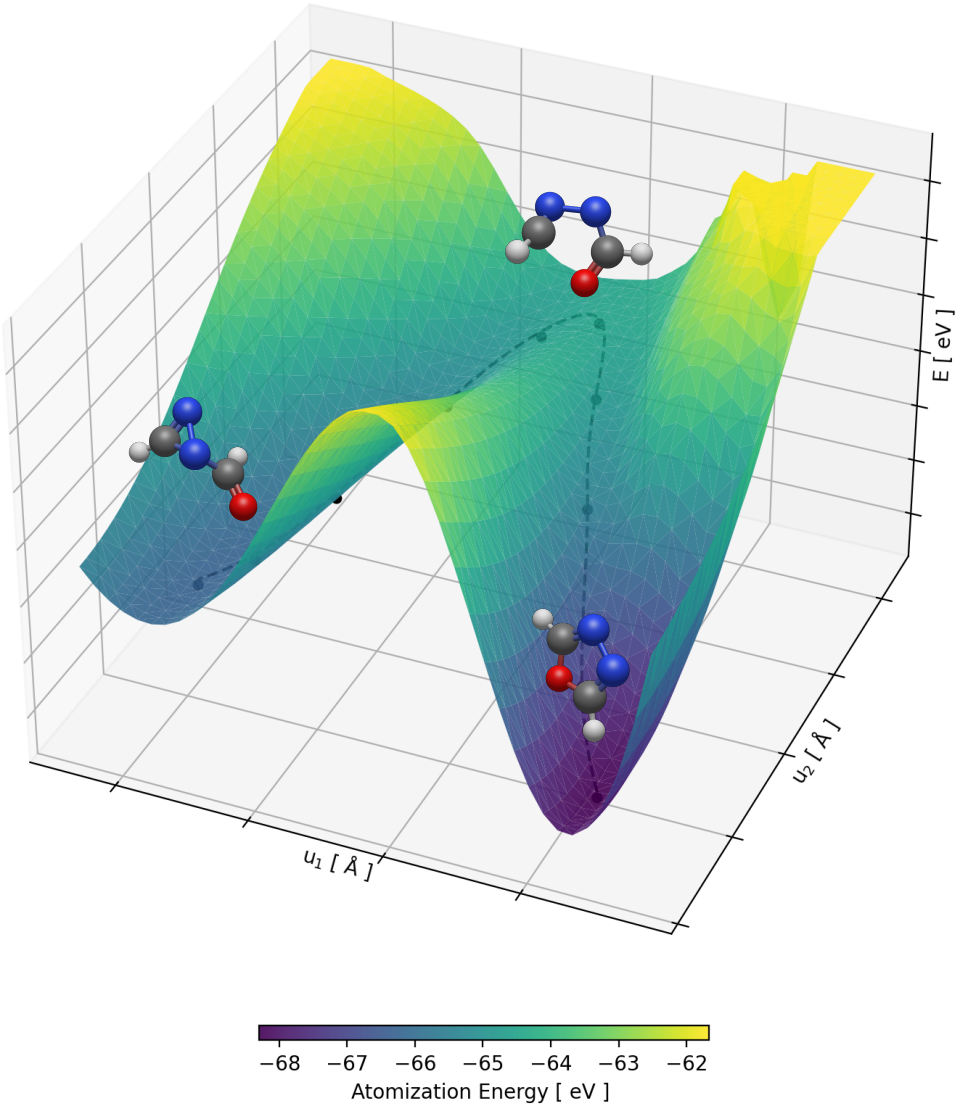
Machine Learning for Molecular Science

Schreiner, Mathias

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Doctor of Philosophy
Doctoral thesis in Computer Science



DTU Compute
Department of Applied Mathematics and Computer Science

Machine Learning for Molecular Science

Author: Mathias Schreiner

Supervisor: Ole Winther
Co-Supervisor: Tejs Vegge
Kongens Lyngby, October, 2023

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 321
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Preface

This Ph.D. thesis is the culmination of three years and one month of research, supervised by Ole Winther and Tejs Vegge, carried out in the Section for Cognitive Systems at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark. The research was conducted to fulfill the requirements for obtaining a Ph.D. degree in Computer Science.

Additionally, a two-month visit was made to the Artificial Intelligence and Machine Learning in the Natural Sciences (AIMLeNS) group, led by Simon Olsson, at the Data Science and AI Division at the Computer Science and Engineering Department at Chalmers University of Technology in Gothenburg, Sweden.

The thesis includes 7 chapters. Chapters 1-4 provide a foundational overview of relevant topics in physics and machine learning. Chapters 5-7 introduce and conclude my research. The appendix contains all of my published papers, as well as an attempt to explain to my mom, in Danish, what my thesis is about.

Kongens Lyngby, October 7, 2023

Mathias Schreiner

Acknowledgements

First and foremost, I would like to thank you, Ole Winther, my thesis supervisor, for your continued support and for the one million encouraging speeches that you have held for me over the last 3 years. You have a gift for encouragement. It has been a pleasure working with you, and I hope that we will continue doing so in the future.

I would like to thank Tejs Vegge, my thesis co-supervisor, Arghya Bhowmik and all the people at DTU Energy that has been involved with my work. I hope to one day come back to DTU Energy and continue where we left off.

A big thanks to you, Simon Olsson, who supervised the *Implicit Transfer Operator Learning* paper. I am so happy to have met you during my Ph.D. studies. Working with you has been an absolute pleasure and I am glad that we will continue doing so for the next two years during my postdoc!

A heartfelt thanks to my parents. I am grateful for your unwavering support and I think that you are truly amazing parents.

I also want to express my appreciation to the wonderful people who have shared my home over the last few years. Thank your for your support during my ups and downs and for, most of time, pretending to be interested in springs and electron clouds.

A big thanks to you, Christine Munch-Petersen for letting me borrow your summerhouse for the better half of the summer 2023 so I could write my thesis in peace. Garden work has proven to be excellent for restarting tired brain [1], which I have confirmed experimentally. You are the kindest. Thank you.

Summary

Methods from Machine Learning (ML) and in particular Neural Network (NN) models have in recent years proved to be capable emulators of expensive ab inito methods for electronic-structure calculations, while operating several orders of magnitude faster. These models are slowly transforming the field of computational quantum chemistry as accurate predictions of molecular properties can be obtained at unprecedented speeds, opening up for an exciting array of new possibilities.

In this thesis, I explore how NNs can be used to accelerate Transition State (TS)-search and multi time-scale simulation of molecular systems. It covers fundamental topics in physics for stochastic processes and quantum mechanics, methods and challenges related to calculating electronic structure in molecules, and an introduction to the NNs architectures used in this work.

My work has resulted in three notable scientific contributions which are presented in the thesis. The first of these contributions is the Transition1x dataset. This consists of Density Functional Theory (DFT) calculations for 10M molecular configurations, sampled with Nudged Elastic Band (NEB), around reaction pathways for 10K different reactions involving H, C, N, and O. This dataset provides valuable data for training NN models for tasks related to chemical reactions.

In the next contribution, NeuralNEB, NNs are trained on various datasets and evaluated on their ability to act as Potential Energy Surfaces (PESs) for NEB. Here it is shown that models trained on Transition1x outperform models trained on other datasets, underlining the importance of specific data relevant for the task.

Finally, the Implicit Transfer Operator Learning framework is presented. Here conditional Denoising Diffusion Probabilistic Models (DDPMs) are trained using a new data-augmentation scheme where training data in the form of trajectories from Molecular Dynamics (MD) simulations are augmented by sampling different lag-times during training. With this scheme, our models demonstrate the ability to capture dynamics at a range of timescales, providing a crucial step forward in multiple time-resolution MD.

Resumé

Metoder fra Machine Learning (ML), og især Neural Network (NN) modeller, har vist sig at være i stand til at approximere tunge ab initio beregninger af elektronstruktur mange gange hurtigere end klassiske metoder. Disse modeller er langsomt ved at revolutionere feltet for beregningskemi, da nøjagtige beregninger af molekylære egenskaber kan udføres med hidtil usete hastigheder. Dette åbner op for en lang række spændende, nye muligheder.

I denne afhandling udforsker jeg, hvordan NNs kan bruges til at accelerere Transition State (TS)-søgning og simulering af molekylære systemer på flere tidsskalaer. Afhandlingen dækker grundlæggende emner i fysik for stokastiske processer og kvantemekanik, metoder og udfordringer i forbindelse med beregning af elektronstruktur i molekyler, samt en introduktion til de NNs modeller, der er blevet brugt undervejs.

Mit Ph.D. arbejde har resulteret i tre nævneværdige videnskabelige bidrag. Det første af disse er datasættet Transition1x. Dette datasæt består af Density Functional Theory (DFT) beregninger for 10 millioner molekylære konfigurationer omkring reaktionsveje for 10 tusind forskellige reaktioner, der involverer elementerne Hydrogen, Carbon, Nitrogen og Oxygen. Disse konfigurationer er fundet ved brug af Nudged Elastic Band (NEB) metoden. Dette datasæt er en værdifuld ressource til træning af NN modeller der skal bruges i relation til kemiske reaktioner.

Mit næste videnskabelige bidrag er NeuralNEB algoritmen. I paperet hvor denne bliver præsenteret bliver NNs trænet på forskellige datasæt og evalueret på deres evne til at agere Potential Energy Surfaces (PESs) for NEB. Her ser vi, at modeller trænet på Transition1x opnår bedre resultater end tilsvarende modeller trænet på andre datasæt. Dette understreger vigtigheden af datasæt med data, der er specifikt relevant for opgaven der skal løses.

Endeligt præsenteres Implicit Transfer Operator Learning metoden. Her trænes betingede Denoising Diffusion Probabilistic Models (DDPMs) ved hjælp af et nyt data augmenterings system, hvor træningsdata i form af trajektorier fra Molecular Dynamics (MD) simuleringen augmenteres ved at sample forskellige tidsskridt under træningen. Med denne metode demonstrerer vores modeller evnen til at indfange dynamikker på en række tidsskalaer.

List of Publications

Scientific contributions included in this thesis:

Transition1x - a dataset for building generalizable reactive machine learning potentials article published in *Scientific Data* 2022

NeuralNEB - neural networks can find reaction paths fast article published in *IOP Science* 2022

Multiple Time-Resolution Surrogates for Molecular Dynamics article in proceedings for *NeurIPS* 2023

Machine Learning for Chemical Reactions extended abstract and poster presented at the *Machine Learning and the Physical Sciences Workshop at NeurIPS* 2022

Contents

Preface i

Acknowledgements iii

Summary v

Resumé vii

List of Publications ix

Contents x

1 Introduction 1

2 Statistical Physics 5
2.1 Boltzmann Distribution . . . 5
2.2 Langevin Dynamics . . . 7
2.3 Fokker Planck Equation . . . 9

3 Quantum Mechanics 13
3.1 Schrödinger Equation . . . 13
3.2 Born-Oppenheimer approximation . . . 15
3.3 Ab-inito methods . . . 18
3.4 Hartree-Fock Method . . . 19
3.5 Density Functional Theory . . . 23

4 Neural Networks 29
4.1 Neural Networks 101 . . . 29
4.2 Latent Variable Models . . . 31
4.3 Denoising Diffusion Probabilistic Models . . . 33
4.4 Accelerating Sampling with Probability Flow ODEs . . . 37
4.5 Message Passing Neural Networks . . . 38
4.6 Wrap-up . . . 41

5 ML Accelerated Transition State Search 43

5.1 Transition States 43
5.2 Transition State Search with Machine Learning 47
5.3 NeuralNEB and Transition1x 48

6 Implicit Transfer Operator Learning 53

7 Conclusion 57

Bibliography 61

A Transition1x - a dataset for building generalizable reactive machine learning potentials 71

B NeuralNEB – Neural Networks can find Reaction Paths Fast 83

C Machine Learning for Chemical Reactions 103

D Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics 113

E Mors Vejledning Til Min Afhandling 135
E.1 Transition1x 136
E.2 NeuralNEB 137
E.3 Implicit Transition Operator 138
E.4 Konlusion 138

Glossary

CINEB Climbing Image Nudged Elastic Band. 47

CLT Central Limit Theorem. 9

CNN Convolutional Neural Network. 39

DDPM Denoising Diffusion Probabilistic Model. v, vii, 3, 33, 37, 38, 54, 55, 58

DFT Density Functional Theory. v, vii, 2, 23–26, 47, 49, 51, 57, 58

DFTB Density-Functional Tight-Binding. 58

ELBO Evidence Lower Bound. 32, 35, 36

GGA Generalized Gradient Approximation. 26

GSM Growing String Method. 46, 49

ITO Implicit Transfer Operator. 54, 55, 58

KL Kullback-Leibler. 31, 32, 36

LCAO Linear Combination of Atomic Orbitals. 21

LDA Local Density Approximation. 25, 26

LST Linear Synchronous Transit. 46

MAE Mean Absolute Error. 51, 57

MD Molecular Dynamics. v, vii, 3, 55, 58

MEP Minimal Energy Path. 3, 44–49, 51

ML Machine Learning. v, vii, 2, 3, 26, 27, 33, 39, 41, 42, 47–50, 53, 57, 58

MPNN Message Passing Neural Network. 39–41

NEB Nudged Elastic Band. v, vii, 2, 3, 46–49, 51, 57, 58

NN Neural Network. v, vii, 2, 3, 26, 27, 29, 30, 35, 37–39, 47–49, 51, 57

ODE Ordinary Differential Equation. 8, 37, 38

OOD Out Of Distribution. 48

PaiNN Polarizable atom interaction Neural Network. 40, 51

PES Potential Energy Surface. v, vii, 2, 3, 8, 43–49, 51, 53, 54

QST Quadratic Synchronous Transit. 46

RMSE Root Mean Squared Error. 51, 58

SCF Self Consistent Field. 22, 23, 26

SDE Stochastic Differential Equation. 7, 37, 38

TS Transition State. v, vii, 2, 3, 44–49, 51, 53, 57

TST Transition State Theory. 43

VAE Variational Autoencoder. 32

VI Variational Inference. 32

CHAPTER 1

Introduction

All matter encountered in the natural world is composed of a variety of elements, with atoms as their fundamental units. The term *atom* derives from its Greek roots meaning *uncuttable* or *indivisible*. This concept, which dates back to ancient Greece, originated as a philosophical idea rather than a scientifically reasoned one. It suggests that if one continues to divide an element into smaller portions, eventually, one would end up with the smallest possible amount - a single atom. Today, we know that atoms can indeed be decomposed further into more fundamental subatomic particles, electrons, neutrons, and protons, which themselves are composed of quarks. Our current model of atomic structure is based on the atomic model proposed by Ernest Rutherford in 1911 [2]. He suggested that an atom's positive charge and the majority of its mass is confined within a tiny nucleus at its center, with electrons occupying a relatively larger spatial volume around this nucleus. It is the number of protons within the nucleus that decides the number of electrons and chemical properties of an element. Rutherford's early atomic model proposed that electrons orbit the nucleus as planets around the sun, however, a more accurate description of the electron is rather as a vibrating cloud or standing wave in 3-dimensions. These waves can take many forms which are referred to as orbitals, and each electron occupies a uniquely shaped orbital within an energy shell which are filled in order of increasing energy. Electrons in the outer shells are called *valence* electrons and the shape of the outer shell, referred to as the electron cloud, is what determines how it can interact with other atoms. Electron clouds around different nuclei can form bonds, linking atoms together resulting in the formation of larger structures called molecules. Atoms on their own have many properties such as electron affinities, ionization energies etc., however, combining atoms into molecules yield properties that are certainly greater than the sum of its parts. For instance the remarkable heat capacity of water or its ability to dissolve many substances, or the extremely complex functions expressed by some proteins are results of properties emerging when atoms form molecules. Our world is built of molecules and governed by their properties. Understanding them lets us predict and manipulate the behavior of matter, leading to the development of new materials, drugs, energy sources, etc. However, capturing the full complexity of molecular behavior, is extremely challenging. While the universe 'simulates' molecular systems effortlessly at an unfathomable scale, both in terms of speed and incomprehensible number of particles and interactions, our human-made computer models rely on formidable calculations to replicate even the smallest systems of sin-

gle molecules.

In quantum mechanics, all matter exhibits a certain particle-wave duality. This means that anything from electrons and protons to molecules and footballs behave like 3-dimensional vibrating waves and are not just confined to exact locations in space. Perhaps one of the most famous equations of all time, the Schrödinger equation, describes how these waves evolve in the presence of a potential. This potential can arise, for example, as the result of electrostatic interaction between charged particles. The Schrödinger equation is extremely difficult to work with, and any attempt at solving it for molecular systems relies on a range of approximations. The most important of these arises from the fortunate fact that nuclei are much more massive than electrons. This means that their wave-like nature is less pronounced, allowing us to approximate them as extremely slow moving particles and solely focus on the electronic structure. Most interesting properties of these systems, such as internal forces, energy stored in bonds, etc., as a function of nuclear positions can be derived from electronic structure calculations. This has led to a variety of approaches for solving these systems including Density Functional Theory (DFT) [3, 4] methods which deals with an overall electron-density, or wave function methods [5, 6] that attempt to solve for molecular orbitals of individual electrons [7, 8]. Each method comes with its own set of trade-offs, and generally, increased accuracy comes at the expense of computational resources. This trade-off is a bottleneck preventing accurate large scale simulation or application of algorithms requiring evaluation of many configurations.

In recent years the Machine Learning (ML) field has seen a renaissance of the Neural Network (NN) model. These models are incredibly versatile and easy to fit to large datasets, and they have been showing promising results as surrogates for traditional computational quantum chemistry methods [9–15]. Though expensive to train, they are quick to evaluate on new data. For example, they can predict energies of molecular configurations, forces acting atoms, or more complex properties like polarizability with a computational cost which is a fraction of traditional methods, while retaining high accuracy. This acceleration opens up various new avenues for exploring chemical space, optimizing molecular structures for desired properties, or simulate systems on timescales or of sizes that were previously infeasible.

In this thesis we explore some of the possibilities and challenges associated with applying NN models in various molecular contexts. During the first two years of my Ph.D. I focused on training NNs to accurately predict Potential Energy Surfaces (PESs) of molecules such that these could be applied in Transition State (TS)-search, specifically in the Nudged Elastic Band (NEB) [16, 17] algorithm. TSs are important when studying chemical reactions. They represent turning points at which atoms involved in the reaction are more likely to proceed to the product rather than revert to the reactant configuration. Moreover, the barrier height - or energy difference between reactant and TS configurations plays an important role in describing the rate of a reaction [18]. The electronic structure at the TS is particularly challenging to

describe quantum mechanically, as at this point, some chemical bonds are partially formed, while others are partially broken. The TS is a saddle-point on the PES and it represents the highest point on the Minimal Energy Path (MEP), or, the minimum of energy required for the reaction to take place. However, the PES is a $3N$ dimensional surface, describing potential energy as a function of x, y, z coordinates of N atoms. Locating saddle points on such high dimensional, complex surfaces requires rather sophisticated methods evaluating many intermediate configurations [16, 19–21]. Carrying out these calculations at large scale becomes prohibitively expensive due to the computational cost of evaluation of the PES using classical quantum mechanical computational methods. This research explored the potential for accelerating such methods through the use of ML models which led to two published papers. The first paper introduces the *Transition1x* [22] dataset consisting of molecular configurations along reaction pathways for 10,000 reactions, which provides NNs with relevant training data for this type of application. The second paper presents the *NeuralNEB* [23] method, which employs NN-potentials to guide NEB-algorithm in finding TSs.

During the last part of my Ph.D. I shifted my focus from identifying TSs with NNs to study the dynamics and time-dependent behavior of complex systems. In the first project we studied reaction pathways from a rather idealized point of view, identifying exact energies and TSs associated with specific reactions. We did this without considering interactions with the surrounding medium and the effects of thermal fluctuations. But molecules are seldom alone, they interacting with its surroundings in a complex manner with energy constantly, erratically, and randomly transferred between various degrees of freedom such as kinetic, vibrational, and rotational energies, which makes their behavior difficult to predict. A practical approximation for simulating such systems is through Langevin dynamics [24]. Here the internal forces within the molecule are treated separately from the random forces arising from its interaction with the environment, simplifying the complexity of simulation. However, due to their stochastic nature, such systems can evolve in a variety of ways, and the challenge becomes modeling the evolution of the probability distribution of the systems' state-space, rather than simulating specific trajectories. In the context of Langevin dynamics, this evolution is described by the Fokker-Planck [25] equation, and trajectories simulated using this Langevin dynamics can be understood as realizations of the integrated Fokker-Planck equation. Owing to its linearity, the Fokker-Planck equation can, in theory, be integrated in closed form corresponding to the action of the Transfer Operator [26]. Frames at various time-intervals on the trajectory can be viewed as samples from the conditional probability distribution, conditioned on the first frame and integration time. We train models on various systems to implicitly learn the transfer operator with Denoising Diffusion Probabilistic Models (DDPMs) in the *Implicit Transfer Operator Learning* approach [27]. These models are trained to simulate various systems including Langevin dynamics in the simple 2-D Müller Brown potential and Molecular Dynamics (MD) simulations of Alanine-dipeptide [28] and of a set of fast folding proteins. We use these models to simulate trajectories and validate them by comparing transition-densities, equilibrium distributions, and

various observables such as free energy differences and mean-first passage times for protein folding.

CHAPTER 2

Statistical Physics

The first law of thermodynamics states that energy cannot be created or destroyed, it can only change form. Energy in a closed system of molecules causes the molecules to move and interact in a chaotic and unpredictable way. The kinetic energy of one molecule might, in an instance, change into rotational energy of another upon collision, just to turn into vibrational energy in a chemical bond a split-second later. In that way, energy travels randomly between the degrees of freedom in the system, but the total energy always stays the same.

While the chaotic nature of molecules in a system makes it impossible to predict how they will behave individually, the sheer number of molecules allows for robust statistical predictions about the system as a whole. This is the essence of statistical physics, which is foundational for our understanding of a wide range of physical processes such as chemical reactions. In this chapter, we will first explore the Boltzmann distribution and its role as equilibrium distribution in relation to the second law of thermodynamics. Next we will describe Langevin Dynamics, a framework for modeling the complex interaction between a molecule and its surroundings without having to explicitly model the surrounding medium. Then we will cover the Fokker-Planck equation which describes the statistical evolution of systems simulated with Langevin dynamics.

2.1 Boltzmann Distribution

Each fully specified 'arrangement' of energy in the system is referred to as a *microstate*, and as the system evolves it randomly, and without preference, transitions between microstates. A *macrostate*, on the other hand, offers a macroscopic description of the system. It can be a statement about pressure or temperature or a general distribution of energy. The log-multiplicity of a macrostate in terms of microstates is called *entropy*, and as the system evolves, randomly traveling between microstates, it will statistically tend towards macrostates with higher entropy. The second law of thermodynamics captures this by stating that, for an isolated system, the entropy tends to increase over time, eventually reaching a maximum value at equilibrium. The equilibrium state is described by the Boltzmann distribution. To find the Boltzmann distribution, let us consider the problem of arranging N molecules in a closed

system over m possible states, finding the distribution with the highest multiplicity [29]. Here, a macrostate is described by the number of molecules n_i in each state i, with $N = \sum_{i=1}^{m} n_i$. This is a combinatorics problem, and the multiplicity of a given macrostate is

$$C = \frac{N!}{\prod_{i=1}^{m} n_i!}, \tag{2.1}$$

or its log-multiplicity

$$\ln C = \ln N! - \sum_{i=1}^{m} \ln n!. \tag{2.2}$$

$\ln N!$ can be approximated with the integral

$$\ln N! = \sum_{n=0}^{N} \ln n \simeq \int_0^N \ln n dn = [n \ln n - n]_0^N = N \ln N - N, \tag{2.3}$$

which converges towards the true value for large N. Using this approximation (2.2) is rewritten as

$$\ln C = N \ln N - \sum_{i=1}^{m} n_i \ln n_i \tag{2.4}$$

$$= N \ln N - N \sum_{i=1}^{m} p_i \ln p_i - N \ln N \sum_{i=1}^{m} p_i \tag{2.5}$$

$$= -N \sum_{i=1}^{m} p_i \ln p_i, \tag{2.6}$$

where p_i is the probability that a given molecule is in the i-th state. Here, between the first and second line, we have used the fact that $n_i = p_i N$, and in the third line we arrive at the log-multiplicity of the system which we recognize as the entropy of the entire system. To find the equilibrium configuration of our entire system, we maximize the entropy of the probability distribution of a single molecule

$$S = -\sum_{i=1}^{m} p_i \ln p_i, \tag{2.7}$$

Under the constraint that

$$\sum_{i=1}^{m} p_i = 1 \quad \text{and} \quad \sum_{i=1}^{m} p_i E_i = E, \tag{2.8}$$

where E_i is the energy of a molecule in the i-th state and E is the average energy per molecule in the system. These constraints ensure that the distribution is normalized, and that the system is has the correct total energy. We use the method of Lagrangian

multipliers to minimize the negative entropy along the contour lines where these constraints are fulfilled;

$$F = \sum_{i=1}^{m} p_i \ln p_i + \beta \left[\sum_{i=1}^{m} p_i E_i - E \right] + \alpha \left[\sum_{i=1}^{m} p_i - 1 \right]. \tag{2.9}$$

We differentiate with respect to p_i and find the maximum

$$\frac{\partial F}{\partial p_i} = \ln p_i + 1 + \beta E_i + \alpha = 0. \tag{2.10}$$

Solving for p_i yields

$$p_i = e^{-\beta E_i} e^{-\alpha - 1}, \tag{2.11}$$

where β is inverse temperature, and

$$e^{\alpha+1} = Z \tag{2.12}$$

is the partition function which normalizes the distribution. Thus we arrive at the well known, omnipresent Boltzmann distribution

$$p_i = \frac{e^{-\beta E_i}}{Z}. \tag{2.13}$$

Or in terms of an energy function $E(x)$

$$p(x) = \frac{e^{-\beta E(x)}}{Z} \tag{2.14}$$

This is an extremely important result in statistical mechanics with a wide range of applications in various scientific disciplines. It describes the equilibrium distribution that a system will tend towards, and provides a direct link between probabilities and energies of states.

