the system coordinate. The $R(t)$ terms represents the random force acting on the particles due to friction (sampled from a Gaussian distribution). The convolution integral is called the memory integral, since it it depends on the history of q. This term represents the finite

time required for the bath to respond to a fluctuation in the system. In the limit of the bath responding infinitely quickly to the motion of the system, the friction kernel, $\zeta(t)$, can be written as $\zeta(t) = \lim_{\varepsilon \to 0} \zeta_0 \delta(t - \varepsilon)$. The equations of motion therefore become

$$\mu \ddot{q} = -\frac{dW}{dq} - \dot{q}(t)\zeta_0 + R(t). \tag{A.11}$$

A.2 Enhanced sampling methods

A.2.1 Metadynamics

In metadynamics, sampling is accelerated along a chosen set of collective variables (CV). [194] The energy basins in the free energy surface are filled up by a time-dependent potential that is constructed from the system's trajectory history. The aim is to create a biased landscape that is flat, allowing uniform sampling across the CV and reconstruction of the free energy profile. A bias of the form

$$U_G(\mathbf{r}_1, \ldots, \mathbf{r}_N) = W \sum_{t=\tau_G, 2\tau_G, \ldots,} \exp\left[-\sum_{\alpha=1}^{n} \frac{(f_\alpha(\mathbf{r}) - f_\alpha(\mathbf{r}_G(t)))^2}{2\Delta s^2} \right], \tag{A.12}$$

is added to the potential energy at time intervals τ_G. [118] Here, $\mathbf{r}_G(t)$ is the time evolution of the set of Cartesian coordinate up to time t under the biased potential. The effect of this bias is to add Gaussians gradually to fill up the energy basins. Once the landscape is flat, the free energy profile is simply the negative of this bias at each CV point q_i. One of the main challenges in metadynamics is selecting a suitable variable can be challenging.[195] In addition, the free energy is sensitive to the Gaussian parameters, so they also need to be carefully chosen.[196] Many variations of metadynamics have been developed, including well-tempered metadynamics[197], parallel-bias metadynamics[198], infrequent metadynamics[199] and adaptive Gaussian metadynamics.[200]

A.2.2 Replica-exchange

Another popular enhanced sampling technique is the parallel-tempering replica-exchange Monte Carlo method.[118] Here, the goal is to accelerate conformational sampling over the entire potential energy landscape in systems that are defined by numerous local minima. There is no collective variable in replica-exchange, so the system is not biased to any particular region but rather explores the entire potential energy surface more quickly than conventional MD or Monte Carlo. The method works by creating M replicas of the system at different temperatures, starting from $T1$ (the target canonical ensemble) to T_M (the highest temperature replica). Replica simulations are performed at the different temperatures, where T_M can quickly scan across the potential energy surface. Then at periodic intervals, an exchange is attempted, where the coordinates from adjacent replicas ($\mathbf{r}^{(K)}$ and $\mathbf{r}^{(K+1)}$) are exchanged with an acceptance probability of

$$A(\mathbf{r}^{(K+1)}, \mathbf{r}^{(K)} | \mathbf{r}^{(K)}, \mathbf{r}^{(K+1)}) = \min \left[1, \mathrm{e}^{-(\beta_K - \beta_{K+1})\left[U\left(\mathbf{r}^{(K)}\right) - U\left(\mathbf{r}^{(K+1)}\right)\right]} \right]. \tag{A.13}$$

Parallel-tempering replica-exchange enables accelerated sampling across a rough PES while preserving the canonical distribution. While replica-exchange samples the potential energy surface faster than conventional sampling, it does not bias sampling along a reaction coordinate required for free energy calculations.

A.2.3 Additional methods

Many other enhanced sampling methods exist, some of which are variations of others, e.g., umbrella sampling is a restrained approach to the blue moon ensemble approach.[118] Wang–Landau sampling, where the microcanonical ensemble is sampled according to the Metropolis acceptance probability, has been shown to be mathematically related to the well-tempered ensemble metadynamics methods. [201] Additional methods include simulated annealing[202], adaptive biasing force[203] and corresponding hybrid methods.[195]

A.2.4 Umbrella integration

An alternative approach to WHAM, introduced by Kästner and Thiel, combines umbrella sampling and thermodynamic integration in a method called umbrella integration (UI).[120] Here, if we assume that the bias is sufficiently strong (via a large κ), each window distribution will approximately follow a Gaussian centred at the reference $s^{(k)}$, **Eqn. A.14**.