2.2 Langevin Dynamics

The Langevin equation was formulated by French physicist Paul Langevin [24] in the early 20th century to describe the random motion of particles suspended in water. However, its applicability extends far beyond this original context and has been used to model a wide range of stochastic processes [30–34]. The Langevin equation is a Stochastic Differential Equation (SDE) that incorporates a stochastic term, a deterministic drift term, and a damping term,

$$m\ddot{x} = R(t) - \eta\dot{x} + F(x). \tag{2.15}$$

Here, m is the mass of the particle, x is its position, η is the drag coefficient, and dots represent time derivatives $\dot{x} = \frac{dx}{dt}$. $F(x)$ is the force induced by the gradient of

the Potential Energy Surface (PES), $U(x)$, on which the particle is located, denoted as $F(x) = -\nabla U(x)$. We call it free Brownian motion when $F(x) = 0$, or the PES is constant. The stochastic term $R(t)$ represents thermal fluctuations e.g. random forces due the bombardment of water molecules in the surrounding medium. $R(t)$ is Gaussian distributed noise with

$$\langle R(t)R(t')\rangle = 2D\delta(t - t') \tag{2.16}$$

meaning that $R(t)$ at time t is independent from past and future random forces, and that the force at time t has a variance of $2D$, with D being the *diffusion coefficient* describing the magnitude of the fluctuations. If the particle is moving, the bombardment from the medium will tend to come from the direction opposite to the particle's movement resulting in a drag force. This effect is captured in the Langevin equation as the damping term, $-\eta\dot{x}$. Since the drag and the diffusion coefficients originate from the same bombarding effect they are related. This is called the Einstein-Smoluchowski [35, 36] relation $D = k_B T/\eta$, where k_B is the Boltzmann constant and T is the temperature of the system. It is important because it relates the microscopic movement of a single particle to macroscopic properties of the system such as friction and temperature. By ignoring the drift term the expectation value of the Langevin equation reduces, to an Ordinary Differential Equation (ODE)

$$m\langle\ddot{x}\rangle = -\eta\langle\dot{x}\rangle \tag{2.17}$$

since $\langle R(t)\rangle = 0$. This is easy to solve equation is easy to solve:

$$\langle\dot{x}\rangle = \dot{x}_0 e^{-\eta t/m}. \tag{2.18}$$

This equation describes how the expected velocity of the particle, $\langle\dot{x}\rangle$, changes over time. In the absence of an external force, the velocity decays exponentially with a characteristic timescale of m/η. If our timescale of interest is much longer than this, the inertia of the particle, $m\ddot{x}$, can be ignored leading to the *overdamped* Langevin equation;

$$\eta\dot{x} = F(x) + R(t). \tag{2.19}$$

This equation effectively describes the motion of a particle that experiences a strong damping due to its interaction with the surrounding medium and is influenced by a deterministic force $F(x)$ and stochastic forces $R(t)$. It provides a simplification that can help study stochastic systems more efficiently, without the need to simulate high-frequency motions that do not contribute significantly to the properties of interest. We absorb $1/\eta$ into $F(x)$ and $R(x)$ to obtain an expression for the dynamics, which can be integrated to evolve the system in time. The Euler-Maruyama [37] method is one of the simplest methods for approximating such integrals;

$$\Delta x = \int_{t_0}^{t_0+\Delta t} [F(x) + R(t)]\, dt \simeq \Delta t F(x) + \Delta w, \tag{2.20}$$

where $\Delta t F(x)$ is the contribution from the deterministic part of the integral, equivalent to Euler integration, and Δw is the stochastic part of the integral with the properties;

$$\Delta w = \int_{t_0}^{t_0+\Delta t} R(t)dt \tag{2.21}$$

$$\langle \Delta w \rangle = 0 \tag{2.22}$$

And, given (2.16)

$$\langle \Delta w^2 \rangle = \int_{t_0}^{t_0+\Delta t}\int_{t_0}^{t_0+\Delta t} \langle R(t)R(t')\rangle dtdt' = 2D\int_{t_0}^{t_0+\Delta t}\int_{t_0}^{t_0+\Delta t} \delta(t-t')dtdt' = 2D\Delta t \tag{2.23}$$

The Central Limit Theorem (CLT) [38] states, that the sum of a large number of independent and identically distributed random variables converges towards a normal distribution, so Δw is normally distributed given that it is an infinite sum of infinitesimal noise

$$\Delta w = \epsilon\sqrt{2D\Delta t} \quad \text{and} \quad \epsilon \sim \mathcal{N}(0, I) \tag{2.24}$$

where ϵ is a standard normal distributed random variable and we can simulate

$$\Delta x = \Delta t F(x) + \Delta w \tag{2.25}$$

This provides an easy framework for integrating dynamics of systems that are influenced by deterministic and stochastic forces.

2.3 Fokker Planck Equation

While the Langevin equation provides a framework for understanding the stochastic motion of individual particles, its perspective is too narrow to draw general conclusions about the system at large. Ensembles offer a more meaningful description of the system's behavior, and often we are more interested in the time-evolution of the distribution $p(x,t)$ of x at time t governed by Langevin dynamics than particular trajectories. This is described by the Fokker Planck equation [25]. Our strategy for deriving it is to compare two expressions for the time derivative of the expectation value of an arbitrary, time-independent function $\langle f(x)\rangle$, and extract the Fokker-Planck equation from this [39]. Since $f(x)$ is time-independent, we can express its time-derivative as

$$\frac{d\langle f(x)\rangle}{dt} = \int_{-\infty}^{\infty} f(x)\frac{\partial}{\partial t}p(x,t)dx, \tag{2.26}$$

moving the time-derivative inside the integral acting on $p(x,t)$.

An alternative expression for the same quantity can be derived by instead considering the time propagation of the system. We have

$$\frac{d\langle f(x)\rangle}{dt} = \lim_{\Delta t\to 0} \frac{\langle f(x+\Delta x)\rangle - \langle f(x)\rangle}{\Delta t}, \tag{2.27}$$

where Δx is caused by propagating the system Δt forward in time. To relate these two equations we start by expanding $\langle f(x)\rangle$ around x;

$$\langle f(x+\Delta x)\rangle = \langle f(x)\rangle + \left\langle \frac{df(x)}{dx}\Delta x \right\rangle + \left\langle \frac{d^2 f(x)}{dx^2}\frac{\Delta x^2}{2} \right\rangle + O(\Delta x^3), \tag{2.28}$$

and by substituting (2.25), the Langevin description of Δx, we get

$$\langle f(x+\Delta x)\rangle = \langle f(x)\rangle + \left\langle \frac{df(x)}{dx}(F(x)\Delta t + \Delta w) \right\rangle + \left\langle \frac{d^2 f(x)}{dx^2}\frac{(F(x)\Delta t + \Delta w)^2}{2} \right\rangle + O(\Delta t \Delta w). \tag{2.29}$$

Since the expectation value for any function $g(x)$ involving

$$\langle g(x)\Delta w\rangle = 0 \qquad \langle g(x)\Delta w^2\rangle = \langle g(x)\rangle 2D \qquad \langle g(x)\Delta t\rangle = \langle g(x)\rangle \Delta t, \tag{2.30}$$

We can simplify (2.29) to

$$\langle f(x+\Delta x)\rangle = \langle f(x)\rangle + \left\langle \frac{df(x)}{dx}F(x) \right\rangle \Delta t + D\left\langle \frac{d^2 f(x)}{dx^2} \right\rangle \Delta t + O(\Delta t^2), \tag{2.31}$$

and by plugging into eq. 2.27 we get an expression for the time derivative:

$$\frac{d\langle f(x)\rangle}{dt} = \left\langle \frac{df(x)}{dx}F(x) \right\rangle + D\left\langle \frac{d^2 f(x)}{dx^2} \right\rangle. \tag{2.32}$$

Our goal now is to rewrite this in terms of $p(x,t)$, so that we can compare it with (2.26) and extract the Fokker-Planck equation. We can rewrite both terms on RHS through partial integration. We have

$$\left\langle \frac{df(x)}{dx}F(x) \right\rangle = \int_{-\infty}^{\infty} \frac{df(x)}{dx}F(x)p(x,t)dx \tag{2.33}$$

$$= f(x)p(x,t)F(x)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} f(x)\frac{\partial}{\partial x}\left[F(x)p(x,t)\right]dx \tag{2.34}$$

where the first term vanishes because the probability density function, $p(x,t) \to 0$ when $x \to \pm\infty$. We can evaluate the second term on RHS of (2.32) as well, this time using partial integration twice and noticing that $\frac{\partial x}{\partial p}(x,t) \to 0$ when $x \to \pm\infty$

$$\left\langle \frac{d^2 f(x)}{dx^2} \right\rangle = \int_{-\infty}^{\infty} \frac{df(x)^2}{dx}p(x,t)dx \tag{2.35}$$

$$= \frac{df(x)}{dx}p(x,t)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \frac{df(x)}{dx}\frac{\partial p(x,t)}{\partial x}dx \tag{2.36}$$

$$= -f(x)\frac{\partial p(x,t)}{\partial x}\Big|_{-\infty}^{\infty} + \int_{-\infty}^{\infty} f(x)\frac{\partial^2 p(x,t)}{\partial x^2}dx \tag{2.37}$$

If we plug eq. 2.34 and eq. 2.37 into eq 2.32 we get:

$$\int_{-\infty}^{\infty} f(x)\frac{\partial}{\partial t}p(x,t)dx = \int_{-\infty}^{\infty} f(x)\frac{\partial}{\partial x}\left[D\frac{\partial p(x,t)}{\partial x} - F(x)p(x,t)\right]dx \tag{2.38}$$

Since this is true for any function $f(x)$, we extract the time evolution of $p(x,t)$

$$\frac{\partial}{\partial t}p(x,t) = \frac{\partial}{\partial x}\left[D\frac{\partial p(x,t)}{\partial x} - F(x)p(x,t)\right]. \tag{2.39}$$

This is the Fokker Planck equation, a fundamental tool for understanding stochastic systems. Remarkably it connects the microscopic behavior of single trajectories governed by deterministic and random forces with macroscopic statistical properties of the system captured by $p(x,t)$. We shall see it multiple times in various contexts in the following chapters.

CHAPTER 3

Quantum Mechanics

Quantum Mechanics deals with the behavior of particles at atomic and subatomic scales. At these scales, our natural intuition of physics breaks down as particles exhibit fundamentally different properties, acting simultaneously as waves and particles. Quantum effects govern molecules and their electrons, and any attempt at describing molecules faithfully must start with an understanding of the underlying quantum mechanics.

The Schrödinger equation dictates the temporal evolution of quantum systems such as molecules and it is intimately connected with their energetics. Therefore, it is one of the key equations to solve in computational quantum chemistry. However, it is notoriously hard to work with, and much research has been focusing on approximating its solutions for molecular systems. In this chapter we will briefly describe fundamental quantum mechanics and explore some of the methods that have been developed over the years for dealing with the quantum mechanics governing molecular systems.

3.1 Schrödinger Equation

The nature of light has been a topic of scientific and philosophical debate since the ancient Greeks. At the beginning of the 20th century, the wave-like nature of light was both rigorously described by Maxwell's equations [40, 41] and a well-founded experimental fact. However, an unexpected observation in a simple experiment showcased the photoelectric effect. This challenged the wave-like picture of light, indicating that light could also exhibit particle-like properties. Essentially, it was observed that upon shining light on a cathode in a vacuum, electrons would be emitted, and they would all, regardless of the intensity of the light, carry kinetic energy given by;

$$E_K = h\nu - \phi, \tag{3.1}$$

where h is the Planck constant, the conversion factor between wave frequency and energy, ν is the frequency of the light and ϕ is the required energy to kick off an electron from the cathode. Crucially, this equation does not depend on the intensity of the light, only its wavelength. It was found that increasing the light intensity would increase the number of electrons emitted, but not affect the kinetic energy of each. This suggested that each electron was emitted because of a collision with a

single photon posing a conflicting view with the well established notion of the wave-like nature of light. This gave rise to Einstein's revolutionising idea of particle-wave duality [42]. The question then arose: If light could exhibit this duality, could the same be true for electrons and matter? This question was posed by Louis de Broglie in his Ph.D. thesis [43] in 1924, where he hypothesized that particles like electrons could also exhibit wave-like properties. Erwin Schrödinger then took it upon himself to find a wave equation of matter, culminating in the formulation of the famous Schrödinger equation[44]. A tentative derivation of the Schrödinger equation for the free particle, may be obtained by considering its classical energy and substituting the expressions for momentum and energy by their wave analogs. These are expressed through the de Broglie and Einstein postulates, $p = \hbar k$ and $E = \hbar\omega$, respectively, where k is the wave-number, ω is angular frequency and $\hbar = h/2\pi$. The classical energy of a particle is the sum of its kinetic and potential energy

$$E = \frac{p^2}{2m} + V. \tag{3.2}$$

Or, if we substitute in the de Broglie and Einstein postulates,

$$\hbar\omega = \frac{\hbar^2 k^2}{2m} + V. \tag{3.3}$$

There is no force acting on the free particle since it is in a constant potential, so its momentum, and hence wavelength is constant. Generally, a wave can be represented by the equation

$$\Psi(x,t) = A\cos(kx - \omega t) + B\sin(kx - \omega t), \tag{3.4}$$

with $\Psi(x,t)$ denoting the wave-function in terms of x and t. This suggests that, in order to get the k and ω from equation (3.3), we should differentiate twice with respect to position on LHS and once with respect to time on the RHS,

$$\alpha\frac{\partial^2}{\partial x^2}\Psi(x,t) + V\Psi(x,t) = \beta\frac{\partial}{\partial t}\Psi(x,t) \tag{3.5}$$

$$(-\alpha k^2 + V)\left[A\cos(kx - \omega t) + B\sin(kx - \omega t)\right] = -\frac{\beta\omega}{B}\left[B^2\cos(kx - \omega t) - AB\sin(kx - \omega t)\right], \tag{3.6}$$

where α and β provides flexibility to the solution. By setting $A = 1$ and $B = i$, the equation simplifies to

$$(\alpha k^2 + V) = -i\beta\omega. \tag{3.7}$$

Comparing this with equation (3.3), relating the total energy of the wave to its classical counterpart it is clear that $\alpha = -\frac{\hbar^2}{2m}$ and $\beta = i\hbar$. Plotting these values into eq. (3.6) yields Schrödinger's famous equation:

$$-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\Psi(x,t) + V\Psi(x,t) = i\hbar\frac{\partial}{\partial t}\Psi(x,t). \tag{3.8}$$

Once a potential $V(x,t)$ is specified, the Schrödinger equation allows us to determine wave-function $\Psi(x,t)$ that governs the quantum system and how it will evolve. The exact meaning of the wave-function has been a topic of debate ever since. Max Born proposed the practical statistical interpretation that the squared magnitude of the wave-function, $|\Psi(x,t)|^2$, represents the probability density of finding the particle at position x at time t[45]. The equation has only been solved analytically for a precious few theoretical potentials, where the only real-world problem is a single electron around a positive point-charge. This is a central force problem, where the force acting on the electron arises from the Coloumb interaction between the point-charge and the negatively charged electron. If more electrons are introduced, the problem becomes more complex as here, while a central force approximation can still be used, the electrostatic repulsion between the electron clouds must be taken into account. When dealing with molecules, the complexity escalates even further, as tools associated with central force problems, such as spherical symmetry and angular momentum, are no longer applicable.

3.2 Born-Oppenheimer approximation

The motion of nuclei and electrons is inherently connected in molecular systems. However, because nuclei are much heavier than electrons, their motion is so slow that they effectively can be considered fixed in relation to the motion of electrons. This is the Born-Oppenheimer approximation[46]. The nuclei in a molecule vibrate due to the spring-like nature of molecular bonds. The total energy of this vibration can be represented as the sum of kinetic and potential energy associated with the vibration:

$$E_{\mathrm{n}} = \frac{p_{\mathrm{n}}^2}{2m_{\mathrm{n}}} + \frac{1}{2}k_n x_{\mathrm{n}}^2, \tag{3.9}$$

where $x_{\mathrm{n}}, p_{\mathrm{n}}, m_{\mathrm{n}}$ and E_{n} are position, momentum, mass and total energy of a nucleus associated with the vibration. k_{n} is the spring constant, and the restoring force arises from the gradient of energy stored in the electron clouds and Coulomb forces between the nuclei. Consider a small molecule characterized by the length scale s, the energy in its electron cloud is on the scale of

$$E_{\mathrm{e}} = \frac{p_{\mathrm{e}}^2}{2m_{\mathrm{e}}} = \frac{\hbar^2}{2m_{\mathrm{e}}s^2}. \tag{3.10}$$

The spring constant has units of energy per length squared. Approximating k_n in terms of the relevant length scale and kinetic energy of the electron cloud yields

$$k_{\mathrm{n}} \simeq \frac{\hbar^2}{2m_{\mathrm{e}}s^4}. \tag{3.11}$$

We can plug into this expression into the formula for angular frequency allowing for a comparison between the timescale of the oscillations in the electron cloud with that

of the oscillations of nuclei

$$\omega_{\mathrm{n}} = \sqrt{\frac{k_{\mathrm{n}}}{m_{\mathrm{n}}}} = \sqrt{\frac{m_{\mathrm{e}}}{m_{\mathrm{n}}}}\sqrt{\frac{\hbar^2}{2m_e^2 s^4}} = \sqrt{\frac{m_{\mathrm{e}}}{m_{\mathrm{n}}}}\frac{E_{\mathrm{e}}}{\hbar} = \sqrt{\frac{m_{\mathrm{e}}}{m_{\mathrm{n}}}}\omega_{\mathrm{e}}, \tag{3.12}$$

Where ω_{n} and ω_{e} are the angular frequencies associated with the oscillations of the nuclei and the kinetic energy in the electron cloud, respectively. This rough calculation shows that the oscillations of the nuclei are extremely slow compared to the oscillations in the electron cloud[47]. This result motivates the Born-Oppenheimer approximation, in which the external potential caused by the nuclei in a molecule is treated as time-independent. In a way it is lucky that there is such significant difference in the mass of these fundamental particles as the disparity allows the crucial approximation that the nuclei in molecules are static. Without it, it would be very difficult to do computational quantum chemistry.

3.2.1 Time-Independent Schrödinger equation

Building on the Born-Oppenheimer approximation, we are motivated to reformulate the Schrödinger equation with a time-independent potential. Since we consider the potential, which is caused by the nuclei in the molecule fixed, we expect that the solutions should also be time-independent stationary states. We can simplify the Schrodinger equation using separation of variables[44], expressing the wave-function as a product of spatial and temporal components $\Psi(x,t) = \psi(x)\phi(t)$ and substitute them into the Schrödinger equation:

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x)}{\partial x^2}\phi(t) + V\psi(x)\phi(t) = i\hbar\frac{\partial\phi(t)}{\partial t}\psi(x), \tag{3.13}$$

which can be rewritten as

$$-\frac{1}{\psi(x)}\frac{\hbar^2}{2m}\frac{\partial^2\psi(x)}{\partial x^2} + V = \frac{1}{\phi(t)}i\hbar\frac{\partial\phi(t)}{\partial t}. \tag{3.14}$$

Since this holds true for all x and t, both sides must be constant. This constant is, as hinted by the potential energy term on the LHS, the total energy of the system. Thus we can express the Schrödinger equation in a time-independent potential as;

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x)}{\partial x^2} + V\psi(x) = E\psi(x). \tag{3.15}$$

This is the time-independent Schrödinger equation and it plays a central role in computational quantum chemistry, serving as the principal tool for studying electronic structure in molecules. The time-independent Schrödinger equation hinges on the Born-Oppenheimer approximation and it is extremely important in quantum chemistry, as it allows us to simplify the study of molecules since we do not need to take the changing potential induced by moving nuclei into account.

3.2.1.1 Identical particles and Pauli's Exclusion principle

In systems with multiple particles, the wave-function must be extended to account for each particle as well as interactions between them. Let us briefly step aside and examine Pauli's exclusion principle[48] - a phenomenon that arises when modelling the multi-particle wave-function with identical particles. Pauli's exclusion principle has no analogy in classical physics, and it arises due to the fact that fundamental particles are completely identical and indistinguishable in every way. Let us investigate a system with two identical particles described by the wave-function $\Psi(r_1, r_2)$, where r_1 and r_2 are the positions of the first and second particle. Because the particles are indistinguishable, no physical measurement can tell them apart or detect the interchange of the two particles. This implies that the probabilistic structure of the wave-function must be invariant under permutation. Mathematically, this is expressed as

$$|\Psi(r_1, r_2)|^2 = |\Psi(r_2, r_1)|^2 \tag{3.16}$$

The wave-function $\Psi(r_1, r_2)$ can fulfill this requirement in one of two ways, depending on the nature of the particles involved:

1. **Symmetric wave-function**: $\Psi(r_1, r_2) = \Psi(r_2, r_1)$. In this case, the wave-function remains unchanged when the particles are exchanged. This property is characteristic of the family of particles called *bosons* such as photons.
2. **Antisymmetric wave-function**: $\Psi(r_1, r_2) = -\Psi(r_2, r_1)$. In this case, the wave-function changes sign when the particles are exchanged. This property is characteristic of the family of particles called *fermions* such as protons, neutrons and electrons.

Since the particles are indistinguishable from one another, the wave-functions are indistinguishable too, and it is meaningless to treat them as separate functions. We must therefore write the wave-function as a normalized linear combination of the two:

$$\Psi = \frac{1}{\sqrt{2}} \left[\Psi(r_1, r_2) \pm \Psi(r_2, r_1)\right]. \tag{3.17}$$

Where the $\pm$ sign depends on whether the wave-function is symmetric $(+)$ or anti-symmetric $(-)$. Let us now consider the special case where the wave-function is the product of two identical states $\Psi_{\text{id}}(r_1, r_2) = \psi(r_1)\psi(r_2)$. By plugging this into the anti-symmetric wave-equation, we find:

$$\Psi_{\text{asym}} = \frac{1}{\sqrt{2}} \left[\psi(r_1)\psi(r_2) - \psi(r_2)\psi(r_1)\right] = 0. \tag{3.18}$$

which implies that $|\Psi(r_1, r_2)|^2 = 0$, or, that the probability of finding two fermions in the same state is zero. This is Pauli's exclusion principle. Pauli's exclusion principle is a quantum mechanical effect, and it is as important as the Coulomb interaction when treating quantum systems. Any attempt at constructing wave-function solutions for quantum systems must take these fundamental constraints dictating the electronic structure into account.

3.3 Ab-inito methods

In ab initio methods the goal is to solve the Schrödinger equation from first principles. In this section we will dive deeper into some of these methods. It is an extremely difficult problem with most of its complexity arising from the intricacies of multiple interacting electrons, and the methods shown here have been developed over many years. Let us start by writing the Hamiltonian of the Schrödinger equation in a suggestive way for how we are going to approach this:

$$\hat{H} = \hat{T}_{\mathrm{e}} + \hat{T}_{\mathrm{n}} + \hat{V}_{\mathrm{ee}} + \hat{V}_{\mathrm{nn}} + \hat{V}_{\mathrm{ne}}, \tag{3.19}$$

where $\hat{T}_{\mathrm{e}}$ and $\hat{T}_{\mathrm{n}}$ are the kinetic energy operators of electrons and nuclei, respectively, and $\hat{V}_{\mathrm{ee}}$, $\hat{V}_{\mathrm{nn}}$ and $\hat{V}_{\mathrm{ne}}$ are the potential energy terms corresponding to electron-electron, nucleus-nucleus, and electron-nucleus interactions. Because of the Born-Oppenheimer approximation, we can treat the potential caused by the nuclei as time-independent, and we are primarily interested in solving the terms that include the electronic structure leading to the electronic Hamiltonian:

$$\hat{H}_{\mathrm{e}} = \hat{T}_{\mathrm{e}} + \hat{V}_{\mathrm{ee}} + \hat{V}_{\mathrm{ne}}, \tag{3.20}$$

Now, the task becomes to solve the time-independent Schrödinger equation for the many-electron wave-function.

$$\hat{H}_{\mathrm{e}}|\psi_{\mathrm{e}}\rangle = E_{\mathrm{e}}|\psi_{\mathrm{e}}\rangle, \tag{3.21}$$

where $|\psi_{\mathrm{e}}\rangle$ is the state vector in Hilbert space describing the multi-electron wave-function. While we can set up the equation for any molecular system, solving it reaches far beyond what can be achieved by scratching the good old Schrödinger equation on a blackboard and algebraic finesse. The primary difficulty lies in the electron-electron interaction. Solving the Schrödinger equation for multiple electrons is a many body problem. This has not even been solved analytically in classical mechanics, where particles have well-defined positions. In quantum mechanics we face a significantly more complex variant of the problem since particles are not localized at single points but are instead described by their wave-functions spanning all of 3D space. Interactions between particles have to be integrated accordingly, and as such the problem scales with $\mathcal{O}(d^{3N})$ as each new particle added to the system introduces an additional three dimensions to the state space. Even if we naively discretized each dimensions with 10 grid points for a numerical solution, it would require an astronomical $10^{3\cdot 24} = 10^{72}$ grid points to represent electrons of the state space for a *single* Iron atom, which has 24 electrons. For reference, there are about 10^{56} atoms in the solar system. The biggest challenges in tackling any quantum mechanical system is dealing with the extremely high-dimensional spaces required to describe the system, sometimes referred to as the *quantum nightmare*. Any attempt to solve the Schrödinger equation must first address this issue.

3.4 Hartree-Fock Method

One of the first approaches to approximate solutions to the Schrödinger equation for molecules was the Hartree-Fock Method [5–7].

3.4.1 Hartree-Approximation

The many-electron wave-function contains such vast amounts of information that it is futile to solve it for any but the simplest systems. The complexity arises from the pairwise interactions between electrons over all space. One way to simplify the problem of the many-electron wave-function is the Hartree approximation [8]. This approach approximates the many-body wave-function as a simple product of independent single-electron wave-functions. We call these single-electron wave-functions molecular orbitals, denoted $\phi_i(r)$, as they represent the available orbitals for electrons, possibly spanning the entire molecule.

$$\Psi(r_1, \ldots, r_n) = \prod_{i=1}^{n} \phi_i(r_i) \tag{3.22}$$

The idea is to solve the Schrödinger equation using this somewhat naive wave-function. In this case, interactions between electrons are reduced to interactions with an average electron-field. However, the Hartree approximation does not capture the antisymmetric nature of the electron wave-function, which is essential since any solution must adhere to Pauli's exclusion principle.

3.4.2 Slater determinants

The rules governing the behavior of identical particles go beyond simple two-particle systems as described in section 3.2.1.1. When dealing with systems of multiple electrons, any exchange of particles must result in a change of sign: $\Psi(.., r_i, .., r_j, ..) = -\Psi(.., r_j, .., r_i, ..)$. The full anti-symmetric n-electron wave-function for a system with n electrons is a linear combination of all possible permutations of all particle indices, and their corresponding sign. In practice, Slater determinants [49] provide a useful mathematical shorthand;

$$\Psi_{\text{Slater}}(r_1, r_2, \ldots, r_n) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \phi_1(r_1) & \phi_1(r_2) & \ldots & \phi_1(r_n) \\ \phi_2(r_1) & \phi_2(r_2) & \ldots & \phi_2(r_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n(r_1) & \phi_n(r_2) & \ldots & \phi_n(r_n) \end{vmatrix}. \tag{3.23}$$

If there are identical states in the system, i.e., $\phi_i(r) = \phi_j(r)$, rows i and j will also be identical, and the determinant becomes zero. Therefore, the Slater-determinant wave-functions respects Pauli's exclusion principle by construction. The factor of

$\frac{1}{\sqrt{n!}}$, where $n!$ is the number of possible permutations of n electrons, normalizes the wave-function .

So by extending the idea of the Hartree approximation with the Slater determinant, we obtain a simplified wave-function that by construction respects the anti-symmetric property dictated by Pauli's exclusion principle.

3.4.3 Basis sets

We have seen how to construct wave-functions out of molecular orbitals and how wave-function solutions can be designed to satisfy Pauli's exclusion principle, but not how to approximate the molecular orbitals themselves. Basis sets serve as the building blocks that bridge the gap between the abstract formulation of quantum problems and practical computational solutions. Molecular orbitals are approximated as linear combinations of simpler basis functions. Mathematically, this is represented as

$$|\phi_i\rangle = \sum_{\mu} C_{i\mu} |\chi_\mu\rangle. \tag{3.24}$$

This turns the problem of finding the complex continuous-space wave-functions $|\phi_i\rangle$ to solve the Schrödinger equation into a simpler one of determining the coefficients $C_{i\mu}$ for the basis functions. These basis functions are approximations of the orbitals that solve the one-electron Schrödinger equation and are called atomic orbitals. These atomic orbitals are decomposed into radial and angular components, where the angular components are represented by spherical harmonics [50], and the radial components are approximated in various ways, including Gaussian-type [51] orbitals or Slater-type [52] orbitals. The most popular basis sets are the Gaussian type because their integrals can be solved in closed form, making them easy to handle in the optimisation process. The radial components in Gaussian-type basis sets are often constructed as sums of contracted Gaussian functions.

In a minimal basis set, a single atomic orbital is included for each electron. This is generally sufficient for electrons in the inner shells as they are not involved in chemical interactions, but it is often insufficient for accurate modeling of valence electrons, since these require additional flexibility to account for multiple bonds, resonance structures and other complex behaviors. Basis sets with multiple available atomic-orbitals for valence electrons are called single-, double-, and triple-zeta, denoting the multiplicity of valence orbital shells. Polarized basis sets include higher order atomic orbitals to the heavy atoms for increased flexibility. In the papers, NeuralNEB [23] and Transition1x [22], presented in this thesis, we have used the 6-31G(d) [53], which is a double-zeta polarized basis set. Basis sets turn the task of finding complex continuous-space functions into a much more tractable problem of finding a set of coefficients. The choice of basis set is a trade-off between computational efficiency and accuracy. Generally, more basis functions allows to express more complex molecular orbitals, but demands more calculations. The ability to choose an appropriate basis set depending

on the specific problem is an important part of successful application of methods in computational quantum chemistry.

3.4.4 Variational Principle in Quantum mechanics

The variational principle [8, 44] of quantum mechanics is an important concept that offers a practical guideline for how to find approximate solutions to the Schrödinger equation. It states that the energy E_ϕ of any trial wave-function ϕ is higher or equal to the ground-state energy E_0 of the Hamiltonian;

$$E_0 \leq E_\phi = \langle \phi | \hat{H} | \phi \rangle. \tag{3.25}$$

The eigenfunctions $\psi_i(r)$ of the Hamiltonian is a complete orthonormal basis set with eigenvalues $E_i \geq E_0$. Any trial wave-function $\phi(r)$ can be written as a linear combination of these eigenfunctions;

$$\phi(r) = \sum_i c_i \psi_i(r). \tag{3.26}$$

Inserting the expression for the trial wave-function into equation (3.25) we get

$$E_\phi = \sum_i |c_i|^2 \langle \psi_i | \hat{H} | \psi_i \rangle \tag{3.27}$$

$$= \sum_i |c_i|^2 E_i \tag{3.28}$$

$$\geq E_0, \tag{3.29}$$

where we have used the fact that the wave-functions are orthonormal eigenfunctions to the Hamiltonian, so that only wave-functions with matching index contributes to the sum. The variational principle offers a framework for systematic minimisation of an approximate ground-state wave-function that minimizes the ground-state energy of the system.

3.4.5 Hartree-Fock

By representing molecular orbitals as Linear Combination of Atomic Orbitalss (LCAOs) we reduce the problem of solving the Schrödinger equation to a question of determining coefficients of the basis functions in the basis set. An early attempt at this problem was the Hartree-Fock method. Here the variational principle of quantum mechanics is applied and the ground-state is sought by iteratively updating the multiple-electron wave-function, represented by a Slater determinant. Minimising the energy, following the variational principle in quantum mechanics, with the constraint that the molecular orbitals should be orthonormal can be expressed as a Lagrangian multipliers problem:

$$E = \langle \Psi | \hat{H} | \Psi \rangle - \sum_{ij} \epsilon_{ij} (\langle \phi_i | \phi_j \rangle - \delta_{ij}) \tag{3.30}$$

Where The functional derivative is

$$\frac{\delta E}{\delta \phi_i^*} = \hat{F}|\phi_i\rangle - \epsilon_i|\phi_i\rangle = 0 \tag{3.31}$$

This turns into an eigenvalue problem;

$$\hat{F}|\phi_i\rangle = \epsilon_i|\phi_i\rangle. \tag{3.32}$$

where $\hat{F}$ is the Fock operator which acts on a single molecular orbital and extracts its energy contribution

$$\hat{F} = \hat{h} + \hat{J} - \hat{K}. \tag{3.33}$$

Here $\hat{h}$ is the operator for the kinetic energy of the electron and the potential energy associated with the electron's electrostatic interaction with the nuclei. $\hat{J}$ and $\hat{K}$ are the energies associated with the Coloumb and exchange potential from the electron density and they both depend on the set of molecular orbitals. We can expand our molecular orbitals in terms of the basis set;

$$|\phi_i\rangle = \sum_\mu C_{\mu i}|\chi_\mu\rangle \tag{3.34}$$

And inserting in (3.32) yields

$$\hat{F}\sum_\mu C_{\mu i}|\chi_\mu\rangle = \epsilon_i \sum_\mu C_{\mu i}|\chi_\mu\rangle. \tag{3.35}$$

Projecting this onto the basis set $\langle\chi_\nu|$ we get

$$\sum_\mu C_{\mu i}\langle\chi_\nu|\hat{F}|\chi_\mu\rangle = \epsilon_i \sum_\mu C_{\mu i}\langle\chi_\nu|\chi_\mu\rangle \tag{3.36}$$

which can be cast in matrix form;

$$FC = \epsilon SC. \tag{3.37}$$

Here S is called the overlap matrix with elements $S_{\mu\nu} = \langle\chi_\nu|\chi_\mu\rangle$. F is the Fock matrix with elements $F_{\mu\nu} = \langle\chi_\nu|\hat{F}|\chi_\mu\rangle$, and ϵ is a diagonal matrix with the energy-eigenvalues on the diagonal. After updating the molecular orbitals, the Coulomb $\hat{J}$ and exchange $\hat{K}$ operators change, as they are integrals over the molecular orbitals. This means that the problem that we have solved is finding a new wave-function that minimizes the energy in the field caused by the old wave-function. Of course if the system is to represent a stationary state, the two wave-functions must be equal, and the molecular orbitals has be minimized with the Self Consistent Field (SCF) method.

3.4.5.1 Self Consistent Field

The SCF method [7, 8] is used to find stationary and self-consistent solutions for the wave-function of a given system. The approach is to initiate the algorithm with a guess for the wave-function and then iteratively improve on it. This guess, together with the positions and types of nuclei, is used to calculate an effective potential. We can then solve the Schrödinger equation for this potential, yielding a new wave-function. The process is then repeated with the new wave-function and the process iterates until convergence is reached and the wave-function becomes self-consistent with the effective potential that it generates.

The Hartree-Fock method allows for the wave-function to be calculated with arbitrary numerical precision, subject to the limitations of the algorithm and computational resources. However, the method is based on a mean-field approximation that neglects electron-electron correlation, and the quality of the results is constrained by the choice of basis set.

3.5 Density Functional Theory

Density Functional Theory (DFT) [3, 4] provides an alternative approach of solving the many-body problem in quantum mechanics. Unlike wave-function-based methods such as Hartree-Fock, DFT focuses on the electron density as its fundamental variable, offering a more computationally efficient way of studying systems with many electrons. The electron density $\rho(r)$ is given by

$$\rho(r) = \sum_{i=1}^{n} |\psi_n(r)|^2, \tag{3.38}$$

where n is the number of electrons in the system, and $\psi_n(r)$ is the orbital occupied by the n-th electron. Since $|\psi_n(r)|^2$ is the probability density of finding electron n at r, we can interpret the sum over $|\psi_n(r)|^2$ as an electron density at r. The total energy of the system is

$$E = \int V_{\text{ne}}(r)\rho(r)dr + F[\rho(r)], \tag{3.39}$$

where F is a functional of $\rho(r)$ which extracts all energy contributions from the electron density, including kinetic energy, exchange energy and electron-electron correlation energy. Given a functional F we can minimize the energy of the electron density using the SCF method until we find a ground-state. The formulation of DFT is straight forward, but it relies on a pair of important theorems proved by Hohenberg and Kohn that such a functional exists and that it is a variational minimum of the energy.

3.5.1 Hohenberg-Kohn Theorems

The Hohenberg-Kohn theorems are the cornerstones of DFT. They prove that the ground-state electron density is as sufficient a variable to describe the properties of a many-electron system as the wave-function, and that the ground-state electron density is a variational minimum of the energy.

3.5.1.1 HK I

The first Hohenberg-Kohn theorem shows that the ground-state electron density uniquely determines the external potential. Since the ground-state wave-function is uniquely determined by the external potential, and it contains all information about the system, if the ground-state electron density can uniquely determine the external potential, the same amount of information must be contained within the ground-state electron density.

The proof of this theorem goes via reductio ad absurdum. Suppose that we have two different, external potentials, $V_1(r)$ and $V_2(r)$ that are both consistent with a ground-state electron density $\rho(r)$. These potentials yield different Hamiltonians $\hat{H}_1$ and $\hat{H}_2$ with corresponding ground-state wave-functions Ψ_1 and Ψ_2 with eigenvalues E_1 and E_2. Following the variational principle of quantum mechanics, since Ψ_2 is the ground-state for $\hat{H}_2$;

$$E_1 < \langle \Psi_2 | \hat{H}_1 | \Psi_2 \rangle \tag{3.40}$$

Or writing the Hamiltonian in terms of its components, $\hat{H} = \hat{T}_\mathrm{e} + \hat{V}_\mathrm{ee} + \hat{V}$

$$E_1 < \langle \Psi_2 | \hat{T}_\mathrm{e} + \hat{V}_\mathrm{ee} | \Psi_2 \rangle + \int \rho(r) V_1(r) dr. \tag{3.41}$$

Now, add and subtract the Hamiltonian for the second system, and rewrite RHS of the inequality;

$$\begin{aligned} E_1 &< \langle \Psi_2 | \hat{H}_1 - \hat{H}_2 + \hat{H}_2 | \Psi_2 \rangle && (3.42)\\ &= \langle \Psi_2 | \hat{T} - \hat{T} + \hat{V}_{ee} - \hat{V}_{ee} + \hat{V}_1 - \hat{V}_2 | \Psi_2 \rangle + E_2 && (3.43)\\ &= \langle \Psi_2 | \hat{V}_1 - \hat{V}_2 | \Psi_2 \rangle + E_2 && (3.44)\\ &= \int \rho(r)[V_1(r) - V_2(r)] dr + E_2. && (3.45) \end{aligned}$$

Equivalently, by swapping the first and second system we would arrive at

$$E_2 < \int \rho(r)[V_2(r) - V_1(r)] dr + E_1. \tag{3.46}$$

But adding equations (3.45) and (3.46) we get

$$E_2 + E_1 < E_2 + E_1, \tag{3.47}$$

which is an absurd result. This means that the assumption that there is a single density associated with both V_1 and V_2 must be wrong, and thus each potential must uniquely determine an electron density. Therefore, the information contained within the wave-function must also be contained within the electron density, and it can be accessed via a functional of the electron density $F[\rho(r)]$. However, this theorem only proves the *existence* of such a functional but does not offer any suggestions as to what it might be.

3.5.1.2 HK II

The second Hohenberg-Kohn theorem extends the variational principle from quantum mechanics, stating that the ground-state electron density minimizes the total energy functional. If there is a one-to-one correspondence between the ground-state wave-function and the ground-state electron density, as stated by the first theorem, then any variational principle that applies to the wave-function must similarly apply to the electron density. Thus it is possible to variationally determine the ground-state electron density, provided one knows the correct functional $F[\rho(r)]$. Finding the correct functional is the main challenge in any practical application of DFT.

3.5.2 Kohn Sham equations

The Hohenberg-Kohn theorems provide theoretical foundation for DFT, but they do not suggest a method for finding the correct functional $F[\rho(r)]$. The Kohn-Sham equations [3] map the many-body problem of interacting electrons onto a set of non-interacting particles that can reproduce the correct ground-state. In this approach, the total energy of the system is expressed as a sum of energy contributions

$$E[\rho(r)] = T_{\mathrm{ni}}[\rho(r)] + \int V(r)\rho(r)\,\mathrm{d}r + \frac{1}{2}\iint \frac{\rho(r)\rho(r')}{|r-r'|}\,\mathrm{d}r\,\mathrm{d}r' + E_{\mathrm{xc}}[\rho(r)], \quad (3.48)$$

where $T_{\mathrm{ni}}[\rho(r)]$ is the kinetic energy of the non-interacting electrons and $E_{\mathrm{xc}}[\rho(r)]$ is the exchange-correlation energy which includes corrections for the non-interacting kinetic energy and self-interaction, as well as accounting for quantum mechanical exchange and correlation effects. The first three terms are 'easy' to calculate, the challenge lies in finding the proper exchange correlation functional.

3.5.2.1 Exchange-Correlation functionals

There are various approaches to approximate exchange-correlation functional $E_{\mathrm{xc}}[\rho(r)]$. Perhaps the simplest is the **Local Density Approximation (LDA)** [4]. In LDA approaches, the exchange-correlation at r is computed solely by the electron density at r. Typically this is chosen as the exchange-correlation energy density of a uniform electron gas of the same density. The functional form is:

$$E_{\mathrm{xc}}^{\mathrm{LDA}}[\rho(r)] = \int \rho(r)\epsilon_{\mathrm{xc}}^{\mathrm{unif}}(\rho(r))\,\mathrm{d}r \quad (3.49)$$

LDA yields accurate geometries but is inaccurate when it comes to energies. Of course LDA does not tell the full story as the electron cloud around a molecule is not a uniform electron gas. The Generalized Gradient Approximation (GGA) [54] includes the gradient of the electron density similar to a first order Taylor approximation to encompass the fact that the electron density is changing. The functional form is:

$$E_{\mathrm{xc}}^{\mathrm{GGA}}[\rho] = \int \rho(r) f(\rho(r), \nabla\rho(r))\,\mathrm{d}r \tag{3.50}$$

Finally, there are the **Hybrid** functionals [55] that combine GGA or LDA, or higher order functionals, with the exact Hartree-Fock exchange energy:

$$E_{\mathrm{xc}}^{\mathrm{Hybrid}} = aE_{\mathrm{x}}^{\mathrm{HF}} + (1-a)E_{\mathrm{xc}}^{\mathrm{DFT}}. \tag{3.51}$$

The way of combining these terms vary depending on the functional. The choice of functional is again a compromise between computational cost and accuracy, and the choice of functional significantly influences the quality of the results.

3.5.2.2 Solving the Kohn-Sham Equations

The Kohn-Sham equations are solved using the SCF method, similar to Hartree-Fock theory. The SCF procedure in DFT aims to find the ground-state electron density by solving the Kohn-Sham equations iteratively until convergence is achieved.

3.5.3 Machine Learning in Computational Quantum Chemistry

The variational principle in quantum mechanics and its counterpart in HK II combined with the SCF provides a clear and systematic framework for computing ground-state wave-functions and electron densities. Solving electronic structure problems for molecular systems provides valuable insights into energetics and forces essential for simulation and further investigation of molecular properties. However, these methods are prohibitively expensive due to the iterative nature of the SCF methods that drives solutions to self-consistency through repeated calculations of multiple electron-electron integrals. These calculations typically scales in complexity as $\mathcal{O}(N^3)$ or worse, where N is the number of electrons in the system. The computational cost of these methods poses a fundamental bottleneck preventing large scale exploration of chemical space.

In recent years, the field of Machine Learning (ML) has seen the rise of the powerful Neural Network (NN) models. These models are capable of fitting extremely complex functions while operating several orders of magnitude faster than traditional methods described in this section. When trained on datasets of high-quality electronic structure calculations, these NNs have shown promising results as emulators of ab initio methods, potentially opening up new avenues of research that were previously computationally infeasible. In the next section we will explore the inner workings

of NNs and methods from ML, investigating how they can be applied to molecular systems.

CHAPTER 4
Neural Networks

4.1 Neural Networks 101

Neural Networks (NNs), also known as Artificial Neural Networks, are mathematical models inspired by the web of interconnected neurons in the human brain. They are incredibly versatile and flexible function approximators that can be used to analyze, generate, control, and predict features in a wide range of domains, from images and molecules to the stock market and games, and many more. Whether we know it or not, we all use neural networks every single day. They hide behind the scenes of various applications such as search engines [56], music and movie recommendations on various platforms [57–59], targeted advertisements on social media [60, 61], and they even empower seemingly intelligent entities like ChatGPT [62] and other large language models. They are arguably one of the most important algorithms in the modern world.

The NN concept is not new; research has been ongoing for many years. The simplest version of a NN, called the perceptron, is essentially just a linear combination of inputs fed through a nonlinearity, representing a single *'layer'*, and it was proposed as early as the late fifties [63]. The idea of training larger models was there, but the necessary mathematical framework, which allowed for training much more complex models, called the back-propagation algorithm [64], was not invented until the eighties. However, training and using neural networks requires vast computational resources, and only at the beginning of the 2010s, with the advent of powerful GPUs, it became feasible to train large NN models.

The "vanilla" NN is the feedforward NN. It consists of multiple layers of neurons that feed a signal forward by sequentially activating each layer of neurons, simulating a cascade of firing neurons in the brain. The input to this type of model is a vector of features. Each neuron in the first layer has a certain sensitivity, called a weight, to each of the input features. The activation of a neuron is calculated as a weighted sum of the inputs, determined by these weights. Then, by adding a *bias* and applying an *activation function*, the final activation of the layer is obtained. This process is repeated for each neuron in the layer until the layer's full activation is calculated. The activation of the next layer can be computed in an equivalent fashion, but using the activation of the first layer as its input. This process is repeated for each layer in

order to propagate a signal forward in the network. Mathematically, the activation of the i-th data point $x_i^{(l)}$ in the l-th layer can be written using matrix notation as;

$$x_i^{(l)} = \sigma(W^{(l)} x_i^{(l-1)} + b^{(l)}), \tag{4.1}$$

where $W^{(l)}$ is called the weight matrix, and it allows for a compact description of all weights connecting neurons from the l-th layer to activations in the $(l-1)$-th layer. The $W_{ji}^{(l)}$ index is the i-th neuron in the l-th layer's sensitivity to the j-th activation in the $(l-1)$-th layer. $b^{(l)}$ is a bias vector that allows the network to learn biases in the data, and σ is a nonlinear activation function. By propagating the signal forward through the network, the model builds increasingly abstract representations of the data, which can be transformed into features of interest, and read off of the final layer.

The term *deep* NN refers to networks with multiple layers making them more complex - a single layer perceptron is only able to model linearly separable patterns. The primary challenge in fitting the model to data lies in finding the model's parameters $\theta = \{W^{(l)}, b^{(l)} \mid l = 1, \ldots, L\}$, where L is the number of layers. This is done through the process of *training* the network. To train a feedforward NN, we need a labeled dataset, $\{(x_i, y_i) \mid i = 1, \ldots, N\}$, where N is the number of samples, x_i represents the features of each data point, and y_i is called the ground truth of the dataset. A classic example is the MNIST dataset [65]. Within this dataset, there are 60,000 training images and 10,000 testing images of handwritten digits, each depicted as a 28x28 pixel grayscale image. The neural network's task is to recognize these digits. Here, the features, x_i, are pixel values, and the ground truth, y_i, corresponds to the digit label, represented as 10-dimensional one-hot encoded vectors [66]. The training process starts with a random initial guess for the parameters. In order to guide the training process, a loss function $L(p_\theta, D)$ is defined that quantifies how well the model aligns with the dataset. Training the network amounts to minimising the loss by repeatedly updating its parameters. A simple update scheme can be expressed as

$$\theta_{t+1} \leftarrow \theta_t - \alpha \nabla_\theta L(p_{\theta_t}, D) \tag{4.2}$$

Where α is called the learning rate. This process is called gradient descent, because the minimum is sought by always taking steps in the direction opposite to the gradient. We seek to find the lowest possible loss given the model;

$$p_\theta^{\text{trained}}(D) = \underset{\theta}{\operatorname{argmin}} L(p_\theta, D). \tag{4.3}$$

The loss surface is extremely rugged and often the gradient descent algorithm gets caught in a local minimum. There is plethora of methods for training neural networks and alleviate this problem, for example by including a notion of momentum in the updates [67], or alternating between learning rates [68], to enable the model to escape local minima.

A general strategy for designing loss functions is to define a probability distribution that is thought to reflect the structure of D, and then minimize the difference between this and an approximated distribution, parameterized by the model. In the case of supervised models, this distribution is a conditional probability distribution, $p(y \mid x)$. The quality of this approximation is measured by the Kullback-Leibler (KL) divergence [69];

$$D_{\mathrm{KL}}\left[\hat{p}_\theta(y \mid x) \mid\mid p(y \mid x)\right] = -\iint p(y \mid x) \ln p(y \mid x) dxdy + \iint p(y \mid x) \ln \hat{p}_\theta(y \mid x) dxdy. \tag{4.4}$$

Here, the first term on RHS is the entropy of the data distribution - it is the expected information of a sample from $p(y \mid x)$. The second term is the negative crossentropy, or the expected information of a sample from $p(y \mid x)$ if probabilities are evaluated with $p_\theta(y \mid x)$. The difference between the two terms is a measure of how wrong the model assigns probabilities. The entropy of the data distribution is constant, and minimising the KL-divergence comes down to minimising the second term in (4.4). However, the true data distribution is not available, only samples from it, so effectively the training procedure is minimising a Monte Carlo estimate of the cross-entropy;

$$\mathcal{L}(D, \hat{p}_\theta) = \iint p(y \mid x) \ln \hat{p}_\theta(y \mid x) dxdy \simeq \frac{1}{M} \sum_{i}^{M} \ln \hat{p}_\theta(y_i \mid x_i), \tag{4.5}$$

which is the negative log-likelihood of the dataset under the model. Or equivalently by directly optimizing the log-likelihood you'll find out when you reach the top, you're on the bottom of the loss function. It is costly to evaluate the entire dataset at each iteration, so it is instead approximated as an average of a mini-batches of M samples. For classification tasks the distribution $p(y \mid x)$ is simply a multiclass-bernoulli distribution. For regression tasks the data is assumed to be normally distributed such that the negative log likelihood of batch becomes the mean squared error.

4.2 Latent Variable Models

The architecture discussed in the previous section, had a particular focus on learning conditional probability distributions, $p(y \mid x)$, or relating the features y to the features x, without taking the distribution of x into account. This type of learning is called discriminative learning, a natural term in the context of classification. In this section, we discuss a different framework, known as generative modeling, particularly in the context of latent variable models. It is important to recognize, that any dataset is merely a sample drawn from an underlying distribution, $p(x)$. In generative modeling the task is, given the dataset, to infer the underlying patterns of the distribution and approximate it with a probabilistic model $p_\theta(x)$, parameterized by θ. One popular class of models are latent variable models, here the true distribution is described by

a much simpler underlying distribution of latent variables;

$$p_\theta(x) = \int p_\theta(x \mid z) p_\theta(z) dz \tag{4.6}$$

This type of model provides an insight into the generation process, and once trained, we have a probabilistic decoder $p_\theta(x \mid z)$, that can decode latent samples from $p_\theta(z)$ into a distribution over data. When $p_\theta(z)$ is Gaussian, the sampling procedure is straightforward, the target distribution can be obtained easily by decoding samples from the prior $p_\theta(z)$. As discussed in the previous section, our best bet for optimising the model is to optimize the log-likelihood of the data under our model. Theoretically, a valid training strategy would be to sample multiple values of z and approximate (4.6) with a Monte Carlo estimate.

$$p_\theta(x) \simeq \frac{1}{M} \sum_{i=1}^{M} p_\theta(x_i \mid z_i) \quad \text{where} \quad z_i \sim p_\theta(z). \tag{4.7}$$

However, the vast majority of samples will contribute negligibly to this integral, and consequently an infeasible amount of samples would have to be taken for a clear training signal. A better approach is to use importance weighted samples from a distribution $q_\phi(z)$, that overlap with the posterior $p_\theta(z \mid x)$. In Variational Autoencoders (VAEs) [70, 71], $q_\phi(z)$ is conditioned on x yielding an encoding network. Variational Inference (VI) [72–74] is an approach for approximating such complex posteriors. As discussed in previous sections, the training strategy is to maximize the log likelihood of the dataset under the model p_θ. This is usually infeasible, and in VI a tractable, lower bound for $\ln p_\theta(x)$, called the Evidence Lower Bound (ELBO), is optimized instead. We can arrive at this bound by analyzing the KL-divergence between $q_\phi(z)$and the posterior $p_\theta(z \mid x)$. The KL divergence is given by

$$D_{KL}\left[q_\phi(z) \,||\, p_\theta(z \mid x)\right] = \mathbb{E}_{q_\phi(z)}\left[\ln \frac{q_\phi(z)}{p_\theta(z \mid x)}\right]. \tag{4.8}$$

Expanding this with Bayes rule $p_\theta(z \mid x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$ and logarithm rules $\ln ab = \ln a + \ln b$ yields an expression for the log-likelihood

$$D_{KL}\left[q_\phi(z) \,||\, p_\theta(z|x)\right] = \mathbb{E}_{q_\phi(z)}\left[\ln q_\phi(z)\right] - \mathbb{E}_{q_\phi(z)}\left[\ln p_\theta(x \mid z)\right] - \mathbb{E}_{q_\phi(z)}\left[\ln p_\theta(z)\right] + \ln p_\theta(x). \tag{4.9}$$

By noticing that $D_{KL}\left[q_\phi(z) \,||\, p_\theta(z \mid x)\right] \geq 0$, this can be rearranged into an inequality;

$$\ln p_\theta(x) \geq \mathbb{E}_{q_\phi(z)}\left[\ln p_\theta(x \mid z)\right] - D_{KL}\left[q_\phi(z) \,||\, p_\theta(z)\right]. \tag{4.10}$$

This lower bound is the ELBO. The first term is a regularisation term ensuring that the $q_\phi(z)$ does not deviate too much from the prior, and the second term is the reconstruction term that ensures that the model can faithfully reproduce the data distribution given the latent variables. This lower bound can be optimized as a proxy for the log-likelihood. Generative modeling provides a framework for understanding and generating samples from the underlying distribution from which our observed samples are drawn.

4.3 Denoising Diffusion Probabilistic Models

One class of generative model that has seen a lot of success in recent years is the Denoising Diffusion Probabilistic Model (DDPM). DDPMs offer a compelling approach for modeling complex data distributions, rooted in statistical physics and non-equilibrium thermodynamics [75–77]. Unlike traditional generative models that sample data directly, these models gradually transform data from a simple noise distribution into a complex target distribution by learning an inverse diffusion process. These models have shown remarkable capabilities for realistic and creative image synthesis [78–81], resulting in a range of tools [82–84] that have attracted a lot of attention, even outside the confines of the Machine Learning (ML) community. Additionally DDPMs have shown promising results in generating 3D structures such as molecular configurations [85–87] or proteins [88, 89]. There are two important processes in DDPMs; the forward and the reverse diffusion process. The forward diffusion process provides a simple way to encode latent variables for data samples by slowly corrupting them towards a standard Gaussian target distribution. The learned part of the model is the reverse diffusion process. Here the model is trained to predict the inverse step taken by the forward diffusion process at each time step. At generation time, a latent variable can be sampled from the target distribution of the forward diffusion process and the DDPM can iteratively remove noise from samples to reconstruct data following the original data distribution.

4.3.1 Forward Diffusion

DDPMs encode latent variables by slowly corrupting data through Langevin Dynamics in a harmonic oscillator potential. The change Δx_t in the latent variable x_t at the t-th step can be expressed as

$$\Delta x_t = -k_t x_t + \sqrt{\beta_t}\epsilon, \tag{4.11}$$

where $-k_t x_t$ is a drift term resembling a spring force. The second term $\sqrt{\beta_t}\epsilon$ is Gaussian noise, where ϵ is sampled from a standard Gaussian distribution. The updated latent variable at step $t+1$ is

$$x_{t+1} = x_t + \Delta x_t = (1-k_t)x_t + \sqrt{\beta_t}\epsilon. \tag{4.12}$$

The latent variable should converge towards a standard Gaussian such that an x_T can be easily sampled for the generation process. This is ensured by choosing an appropriate spring constant, which can be found by inspecting the variance of the latent variable at each step and its behavior at equilibrium. The variance of x_{t+1} is

$$\langle x_{t+1}^2 \rangle = \langle (1-k_t)^2 x_t^2 + \beta_t \epsilon^2 + (1-k_t)x_t\sqrt{\beta}_t\epsilon \rangle \tag{4.13}$$

$$= (1-k_t)^2 \langle x_t^2 \rangle + \beta_t, \tag{4.14}$$

where we have used that $\langle \epsilon \rangle = 0$, and $\langle \epsilon^2 \rangle = 1$. At equilibrium the variance is constant, $\langle x_t^2 \rangle = \langle x_{t+1}^2 \rangle$, and since x_t should follow a standard Gaussian with $\langle x_t^2 \rangle = 1$, we can plug this into the equation above to get

$$1 = (1 - k_t)^2 + \beta_t, \tag{4.15}$$

so with a spring constant $k_t = 1 - \sqrt{1 - \beta_t}$, the forward diffusion process will tend towards a standard Gaussian. Plugging this into (4.12) we get

$$x_{t+1} = \sqrt{1 - \beta_t} x_t + \sqrt{\beta_t} \epsilon_t, \tag{4.16}$$

which can be packed neatly in a normal distribution

$$q(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1} \mid \sqrt{1 - \beta_t} x_t, \beta_t I), \tag{4.17}$$

where I is the identity matrix. We denote the trajectory of the entire diffusion process with $x_{0:T} = (x_0, \ldots, x_T)$. Each state x_t only depends on the previous step x_{t-1} allowing for an expression for the probability of the entire trajectory, given x_0 as

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}). \tag{4.18}$$

However, repeatedly calculating this at training time is extremely costly, especially when T is large. Here, the reparameterization trick [70, 71] comes to our rescue - the forward process is a Markov chain governed by a series of Gaussian transition-kernels and due to this structure we can directly sample x_t from $q(x_t \mid x_0)$ at any time step t, given the initial x_0, instead of having to repeatedly apply $q(x_t \mid x_{t-1})$. By defining $\alpha_t = 1 - \beta_t$, allows x_{t-1} to be expressed as:

$$x_{t-1} = \sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1 - \alpha_{t-1}} \epsilon_{t-2}. \tag{4.19}$$

By plugging this into (4.16) and expanding we get

$$x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{\alpha_t (1 - \alpha_{t-1})} \epsilon_{t-2} + \sqrt{1 - \alpha_t} \epsilon_{t-1}. \tag{4.20}$$

The sum of two normally distributed random variables, $X + Y = Z$ is itself a normally distributed random variable with mean and variance $\mu_Z = \mu_X + \mu_Y$ and $\sigma_Z^2 = \sigma_X^2 + \sigma_Y^2$. With this (4.20) can be rewritten as:

$$x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \epsilon. \tag{4.21}$$

By applying this repeatedly, and defining $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ we obtain an expression for x_t given x_0

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \tag{4.22}$$

or

$$q(x_t \mid x_0) = \mathcal{N}(x_t \mid \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I). \tag{4.23}$$

This drastically reduces computational complexity for calculating losses as we do not have to sample the full Markov chain to evaluate samples of the forward diffusion at time t.