$$P(q) \approx \frac{1}{\sqrt{2\pi\sigma_k^2}} \mathrm{e}^{(q-\bar{q}_k)^2/2\sigma_k^2} \tag{A.14}$$

Using this expression, the derivative of the free energy, A_k, along the reaction coordinate, q, can be derived, **Eqn. A.15**.

$$\frac{\mathrm{d}A_k}{\mathrm{d}q} = \frac{1}{\beta\sigma_k^2}(q - \bar{q}_k) - \kappa(q - s^{(k)}) \tag{A.15}$$

The derivatives can be glued through a linear combination of the window derivatives, **Eqn. A.16**, weighted by coefficients, $C_k(q)$ satisfying **Eqn. 1.41**.

$$\frac{\mathrm{d}A_k}{\mathrm{d}q^{(i)}} = \sum_{k=1}^{n_w} C_k(q^{(i)}) \frac{\mathrm{d}A_k}{\mathrm{d}q}\Big|_{q=q^{(i)}}, \tag{A.16}$$

Where n_w is the number of windows. Finally, numerically integrating **Eqn. A.16** gives the full free energy profile. One limitation of UI is the assumption and reliance on Gaussian distribution, which for windows around steep energy regions can break down.

B

Computational methods

B.1 Methods for Chapter 2

DFT calculations were performed using ORCA (v. 5.0.3).[6] Geometry optimisations were run using the default ORCA parameters, while transition state optimisations calculated the Hessian at the beginning of the run and then after every five steps. Intrinsic reaction coordinate (IRC) calculations were performed in ORCA by invoking the IRC keyword and using a TightSCF. The initial Hessian was calculated using analytic frequencies if available. DFTB calculations were performed with DFTB+[204] using the 3ob parameters.[205] The Gaussian Approximation Potential (GAP) machine learning potential (MLP) was used within the *gap-train* Python package (v. 1.0.0)[1] and atomic cluster expansion (ACE) within *mlp-train*

(v. 1.0.0).[2] MLPs for the ACE[70] MLP were trained using the ACE.jl code (v. 0.8.4, Julia v. 1.6.3) with the Julia library for Interatomic Potentials (JuLIP, v. 0.10.1), wrapped by pyjulip (v. 0.1, using PyCall). The ACE loss function used a square norm of the difference in true and predicted energies and forces, with $w^E = 20$ and $w^F = 1$. A maximum correlation order of four was used, with a maximum polynomial degree of 12. An inner cutoff radius was chosen as greater than the minimum distance in the dataset, while the outer cutoff was set to 4.0 Å. Potentials generated by the GAP[69] used the QUIP (v. 0.9.10)[206] code, wrapped using quippy[207], with the Smooth Overlap of Atomic Positions (SOAP) kernel.[66] An expected error in energies of 0.316 meV atom^{-1} and forces of 0.1 eV Å^{-1} was used. A power coefficient for the kernel of four was used. The sparse method used CUR selection with 500 points. The σ for SOAP for each atom was 0.5 Å and $n_{\text{max}} = l_{\text{max}} = 6$. Finally, the cutoff radius in the short range descriptor was set to 4.0 Å. PhysNet[54] training was run directly from the GitHub distribution of Physnet (commit e243e2c) using the default parameters. Final training was run for 2000 epochs, with a training to validation split of 90:10. Potentials were trained on energies (weight 1) and force (weight 53). For active learning in *mlp-train*, 10 initial configurations are generated.[1, 2] The energy error selector was used with an $E_T = 0.1$ eV, a minimum number of active iterations of one and maximum of 50.

MLP dynamics were propagated using ASE[88] wrapped in *gap-train* (for GAP) and *mlp-train* (for ACE) and the PhysNet Python package (for PhysNet). The Langevin thermostat was used (0.02 friction coefficient, ASE units) at constant NVT, with initial velocities sampled from a Maxwell-Boltzmann distribution, 0.5 fs timestep. Molecular mechanics dynamics for water were run using GROMACS 2019.2 TIP3P parameters.[8] The system was minimised using the steepest decent mechanism, followed by NVT dynamics at 300 K for 100 ps, 1 fs timestep. Umbrella sampling simulations were performed using *mlp-train* and reweighted using the weighted histogram analysis method, also in *mlp-train*.[2] Python v. 3.7.12 was used for analysis and code development, along with numpy v. 1.21.6.[208] Plots were generated with matplotlib Python library.[209]