4.3.2 Reverse Diffusion

As $T \to \infty$, the latent variable x_T tends towards a standard Gaussian. Samples can be generated from the original distribution by first sampling x_T from $\mathcal{N}(x_T \mid 0, I)$ and then gradually denoise through the reverse diffusion process, $q(x_{t-1} \mid x_t)$. However, we do not have access to $q(x_{t-1} \mid x_t)$ as its computation requires knowledge of the true data distribution. The essential task in training diffusion models is to approximate $q(x_{t-1} \mid x_t)$ with a parameterized model $p_\theta(x_{t-1} \mid x_t)$, e.g. a NN. In the limit of small values of β_t the forward and reverse diffusion process has the same functional form [90], and it suffices to parameterize $p_\theta(x_{t-1} \mid x_t)$ with mean and variance of a normal distribution. With this setup, we can calculate the probability of the reverse trajectory

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \tag{4.24}$$

and training the model amounts to learning the mean $\mu_\theta(x_t, t)$ and covariance $\Sigma_\theta(x_t, t)$ of the distribution

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1} | \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{4.25}$$

This yields a powerful model for approximating the true data distribution with a conceptually straightforward mechanism for sampling.

4.3.3 Training DPPMs

As usual, our best bet to train the model is to optimize the log-likelihood of the dataset.

$$\ln p_\theta(x_0) = \ln \left[\int p_\theta(x_0 | x_1, \ldots, x_T) dx_{1:T} \right] = \ln \left[\int p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t) dx_{1:T} \right] \tag{4.26}$$

Again, this problem is tackled by importance sampling, using the forward diffusion process as the encoding distribution.

$$\ln p_\theta(x_0) = \ln \left[\int p_\theta(x_T) \prod_{t=1}^{T} q(x_t \mid x_{t-1}) \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} dx_{1:T} \right] \tag{4.27}$$

The ELBO, denoted $\mathcal{L}$, is obtained by using Jensen's inequality [91]

$$\mathcal{L} = \mathbb{E}_{q(x_{1:T}|x_0)} \left[\ln p_\theta(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} \right]. \tag{4.28}$$

Since x_t is Markov, $q(x_t \mid x_{t-1}) = q(x_t \mid x_{t-1}, x_0)$, and rewriting with Bayes rule yields

$$q(x_t \mid x_{t-1}, x_0) = \frac{q(x_{t-1} \mid x_t, x_0) q(x_t \mid x_0)}{q(x_{t-1} \mid x_0)}. \tag{4.29}$$

Which is plugged into the ELBO

$$\mathcal{L} = \mathbb{E}_{q(x_{1:T}|x_0)} \left[\ln p_\theta(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1} \mid x_t) q(x_{t-1} \mid x_0)}{q(x_{t-1} \mid x_t, x_0) q(x_t \mid x_0)} \right]. \tag{4.30}$$

Using logarithm rules and canceling out terms of $q(x_t \mid x_0)$, this can be rewritten as

$$\mathcal{L} = \mathbb{E}_{q(x_{1:T}|x_0)} \left[\ln p_\theta(x_0 \mid x_1) + \sum_{t=2}^{T} \ln \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_{t-1} \mid x_t, x_0)} + \ln \frac{p_\theta(x_T)}{q(x_T \mid x_0)} \right]. \tag{4.31}$$

Here, the first term in the expectation is simply a reconstruction error, quantifying how well the model can reconstruct data x_0 from the latent variable x_1. The second term is the KL-divergence between the approximated and true reverse diffusion process, and the final term measures how close x_T is to a standard Gaussian. The last term does not have trainable parameters and is ignored during training. The denoising step is normally distributed

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1} \mid \tilde{\mu}, \tilde{\beta} I). \tag{4.32}$$

Obtaining expressions for $\tilde{\mu}$ and $\tilde{\beta}$ is cumbersome but straightforward. It involves writing up the full expression for the normal distributions in (4.29), expanding exponents and extracting mean and variance.

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} x_0 \quad \text{and} \quad \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t. \tag{4.33}$$

This can be simplified further by inserting the expression for x_0 isolated from (4.22).

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right). \tag{4.34}$$

Approximating $\tilde{\mu}_t$ with the model comes down to predicting ϵ_t given x_t. Given that $p_\theta(x_{t-1}|x_t)$ and $q(x_{t-1}|x_t, x_0)$ are normal distributions, the t-th term (4.31).

$$\mathcal{L}_t = \mathbb{E}_{p(x_0), \mathcal{N}(\epsilon_t; 0, I)} \left[\frac{|\tilde{\mu}_t - \mu_\theta(x_t, t)|^2}{2\sigma_t^2} \right] \tag{4.35}$$

where $x_t = \sqrt{\bar{\alpha}_t x_0} + \sqrt{1 - \bar{\alpha}_t}\epsilon$. In this case the variance σ_t follows a fixed schedule. However, it has been shown that learning σ_t leads to better likelihoods and faster sampling [92]. (4.34) reveals that $\mu_\theta(x_t, t)$ can be approximated simply by predicting ϵ_t with a model $\epsilon_\theta(x_t, t)$. By isolating ϵ_t and $\epsilon_\theta(x_t, t)$ from (4.35) we arrive at an expression that can be used to calculate the t-th term in the loss (4.31):

$$\mathcal{L}_t = \mathbb{E}_{p(x_0), \mathcal{N}(\epsilon_t; 0, I)} \left[\frac{\beta_t}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \left| \epsilon_t - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t x_0} + \sqrt{1 - \bar{\alpha}_t}\epsilon, t \right) \right|^2 \right]. \tag{4.36}$$

The t-dependent weighting puts extra weight on steps that correspond to small t. Ho et al. [76] demonstrated discarding this weighting and using the simplified loss

$$\mathcal{L}_t = \mathbb{E}_{p(x_0),\mathcal{N}(\epsilon_t;0,I)}\left[\left|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\right|^2\right], \tag{4.37}$$

leads to a better sample quality.

4.4 Accelerating Sampling with Probability Flow ODEs

Although DDPMs produce high-quality samples, their sampling process is slow when compared to many other types of generative models. Typically it takes hundreds or thousands of sequential evaluations of a NN estimating scores to produce a single sample from the model. Whereas for many other models the generation is done in a single pass. This inefficiency poses a significant bottleneck for using DDPMs for downstream tasks. DDPMs as formulated in the previous chapter can be seen as discrete realisations of an underlying stochastic process described by a Stochastic Differential Equation (SDE). In its continuous form, the forward diffusion is described by

$$dx = f(x,t)dt + g(t)dw, \tag{4.38}$$

where dw is the Wiener process. Interestingly, it can be shown [93] that the inverse process has the same functional form;

$$dx = \left[f(x,t) - g(t)^2\nabla \ln p_\theta(x,t)\right]dt + g(t)dw. \tag{4.39}$$

Samples can be obtained by integrating standard Gaussian noise backwards in time using (4.39). Here the *score* is related to the noise by $\nabla \ln p_\theta(x,t) = \epsilon_\theta(x,t)/\sigma(t)$. When integrating SDEs with numerical methods, a large number of short integration steps, have to be taken relative to Ordinary Differential Equations (ODEs) for convergence [94], due to the randomness of the Wiener process. This motivates finding an ODE that has the same marginal distribution as (4.39). The Fokker Planck equation governing (4.39) can be shown to be [95]

$$\frac{\partial p_\theta(x,t)}{\partial t} = -\nabla \cdot [\tilde{f}(x,t)p_\theta(x,t)] \tag{4.40}$$

where

$$\tilde{f}(x,t) = f(x,t) - \frac{1}{2}g(t)^2\nabla \ln p_\theta(x,t). \tag{4.41}$$

But (4.40) is the Fokker Planck equation governing

$$dx = \tilde{f}(x,t)dt + \tilde{g}(t)dw, \tag{4.42}$$

when setting $\tilde{g}(t) = 0$, so

$$dx = \left[f(x,t) - \frac{1}{2}g(t)^2\nabla \ln p_\theta(x,t)\right]dt \tag{4.43}$$

describes an ODE with the same marginal distribution as (4.39). This provides a compelling approach for efficient sampling of DDPMs models. As the ODE described by (4.43) and the SDE described by (4.39) are governed by the same Fokker Planck equation, we can potentially accelerate the sampling procedure by adopting the ODE formulation in place of the SDE. This allows for larger step sizes and therefore faster numerical integration. The ODE is composed of a linear and non-linear term - with the NN model being responsible for the nonlinearity. The linear term can be solved analytically leaving only the non-linear term to be handled by black-box ODE solver. This approach offers a way to obtain high-quality samples at a significantly reduced computational cost. In fact, when applied for down-stream tasks, high quality samples can be obtained with as few as 10–20 evaluations of the score model [96]. Moreover, this approach is agnostic to the training procedure of the score model and does not require any modification of the training procedure outlined in the previous section. Other methods such as score matching can be used as well [97, 98]. As a result, models trained using various strategies can be integrated directly into this efficient sampling framework.

4.5 Message Passing Neural Networks

4.5.1 Molecules and Proteins as graphs

Graphs are mathematical structures consisting of nodes and edges. They can efficiently encode entities and their spatial and relational characteristics. A graph is defined as $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each edge $e = (u, v)$ represents a connection between node u and v. Graphs offer a natural way of encoding molecules [99] and proteins with atoms or amino acids represented by nodes, and bonds and interactions between them represented as edges, allowing for a rich featurization while retaining 3D structure. When working with molecules and proteins, the spatial arrangement of nodes in the graph is referred to as the geometry or sometimes configuration. However, the meaning of configurational changes refers to changes in chemical bonds. Even when bonds are fixed, there's still some flexibility coming from rotation around or stretching of them, which can allow the configuration to adopt different shapes or spatial arrangements. These different shapes that a configuration can take are referred to as conformers or conformations.

Typically, geometries are encoded as lists of (x, y, z) coordinates with corresponding atom or amino acid types. Sometimes, more detailed features such as specific bonds, partial charges, or associations with particular molecular groups or residues are also encoded in the data. The input, of course, depends on the task at hand. For the work presented in this thesis, all inputs to the models have been pure geometries with no information about bonds.

4.5.2 Message Passing Neural Networks

Message Passing Neural Networks (MPNNs) [100–102] is a class of NNs designed specifically for handling graphs, which is why they have been widely adopted in the field of ML for molecules and proteins [100, 103–106]. MPNNs operate by iteratively updating node and edge features through interactions along the edges of the graph until a prediction is made based on the final features. This process is similar to, or in fact a generalisation of, how traditional Convolutional Neural Networks (CNNs) [107] work. In the context of CNNs, kernels iteratively generate higher order feature maps, ultimately leading to predictions based on these feature maps. These maps can be viewed as graphs, where pixels are connected with their neighboring pixels. If all nodes are to interact in each iteration, the computational complexity scales with the number of nodes, n, as $\mathcal{O}(n^2)$. Since, in molecules, the most significant interactions are typically between nodes that are spatially close, it can be computationally advantageous to define a smaller neighborhood for each node, u, denoted as $\mathcal{N}(u) = \{v \in V \mid (u, v) \in E\}$. This approach reduces the complexity scaling to $\mathcal{O}(n)$ at the cost of long range interactions. The neighborhood can be defined in various ways, for example through a radius cutoff, where nodes are connected if the distance between them is less than a predefined radius r. Note that the neighborhood and chemical bonds are different; the neighborhood defines which nodes are allowed to interact regardless of whether are connected by a chemical bond. MPNNs consist of four fundamental blocks: the embedding block, the message passing block, the update block, and the readout block.

The **Embedding block** is responsible for embedding the input to the model. Often, the model input consists of just the coordinates and node types. In the case of molecules, the embedding block could contain a lookup table of trainable element embeddings, allowing the model to build an internal representation of properties such as valence, electronegativity, or other useful attributes of each element. Usually, the graph and its neighborhood are also defined in this block.

The **Message Passing Blocks** are responsible for propagating information between nodes and their neighbors. They do so by combining node and edge attributes with spatial information to compute messages, that are exchanged between nodes. The information exchange through the graph comes from these messages. The message received by node u at the l-th iteration is given by

$$m_u^l = \phi^l \left(\{ M^l(h_u^l, h_v^l, e_{uv}^l, r_{uv}) \mid v \in \mathcal{N}(u) \} \right), \tag{4.44}$$

where $M^l(h_u^l, h_v^l, e_{uv}^l, r_{uv})$ computes the message sent from v to u, given the features of the nodes, h_u^l, h_v^l, the edge connecting them, e_{uv}^l, and their separation vector r_{uv}. ϕ^l denotes an aggregation function that aggregates incoming messages. This can be a simple sum or more sophisticated functions.
The **Update Blocks** are responsible for updating the internal representation of nodes

and edges based on the incoming messages. The updates are given by:

$$h_u^{l+1} = U_h^l(h_u^l, m_u^l), \tag{4.45}$$

and

$$e_{uv}^{l+1} = U_e^l(h_u^l, h_v^l, e_{uv}^l, r_{uv}), \tag{4.46}$$

where r_{uv} is the separation vector between the u and v. During the *message passing* phase of the MPNN, the internal representation of the graph is iteratively updated by alternating between message passing and update blocks. The main challenge in designing a MPNN often lies in defining these blocks.

Finally, once the message passing phase is over, the **Readout Block** calculates global features on all nodes of the graph:

$$y = \phi^L\left(h_u^L \mid u \in V\right), \tag{4.47}$$

such as the total energy of a molecule. Here, L represent the last iteration of the message passing steps, and ϕ^L is an aggregation function that aggregates the final features, which could be, for example, atomic energy or other relevant characteristics.

4.5.3 Equivariance

A function f is said to be equivariant under a group G if

$$f(g(x)) = g(f(x)) \tag{4.48}$$

for any transformation $g \in G$. Molecules are embedded in 3D space and are governed by translational and rotational symmetries. If for example a molecule is rotated, we expect the forces on atoms to rotate equivariantly. These symmetries are referred to as the Euclidian E(3) group and models have to be designed carefully to inherently respect them. Incorporating these symmetries into models provides an important inductive bias enhancing interpretability and generalization [108–110] Recent advances has seen the development of multiple E(3)-equivariant architectures [110–113]. The work presented in this thesis has all been applying the Polarizable atom interaction Neural Network (PaiNN) [111] architecture.

4.5.4 Comparison with the electrostatic potential

MPNNs are analogous to the electrostatic potential [113, 114], providing some justification for them as an inductive bias for physical interactions in molecules, protein, and other many body systems. Coulomb's law states that the force acting on a particle i under the influence of another particle, j, is

$$F_i^{\text{es}} = -k_e \frac{q_i q_j}{r_{ij}^2} \hat{r}_{ij}, \tag{4.49}$$

where q_i and q_j are the charges of i and j, $k_e = 1/4\pi\epsilon_0$ is Coulomb's constant and ϵ_0 is the vacuum permittivity. By integrating over $\hat{r}_{ij}dr_{ij}$ and summing over all particles in an N-particle system, we get the total electrostatic energy

$$E^{\text{es}} = \frac{1}{2}\sum_{i=1}^{N} q_i V_i^{\text{es}}(x_i) \tag{4.50}$$

where V_i^e is the electrostatic potential due to all charged particles except for i,

$$V_i^{\text{es}}(x_i) = k_e \sum_{j\neq i} \frac{q_j}{r_{ij}} \tag{4.51}$$

The analogy between electrostatic potential and message passing neural networks can be broken down into the following components:

- **Charge as Node Features**: The charge q_i of a particle corresponds to the features h_i of node i.

- **Electrostatic potential as Message function**: The electrostatic potential $V_i^{es}(x_i)$ at x_i, caused by all other particles except i, corresponds to the incoming message to the node i, where the message and aggregation function is

$$M(q_j, r_{ij}) = \frac{q_j}{r_{ij}} \quad \text{and} \quad \phi = k_e \sum_{j\neq i} \cdot, \tag{4.52}$$

- **Energy Calculation and Update Rule**: The calculation of electrostatic energy of the particle at x_i corresponds to the update function of node i,

$$U(q_i, V_i^{\text{es}}(x_i)) = \frac{1}{2} q_i V_i^{\text{es}}(x_i) \tag{4.53}$$

- **Energy and Readout Function**: The sum of the electrostatic energy contributions from each particle corresponds to the readout function in an MPNN, aggregating individual contributions to yield a global feature of the graph or system.

While motivating MPNNs for physical systems, this analogy also highlights an important potential issue concerning the errors that a cutoff function might introduce.

4.6 Wrap-up

In summary, the ML field offers a rich array of methods that can be used in various contexts related to computational quantum chemistry. MPNNs are especially well-suited for working on molecular structures because of their graph-like nature and they

can be employed for various complex tasks such as predicting molecular properties, estimating distributions or generating structures. Having previously explored some of the challenges in computational chemistry and the bottlenecks posed by computationally intensive algorithms, we will now turn our focus towards how to incorporate techniques from the field of ML for various tasks.

CHAPTER 5

ML Accelerated Transition State Search

Based on the papers "NeuralNEB - Neural Networks can find Reaction Paths Fast", published in IOP Science, 2022, "Transition1x - a dataset for building generalizable reactive machine learning potentials", published in Scientific Data, 2022, and work done by Mathias Schreiner, Arghya Bhowmik, Peter Bjørn Jørgensen, Jonas Busk, Tejs Vegge, and Ole Winther published in Scientific Data, 2022 and IOP Science, 2022

5.1 Transition States

The Born-Oppenheimer approximation provides a conceptual framework in which chemical reactions can be viewed as the motion of nuclear configurations on a Potential Energy Surface (PES). From this perspective minima represent stable configurations where, if perturbed the configuration will experience a restoring force, pulling it back to its initial state. Chemical reactions can be viewed as configurations transitioning between these energy minima, by traversing energetic barriers separating them. The height and shape of the barriers play a crucial role in determining reaction rates and mechanisms. Within the Born-Oppenheimer approximation, electrons are assumed to adjust instantaneously, or adiabatically, to the positions of the moving nuclei, allowing us to focus solely on the nuclei when analyzing reaction mechanisms. The nuclei, are treated classically and therefore, Transition State Theory (TST) is a classical theory - even if forces and energies etc. are calculated using sophisticated methods from computational quantum chemistry. The Boltzmann distribution describes how energy is distributed among the possible states, x, of a system. The probability that a system is in a configuration with energy E is given by

$$p(x) = \frac{e^{-\beta E(x)}}{Z}, \tag{5.1}$$

where Z is the partition function that normalizes the distribution by accounting for all possible microstates, and β is the inverse temperature $\beta = 1/k_B T$, with k_B being the Boltzmann constant and T the temperature. Increasing the temperature effectively

reduces the exponent $-\beta E(x)$, thereby making higher-energy states more probable. At high temperatures configurations can move more freely on the PES as there is more energy available in the system. Regardless of the specific way in which energy is stored or used to break or form bonds or in other ways modify the geometry, the requirement for a certain amount of energy to be present creates a substantial bottleneck for a reaction to take place. Regardless of form of the energy, whether for breaking or forming bonds, or for altering molecular geometry, the necessity for a minimal energy for the reaction to take place represents a significant bottleneck for any reaction to occur.

The term transition state formally refers to any configuration on the hyperplane where the force acting on the configuration does not point towards either the reactant or the product. A small perturbation of a configuration out of this hyperplane would cause the configuration to fall into the corresponding minimum. However, most commonly when a Transition State (TS) is mentioned in the literature, it is referring to the specific TS on the hyperplane with the lowest energy. To calculate the true reaction rate of a reaction, one would have to integrate over the entire hyperplane separating the product and reactant and weigh all possible TSs with their probability dictated by Boltzmann distribution. However, since the Boltzmann distribution is dominated exponentially by the energy, most of the contribution to this integral will come from the lowest energy configuration TS, which makes it particularly interesting. The Minimal Energy Path (MEP) for a reaction is the trajectory on the PES that follows the minimum energy valley connecting the reactant state to the product state, passing through the TS, and gradient perpendicular to the MEP is 0 along its entire length. The MEP provides crucial insights into the reaction pathways and allows for understanding the energetics and reaction mechanisms of the reaction process. Consider a chemical reaction $A + B \rightarrow C$. The rate of this reaction can be described as

$$-\frac{d[A]}{dt} = -\frac{d[B]}{dt} = \frac{d[C]}{dt} = \kappa[A][B], \tag{5.2}$$

where κ is the rate constant and $[X]$ denotes the concentration of species X. The rate depends on the concentration of the reactants since these have to collide in the medium to react. The rate constant κ is described by the Arrhenius equation:

$$\kappa = Ae^{-\beta E_a}, \tag{5.3}$$

where E_a is called the *activation* energy of the reaction, representing the energy difference or the barrier height between the reactant and the transition state. It describes how much energy must be added to the reactant configuration to overcome the energy barrier. The Arrhenius equation essentially recasts the Boltzmann distribution in terms of the activation energy, offering insights into how temperature and energy barriers affect the speed of reactions. A is a scales the rate taking specific by considering various mechanistic details. Transition states are of particular interest, not just because they offer insights into the reaction mechanism when studied alongside

the MEP, but most importantly because they define the energy barrier that must be overcome for the reaction to occur.

5.1.1 Transition State Search

The PES is a an extremely complex and high-dimensional landscape with $3N$ dimensions, describing the x, y, z coordinates of each of N nuclei. Local minima on the PES are 'easy' to find if we can calculate its gradient - then we can simply relax configurations, applying any gradient descent algorithms. Transition states on the other hand are extremely hard to locate as the gradients on the PES does not point towards them. There is largely two methods for finding TSs, *local* methods which search for TSs based on an initial guess and *interpolation* methods that starts from the product and/or reactant.

5.1.2 Local Methods

Local methods are typically used to find a TS starting from an initial guess and then refining it within a local region of the PES. An important property that is used for most local method is, that the TS is a first-order saddle point. This means that the curvature is positive in all but one direction and that the gradient is zero. Perhaps the most notable methods is the *Newton-Raphson* method [115]. This method is known to converge quickly [7] if the initial guess is in a region where the Hessian matrix of the PES has only one negative eigenvalue. The Hessian matrix describes the curvature in different directions and positive eigenvalues correspond to directions with positive curvature. Perturbing a configuration along an eigenvector with a positive eigenvalue will result in a restoring force on the configuration. The eigenvalues correspond to angular frequencies along the direction of the eigenvectors, and negative eigenvalues correspond to imaginary angular frequencies or repulsive forces along those eigenvectors. The Hessian, or equivalently, the negative Jacobian J of the force field, quantifies how a small perturbation of a configuration x results in a small change in the force $F(x)$. Given a desired change in the force ΔF, the inverse Jacobian computes the required change in the input Δx to produce it:

$$\Delta F(x) = J(x)\Delta x \tag{5.4}$$

$$\Delta x = J^{-1}(x)\Delta F(x). \tag{5.5}$$

Since the force acting on a saddle point is 0, if there are any forces $F(x) \neq 0$ on the configuration, the desired change in output is $\Delta F = F(x)$. Assume that the Jacobian is constant, Δx is calculated simply by

$$\Delta x = -J^{-1}(x)F(x). \tag{5.6}$$

And thus we can update our guess for x

$$x_{\text{new}} = x_{\text{guess}} - J^{-1}(x)F(x_{\text{guess}}). \tag{5.7}$$

There is going to be numerical errors as it is unlikely that the Jacobian is constant, but the closer our x_{guess} is to the TS the better the approximation becomes. By iteratively updating this guess we can quickly converge towards the correct TS. To prevent too large jumps, a *trust radius* is sometimes defined so x_{new} is restricted to some range. At each step it is checked whether the Hessian still only has a single negative eigenvalue. Computing the Hessian is computationally expensive and this method is best used as a refinement to TSs found by other methods.

5.1.3 Interpolation and Growing String methods

These methods instead start from the product and reactant configurations. Perhaps the simplest is the Linear Synchronous Transit (LST)[7] method where all nuclei are assumed to move linearly between the reactant and product throughout the reaction. Configuration energies are then calculated along this path and the maximal energy configuration is reported as the TS. An improvement to the LST is the Quadratic Synchronous Transit (QST) algorithm which improves on the guess of TS and MEP from LST by relaxing the TS perpendicularly to the LST path. This yields a product, reactant, and TS between which a second order polynomial is fitted.

5.1.3.1 Growing String Methods

A more sophisticated family of TS search algorithms are the Growing String Methods (GSMs) algorithms. These algorithms can be categorized into single-ended[116, 117] and double-ended[118, 119] methods. In single ended methods a MEP is grown from the reactant alone, and in double ended methods a MEP is grown simultaneously from the product and reactant. There are several methods for growing the strings in GSM. One approach is searching for energetic minima on segments of the hypersphere pointing in the general direction of the reaction. Another is the ”step-and-slide” method, where the step is taken either in the direction of the product of the reaction, or the end of the opposite string. After taking this step, the configuration is relaxed under the constraint that the distance to the previous point is kept fixed. GSMs can also be employed to explore the PES without having any specific target, but simply starting from a given minima and finding possible TSs in its vicinity.

5.1.3.2 Nudged Elastic Band

Another algorithm is the Nudged Elastic Band (NEB)[16, 17]. NEB uses a series of configurations referred to as images to represent the path between product and reactant states. These images are connected by virtual elastic bands, pulling adjacent images toward each other. Initially a guess MEP is proposed by a cheap algorithm such as LST or similar. Then, at each step, the gradient of the PES is calculated at each image, and the images are nudged in the direction of the component of the gradient perpendicular to the path MEP. This causes the path to relax along a valley such that all perpendicular forces acting on it is 0. The springs make sure that the

images stay evenly spaced. At convergence the true TS may lie between two images causing NEB to systematically underestimate the energy of the TS. Climbing Image Nudged Elastic Band (CINEB)[120] adds a slight modification to the algorithm that addresses this problem. In CINEB an extra phase is added between path optimization steps. In this phase, the highest energy image along the path is nominated as a candidate for the TS. This image climbs to the highest point on the path, freed from all spring forces, by following the component of the gradient that is parallel to the current iteration of the MEP. CINEB then alternates between the climbing step and the path optimization step in order to predict the final TS.

5.2 Transition State Search with Machine Learning

Accurate calculations of PESs for molecular systems are important for understanding a wide range of phenomena such as molecular interactions, reaction mechanisms, transition states and many more. The PES is essentially a function that maps molecular systems with N atoms, each described by 3 spatial coordinates (x, y, z), to a to a potential energy: $E : \mathbb{R}^{3N} \rightarrow \mathbb{R}$. Here, $\mathbb{R}^{3N}$ represents the $3N$-dimensional configuration space, and $\mathbb{R}$ represents energy. Given a configuration $x \in \mathbb{R}^{3N}$, $E(x)$ yields the potential energy of that configuration. In general, the force, $F(x)$ on any generalized coordinate, x, can be calculated by the gradient of the PES with respect to that coordinate $F(x) = -\nabla E(x)$. This can be used to find minima, or stable configurations on the PES or perform simulations, e.g. with Langevin Dymamics. There exists a wealth of methods for calculating the PES. These range from wave-function methods and Density Functional Theory (DFT) which deal seriously with the quantum mechanical nature of molecules, to force-field methods that abstracts quantum mechanical nature of electrons away and essentially treat the entire molecule classically by approximating all nuclei as charged particles connected with springs. The plethora of methods for approximating energetics of molecular configurations reflects the importance of the PES. In recent years a new player has entered the field - Machine Learning (ML) potentials, and specifically Neural Networks (NNs) trained to accurately emulate expensive methods from computational quantum chemistry. NNs are extremely flexible function approximators ideal for the task. The most important algorithm in the context of NNs is the *back propagation*-algorithm to which the field of NNs at large owes its success. The back propagation algorithm lets us calculate the gradient of any output of the model with respect to the model parameters or model inputs. In the context of NNs emulating PESs the back-propagation algorithm plays double duty as it provides a framework to obtain the gradient of the potential energy with respect to the nuclear coordinates, which are the forces acting on the nuclei.

5.3 NeuralNEB and Transition1x

The initial focus of my PhD research was to develop an approach for accelerating transition-state search in molecular systems by combining fast and accurate ML methods for approximating PESs with the well-established NEB algorithm. This method was dubbed *NeuralNEB*[23] The NEB method, though robust when operating on smooth PESs, is computationally intensive to run. This is because it requires expensive calculations of energy and force along entire reaction paths at each iteration. This becomes particularly heavy for complex systems. Moreover, the MEP of these complex systems tend to converge slowly, resulting in further iterations of the algorithm, exacerbating the computational cost. The success of any TS-search method ultimately relies on the quality of the PES that it is operating on. TS search methods are agnostic to the physical significance of the TSs - they are simply algorithms for locating first order saddle points in high-dimensional scalar fields. If the model for the PES does not capture the physics of the system adequately, saddle points on the PES will not reflect the true TSs, and derived barrier heights and reaction mechanism will be irrelevant or misleading. The NEB method is particularly sensitive to the behavior of the PESs over large regions. NEB is not just an algorithm for finding TSs but entire MEPs, and doing so it essentially "sweeps" the entire surface. Any regions encountered during the search with high fluctuations or unphysical gradients can throw the algorithm severely off track resulting in convergence to wrong TSs or an inability to converge all together.

Unfortunately, this particular requirement of the NEB method aligns with one the inherent weaknesses of NNs. NNs are extremely flexible function approximators owing to their numerous parameters, which makes them well-suited for approximating complex PESs over a known data distribution. However, NNs are notoriously bad at Out Of Distribution (OOD)[121, 122] tasks owing to the fact that their interplay many parameters can not necessarily be extrapolated, and consequently, in order to train NN-models to work with NEB large regions of the PES has to be available during training.

Initially we decided to use the QM9[123] dataset for training our models. QM9 is a popular benchmarking dataset in the field of computational chemistry and it is often used as a starting point for developing new models and methods in the field, especially in the context of ML based methods. It contains 135k equilibrium configurations for small organic molecules consisting of up to 9 heavy atoms, including N, O, C, and F and associated properties such as energies, polarizability, dipole moments, etc., all optimized with the B3LYP [124] functional and 6-31G(2df,p) basis set. QM9 is based on GDB-17 [125] - a vast database of molecular configuration based on a combinatorial exploration of chemical space. A problem with the QM9 dataset in the context of learning general PESs with NNs is that we can only reasonably expect the models to accurately predict energies of equilibrium-configurations. As outlined above, NN models trained on QM9 are ill-suited to act as PES-approximators for NEB as they have a limited understanding of the PES in the relevant regions between equilibrium

configurations. The results when applying these models, trained on QM9, as PESs for the NEB algorithm were poor. The algorithm would only converge around 35% of the time, and when it did, it would underestimate barrier heights, and find MEPs with irregular energy profiles displaying sudden spikes or dips that were not physically meaningful.

Following the shortcomings of our initial approach, we shifted to another, more suitable dataset for the task. ANI1x [126] is part of a series of datasets designed for training and validating ML models for tasks in quantum chemistry. The focus of these datasets is to provide a diverse range of molecular configurations including off-equilibrium geometries spanning a wider range of the PES. The ANI1x dataset contains 5M geometries of up to 8 heavy atoms, consisting of C, N, and O and includes energies and forces calculated with the wb97x [127] functional and 6-31G(d) [128] basis set. The configurations in ANI1x are proposed through various methods such as pseudo-molecular dynamics and perturbations of existing configurations. Candidate configurations are evaluated through the query by committee algorithm, where new data is included or rejected based on the variance of an ensemble of models trained on the dataset. The assumption is that if the ensemble prediction has a high variance, the data is not represented well enough in the dataset and it should be included. Only then is it necessary to perform expensive DFT calculations. This approach allows for a systematic way of expanding the dataset to extend the representational capability of the NN-models. This makes it suitable for models that require a more complete sampling of configuration space such as the PES-models in NeuralNEB. We trained a new batch of models on the ANI1x dataset and they yielded significantly better results than our initial efforts using QM9. The models would converge on 70% of reactions and predict physically plausible MEPs.

At this point we used a third dataset of chemical reactions for validating NeuralNEBs TSs-search capabilities. This dataset consists of reactants, products and TSs for 12k organic reactions of up to 7 heavy atoms including C, N, and O, all generated with the GSM. By initiating NeuralNEB in the endpoints of these reactions we could find TSs and compare them with TSs found by traditional methods. This assessment had an inherent systematic error that could not be accounted for, due to an inconsistency in the level of DFT used for the reaction dataset and the ANI1x configurations, on which the NeuralNEB models were trained.

To properly assess the quality of barrier heights and TSs predicted by NeuralNEB, we had to recalibrate the reaction dataset to align with the level of DFT from ANI1x. We did this using NEB as search algorithm. We soon realized that recalibrating the entire dataset was an immense computational task, estimated to require in the order of 25 years of wall-clock CPU-time. However, it was clear that the millions of DFT calculations for configurations generated while running NEB would be the ideal training data for NeuralNEB models. NEB excessively samples the particular regions of the PESs interesting for studying reaction mechanisms, thus making it an excellent

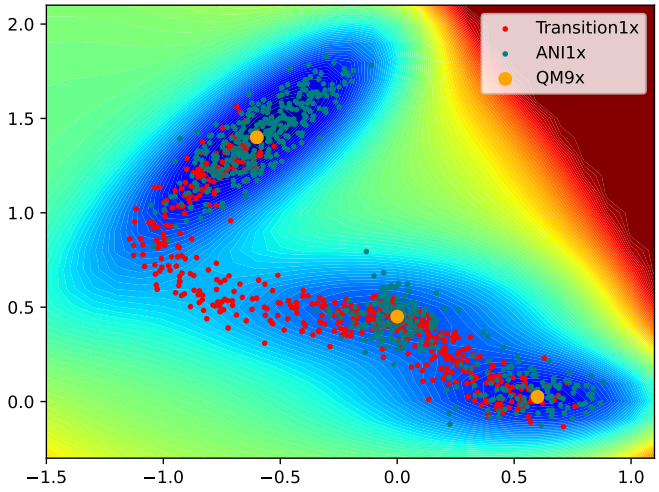


Figure 5.1: Distribution of data in the QM9, ANI-1x, and Transition1x datasets, illustrated on the Muller-Brown potential energy surface. Transition1x dataset includes data along reaction pathways and at transition states, providing valuable data for reaction barrier calculations. The ANI-1x dataset samples the energy minima thoroughly but provides sparse data around transition states. The QM9 dataset only provides a single data point the bottom of each minimum..

data proposal mechanism for our application. This naturally led to the creation of the *Transition1x* [22] dataset, owing its postfix *1x* to the ANI1x dataset. This amounted to the first publication during my Ph.D. *"Transition1x - a dataset for building generalizable reactive machine learning potentials"*, published in *Scientific Data* in 2022, which is appended in appendix A. The publication also includes a recalibration of the QM9 dataset dubbed QM9x, allowing for various comparisons of the datasets. Note that transitions are rare events and are therefore only sparsely represented in datasets such as ANI1x, that relies on proposal methods imitating the dynamics of the system. Figure 5.3 is an illustration of the characteristics of QM9, ANI1x and Transition1x, demonstrated on the Müller-Brown dataset, which is a theoretical 2-D toy potential, often used for showcasing and testing methods in computational chemistry.

The Transition1x dataset is useful in a wide range of ML-based methods concerned

with chemical reactions and it is a tailor-made fit for training NeuralNEB models. Building on Transition1x the NeuralNEB project progressed to its third iteration which resulted in the paper *"NeuralNEB - Neural Networks can find Reaction Paths Fast"*, published in *IOP Science*, 2022, which is appended in appendix B. Here, the Transition1x dataset was split into ten subsets, each consisting of a unique set of isomers participating in various elementary reactions. We employed 10-fold cross validation on these splits. However, we did not simply evaluate the models on the test-sets, as is normally done in cross-validation. Instead the models were applied as PES-approximators within the NeuralNEB framework and evaluated based on their ability to guide NEB to the correct TSs of the reactions in the test set. These models would converge on 80% of the reactions examined, along with a Mean Absolute Error (MAE) of 0.23 eV and Root Mean Squared Error (RMSE) of 0.52 eV for barrier heights. Although these results did not achieve chemical accuracy, the models significantly accelerated the computations. The average wall-clock time for convergence using NeuralNEB was only 33 seconds, compared to 12 hours and 15 minutes required for Density Functional Theory (DFT) calculations.

These results are, encouraging nonetheless as they open up for various avenues for improvement. Particularly in the choice of NN architecture. In our early experiments we used the SchNet [129, 130] architecture, and were rarely able to converge on MEPs. It was upon replacing the architecture with Polarizable atom interaction Neural Network (PaiNN) [111] that we saw signs of a functional framework, indicating that the architecture plays a huge role and there are many architectures to try [108, 112, 113]. The results found in the work with Transition1x and NeuralNEB was sent summarized in the extended abstract *"Machine Learning for Chemical Reactions"* and poster presented at the *Machine Learning and the Physical Sciences Workshop at NeurIPS* 2022, appended in C.

NNs run on GPUs as opposed to CPUs as classical methods in computational chemistry. This allows for calculations of hundreds of configurations at a time at only little extra computational cost. Traditionally, the images used to represent the MEP is a a compromise as computational cost scales linearly with the number of images. However, this is not the case for NNs, so MEPs could potentially by modeled with high resolutions, potentially leading to much higher convergence rates.

CHAPTER 6

Implicit Transfer Operator Learning

Based on the paper "Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics", work done by Mathias Schreiner, Ole Winter, and Simon Olsson, in proceedings for NeurIPS 2023. The focus of the *Transition1x* and *NeuralNEB* papers was to generate relevant data and employ Machine Learning (ML) models to accelerate Transition State (TS) search. TSs represent specific configurations on the Potential Energy Surface (PES) and serve as critical points for understanding reaction kinetics. By identifying the TSs we can unravel reaction mechanisms and calculate barrier heights, which, through the Arrhenius equation, can be used to estimate reaction rates.

An alternative approach for studying properties such as reaction rates involves directly simulating the system's evolution over time. Langevin dynamics is an important tool for such simulations. Here it is often assumed that the systems momentum experiences a rapid decay due to frequent interactions with the surrounding medium, operating on a time scale much faster than the sampling rate of frames in the simulated trajectory. This leads to dynamics, where the system is acted upon by a drift term, arising from the gradient of the PES and random forces that, through the central limit theorem, converges towards Gaussian noise between frames. This strategy enables realistic simulations of complex systems allowing us to study a variety of properties, some of which were previously revealed by the TSs. Such stochastic systems can evolve in a multitude of different ways, necessitating either numerous or extremely long trajectories to draw statistically significant conclusions. In reality, we are interested in a model that describes the evolution of the time dependent probability distribution of the system, $p(x,t)$, over configuration states, representing the behavior of the ensemble of the trajectories. As previously discussed, in the context of Langevin Dynamics, this is precisely the Fokker Planck equation, which we will write in terms of the Fokker-Planck operator $\mathcal{L}$

$$\frac{d}{dt}p(x,t) = \mathcal{L}p(x,t). \tag{6.1}$$

We can study the system at a later time by integrating the Fokker Planck operators action on the system:

$$p(x, t_0 + \tau) = p(x, t_0) + \int_{t_0}^{t_0+\tau} \mathcal{L}p(x, t)dt, \tag{6.2}$$

and since the Fokker-Planck operator is a linear operator, it allows for a closed solution of the integral:

$$p(x, t_0 + \tau) = T_\tau p(x, t_0), \tag{6.3}$$

where T_τ is called the transfer operator. This can be rewritten in its spectral form

$$T_\tau = \sum_{i=1}^{\infty} e^{\tau\lambda_i} |\psi_i\rangle\langle\phi_i| \tag{6.4}$$

where ϕ_i and ψ_i are left and right eigenfunctions of $\mathcal{L}$. This formulation reveals a rather straight-forward dependency on the integration time τ motivating the use of multiple time-resolutions as a potential powerful data augmentation for a conditional generative model of the form

$$p(x_\tau|x_0, \tau) = \sum_{i=0}^{\infty} e^{\kappa_i\tau} \langle\delta_{x_\tau}|\psi_i\rangle\langle\phi_i|\delta_{x_0}\rangle, \tag{6.5}$$

that implicitly learns the transfer-operator. Here, δ_x denotes the dirac-delta distribution. Apart from serving as a useful inductive bias, this method of data augmentation is easy to implement; it simply involves sampling a lag τ and a state from a trajectory x_t, and then train the model to predict $x_{t+\tau}$ from the same trajectory. This is exactly the motivation for and strategy in the paper introducing the Implicit Transfer Operator (ITO) framework *"Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics"* [27].

In the paper we train conditional Denoising Diffusion Probabilistic Models (DDPMs) models for learning transfer operators for various systems. In our first experiments we trained models on trajectories of Langevin Dynamics simulated on the Müller Brown potential. This is a theoretical 2-D potential commonly used as toy-potential for testing various methods related to the behavior of physical systems on PESs.

In order to test our hypothesis, that augmenting the training data with multiple time-scales would enhance the robustness of the models we designed two experiments and compared their results. In the first we trained a set of models using a fixed lag, meaning that each model was trained to propagate the system forward on a specific timescale. In the second experiment we trained models to learn multiple time-scales at once, by sampling various lag times at training. We simulated trajectories with the fixed-lag and stochastic-lag models and evaluated their performance using VAMP2-scores, comparing these scores with those from the original Langevin

Dynamics simulations. The stochastic-lag models outperformed the fixed-lag models across all timescales. Remarkably, a single model employing the data augmentation strategy would outperform multiple fixed-lag models on domains on which they were specialized.

In the next experiment we set out to evaluate the self-consistency of our stochastic-lag models according to the Chapman-Kolmogorov equations. Specifically whether $p(x_{N\tau} \mid x_0) = \prod_{i=1}^{N} p(x_{i\tau} \mid x_{(i-1)\tau})$, or, in other words, we wanted to verify that the final state in a trajectory $x_{N\tau}^{\text{anc}}$, obtained by ancestrally sampling a sequence of N intermediate states at time intervals τ is identical to the distribution of $x_{N\tau}^{\text{dir}}$ obtained by direct sampling from the same model with the longer time-step $N\tau$. In this experiment we train models on molecular dynamics simulations of alanine-dipeptide with a 1ps resolution. We plot transition densities projected on to the torsion angles ϕ and ψ on the backbone and show a strong consistency between samples at various intervals, whether sampling up to 512ps directly or ancestrally sampling increments of 1ps at a time. 512 ps of dynamics correspond to 10^6 integration steps in the Molecular Dynamics (MD) simulations that generated the data. We obtained our samples with 50 model evaluations by applying the improved sampling techniques of DDPMs involving probability flow ODE framework described earlier.

In our last experiment we evaluate the capabilities of our models to realistically simulate various systems of fast-folding proteins. These proteins are originally simulated using all-atom molecular dynamics in explicit solvent. However, we train our model using a coarse-grained representation based on the positions of the C_α atoms. We calculate free energies of folding as well as mean first passage times of both folding and unfolding, based on trajectories simulated by our models. We then compare these values with values derived from the original simulations and show good agreement between models and simulation.

An interesting avenue for continued research is generalising ITO models across chemical space. Currently, we can simulate training the data, but if ITO models could generalize from high-quality dynamics simulation dataset that includes various chemical systems, they could potentially provide a powerful for studying dynamics of unseen systems - essentially serving as a virtual chemical laboratory.

CHAPTER 7

Conclusion

In this thesis I provide a brief overview of central methods and concepts in the field of Machine Learning (ML) for Molecular Science and present my contributions to the field as a result of three years of Ph.D. studies at the Technical University of Denmark.

Accurate modeling of electronic-structure offers an invaluable insight into the microscopic behavior of molecules, paving the way for in-silico design of drugs and materials and the study of phenomena in chemistry, physics, and biology. While classical electronic structure methods are capable of accurate prediction of molecular properties, they are prohibitively expensive for large-scale exploration. Neural Networks (NNs) have turned out to be excellent emulators of such methods while operating orders of magnitude faster. ML methods open up for fascinating possibilities in computational chemistry, some of which I explore in my publications listed here:

Transition1x - a dataset for building generalizable reactive machine learning potentials. Though powerful function approximators, NNs has poor extrapolation capabilities, and they require vast amounts of relevant training data to perform well at a given task. At a microscopic scale, chemical reactions are rare events and datasets that rely on dynamical simulation or perturbation methods for their data-generation procedure often lack sufficient examples of reactive configurations. *Transition1x* uses Nudged Elastic Band (NEB) as an efficient sampling algorithm for relevant data for training NN models to work on reactive systems. Transition1x can hopefully serve as a resource facilitating exploration of reactive systems and reaction networks with NNs. The data generation procedure for Transition1x is straightforward and has been made publicly available to encourage further iterations of the dataset with larger systems or reactions involving different elements.

NeuralNEB – Neural Networks can find Reaction Paths Fast. This paper introduces the *NeuralNEB* algorithm which significantly accelerates Transition State (TS)-search by replacing expensive Density Functional Theory (DFT) calculations with cheap NN-based potentials. We trained models on various popular electronic structure datasets, and those trained on Transition1x significantly outperformed the others, both in terms of accuracy and convergence rates. This illustrates the importance of specialized datasets such as Transition1x in the literature. NeuralNEB would on average locate TSs 1350 times faster with a Mean Absolute Error (MAE)

of 0.23 eV and Root Mean Squared Error (RMSE) of 0.52 eV on barrier heights, when compared to NEB using traditional DFT. We also compared the NeuralNEB against Density-Functional Tight-Binding (DFTB)-based NEB. DFTB [131] is a popular "fast" potential for quick screening of large amounts of molecules, and it was equivalently outperformed by NeuralNEB in terms of speed, accuracy and convergence rates.

Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics. In this paper we introduce the *Implicit Transfer Operator (ITO)* framework which employs conditional Denoising Diffusion Probabilistic Models (DDPMs) along with a data augmentation scheme that allows models to learn dynamics at multiple time-scales. We train and test ITO models on a variety of dynamical systems, including Molecular Dynamics (MD) simulations, and demonstrate these models' ability to faithfully capture both fast and slow dynamics. By augmenting training data with samples at multiple time scales we provide an inductive bias, encouraging the model to implicitly learn the eigenfunctions and eigenvalues of the dynamics. We find that, using our data augmentation strategy, ITO models can, across timescales, outperform equivalently trained fixed-lag models on their specific domains. The ITO framework provides a virtual a microscopic laboratory that allows us to study and simulate dynamics of molecular systems. Avenues of further research includes generalising over temperatures, chemical space, and explicitly learning the eigenfunctions and eigenvalues of the transfer operator.

In summary, this thesis has been exploring some of the possibilities that is opening up as we integrate ML methods into computational quantum chemistry. The effort of merging these fields is an exciting ride, and it may slowly be revolutionizing quantum chemistry, leading to novel methods with far-reaching applications in industry, technology and scientific research.

Bibliography

[1] S Park et al. "Benefits of gardening activities for cognitive function according to measurement of brain nerve growth factor levels." In: *International Journal of Environmental Research and Public Health* 16.5 (2019).

[2] E. Rutherford. "LXXIX. The scattering of and particles by matter and the structure of the atom." In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 21 (125 1911). ISSN: 1941-5982. DOI: 10.1080/14786440508637080.

[3] W. Kohn and L. J. Sham. "Self-consistent equations including exchange and correlation effects." In: *Physical Review* 140 (4A 1965). ISSN: 0031899X. DOI: 10.1103/PhysRev.140.A1133.

[4] P. Hohenberg and W. Kohn. "Inhomogeneous electron gas." In: *Physical Review* 136 (3B 1964). ISSN: 0031899X. DOI: 10.1103/PhysRev.136.B864.

[5] D. R. Hartree. "The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part I Theory and Methods." In: *Mathematical Proceedings of the Cambridge Philosophical Society* 24 (1 1928). ISSN: 14698064. DOI: 10.1017/S0305004100011919.

[6] D. R. Hartree. "The Wave Mechanics of an Atom with a Non-Coulomb Central Field Part II Some Results and Discussion." In: *Mathematical Proceedings of the Cambridge Philosophical Society* 24 (1 1928). ISSN: 14698064. DOI: 10.1017/S0305004100011920.

[7] Frank Jensen. *Introduction to Computational Chemistry*. 2007. DOI: 10.1007/s00214-013-1372-6.

[8] David Chatfield. "Christopher J. Cramer: Essentials of Computational Chemistry: Theories and Models." In: *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* 108 (6 2002). ISSN: 1432-881X. DOI: 10.1007/s00214-002-0380-8.

[9] Pavlo O. Dral. *Quantum Chemistry in the Age of Machine Learning*. 2020. DOI: 10.1021/acs.jpclett.9b03664.

[10] Julia Westermayr et al. "Perspective on integrating machine learning into computational chemistry and materials science." In: *The Journal of Chemical Physics* 154 (23 June 2021), page 230903. ISSN: 0021-9606. DOI: 10.1063/5.0047760. URL: https://aip.scitation.org/doi/abs/10.1063/5.0047760.

[11] Stuart I Campbell, Daniel B Allan, and Andi M Barbour. "Machine learning for the solution of the Schrödinger equation." In: *Machine Learning: Science and Technology* 1 (1 April 2020), page 013002. ISSN: 2632-2153. DOI: 10.1088/2632-2153/AB7D30. URL: https://iopscience.iop.org/article/10.1088/2632-2153/ab7d30%20https://iopscience.iop.org/article/10.1088/2632-2153/ab7d30/meta.

[12] Jörg Behler and Michele Parrinello. "Generalized neural-network representation of high-dimensional potential-energy surfaces." In: *Physical review letters* 98.14 (2007), page 146401.

[13] Julia Westermayr and Philipp Marquetand. "Machine Learning for Electronically Excited States of Molecules." In: *Chemical Reviews* 121 (16 August 2021), pages 9873–9926. ISSN: 15206890. DOI: 10.1021/ACS.CHEMREV.0C00749. URL: https://pubs.acs.org/doi/full/10.1021/acs.chemrev.0c00749.

[14] Frank Noé et al. "Machine learning for molecular simulation." In: *Annual Review of Physical Chemistry* 71 (2020). ISSN: 15451593. DOI: 10.1146/annurev-physchem-042018-052331.

[15] Pavlo O. Dral. "Quantum chemistry assisted by machine learning." In: volume 81. 2020. DOI: 10.1016/bs.aiq.2020.05.002.

[16] Daniel Sheppard, Rye Terrell, and Graeme Henkelman. "Optimization methods for finding minimum energy paths." In: *The Journal of Chemical Physics* 128 (13 April 2008), page 134106. ISSN: 0021-9606. DOI: 10.1063/1.2841941. URL: https://aip.scitation.org/doi/abs/10.1063/1.2841941.

[17] Graeme Henkelman and Hannes Jónsson. "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points." In: *Journal of Chemical Physics* 113 (22 2000). ISSN: 00219606. DOI: 10.1063/1.1323224.

[18] S. R. Logan. "The origin and status of the arrhenius equation." In: *Journal of Chemical Education* 59 (4 1982). ISSN: 00219584. DOI: 10.1021/ed059p279.

[19] Paul M. Zimmerman. "Single-ended transition state finding with the growing string method." In: *Journal of Computational Chemistry* 36 (9 April 2015), pages 601–611. ISSN: 1096-987X. DOI: 10.1002/JCC.23833. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/jcc.23833%20https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.23833%20https://onlinelibrary.wiley.com/doi/10.1002/jcc.23833.

[20] Jon Baker. "An algorithm for the location of transition states." In: *Journal of Computational Chemistry* 7 (4 1986). ISSN: 1096987X. DOI: 10.1002/jcc.540070402.

[21] P. Culot et al. "A quasi-Newton algorithm for first-order saddle-point location." In: *Theoretica Chimica Acta* 82 (3-4 1992). ISSN: 00405744. DOI: 10.1007/BF01113251.

[22] Mathias Schreiner et al. “Transition1x - a dataset for building generalizable reactive machine learning potentials.” In: *Scientific Data* 9 (1 2022). ISSN: 20524463. DOI: 10.1038/s41597-022-01870-w.

[23] Mathias Schreiner et al. “NeuralNEB - Neural Networks can find Reaction Paths Fast.” In: *Machine Learning: Science and Technology* (November 2022). ISSN: 2632-2153. DOI: 10.1088/2632-2153/ACA23E. URL: https://iopscience.iop.org/article/10.1088/2632-2153/aca23e%20https://iopscience.iop.org/article/10.1088/2632-2153/aca23e/meta.

[24] P Langevin. “Sur la theorie du mouvement brownien.” In: *C.R. Acad. Sci., (Paris)* 146 (1908).

[25] A. D. Fokker. “Die mittlere Energie rotierender elektrischer Dipole im Strahlungsfeld.” In: *Annalen der Physik* 348 (5 1914). ISSN: 15213889. DOI: 10.1002/andp.19143480507.

[26] Stefan Klus, Ingmar Schuster, and Krikamol Muandet. “Eigendecompositions of Transfer Operators in Reproducing Kernel Hilbert Spaces.” In: *Journal of Nonlinear Science* 30 (1 2020). ISSN: 14321467. DOI: 10.1007/s00332-019-09574-z.

[27] Mathias Schreiner, Ole Winther, and Simon Olsson. “Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics.” In: (2023). arXiv: 2305.18046 [physics.chem-ph].

[28] Feliks Nüske et al. “Markov state models from short non-equilibrium simulations - Analysis and correction of estimation bias.” In: *Journal of Chemical Physics* 146 (9 2017). ISSN: 00219606. DOI: 10.1063/1.4976518.

[29] Leonard; Hrabovsky George Susskind. *The Theoretical Minimum*. Penguin Books, 2012.

[30] Eric Paquet and Herna L. Viktor. “Molecular dynamics, monte carlo simulations, and langevin dynamics: A computational review.” In: *BioMed Research International* 2015 (2015). ISSN: 23146141. DOI: 10.1155/2015/183918.

[31] Scott E. Feller et al. “Constant pressure molecular dynamics simulation: The Langevin piston method.” In: *The Journal of Chemical Physics* 103 (11 1995). ISSN: 00219606. DOI: 10.1063/1.470648.

[32] J. P. Bouchaud and R. Cont. “A Langevin approach to stock market fluctuations and crashes.” In: *European Physical Journal B* 6 (4 1998). ISSN: 14346028. DOI: 10.1007/s100510050582.

[33] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry, Third Edition*. 2007. DOI: 10.1016/B978-0-444-52965-7.X5000-4.

[34] Laura De Lorenzis and Alexamder Düster. *Modeling in Engineering Using Innovative Numerical Methods for Solids and Fluids*. Volume 599. 2020.

[35] M. von Smoluchowski. "Zur kinetischen Theorie der Brownschen Molekularbewegung und der Suspensionen [On the kinetic theory of Brownian motion and suspensions]." In: *Annalen der Physik* 326 (14 1906). ISSN: 1521-3889.

[36] A. Einstein. "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen." In: *Annalen der Physik* 322 (8 January 1905), pages 549–560. ISSN: 1521-3889. DOI: 10.1002/ANDP.19053220806. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/andp.19053220806%20https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19053220806%20https://onlinelibrary.wiley.com/doi/10.1002/andp.19053220806.