B.2 Methods for Chapter 3

DFT calculations were performed using ORCA (v. 5.0.3).[6] Geometry optimisations were run using the default ORCA parameters, while transition state optimisations calculated the Hessian at the beginning of the run and then after every five steps. Intrinsic reaction coordinate (IRC) calculations were performed in ORCA by invoking the IRC keyword and using a TightSCF. The initial Hessian was calculated using analytic frequencies if available. For the isoindene IRC, the convergence thresholds were reduced by a factor of ten to 2×10^{-4} (TolMaxG) and 5×10^{-5} (TolRMSG). Static free energy were performed in ORCA using the quasi-rigid rotor harmonic oscillator (QRRHO) method introduced by Grimme[185] and taken as the difference in free energy between the optimised reactant and transition state.

Climbing image nudged elastic band (CINEB)[131] calculations were performed within *autodE*[140] using 15 images. Path integral molecular dynamics (PIMD) simulations were performed using I-PI[143] with a 0.5 fs timestep in the NVT ensemble combined with umbrella sampling from *mlp-train* for free energy calculations or NVE ensemble (initial velocities sampled from a Maxwell-Boltzmann distribution at 298 K) for uphill dynamics for product ratios. For NVT dynamics, the PILE (path integral Langevin equation) thermostats were used with a tau of 510 (I-PI units).

ACE was used within *mlp-train* (v. 1.0.0).[2] MLPs for the ACE[70] MLP were trained using the ACE.jl code (v. 0.8.4, Julia v. 1.6.3) with the Julia library for Interatomic Potentials (JuLIP, v. 0.10.1), wrapped by pyjulip (v. 0.1, using PyCall). The ACE loss function used a square norm of the difference in true and predicted energies and forces, with $w^E = 20$ and $w^F = 1$. A maximum correlation order of four was used, with a maximum polynomial degree of 12. An inner cutoff radius was chosen as greater than the minimum distance in the dataset, while the outer cutoff was set to 4.0 Å.

MLP dynamics were propagated using ASE[88] wrapped in *mlp-train* for ACE. The Langevin thermostat was used (0.02 friction coefficient, ASE units) at constant NVT, with initial velocities sampled from a Maxwell-Boltzmann distribution, 0.5 fs timestep. Umbrella sampling

simulations were performed using *mlp-train* and reweighted using the weighted histogram analysis method, also in *mlp-train*.[2] NVE Uphill dynamics were performed using ASE wrapped in *mlp-train* with additional momentum placed in the C−O breaking bond. Python v. 3.7.12 was used for analysis and code development, along with numpy v. 1.21.6.[208] Plots were generated with matplotlib Python library.[209]

B.3 Methods for Chapter 4

The ACE and GAP MLPs used in this chapter were taken from the chapters 3 and 4 except in the case of the S_N2 reaction, which was taken from ref [1]. Python v. 3.7.12 was used for analysis and code development, along with numpy v. 1.21.6.[208] and scipy v. 1.7.3.[210] Plots were generated with matplotlib Python library.[209] Umbrella sampling simulations were run in the *adaptiveus* Python package.[146] Analytic functions were built and run using Python, numpy and scipy.

Molecular dynamics for Aβ42 protofibril (2BEG PBD code) were run using GROMACS 2019.2 GROMOS force field and TIP3P water[8] The system was minimised using the steepest decent mechanism, followed by 100 ps NVT equilibration, 2 fs timestep. The system was pulled apart along the central of mass of one fibril under a spring constant of 1000 kJ mol^{-1} nm^{-2} between 0 and 5 nm. Umbrella sampling simulations were run within *adaptiveus* for 5 ns, 2 fs timestep between 2.6 and 5.0 nm, 300 K, with a spring constant of 2000 kJ mol^{-1} nm^{-2}. The free energy was calculated using the Grossfield implementation of WHAM.[145]

B.4 Methods for Chapter 5

The ACE and GAP MLPs used in this chapter were taken from the chapters 3 and 4 except in the case of the S_N2 reaction, which was taken from ref [1]. Python v. 3.7.12 was used for analysis and code development, along with numpy v. 1.21.6.[208] and scipy v. 1.7.3.[210] Plots were generated with matplotlib Python library.[209] Umbrella sampling simulations were run in the *adaptiveus* Python package.[146] Analytic functions were built and run

using Python, numpy and scipy.