[37] D. J. Higham. *An algorithmic introduction to numerical simulation of stochastic differential equations*. Volume 43. 2001. DOI: 10.1137/S0036144500378302.

[38] Sang Gyu Kwak and Jong Hae Kim. "Central limit theorem: The cornerstone of modern statistics." In: *Korean Journal of Anesthesiology* 70 (2 2017). ISSN: 20057563. DOI: 10.4097/kjae.2017.70.2.144.

[39] Namiko Mitarai. "Diffusive and Stochastic Processes: Lecture Notes." For the course Diffusive and Stochastic Processes at KU. 2023.

[40] Maxwell James Clerk. *A treatise on electricity and magnetism*. Volume 9781108014038. 1873. DOI: 10.1017/CBO9780511709333.

[41] D J Griffiths. *Introduction to electrodynamics,Griffith-3ed.pdf*. 2010.

[42] A. Einstein. "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt." In: *Annalen der Physik* 322 (6 1905). ISSN: 15213889. DOI: 10.1002/andp.19053220607.

[43] Louis de Broglie. "Recherches sur la théorie des quanta." Ann. de Physique (10) 3, 22–128 (1925). PhD dissertation. Paris, France: Université de Paris, 1924.

[44] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. 2018. DOI: 10.1017/9781316995433.

[45] Max Born. "Zur Quantenmechanik der Stoßvorgänge." In: *Zeitschrift für Physik* 37 (12 1926). ISSN: 14346001. DOI: 10.1007/BF01397477.

[46] M. Born and R. Oppenheimer. "Zur Quantentheorie der Molekeln." In: *Annalen der Physik* 389 (20 1927). ISSN: 15213889. DOI: 10.1002/andp.19273892002.

[47] Prof. Barton Zwiebach. *Quantum Physics III Chapter 6: Adiabatic Approximation*. MIT OpenCourseWare, Physics Department. 2018. URL: https://ocw.mit.edu/courses/8-06-quantum-physics-iii-spring-2018/resources/mit8_06s18ch6/.

[48] W. Pauli. "Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren." In: *Zeitschrift für Physik* 31 (1 1925). ISSN: 00443328. DOI: 10.1007/BF02980631.

[49] J. C. Slater. “The theory of complex spectra.” In: *Physical Review* 34 (10 1929). ISSN: 0031899X. DOI: 10.1103/PhysRev.34.1293.

[50] J. R. Barber. “Spherical Harmonics.” In: volume 172. 2022. DOI: 10.1007/978-3-031-15214-6_25.

[51] Samuel Francis Boys. “Electronic wave functions - I. A general method of calculation for the stationary states of any molecular system.” In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 200 (1063 1950). ISSN: 0080-4630. DOI: 10.1098/rspa.1950.0036.

[52] J. C. Slater. “Atomic shielding constants.” In: *Physical Review* 36 (1 1930). ISSN: 0031899X. DOI: 10.1103/PhysRev.36.57.

[53] R. Ditchfield, W. J. Hehre, and J. A. Pople. “Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules.” In: *The Journal of Chemical Physics* 54 (2 September 2003), page 724. ISSN: 0021-9606. DOI: 10.1063/1.1674902. URL: https://aip.scitation.org/doi/abs/10.1063/1.1674902.

[54] A. D. Becke. “Density-functional exchange-energy approximation with correct asymptotic behavior.” In: *Physical Review A* 38 (6 1988). ISSN: 10502947. DOI: 10.1103/PhysRevA.38.3098.

[55] Axel D. Becke. “A new mixing of Hartree-Fock and local density-functional theories.” In: *The Journal of Chemical Physics* 98 (2 1993). ISSN: 00219606. DOI: 10.1063/1.464304.

[56] Chris Burges et al. “Learning to rank using gradient descent.” In: 2005. DOI: 10.1145/1102351.1102363.

[57] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. “Deep content-based music recommendation.” In: 2013.

[58] Steffen Rendle et al. “Neural Collaborative Filtering vs. Matrix Factorization Revisited.” In: 2020. DOI: 10.1145/3383313.3412488.

[59] Rana Alaa El Deen Ahmed, Manuel Fernández-Veiga, and Mariam Gawich. “Neural Collaborative Filtering with Ontologies for Integrated Recommendation Systems.” In: *Sensors* 22 (2 2022). ISSN: 14248220. DOI: 10.3390/s22020700.

[60] Xinran He et al. “Practical lessons from predicting clicks on ads at Facebook.” In: 2014. DOI: 10.1145/2648584.2648589.

[61] Omer Hanif et al. “Data Mining and Knowledge Discovery Series : RAPID MINER Data Mining Use Cases and Business Analytics Applications.” In: *CEUR Workshop Proceedings* 2345 (3 2019). ISSN: 16130073.

[62] OpenAI. *ChatGPT*. 2022. URL: %5Curl%7Bhttps://platform.openai.com/products/chatgpt%7D.

[63] F. Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological Review* 65 (6 1958). ISSN: 0033295X. DOI: 10.1037/h0042519.

[64] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors.” In: *Nature* 323 (6088 1986). ISSN: 00280836. DOI: 10.1038/323533a0.

[65] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database.” In: *AT&T Labs [Online].* 7 (2010).

[66] Christopher M. Bishop. “Bishop - Pattern Recognition And Machine Learning - Springer 2006.” In: *Antimicrobial agents and chemotherapy* 58 (12 2014). ISSN: 1098-6596.

[67] Diederik P Kingma and Jimmy Lei Ba. *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.*

[68] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic gradient descent with warm restarts.” In: 2017.

[69] S. Kullback and R. A. Leibler. “On Information and Sufficiency.” In: *The Annals of Mathematical Statistics* 22 (1 1951). ISSN: 0003-4851. DOI: 10.1214/aoms/1177729694.

[70] Diederik P. Kingma and Max Welling. “Auto-encoding variational bayes.” In: 2014.

[71] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. “Stochastic backpropagation and approximate inference in deep generative models.” In: volume 4. 2014.

[72] Michael I. Jordan et al. “Introduction to variational methods for graphical models.” In: *Machine Learning* 37 (2 1999). ISSN: 08856125. DOI: 10.1023/A:1007665907178.

[73] J. T. Ormerod and M. P. Wand. “Explaining variational approximations.” In: *American Statistician* 64 (2 2010). ISSN: 00031305. DOI: 10.1198/tast.2010.09058.

[74] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. *Variational Inference: A Review for Statisticians.* 2017. DOI: 10.1080/01621459.2017.1285773.

[75] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics.” In: *32nd International Conference on Machine Learning, ICML 2015* 3 (March 2015), pages 2246–2255. DOI: 10.48550/arxiv.1503.03585. URL: https://arxiv.org/abs/1503.03585v8.

[76] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models.” In: (). URL: https://github.com/hojonathanho/diffusion..

[77] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution.” In: volume 32. 2019.

[78] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models.” In: volume 2022-June. 2022. DOI: 10.1109/CVPR52688.2022.01042.

[79] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision.” In: volume 139. 2021.

[80] Jonas Oppenlaender. “The Creativity of Text-to-Image Generation.” In: 2022. DOI: 10.1145/3569219.3569352.

[81] Prafulla Dhariwal and Alex Nichol. “Diffusion Models Beat GANs on Image Synthesis.” In: volume 11. 2021.

[82] OpenAI. *DALL-E 2*. URL: https://openai.com/dall-e-2.

[83] StabilityAI. *Stable Diffusion*. URL: https://stability.ai/stable-diffusion.

[84] StabilityAI. *Midjourney*. URL: https://www.midjourney.com/.

[85] Emiel Hoogeboom et al. “Equivariant Diffusion for Molecule Generation in 3D.” In: *Proceedings of the 39th International Conference on Machine Learning*. Edited by Kamalika Chaudhuri et al. Volume 162. Proceedings of Machine Learning Research. PMLR, July 2022, pages 8867–8887. URL: https://proceedings.mlr.press/v162/hoogeboom22a.html.

[86] Bowen Jing et al. “Torsional Diffusion for Molecular Conformer Generation.” In: *arXiv preprint arXiv:2206.01729* (2022).

[87] Minkai Xu et al. “GEODIFF: A GEOMETRIC DIFFUSION MODEL FOR MOLECULAR CONFORMATION GENERATION.” In: 2022.

[88] John Ingraham et al. “Illuminating protein space with a programmable generative model.” In: *bioRxiv* (2022).

[89] Joseph L Watson et al. “Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models.” In: *bioRxiv* (2022).

[90] William Feller. “On the Theory of Stochastic Processes, with Particular Reference to Applications.” In: 2015. DOI: 10.1007/978-3-319-16859-3_42.

[91] J. L.W.V. Jensen. “Sur les fonctions convexes et les inegalites entre les valeurs moyennes.” In: *Acta Mathematica* 30 (1 1906). ISSN: 00015962. DOI: 10.1007/BF02418571.

[92] Alex Nichol and Prafulla Dhariwal. “Improved Denoising Diffusion Probabilistic Models.” In: volume 139. 2021.

[93] Brian D.O. Anderson. “Reverse-time diffusion equation models.” In: *Stochastic Processes and their Applications* 12 (3 1982). ISSN: 03044149. DOI: 10.1016/0304-4149(82)90051-5.

[94] S. C. Shiralashetti and Lata Lamani. “Numerical solution of stochastic ordinary differential equations using HAAR wavelet collocation method.” In: *Journal of Interdisciplinary Mathematics* 25 (2 2022). ISSN: 09720502. DOI: 10.1080/09720502.2021.1874085.

[95] Yang Song et al. “SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS.” In: 2021.

[96] Cheng Lu et al. “DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps.” In: *arXiv preprint arXiv:2206.00927* (2022).

[97] Aapo Hyvärinen. “Estimation of non-normalized statistical models by score matching.” In: *Journal of Machine Learning Research* 6 (2005). ISSN: 15337928.

[98] Yang Song et al. “Sliced Score Matching: A Scalable Approach to Density and Score Estimation.” In: volume 115. 2019.

[99] Steven Kearnes et al. “Molecular graph convolutions: moving beyond fingerprints.” In: *Journal of Computer-Aided Molecular Design* 30 (8 2016). ISSN: 15734951. DOI: 10.1007/s10822-016-9938-8.

[100] Justin Gilmer et al. “Neural Message Passing for Quantum Chemistry.” In: (2017).

[101] Davide Bacciu et al. “A Gentle Introduction to Deep Learning for Graphs.” In: *Neural Networks* 129 (December 2019), pages 203–221. DOI: 10.1016/j.neunet.2020.06.006. URL: http://arxiv.org/abs/1912.12693%20http://dx.doi.org/10.1016/j.neunet.2020.06.006.

[102] Franco Scarselli et al. “The graph neural network model.” In: *IEEE Transactions on Neural Networks* 20 (1 2009). ISSN: 10459227. DOI: 10.1109/TNN.2008.2005605.

[103] M. Withnall et al. “Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction.” In: *Journal of Cheminformatics* 12 (1 2020). ISSN: 17582946. DOI: 10.1186/s13321-019-0407-y.

[104] David Duvenaud et al. “Convolutional networks on graphs for learning molecular fingerprints.” In: volume 2015-January. 2015.

[105] Jörg Behler. *First Principles Neural Network Potentials for Reactive Simulations of Large Molecular and Condensed Systems*. 2017. DOI: 10.1002/anie.201703114.

[106] Oliver T. Unke et al. *Machine Learning Force Fields*. 2021. DOI: 10.1021/acs.chemrev.0c01111.

[107] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. “Convolutional networks and applications in vision.” In: 2010. DOI: 10.1109/ISCAS.2010.5537907.

[108] Nathaniel Thomas and Kai Kohlhoff. “Tensor field networks : Rotation- and translation-equivariant neural networks for 3D point clouds arXiv : 1802 . 08219v3 [cs . LG] 18 May 2018.” In: *arXiv preprint* (2018).

[109] Fabian B. Fuchs et al. “SE(3)-transformers: 3D roto-translation equivariant attention networks.” In: volume 2020-December. 2020.

[110] Marc Finzi et al. “Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data.” In: volume PartF168147-5. 2020.

[111] Kristof T Schütt et al. “Equivariant message passing for the prediction of tensorial properties and molecular spectra.” In: *Proceedings of Machine Learning Research* (July 2021), pages 9377–9388. ISSN: 2640-3498. URL: https://proceedings.mlr.press/v139/schutt21a.html.

[112] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E(n) Equivariant Graph Neural Networks.” In: volume 139. 2021.

[113] Brandon Anderson, Truong Son Hy, and Risi Kondor. “Cormorant: Covariant molecular neural networks.” In: volume 32. 2019.

[114] Arthur Kosmala et al. “Ewald-based Long-Range Message Passing for Molecular Graphs.” In: *International Conference on Machine Learning*. 2023. URL: https://api.semanticscholar.org/CorpusID:257405030.

[115] Jean-Pierre Dedieu. “Newton-Raphson Method.” In: *Encyclopedia of Applied and Computational Mathematics*. Edited by Björn Engquist. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pages 1023–1028. ISBN: 978-3-540-70529-1. DOI: 10.1007/978-3-540-70529-1_374. URL: https://doi.org/10.1007/978-3-540-70529-1_374.

[116] Baron Peters et al. “A growing string method for determining transition states: Comparison to the nudged elastic band and string methods.” In: *Journal of Chemical Physics* 120 (17 2004). ISSN: 00219606. DOI: 10.1063/1.1691018.

[117] Paul M. Zimmerman. “Single-ended transition state finding with the growing string method.” In: *Journal of Computational Chemistry* 36 (9 April 2015), pages 601–611. ISSN: 1096-987X. DOI: 10.1002/JCC.23833. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/jcc.23833%20https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.23833%20https://onlinelibrary.wiley.com/doi/10.1002/jcc.23833.

[118] Paul M. Zimmerman. “Growing string method with interpolation and optimization in internal coordinates: Method and examples.” In: *Journal of Chemical Physics* 138 (18 2013). ISSN: 00219606. DOI: 10.1063/1.4804162.

[119] Andrew Behn et al. “Incorporating linear synchronous transit interpolation into the growing string method: Algorithm and applications.” In: *Journal of Chemical Theory and Computation* 7 (12 2011). ISSN: 15499618. DOI: 10.1021/ct200654u.

[120] Graeme Henkelman, Blas P. Uberuaga, and Hannes Jónsson. “A climbing image nudged elastic band method for finding saddle points and minimum energy paths.” In: *The Journal of Chemical Physics* 113 (22 November 2000), page 9901. ISSN: 0021-9606. DOI: 10.1063/1.1329672. URL: https://aip.scitation.org/doi/abs/10.1063/1.1329672.

[121] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." In: volume 2017-December. 2017.

[122] Chuan Guo et al. "On calibration of modern neural networks." In: volume 3. 2017.

[123] Raghunathan Ramakrishnan et al. "Quantum chemistry structures and properties of 134 kilo molecules." In: *Scientific Data 2014 1:1* 1 (1 August 2014), pages 1–7. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.22. URL: https://www.nature.com/articles/sdata201422.

[124] Chengteh Lee, Weitao Yang, and Robert G. Parr. "Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density." In: *Physical Review B* 37 (2 1988). ISSN: 01631829. DOI: 10.1103/PhysRevB.37.785.

[125] Lars Ruddigkeit et al. "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17." In: *Journal of chemical information and modeling* 52.11 (2012), pages 2864–2875.

[126] Justin S. Smith et al. "Less is more: Sampling chemical space with active learning." In: *The Journal of Chemical Physics* 148 (24 May 2018), page 241733. ISSN: 0021-9606. DOI: 10.1063/1.5023802. URL: https://aip.scitation.org/doi/abs/10.1063/1.5023802.

[127] Jeng Da Chai and Martin Head-Gordon. "Long-range corrected hybrid density functionals with damped atom-atom dispersion corrections." In: *Physical Chemistry Chemical Physics* 10 (44 2008). ISSN: 14639076. DOI: 10.1039/b810189b.

[128] P. C. Hariharan and J. A. Pople. "The influence of polarization functions on molecular orbital hydrogenation energies." In: *Theoretica Chimica Acta* 28 (3 1973). ISSN: 00405744. DOI: 10.1007/BF00533485.

[129] Kristof T. Schütt et al. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions." In: (June 2017). URL: http://arxiv.org/abs/1706.08566.

[130] K. T. Schütt et al. "SchNet - A deep learning architecture for molecules and materials." In: *Journal of Chemical Physics* 148 (24 2018). ISSN: 00219606. DOI: 10.1063/1.5019779.

[131] Gotthard Seifert and Jan Ole Joswig. "Density-functional tight binding—an approximate density-functional theory method." In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 2 (3 May 2012), pages 456–465. ISSN: 1759-0884. DOI: 10.1002/WCMS.1094. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/wcms.1094%20https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1094%20https://wires.onlinelibrary.wiley.com/doi/10.1002/wcms.1094.

APPENDIX A

Transition1x - a dataset for building generalizable reactive machine learning potentials

Transition1x - a dataset for building generalizable reactive machine learning potentials

Mathias Schreiner[1], Arghya Bhowmik[2], Tejs Vegge[2], Jonas Busk[2], and Ole Winther[1, 3, 4]

[1]DTU Compute, Technical University of Denmark (DTU), 2800 Lyngby, Denmark
[2]DTU Energy, Technical University of Denmark, 2800 Lyngby, Denmark
[3]Department of Biology, University of Copenhagen (UCph), 2700 Copenhagen N, Denmark
[4]Genomic Medicine, Copenhagen University Hospital, Rigshospitalet, 2100 Copenhagen Ø, Denmark

ABSTRACT

Machine Learning (ML) models have, in contrast to their usefulness in molecular dynamics studies, had limited success as surrogate potentials for reaction barrier search. This is primarily because available datasets for training ML models on small molecular systems almost exclusively contain configurations at or near equilibrium. In this work, we present the dataset Transition1x containing 9.6 million Density Functional Theory (DFT) calculations of forces and energies of molecular configurations on and around reaction pathways at the ωB97x/6-31G(d) level of theory. The data was generated by running Nudged Elastic Band (NEB) with DFT on 10k organic reactions of various types while saving intermediate calculations. We train equivariant graph message-passing neural network models on Transition1x and cross-validate on the popular ANI1x and QM9 datasets. We show that ML models cannot learn features in transition state regions solely by training on hitherto popular benchmark datasets. Transition1x is a new challenging benchmark that will provide an important step towards developing next-generation ML force fields that also work far away from equilibrium configurations and reactive systems.

Background & Summary

ML models for molecular systems have accuracy comparable to quantum mechanical (QM) methods but the computational cost of classical interatomic potentials[1–5]. The development of such data-driven models has ushered in a new age in computational chemistry over the last few years[6–9]. ML potentials have been used for a variety of tasks such as structural optimization[10] or the study of finite-temperature dynamical properties through molecular dynamics[11]. ML potentials are especially suited for screening through large numbers of molecules or simulating systems that are too large for traditional QM methods due to a complexity scaling that is orders of magnitudes lower. The applicability of these models depends on the sampling of training data of chemical and structural space[12]. Fitting ML models to the entire potential energy surface (PES) requires lots of carefully selected data as the underlying electronic interaction between atoms is of a complex, quantum mechanical nature. Thus the focus remains on an efficient sampling strategy of the useful parts of the PES that are relevant to the application at hand. For example, models for optimization tasks should be trained on datasets including small perturbations to equilibrium geometries, and models for molecular dynamics (MD) simulations and reactive systems should be trained on datasets with high energy geometries and states that represent the making and breaking of bonds.

ML potentials that allow accurate modeling of general reaction barriers are challenging to train and only limited demonstrations have been shown to date. Acceptable accuracy has been achieved by focusing on single or few types of reactions involving small molecules with tractable dataset size[13–16] or by studying simple molecular dissociation[17]. ML models that can accurately predict PESs for unseen chemical reactions must be incredibly expressive and have access to training data that extensively samples structures from reactive and high-energy regions (compared to near-equilibrium geometries) of chemical space. Recently, the development of Neural Network (NN) architectures that learn representation and energy/force mapping[6] has tackled the problem of expressive models, but creating datasets with millions of data points sampled around reactions of various types allowing such models to generalize across a large number of reactions has remained an open challenge. Thus, ML potentials have not yet proved capable of accurate and general prediction of reaction barriers and transition states.

Sampling of rare transition events is efficiently done with the NEB method[18]. Here we propose a new dataset for building ML models capable of generalizing across a large variety of reaction PESs. We base our work on a dataset of reaction-product pairs from Grambow et al. 2020[19]. The original dataset contains a wide range of organic reactions representing bond changes between all possible combinations of H, C, N, and O atoms. We leverage NEB-based PES exploration as an efficient data

collection tool and prove its superiority compared to MD-based dataset preparation on reactive molecular configurations by testing the accuracy of ML models built from both types of data. Moreover, Transition1x is compatible with ANI1x in the level of DFT such that ML models can be trained on the two in conjunction to leverage both of their strengths.

Ultra-fast prediction of chemical reaction kinetics, especially for computational modeling of complex reaction networks, is groundbreaking for the entire field of chemical and molecular sciences. We believe that the Transition1x dataset will expedite the development and testing of universal reactive ML potentials that help the community achieve that goal.

Methods

Starting from a set of 11961 reactions[19] with reactants, transition states, and products, NEB is used to explore millions of molecular configurations in transition state regions, using DFT to evaluate forces and energies. The resulting DFT calculations are available in the Transition1x dataset. Figure 1 presents an overview of the workflow. Reactant and product are relaxed for any particular reaction before generating an initial path using Image Dependent Pair Potential (IDPP)[20]. Next, the minimal energy path (MEP) is optimized with NEB[18] and consecutively Climbing Image Nudged Elastic Band (CINEB)[21] until convergence. If the path converges, we save the DFT calculations from the iterations for which the current reaction path moved significantly.

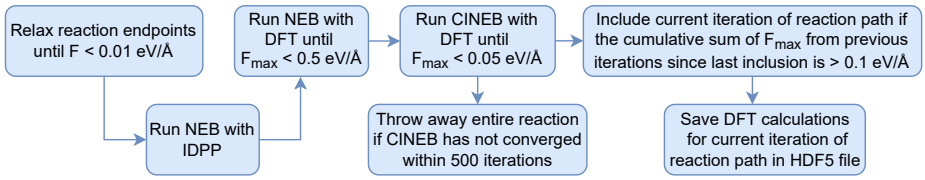


Figure 1. Overview of the data generation workflow. First, reactant and product are relaxed before generating an initial MEP guess with IDPP[20]. Next NEB[18] and CINEB[21] is run on the initial path until convergence. If the MEP does not converge within 500 iterations we discard the reaction, as unphysical configurations may have been encountered. If the reaction converges, all intermediate paths are saved in the dataset, as long as they are sufficiently different from previously saved paths.

Initial Data

The data generating procedure starts by taking an exhaustive database[19] of product-reactant pairs based on the GDB7 dataset[22]. Each reaction consists of up to seven heavy atoms including C, N, and O. The authors of this data used the Growing String Method (GSM)[23] with the ωB97X-D3[24]/def2-TZVP level of theory to generate reactants, products, and transition states for 11961 reactions using Qchem[25].

Density Functional Theory

For compatibility with ANI1x[26], the ωB97x[24] and 6-31G(d)[27] basis set is applied to perform all calculations in ORCA 5.0.2[28].

Optimizer

The BFGS optimizer[29] implemented in Atomic Simulation Environment (ASE)[30] with $\alpha = 70$ and a maximal step size of 0.03 Å is used for all optimization tasks, including relaxing endpoints and running both NEB and CINEB.

Initial Path Generation

Product and reactant geometries are relaxed in the potential before running NEB. The configuration is considered relaxed once the norm of the forces in configurational space is less than a threshold of 0.01 eVÅ^{-1}. After relaxing the endpoints an initial path is proposed, built from two segments – one interpolated from the reactant to the transition state from the original data, and another interpolated from the transition state to the product. Next, the initial path is minimised with NEB using IDPP[20], a potential specifically designed to generate physically realistic reaction paths for NEB at a low computational cost. Finally, the path is proposed as the initial MEP in the DFT potential.

Nudged Elastic Band

NEB[18] is a double ended search method for finding MEPs connecting reactant and product states. It works by iteratively improving an initial guess for the MEP by using information about the PES as calculated by some potential. NEB represents the path as a series of configurations called images connected with an artificial spring force. The energy of the path is minimized by iteratively nudging it in the direction of the force perpendicular to it until convergence. After the path has converged, there is no guarantee that the maximal energy image represents the correct transition state as the maximal energy image may not

correspond with the true maximum along the path. CINEB[21] is an improvement to the NEB algorithm as it imposes, as an additional condition to the convergence, that the maximal energy image has to lie at a maximum. It does so by, in each iteration, letting the image with the maximal energy climb freely along the reaction path. Running CINEB from the beginning, however, can interfere with the optimizer and result in slow (or wrong) convergence of the MEP as the climbing image can pull the current path off the target MEP if the paths are not close. Therefore, first, the path is relaxed with regular NEB until the maximal perpendicular force to the MEP is below a threshold of 0.5 eVÅ^{-1}. At this point, NEB has usually found the qualitatively correct energy valley, and further optimization only nudges the path slightly while finding the bottom. At this point, CINEB is turned on to let the highest energy image climb along the path until it finds an energy maximum. CINEB is run until the path has been relaxed fully and the maximal perpendicular force on the path does not exceed 0.05 eVÅ^{-1}. This threshold was chosen as a compromise between having accurate reaction paths in the dataset and limiting redundant DFT calculations. No further refinement of the transition states was done at this point as the goal is to generate a dataset of molecular configurations close to reaction pathways rather than finding accurate transition states. Ten images were used to represent all reaction paths and the spring constant between them was 0.1 eVÅ^{-2}.

Data selection

When running NEB, unphysical configurations are often encountered in reactions that do not converge. Such images in the data will interfere with model when training, and therefore those reactions are discarded entirely. There are 10073 converged reactions in Transition1x. In the final steps of NEB, the molecular geometries of images are similar between each iteration as the images are nudged only slightly close to convergence. Data points should be spread out so that models do not overfit to specific regions of the data. Updated paths are only included in the dataset if they are significantly different from previous ones. The maximal perpendicular force, F_{max}, to the path is a proxy for how much the path moves between iterations. Once the cumulative sum of F_{max} from previous iterations, since the last included path, exceeds 0.1 eVÅ^{-1} the current path is included. This means that often in the first iterations of NEB every path is included, but as we move towards convergence new data points are included at a lower frequency.

Model and Training

To validate the dataset, we train and evaluate PaiNN[31] models on Transition1x, QM9x and ANI1x and compare the their performances. PaiNN is an equivariant Message Passing Neural Network (MPNN)[32] model specifically designed to predict properties of molecules and materials. Forces are calculated as the negative gradient of the energy wrt. the Cartesian coordinates of the atoms rather than as a direct output from the model. This ensures consistent forces. We used a cut-off distance of 5 Å to generate the molecular graph neighborhood, three message-passing steps, and 256 neurons in each hidden layer of the model. The model was trained using the ADAM[33] optimizer and an initial learning rate of 10^{-4}. During training, the learning rate was decreased by 20% if no improvement was seen on training data for 10^4 batches. The loss is a combination of a squared error loss on force and energy. The force error is the Euclidian distance between the predicted and the true force vector divided by the number of atoms in the molecule, as otherwise, the force term would contribute more to the loss on bigger molecules. All datasets are stratified by molecular formula such that no two configurations that come from different data splits are constituted of the same atoms. Test and validation data each consist of 5% of the total data and are chosen such that configurations contain all heavy atoms (C, N, O). Potentially, models can learn fundamental features faster from simpler molecules, therefore, all molecules with less than three heavy atom types are kept in the training data. The models are trained on the training data with early stopping on the validation data, and we report the mean and standard deviation of Root Mean Square Error (RMSE) and Mean Average Error (MAE) from the evaluation of test data.

QM9 and QM9x

QM9[34] consists of DFT calculations of various properties for 135k small organic molecules in equilibrium configurations. All molecules in the dataset contain up to 9 heavy atoms, including C, N, O, and F. QM9 is ubiquitous as a benchmark for new QM methods, and to enable direct comparison with Transition1x, all geometries from the QM9 dataset is recalculated with the appropriate level of DFT. Since configurations in the original QM9[34] are not relaxed in our potential, there will be forces on some configurations. All recalculated geometries are saved in a dataset that we shall refer to as QM9x.

ANI1x

ANI1x[26] is a dataset of off-equilibrium molecular configurations generated by perturbing equilibrium configurations using pseudo molecular dynamics. Data is included or rejected from the dataset based on the Query by Comittee (QbC) algorithm. In QbC an ensemble (or committee) of models is trained on the dataset, and the relevance of new proposed data is assessed through the variance of the ensemble's predictions without having to perform expensive calculations on the data. It is assumed that data points will contribute new information to the dataset if the committee disagrees. It is cheaper to evaluate the committee on data than running DFT calculations, so it is possible to screen many candidate configurations before calculating force and energy

with more expensive methods. The dataset is generated by alternating between training models and expanding the dataset. The procedure resulted in force and energy calculations for approximately 5 million configurations containing C, O, N, and H.

Data Records

Data records for Transition1x are available in a single Hierarchical Data Format (HDF5)[35] file; Transition1x.h5, hosted by figshare[36]. It can be downloaded at https://doi.org/10.6084/m9.figshare.19614657.v4 or through the repository https://gitlab.com/matschreiner/Transition1x. The HDF5 file structure is as shown in Figure 2. The parent file has four groups, one group contains all data and the three other groups contain symbolic links to the train, test, and validation data – these are the data splits used in this paper. In each data split group, there is a group for each chemical formula under which there is a subgroup for each reaction with the corresponding atoms. Each reaction has four datasets; the atomic numbers of the reaction, the energies of the configurations, the forces acting on the individual atoms, and the positions of atoms. For a reaction with m atoms where we have saved n images, the *atomic_numbers* dataset will have dimensions $(m,)$, one for each atom. The energy dataset will have dimensions $(n,)$, one energy per configuration. The force and position datasets will have dimensions $(n,m,3)$ as we need three components of position and force for each of m atoms in n configurations. Under each reaction group, there is a child group for reactant, transition state, and product that follow the same structure as described above with $n = 1$. Products from some reactions are reactants for the next, and they can be linked with a hash value available for each product, transition state, and reactant in the hash dataset. The data has been uploaded to figshare, and there is a git repository with data loaders that can turn the HDF5 file into an ASE database or save the configurations as .xyz files.

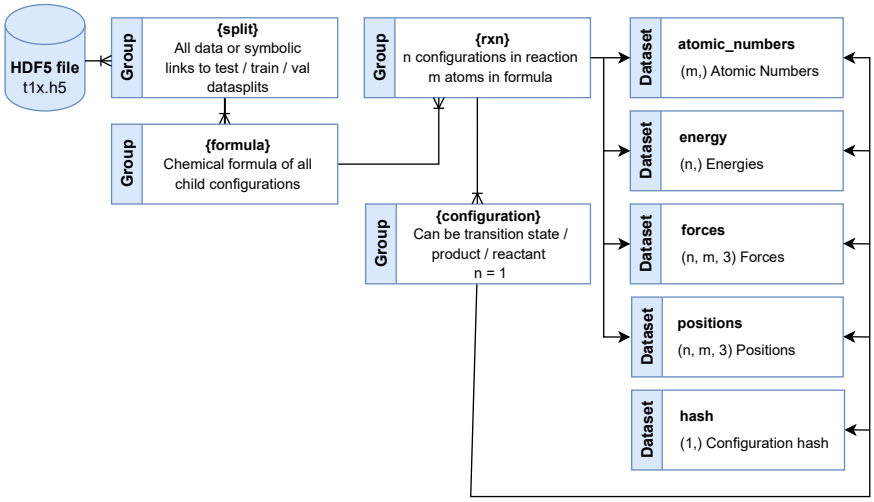


Figure 2. Structure of Transition1x HDF5 file. Parent groups are data/train/val/test. The data group contains all configurations in the set, and the train/val/test groups contain symlinks to the suggested data splits used in this paper. Each split has a set of chemical formulas unique to that split, and each formula contains all reactions with the given atoms. Finally, energy and force calculations can be accessed from the reaction groups for all intermediate configurations, including transition state, product, and reactant.

Data records for QM9x are also available in a HDF5 file; QM9x.h5, hosted by figshare[37]. It can be downloaded at https://doi.org/10.6084/m9.figshare.20449701.v2 or through the repository https://gitlab.com/matschreiner/QM9x. The HDF5 file structure is shown in Figure 3. Energy and force calculations for all configurations in the QM9x dataset consisting of a certain combination of atoms can be accessed as datasets through the formula group. The dimensions and structure of these datasets follow the same logic as described above.

Technical Validation

In figure 4 we show the MEP for a reaction involving C3H7O2 and the convergence of NEB for it. Often the barrier grows initially after turning on the climbing image because we start maximizing the energy in a new degree of freedom. NEB

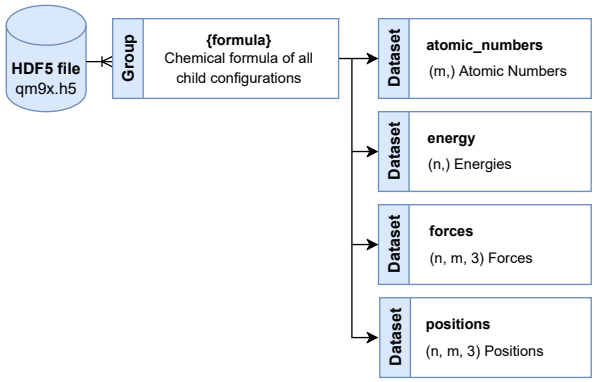


Figure 3. Structure of QM9x HDF5 file. Energy and force calculations for all configurations in the QM9x dataset consisting of a certain combination of atoms can be accessed as datasets through the formula group.

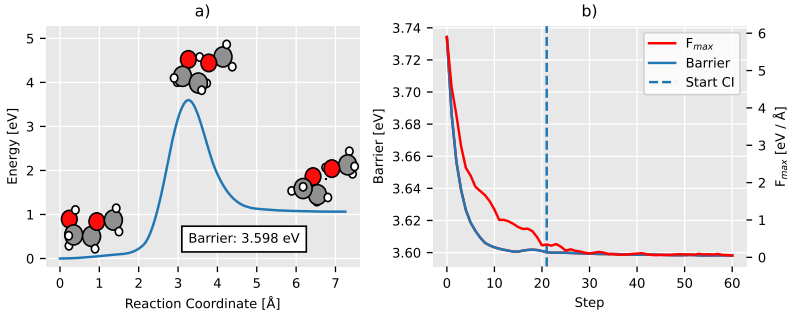


Figure 4. Plot of NEB convergence on example reaction. Panel a) displays the final MEP with reactant, transition state, and product plotted on top with H, C, and O in white, black, and red, respectively. On the x-axis; the reaction coordinate – distance along the reaction path in configurational space, measured in Å. On the y-axis; the difference in potential energy between reactant and current configuration. Panel b) displays the convergence of NEB. On the x-axis; iterations of NEB. On the y-axis; force in eVÅ^{-1} and energy barrier in eV at the current step. F$_{max}$, shown in red, is the maximal perpendicular force acting on any geometry along the path, and Barrier, shown in blue, is the height of the energy barrier found at the current step. Moving right in the plot both F$_{max}$ converges towards zero as NEB finds the saddle point, and the Barrier converges towards the final value of 3.6 eV that can be seen in panel a.

converged on 10073 out of 11961 reactions, and 89 percent of the converged reactions did so within the first 200 iterations. To ensure the cleanliness of the data, we choose to discard all reactions that do not converge – these reactions often contain unphysical structures that do more harm than good as training data.

The dataset includes a wide range of organic reactions. All reactions contain up to 7 heavy atoms including C, N, and O, and up to six bond changes where bonds are breaking and forming between all combinations of heavy atoms. Detailed analysis of the number of bond changes per reaction, number of bond changes involving specific pairs of atoms, the spread of activation energies, and SMARTS strings describing reactive centers of the reactions, is included in the original work Grambow et al. 2020[19].

The perpendicular force drops off rapidly when running NEB and so does the variation in data between iterations as the path is nudged less between iterations towards convergence of the algorithm. F_{max} is used as a proxy for how much the path moves between iterations and once the cumulative F_{max} since the last included path exceeds a threshold of 0.1 eV/Å, the path is

included in the dataset.

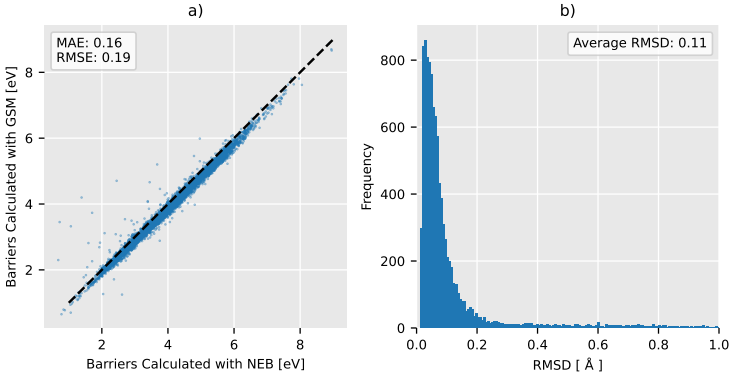


Figure 5. Comparison of transition states and barriers found in this work with NEB and the 6-31G(d) basis set, and in the original work with GSM and the def2-mSVP basis set. Panel (a) displays energies in eV for all transition states calculated using NEB on the x-axis and GSM on the y-axis. Panel (b) displays a histogram of Root Mean Square Deviation (RMSD) between transition states found.

The transition states found with NEB correspond to the transition states from the original GSM data with a MAE of 0.16 eV, RMSE of 0.19 eV, and an average Root Mean Square Deviation (RMSD) of 0.11Å. See Figure 5 in the Appendix for details. Barrier energies match, but the NEB energies tend to be shifted higher. Generally, it is easier to describe electron clouds around relaxed configurations than around transition states where bonds are breaking and other complex interactions take place. Therefore, the more expressive basis set enables us to relax configurations around transition states further than around equilibrium states which results in lower barrier heights. There are more outliers above the $x = y$ line than below it which indicates that GSM was caught in suboptimal reaction pathways more often than NEB.

Figure 6 displays the distribution of forces on each type of atom in Transition1x compared with ANI1x and QM9. Interestingly, even though geometries in Transition1x are further away from equilibrium than in ANI1x (regions between equilibria are actively sought out in Transition1x), the distribution of forces on ANI1x has flatter, wider tails signifying higher variance in forces. Moreover, Transition1x cusps at zero whereas ANI1x maxima lie further out. Large forces are not necessarily involved when dealing with reactive systems. When reaction pathways are minimized, forces are minimized too in all but one degree of freedom. The transition states contribute as much to the force distribution as equilibrium configurations, as the transition state is a saddle point with no net force on any atoms. ANI1x has no inherent bias towards low forces on geometries as it explores configurational space with pseudo-MD and therefore, even though the configurations are closer to equilibrium we see a higher variance in forces. On the heavy atoms, the tails are qualitatively equal between ANI1x and Transition1x, trailing off exponentially, but it is different for hydrogen. In ANI1x, forces on Hydrogen trail off exponentially as with the other atoms, but for Transition1x there is a sudden drop of the distribution. Hydrogen atoms are often at the outskirts of the molecules and are relatively free to move compared to heavier atoms on the backbone. In the case of the Transition1x data generation procedure, energy and forces are minimized, and therefore Hydrogen atoms do not experience large forces as they have lots of freedom to relax in the geometry. In ANI1x, configurations are generated by perturbing the geometries randomly, and hydrogen atoms might end up with unrealistically large forces on them. This might be a general problem with ANI1x and also a reason why ANI1x is not a proper dataset to learn reaction mechanisms.

Even though ANI1x has a wider distribution of forces, the inter-atomic distances between pairs of heavy atoms are less varied than in Transition1x. Figure 7 displays the distribution of distances between pairs of heavy atoms for Transition1x, ANI1x, and QM9x. For QM9x, a dataset of only equilibrium configurations, some inter-atomic distances are not present at all. Distances are measured in units of r_0, the single bond equilibrium distance between the atoms in the smallest possible molecule constructed out of the two. For example, in the case of "C, C" we measure in units of the distance between carbon atoms in ethane. Many of the more extreme inter-atomic distances in Transition1x are difficult to produce by the normal mode sampling

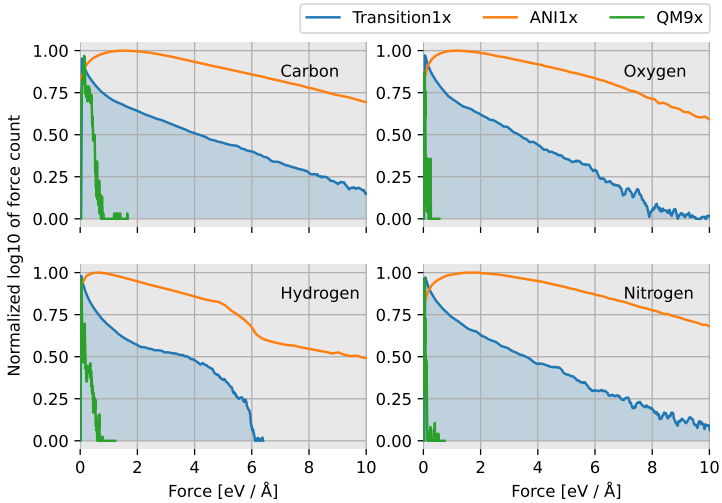


Figure 6. Distribution of forces acting on atom-types in each dataset. The x-axis is the force measured in eV/Å. The y-axis is the base 10 logarithm of the count of forces in each bin, normalized over the full domain so that all sets can be compared. In blue; Transition1x. In yellow; ANI1x. In green QM9x.

technique of ANI1x as many atoms would randomly have to move such that the whole molecule moves along a low-energy valley. However, because NEB samples low-energy valleys by design, we discover likely molecules with inter-atomic distances that are otherwise energetically unfavorable.

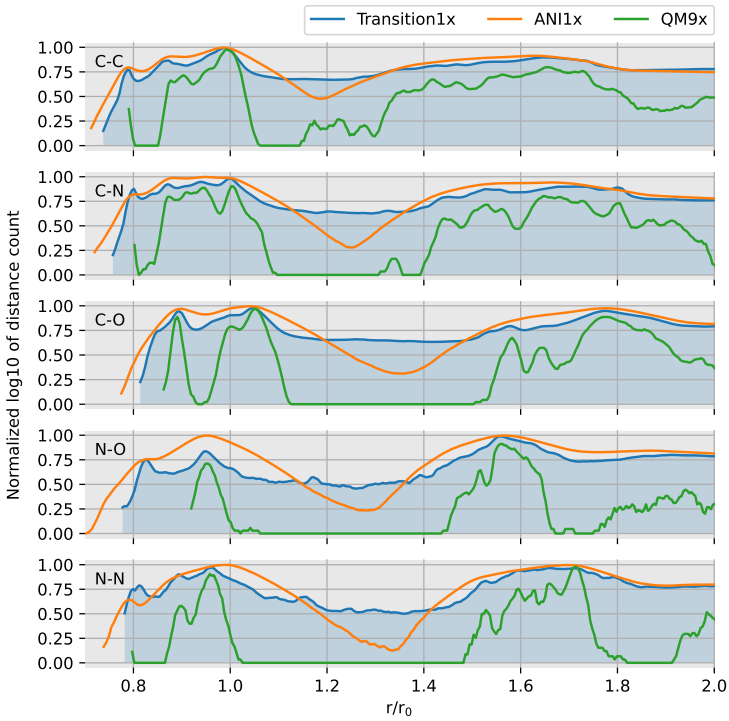


Figure 7. Distribution of interatomic distances between heavy atoms in each dataset. A configuration with n heavy atoms contributes with $n(n-1)/2$ distances in the count. On the y-axis; the log frequency of interatomic distance, normalized between 0 and 1 for comparison as datasets vary in size. On the x-axis; distance given in units of r_0 where r_0 is the equilibrium bond length for a single bond between the smallest possible stable molecule that can be made with the atoms in question. In blue; Transition1x. In yellow; ANI1x. In green; QM9x, recalculated using our level of theory.

		Energy [eV]		Forces [eV/Å]	
Trained on	Tested on	RMSE	MAE	RMSE	MAE
ANI1x		0.629 (11)	0.495 (10)	0.593 (18)	0.331 (7)
Transition1x	Transition States	**0.112 (3)**	**0.075 (1)**	**0.158 (1)**	**0.089 (1)**
QM9x		3.132 (23)	2.957 (25)	0.637 (15)	0.261 (4)
ANI1x		**0.042(8)**	**0.023(2)**	**0.063(2)**	**0.036(1)**
Transition1x	ANI1x	0.362(2)	0.227(9)	0.42(3)	0.176(3)
QM9x		3.06(4)	2.319(11)	1.99(3)	1.284(7)
ANI1x		0.65(7)	0.29(1)	0.57(3)	0.16(6)
Transition1x	Transition1x	**0.068(3)**	**0.04(1)**	**0.12(1)**	**0.053(1)**
QM9x		1.72(3)	1.21(3)	0.476(5)	0.23(2)
ANI1x		0.134(1)	0.124(1)	0.057(1)	0.031(1)
Transition1x	QM9x	0.111(3)	0.074(3)	0.08(1)	0.047(0)
QM9x		**0.032(5)**	**0.015(1)**	**0.011(3)**	**0.006(0)**

Table 1. Test results of PaiNN models trained on ANI1x, QM9x, Transition1x. We report RMSE and MAE on energy and forces. Force error is the component-wise error between the predicted and true force vector. The test sets have been constructed such that all configurations contain C, N, O, and H, and such that no formula has been seen previously in the training data. We show the best performing model in bold in each test-setup.

		Energy [eV]		Forces [eV/Å]	
Trained on	Tested on	RMSE	MAE	RMSE	MAE
ANI1x		0.629 (11)	0.495 (10)	0.593 (18)	0.331 (7)
Transition1x	Transition States	**0.112 (3)**	**0.075 (1)**	**0.158 (1)**	**0.089 (1)**
QM9x		3.132 (23)	2.957 (25)	0.637 (15)	0.261 (4)
ANI1x		**0.044(5)**	**0.023(1)**	**0.041(1)**	**0.016(0)**
Transition1x	ANI1x	0.365(17)	0.226(8)	0.321(25)	0.078(1)
QM9x		3.042(13)	2.313(11)	1.314(8)	0.559(2)
ANI1x		0.628(63)	0.289(13)	0.542(118)	0.13(5)
Transition1x	Transition1x	**0.102(2)**	**0.048(1)**	**0.102(1)**	**0.046(1)**
QMx		2.613(18)	1.421(11)	0.433(3)	0.199(1)
ANI1x		0.134(1)	0.124(1)	0.051(1)	0.025(1)
Transition1x	QM9x	0.111(2)	0.074(3)	0.072(1)	0.038(0)
QM9x		**0.04(2)**	**0.015(1)**	**0.014(0)**	**0.005(0)**

Table 2. Test results of PaiNN models trained on ANI1x, QM9x, Transition1x. We report RMSE and MAE on energy and forces. Force error is the component-wise error between the predicted and true force vector. The test sets have been constructed such that all configurations contain C, N, O, and H, and such that no formula has been seen previously in the training data. We show the best performing model in bold in each test-setup.

We test all resulting models against the test data from each dataset and Transition States from the test reactions. Table 2 displays the results. It is clear from their evaluation of Transition1x and transition states, that models trained on ANI1x do not have sufficient data in transition state regions to properly learn the complex interactions present here. ANI1x has a broad variety of chemical structures, but many of the fundamental interactions found in ANI1x are present in Transition1x, which is why models trained on Transition1x perform better on ANI1x than vice versa. In general, the PES of a set of atoms is an incredibly complex function of quantum mechanical nature. Models trained on QM9x do not perform well on either Transition1x or ANI1x. This is as expected as QM9x contains only equilibrium (or very close to equilibrium in the new potential) structures, so the models trained on it have not seen any of the out-of-equilibrium interactions that are present in the more challenging datasets of ANI1x and Transition1x.

Transition state data is required if we want to replace DFT with cheap ML potentials in algorithms such as NEB[38] or GSM, or train molecular dynamics models to work in transition state regions. NNs are phenomenal function approximators, given sufficient training examples, but they do not extrapolate well. Training examples spanning the whole energy surface are needed to train reliable and general-purpose ML models. Transition1x is a new type of dataset that explores different regions of chemical space than other popular datasets and it is highly relevant as it expands on the completeness of available data in the literature.

Code availability

There are download scripts and data loaders available in the repositories https://gitlab.com/matschreiner/Transition1x and https://gitlab.com/matschreiner/QM9x. See the repositories and README for examples and explanations of how to use the scripts and datasets.

All electronic structure calculations were computed with the ORCA electronic structure packages, version 5.0.2. All NEB calculations were computed with ASE version 3.22.1. Scripts for calculating, gathering and filtering data can be found in the Transition1x repository. `scripts/neb.py` takes reactant, product, output directory, and various optional arguments and runs NEB on the reaction while saving all intermediate calculations in an ASE database in the specified output directory. `scripts/combine_dbs.py` takes an output path for the HDF5 file and a JSON-list of all output directories produced by running the previous script and combines them in the HDF5 file as described in the paper. See the repository for how to install, specific commands, options, and further documentation.

Author contributions statement

M.S., A.B., T.V, and O.W. conceived the study. M.S. wrote the code, conducted the experiments, and wrote the majority of the article. A.B. and M.S. wrote background and summary, A.B. and O.W. provided supervision and reviewed the article, and J.B. reviewed the article and provided helpful discussions.

Competing interests

The authors declare no competing interests.

Acknowledgements

The authors acknowledge support from the Novo Nordisk Foundation (SURE, NNF19OC0057822) and the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 957189 (BIG-MAP) and No. 957213 (BATTERY2030PLUS).

Ole Winther also receives support from Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606) and the Pioneer Centre for AI, DNRF grant number P1.

References

1. Faber, F. A. *et al.* Prediction errors of molecular machine learning models lower than hybrid dft error. *J. Chem. Theory Comput.* **13**, 5255–5264, 10.1021/ACS.JCTC.7B00577/SUPPL_FILE/CT7B00577_SI_001.PDF (2017).
2. Westermayr, J., Gastegger, M., Schütt, K. T. & Maurer, R. J. Perspective on integrating machine learning into computational chemistry and materials science. *The J. Chem. Phys.* **154**, 230903, 10.1063/5.0047760 (2021).
3. Campbell, S. I., Allan, D. B. & Barbour, A. M. Machine learning for the solution of the schrödinger equation. *Mach. Learn. Sci. Technol.* **1**, 013002, 10.1088/2632-2153/AB7D30 (2020).
4. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. review letters* **98**, 146401 (2007).
5. Westermayr, J. & Marquetand, P. Machine learning for electronically excited states of molecules. *Chem. Rev.* **121**, 9873–9926, 10.1021/ACS.CHEMREV.0C00749 (2021).
6. Unke, O. T. *et al.* Machine learning force fields. *Chem. Rev.* **121**, 10142–10186 (2021).
7. Behler, J. Four generations of high-dimensional neural network potentials. *Chem. Rev.* **121**, 10037–10072 (2021).
8. Huang, B. & von Lilienfeld, O. A. Ab initio machine learning in chemical compound space. *Chem. reviews* **121**, 10001–10036 (2021).
9. Deringer, V. L. *et al.* Gaussian process regression for materials and molecules. *Chem. Rev.* **121**, 10073–10141 (2021).
10. Kaappa, S., Larsen, C. & Jacobsen, K. W. Atomic structure optimization with machine-learning enabled interpolation between chemical elements. *Phys. Rev. Lett.* **127**, 10.1103/PhysRevLett.127.166001 (2021).
11. Wang, J., Shin, S. & Lee, S. Interatomic potential model development: Finite-temperature dynamics machine learning. *Adv. Theory Simulations* **3**, 1900210, 10.1002/ADTS.201900210 (2020).
12. von Lilienfeld, O. A., Müller, K. R. & Tkatchenko, A. Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem. 2020 4:7* **4**, 347–358, 10.1038/s41570-020-0189-9 (2020).
13. Lu, X., Meng, Q., Wang, X., Fu, B. & Zhang, D. H. Rate coefficients of the h+ h2o2→ h2+ ho2 reaction on an accurate fundamental invariant-neural network potential energy surface. *The J. chemical physics* **149**, 174303 (2018).
14. Young, T. A., Johnston-Wood, T., Deringer, V. L. & Duarte, F. A transferable active-learning strategy for reactive molecular force fields. *Chem. science* **12**, 10944–10955 (2021).
15. Manzhos, S. & Carrington Jr, T. Neural network potential energy surfaces for small molecules and reactions. *Chem. Rev.* **121**, 10187–10217 (2020).
16. von Rudorff, G. F., Heinen, S. N., Bragato, M. & von Lilienfeld, O. A. Thousands of reactants and transition states for competing e2 and s2 reactions. *Mach. Learn. Sci. Technol.* **1**, 045026, 10.1088/2632-2153/ABA822 (2020).
17. Malshe, M. *et al.* Theoretical investigation of the dissociation dynamics of vibrationally excited vinyl bromide on an ab initio potential-energy surface obtained using modified novelty sampling and feedforward neural networks. ii. numerical application of the method. *The J. chemical physics* **127**, 134105 (2007).
18. Sheppard, D., Terrell, R. & Henkelman, G. Optimization methods for finding minimum energy paths. *The J. Chem. Phys.* **128**, 134106, 10.1063/1.2841941 (2008).
19. Grambow, C. A., Pattanaik, L. & Green, W. H. Reactants, products, and transition states of elementary chemical reactions based on quantum chemistry. *Sci. Data* **7**, 10.1038/s41597-020-0460-4 (2020).

20. Smidstrup, S., Pedersen, A., Stokbro, K. & Jónsson, H. Improved initial guess for minimum energy path calculations. *The J. Chem. Phys.* **140**, 214106, 10.1063/1.4878664 (2014).

21. Henkelman, G., Uberuaga, B. P. & Jónsson, H. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The J. Chem. Phys.* **113**, 9901, 10.1063/1.1329672 (2000).

22. Ruddigkeit, L., Van Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *J. chemical information modeling* **52**, 2864–2875 (2012).

23. Zimmerman, P. M. Single-ended transition state finding with the growing string method. *J. Comput. Chem.* **36**, 601–611, 10.1002/JCC.23833 (2015).

24. Chai, J. D. & Head-Gordon, M. Systematic optimization of long-range corrected hybrid density functionals. *The J. Chem. Phys.* **128**, 084106, 10.1063/1.2834918 (2008).

25. Epifanovsky, E. *et al.* Software for the frontiers of quantum chemistry: An overview of developments in the q-chem 5 package. *The J. Chem. Phys.* **155**, 084801, 10.1063/5.0055522 (2021).

26. Smith, J. S., Nebgen, B., Lubbers, N., Isayev, O. & Roitberg, A. E. Less is more: Sampling chemical space with active learning. *The J. Chem. Phys.* **148**, 241733, 10.1063/1.5023802 (2018).

27. Ditchfield, R., Hehre, W. J. & Pople, J. A. Self-consistent molecular-orbital methods. ix. an extended gaussian-type basis for molecular-orbital studies of organic molecules. *The J. Chem. Phys.* **54**, 724, 10.1063/1.1674902 (2003).

28. Neese, F., Wennmohs, F., Becker, U. & Riplinger, C. The orca quantum chemistry program package. *The J. Chem. Phys.* **152**, 224108, 10.1063/5.0004608 (2020).

29. Broyden, C. G. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA J. Appl. Math.* **6**, 76–90, 10.1093/IMAMAT/6.1.76 (1970).

30. Larsen, A. H. *et al.* The atomic simulation environment—a python library for working with atoms. *J. Physics: Condens. Matter* **29**, 273002, 10.1088/1361-648X/AA680E (2017).

31. Schütt, K. T., Schütt, S., Unke, O. T. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *Proc. Mach. Learn. Res.* 9377–9388 (2021).

32. Bacciu, D., Errica, F., Micheli, A. & Podda, M. A gentle introduction to deep learning for graphs. *Neural Networks* **129**, 203–221, 10.1016/j.neunet.2020.06.006 (2019).

33. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization.

34. Ramakrishnan, R., Dral, P. O., Rupp, M. & Lilienfeld, O. A. V. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data 2014 1:1* **1**, 1–7, 10.1038/sdata.2014.22 (2014).

35. The HDF Group. Hierarchical data format version 5 (2000-2010).

36. Schreiner, M. Transition1x. Figshare., https://doi.org/10.6084/m9.figshare.19614657.v4 (2022).

37. Schreiner, M. QM9x. Figshare., https://doi.org/10.6084/m9.figshare.20449701.v2 (2022).

38. Schreiner, M., Bhowmik, A., Vegge, T., Jørgensen, P. B. & Winther, O. Neuralneb - neural networks can find reaction paths fast. *Mach. Learn. Sci. Technol.* 10.1088/2632-2153/ACA23E (2022).

APPENDIX B

NeuralNEB – Neural Networks can find Reaction Paths Fast

NeuralNEB – Neural Networks can find Reaction Paths Fast

Mathias Schreiner, Arghya Bhowmik, Tejs Vegge,
Peter Bjørn Jørgensen, Ole Winther

Technical University of Denmark (DTU), 2800 Lyngby, Denmark

E-mail: matschreiner@gmail.com
24th June 2022

Abstract. Quantum mechanical methods like Density Functional Theory (DFT) are used with great success alongside efficient search algorithms for studying kinetics of reactive systems. However, DFT is prohibitively expensive for large scale exploration. Machine Learning (ML) models have turned out to be excellent emulators of small molecule DFT calculations and could possibly replace DFT in such tasks. For kinetics, success relies primarily on the models' capability to accurately predict the Potential Energy Surface (PES) around transition-states and Minimal Energy Paths (MEPs). Previously this has not been possible due to scarcity of relevant data in the literature. In this paper we train equivariant Graph Neural Network (GNN)-based models on data from 10.000 elementary reactions from the recently published Transition1x dataset. We apply the models as potentials for the Nudged Elastic Band (NEB) algorithm and achieve a Mean Average Error (MAE) of 0.23 eV and Root Mean Squared Error (RMSE) of 0.52 eV on barrier energies on unseen reactions. We compare the results against equivalent models trained on QM9x and ANI1x. We also compare with and outperform Density Functional based Tight Binding (DFTB) on both accuracy and required computational resources. The implication is that ML models are now at a level where they can be applied to studying chemical reaction kinetics given a sufficient amount of data relevant to this task.

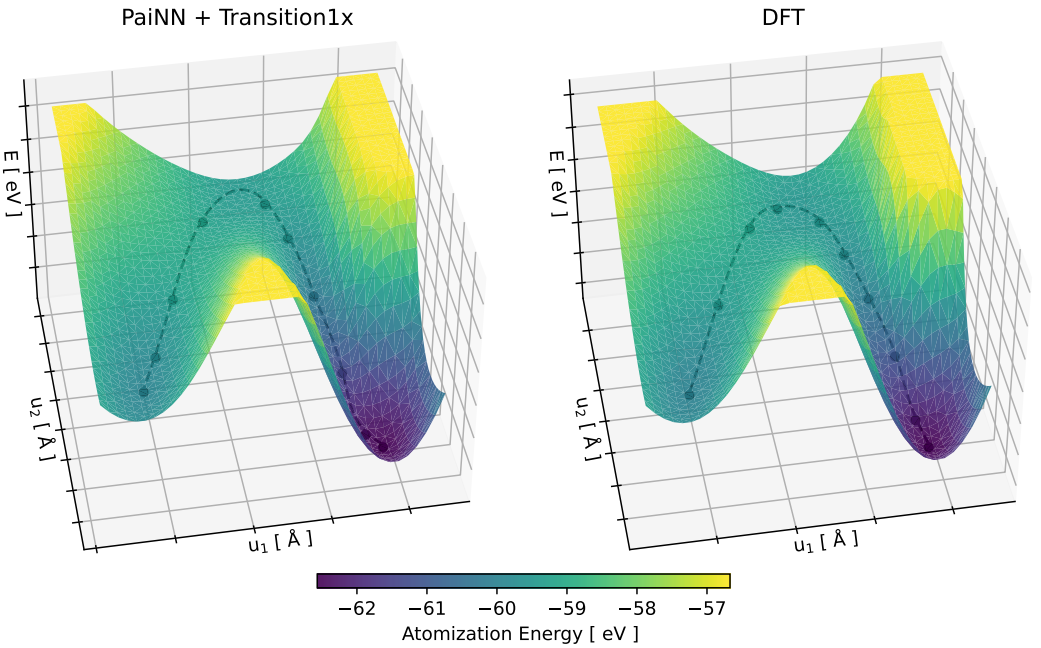


Figure 1. Minimal Energy Paths (MEPs) found with Nudged Elastic Band (NEB) applying the Graph Neural Network (GNN) architecture Polarizable Atom interaction Neural Network (PaiNN) trained on the Transition1x dataset and Density Functional Theory (DFT) as potentials. The MEPs are projected onto planes in structural space, intersecting product, reactant and transition-state of the converged MEPs. The PES has been calculated on the planes in the vicinity of the MEPs with the respective potential and is shown on the z-axis. The x and y-axes are basis vectors describing the plane. The reaction involves a H-transfer coupled with a C-C bond formation on C6H8. The reaction can be seen as a GIF by following this link.

1. Introduction

Machine Learning (ML) models and especially Graph Neural Networks (GNNs) [1, 2] have turned out to be potent emulators of Density Functional Theory (DFT) potentials for small molecules [3, 4, 5, 6, 7], thanks to their remarkable ability to find complex relations in high dimensional data. They have a complexity-scaling orders of magnitudes lower than classic Quantum Mechanics (QM) methods, but have in recent years achieved comparable accuracy [8, 9, 10, 11, 12]. The capability of these models is manifested by their success in tasks beyond simple prediction of molecular features such as structural optimization or studying finite-temperature dynamical properties through molecular dynamics [13, 14]. Despite their achievements, there has only been limited success in applying ML-models as potentials for transition search algorithms. The earliest work studied simple diatomic molecule dissociation and achieved acceptable accuracy with tens of thousands of data points [15]. Other works have had success by limiting their scope to studying single or few reactions but sacrificing the generality of the approach [16, 17, 18]. Attempts to study reactive systems with Gaussian Processs (GPs) [19] have been successful too, but the GP is trained on the particular atomic system, sacrificing speed for generality by requiring expensive DFT calculations at inference time. Transition-states are notoriously hard to find as there is no well-defined gradient on the Potential Energy Surface (PES) to guide traditional optimization algorithms towards them. A wealth of algorithms have been proposed to solve this problem – one is the Nudged Elastic Band (NEB) [20] algorithm, which works by interpolating an initial path between reactant and product and iteratively updating it to minimize energy by using information about the PES. It shares a common bottleneck with other transition search algorithms – the necessity to repeatedly evaluate energy and atomic forces of molecular configurations, which is extremely costly, especially if ab-initio or electron DFT calculations are used[21].

Recent advances in ML have not alleviated the bottleneck as even modern Neural Network (NN) architectures have not proved proficient potential approximators for this type of application. The fault lies primarily with available data in the literature rather than the models' expressiveness [22]. Most quantum mechanical datasets are focused on molecular configurations in or near equilibrium [23, 24, 25, 26]. Without configurations on and around reaction pathways in the training data, ML models cannot learn the interatomic interactions that occur during chemical reactions and cannot reliably be applied for transition-state search.

We compare ML models against Density Functional based Tight Binding (DFTB), [27] a fast approximation to DFT that is often used for fast screening of large quantities of configurations with an acceptable trade-off between accuracy and speed, and our

models outperform DFTB with a factor three in accuracy and a factor two in CPU time.

In this work, we bridge generalization, speed, and accuracy for transition-state search by applying Polarizable Atom interaction Neural Network (PaiNN) models as surrogate potentials for DFT. We build on and showcase the utility of our previous paper [28] where we released Transition1x, a dataset constituted by DFT calculations for 10 million molecular configurations, all sampled around reaction pathways from 10.000 elementary, organic reactions. It is clear from the results of this paper, that for precise modeling of transition-state regions, and, consequently, transition states and barrier energies, hitherto popular benchmark datasets have had insufficient relevant data. On the other hand, training ML potentials on the Transition1x dataset allows for accurate modeling of PESs in transition-state regions, underlining that relevant and available data in the literature is as important as the efficiency of available models.

Reliable and fast analysis of reaction kinetics through ML will bring the whole field of computational chemistry a considerable step closer to the ultimate goal, a virtual laboratory, hyper-accelerating the discovery of reaction mechanisms for synthesizing drugs and materials.

2. Methods

2.1. Nudged Elastic Band

NEB [20] is a method for finding Minimal Energy Path (MEP) and transition-state given product and reactant of a chemical reaction. It does so by iteratively nudging an interpolated path between the reaction endpoints in the direction of the force perpendicular to the path. Once the perpendicular force converges to zero, NEB reports the maximal-energy configuration along the path as the transition-state. The path is represented by an array of molecular configurations called images, and there is no guarantee that, at convergence, the maximal energy image corresponds to the maximal energy along the path. The maximum might lie between two images. Climbing Image Nudged Elastic Band (CINEB) [29] addresses this problem by letting the transition-state candidate (the maximal energy image) further maximize its energy by following the gradient on the PES parallel to the current path between iterations. If the current path has not converged properly, the climbing image can pull the predicted MEP off the true MEP and therefore, the path is first relaxed with regular NEB before turning on CINEB. The MEP is considered converged once the maximal perpendicular force on the path is below a threshold of 0.05 $\mathrm{eV\AA^{-1}}$. The spring constant between images on the path is set to 0.1 $\mathrm{eV\AA^{-2}}$, and ten images are used to represent the path.

2.2. Initial Path Generation

The endpoints of the reaction have to be minimized in their respective minima before running NEB – otherwise the energetic difference between reactant and transition-state cannot be evaluated properly. A configuration is considered relaxed if the norm of the forces acting on it is below 0.01 $\mathrm{eV\AA^{-1}}$. Once the endpoints have been minimized, the initial guess for the MEP is found by running NEB with the Image Dependent Pair Potential (IDPP) [30] on a linearly interpolated path between reactant and product. IDPP is an inexpensive potential specifically designed to generate physically realistic MEP guesses for NEB at an extremely low computational cost.

2.3. Optimizers

Reactants and products are relaxed using the BFGS [31] optimizer with $\alpha = 70$ and a maximal step size of 0.03 Å in configurational space. The MEP is found with an optimizer [32] designed to reduce the computational cost of transition-state search algorithms by applying an adaptive time step selection algorithm with $\alpha = 0.01$ and rtol = 0.1, and a preconditioning scheme to the PES given an estimate of its curvature.

3. Data

We train all models on ANI1x [24], QM9x [33], Transition1x [28]. All datasets are calculated with the 6-31G(d) [34] basis set and ωB97x [35] functional which has an accuracy comparable to the gold standard but expensive high-level CCSD(T) [36] [37] calculations. Given the compatibility of the datasets, it is possible to train on either dataset alone or combinations of them to leverage all of their strengths.

3.1. ANI1x

ANI1x [38] aims to provide varied data of off-equilibrium molecular configurations by perturbing equilibrium configurations with pseudo molecular dynamics. The data is collected through an active learning technique called Query by Committee; an automated data diversification process that trains an ensemble (committee) of models on a dataset and accepts or rejects new proposed data based on the disagreement of models in the committee. The assumption is that if the committee disagrees the data is sufficiently different from what has already been learned, and the proposed data should be included in the analysis. The procedure for proposing data and evaluating it with the committee is cheap compared to the calculation of data using DFT. The dataset is consecutively expanded by alternating between training committees and adding new data points based on the committee uncertainty. In total, ANI1x contains force and energy calculations for approximately 5 million configurations.

3.2. Transition1x

We have recently published Transition1x [28], a dataset providing a collection of molecular configurations on and along reaction paths for approximately 10.000 reactions. The reactions consist of up to 7 heavy atoms, including C, N, and O. Transition events are rare, and it is not possible to collect sufficient data in relevant regions by simple molecular dynamics if the intention is to train NNs models to understand chemical reactions. Transition1x addresses this problem by sampling molecular configurations around reaction pathways proposed by NEB, using DFT as potential. The procedure resulted in approximately 10 million DFT calculations that were collected and saved during the process and constitute the dataset. Transition1x is available through the repository https://gitlab.com/matschreiner/Transition1x which includes data loaders and scripts for downloading the dataset and generating ASE-database files.

3.3. QM9 and QM9x

QM9 [33] is a dataset of 135k small organic molecules with various chemical properties that has served as the benchmark for many existing ML methods for quantum chemistry. All molecules in QM9 are in equilibrium. We have recalculated QM9 with the 6-31G(d) basis set and ωB97x functional to make it compatible with Transition1x and ANI1x, and we refer to the recal-

culated dataset as QM9x. Molecular configurations recalculated in the new potential are not necessarily in equilibrium as the potential shifts when changing functional and basis sets. QM9x is available through the repository https://gitlab.com/matschreiner/QM9x which includes data loaders and scripts for downloading the dataset and generating ASE-database files.

3.4. Models and Training

Message Passing Neural Networks (MPNNs) [39] are a class of GNNs [1, 2] that build their internal graph representation by running a series of message passing steps. A single message passing step consists of two distinct operations: i) *Message Dispatching*, each node computes a message given its state (and possibly information about the edge connecting to – and the state of the receiving node) and sends it to its neighbors. ii) *State Update*, incoming messages are collected with an aggregation function, and are used to simultaneously update the internal representation of all nodes. After the message-passing phase, a readout function extracts the inner representation of the nodes and computes a final feature vector of the graph for downstream tasks. In the case of molecules, interesting properties are energy and forces where conservative force fields can be computed via the back-propagation algorithm as the negative gradient of the energy wrt. coordinates of the atoms.

The PaiNN model [40] was used for all experiments – it is a GNN architecture that implements rotationally equivariant representations for prediction of tensorial properties of graph structures. We refer to the literature for further details [40]. A cut-off radius of 5 Å was used to generate the initial molecular graph. All models have three message passing steps and 256 units in each hidden layer, and are trained using the ADAM [41] optimizer with learning rate 10^{-3} on training examples from QM9x, ANI1x, and Transition1x. A batchsize of 75 was used for all datasets and a maximum of 10^6 training steps was allowed - however, models training on ANI1x and Transition1x reached maximal scores on validation data after around $6 \cdot 10^5$ steps. In order to understand to which extent a PaiNN-model trained on Transition1x can generalize to reactions with unseen atomic compositions, building on an assessment of the substructures or elemental features, Transition1x was stratified by chemical formula such that each formula can only be found in one split. The Transition1x was split in 10, and 10 models were trained such that each split could be set aside once as testing data for the NeuralNEB algotithm and once as validation data for early stopping. ANI1x was stratified by chemical formula such that test, validation and training sets consist of chemical formulas unique to that set. QM9x was split randomly. In the case of QM9x and ANI1x, 80% of the data was used for training, 10% for testing, and 10% was used for validation and early stopping. In QM9x all configurations are unique as they are in distinct equilibria and can therefore be split randomly. No attention was paid to the molecular scaffold. For ANI1x, it is necessary to split on chemical formula to ensure that configurations across splits are significantly different. Each chemical formula contains similar configurations, since data is generated by randomly perturbing identical initial configurations.

4. Results

Table 1 shows the overall findings of the paper. Each row displays the performance

	Barrier [eV]		NEB Convergence		
	MAE	RMSE	Rate	Avg. CPU Time	Avg. Iterations
ANI1x	0.51	1.67	69.3%	37s	149
T1x	**0.23**	**0.52**	80.3%	33s	135
QM9x	3.40	3.59	35.0%	**28s**	111
DFTB	0.70	0.85	65.7%	82s	114
DFT	-	-	**84.1%**	12h14m43s	**100.74**

Table 1. Performance of various potentials used for Nudged Elastic Band (NEB) when compared to Density Functional Theory (DFT). ANI1x, Transition1x and QM9x indicate PaiNN models trained on the respective dataset. The Barrier column displays the Mean Average Error (MAE) and Root Mean Squared Error (RMSE) of barrier predictions, where the individual error is the difference between the barrier as predicted when using DFT as potential vs. using the surrogate potential. The convergence rate is the percentage of reactions that converged. Average CPU time is CPU time spent per reaction. Average iterations is the average number of Minimal Energy Path (MEP) updates before convergence.

of a surrogate potential, where datasets in the leftmost column indicate PaiNN models trained on the given dataset. The barrier error is the difference in barrier heights found when applying DFT as potential for NEB versus when applying the surrogate potential.

As different initializations of parameters in equivalent architectures result in variations in the trained models capabilities, five models were trained on each of QM9x and ANI1x and ten models were trained on the Transition1x dataset. QM9x and ANI1x models were used as potentials for all reactions in the Transition1x dataset, and models trained on Transition1x were used as potentials only for those reactions with atomic compositions from the test split. The best models are trained on Transition1x, with the lowest Mean Average Error (MAE) and Root Mean Squared Error (RMSE) and the highest convergence ratio. The QM9x models have only seen data very close to equilibrium and have not learned the structure of the PES between equilibria which makes it unable to converge in most cases. In general DFT performs the best in terms of convergence rate and average iterations run, but it comes at a steep price, running almost a factor 1500 times slower than the ML potentials. DFTB is the go-to fast potential, but the models trained on Transition1x are twice as fast and three times as accurate. Figure 1, on the frontpage, displays MEPs calculated with NEB using DFT and PaiNN trained on Transition1x side by side. Each MEP is projected onto a plane in configurational space spanned by the reaction's transition-state, product, and reactant. The x and y axes are basis vectors describing the plane in units of Å, and the z-axis and color-coding show the atomization energy of configurations in the plane in eV. Not only does PaiNN trained on the Transition1x accurately calculate the barrier energy for the reaction, but it also correctly identifies the plane spanned by the configurations defining the reaction, and calculates an almost identical PES in the vicinity of the MEP. Each MEP is projected from a high dimensional space onto the plane, and therefore, only the atomization energy of equilibria and transition-states are shown correctly in the plot. At these points, the MEP intersects with the plane. The intermediate points have energies slightly shifted up the sides of the energy valley. The

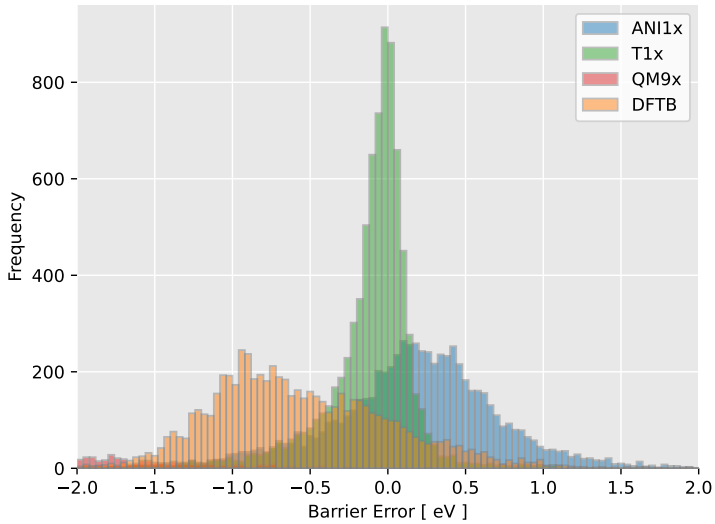


Figure 2. Histogram of barrier errors. The x-axis shows errors between reaction barriers calculated using Density Functional Theory (DFT) and surrogate potentials for Nudged Elastic Band (NEB). The x-axis has been truncated at +/- 2 eV error (see appendix for the full plot). The y-axis shows the frequency of each bin. Green, red and blue display results from PaiNN models trained on Transition1x, QM9x and ANI1x, respectively. Yellow displays results from Density Functional based Tight Binding (DFTB). The QM9x model has such a low convergence frequency, and general barrier error, that the model does not show in the plot.

MEP does not necessarily lie in the plane, and since the MEP represents the energy valley, projecting it onto the plane, will increase the energy. The $\times$ symbols on the surfaces are projections of images predicted by NEB and the dashed lines connecting them are cubic spline interpolations. The importance of accurate predictions in the vicinity of the MEP is clear, as these calculations will guide the search for the transition-state. The Transition1x model predicts smooth and well-behaved PESs resembling DFT.

Figure 2 and B2 tell similar stories. Figure 2 is a histogram of barrier errors where the error is the difference between activation energy found using the surrogate potential and DFT. The Transition1x model is precise and accurate, with a sharp peak around zero, whereas DFTB and ANI1x have wider spreads with means below and above zero, respectively. The QM9x model is plotted on the histogram, but due to high errors and low convergence, only a few calculated barriers fall within an error of ± 2 eV, as shown in the figure. See appendix for an equivalent figure without truncated x-axis.

Figure B2 compares activation energies found with DFT on the x-axis with those found using various surrogate potentials on the y-axis. Each marker represents a single reaction. Predictions from the model trained on Transition1x follow the $x = y$ line with a MAE of only 0.23 eV. The QM9x model does not have a proper representation of the

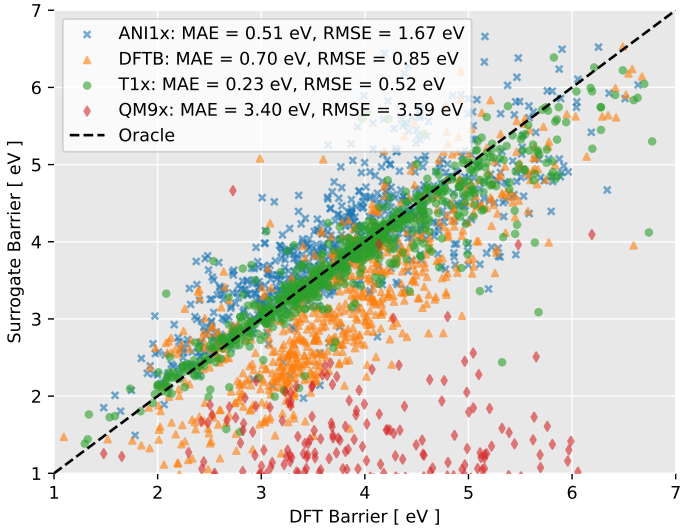


Figure 3. Comparison of reaction barriers found with Nudged Elastic Band (NEB) using Density Functional Theory (DFT) as potential on the x-axis vs. various surrogate potentials on the y-axis. Green, red and blue markers are PaiNN models trained on The Transition1x, QM9x, and ANI1x datasets respectively. Yellow is Density Functional based Tight Binding (DFTB). Points on the dashed line have been calculated perfectly. The figure displays a subsample of 500 reactions - see appendix for the full scatter plot.

	Corrected Barrier [eV]		Systematic Error [eV]
	MAE	RMSE	
ANI1x	0.48	1.66	0.23
T1x	**0.23**	**0.51**	**-0.10**
QM9x	0.89	1.14	-3.40
DFTB	0.48	0.62	-0.58

Table 2. Mean Average Error (MAE) and Root Mean Squared Error (RMSE) of barrier errors found by PaiNN trained on Transition1x and ANI1x and DFTB, after correcting for systematic error.

transition-state regions as it has not seen that type of data during training. Often, the QM9x model does not recognize nearby initial equilibria as minima on the PES, and even before optimizing the MEP, the reaction endpoints have dropped further on the PES to qualitatively different endpoints which results in the model calculating the MEP for a completely different reaction. The algorithm is not set up to detect this, and as long as the reaction converges, it is included in the analysis. Even when the QM9x model relaxes the endpoints of the reaction correctly, it either finds low energy shortcuts in the faulty potential or does not converge, and as a result the converged reactions are often only the energy difference between reactant and product. The QM9x dataset was not

designed with any type of molecular dynamics or reaction kinetics in mind, and comparing it to ANI1x and Transition1x for reaction path search is perhaps inappropriate. However, given the ubiquity of QM9 in the literature, it is an important point to convey, that new datasets are required for solving higher order problems in computational chemistry. The Transition1x and ANI1x models drop in performance above 5 eV. Data becomes scarcer at higher energies and consequently, models are less accurate in high energy regions. DFTB and the ANI1x models have systematic errors in their predictions. The ANI1x models are biased towards high energies in the transition regions as they have not seen the low energy valleys connecting equilibria. The DFTB potential systematically predicts energies too low. In Table 2 the systematic errors are corrected based on the training data. This leads to a lower test error for the ANI1x and DFTB, but equal test error for Transition1x underlining that Transition1x models are already very accurate.

5. Discussion

To train models that can properly step in as surrogate potentials for DFT when running NEB, it is necessary to have datasets with appropriate data in and around transition-state regions. Finding reaction barriers with ML models and NEB is a non-trivial test. ML models, and especially NNs, are known to perform poorly for out of distribution tasks [42, 43]. Table A1 illustrates this clearly with results for training and testing ML models on various datasets.

Finding reaction barriers with NEB is a much more demanding test of the models' capabilities. When running NEB, the PES is swept by the path connecting endpoints, and data encountered in the process can diverge wildly from any data seen during testing and training. The model can get caught in even a small region of high error, or it can be thrown off the correct MEP and be unable to converge altogether, so the model must be accurate across the entire PES.

The reaction paths are represented by ten images in all reactions. A core strength of NNs is their ability to utilize GPUs to evaluate multiple data points at once, and in principle, NEB can be run with hundreds of images instead of tens at little to no additional cost when using NNs as potentials. We ran experiments with high density paths with the rest of the setup fixed but saw no improvement in neither accuracy nor convergence speed. The preconditioning scheme of the NEB optimizer relies on a sparsely populated path. But this approach could possibly produce robust results by applying other optimizers.

A clear application of this work is as a screening procedure for complex reaction networks. Cheap methods, such as permuting bond order matrices, can be used to automatically generate nodes for entire reaction networks. The individual reactions can be screened fast using the method before recalculating entire reaction networks with expensive methods. Usually this is done with DFTB [27] but running NEB with NNs is faster and more accurate.

6. Conclusion

We have trained GNN potentials on various datasets and used them as surrogate potentials for DFT when running NEB for transition-state search. A MAE of 0.23 eV and Root Mean Squared Error (RMSE) of 0.52 eV is achieved with the best model, compared against running the same set up with DFT.

The models converge 80.3% of the time on unseen reactions. We show that expressive models alone are not sufficient for solving complex tasks in quantum chemistry moving forward, but just as much care has to be put into designing and generating datasets. We tested 3 different datasets: ANI1x, QM9x and Transition1x and only models trained on the latter could reliably solve the transition search task.

Our results show that the future development of the field of ML for quantum chemistry stands on two legs – the completeness of the available data, and the expressiveness of the available models. Transition1x deals with only four types of atoms. To apply the results of this paper to general chemistry, larger datasets with more atom types should be produced. Our results indicate that the machine learning approach scales: With the right amount of the right data, accuracies at a sufficient level can be achieved.

Data Availability

The data that support the findings of this study are openly available.

Acknowledgements

The authors acknowledge support from the Novo Nordisk Foundation (SURE, NNF19OC0057822) and the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 957189 (BIG-MAP) and No. 957213 (BATTERY2030PLUS).

Ole Winther also receives support from Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606) and the Pioneer Centre for AI, DNRF grant number P1.

7. Code Availability

Code for training PaiNN models and running NEB is available through the repository https://gitlab.com/matschreiner/neuralneb.

References

[1] Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 12 2019. doi: 10.1016/j.neunet.2020.06.006. URL http://arxiv.org/abs/1912.12693http://dx.doi.org/10.1016/j.neunet.2020.06.006.

[2] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. 2021. doi: 10.1016/j.aiopen.2021.01.001. URL https://doi.org/10.1016/j.aiopen.2021.01.001.

[3] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole Von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid dft error. *Journal of Chemical Theory and Computation*, 13:5255–5264, 11 2017. ISSN 15499626. doi: 10.1021/ACS.JCTC.7B00577. URL https://pubs.acs.org/doi/abs/10.1021/acs.jctc.7b00577.

[4] Julia Westermayr, Michael Gastegger, Kristof T. Schütt, and Reinhard J. Maurer. Perspective on integrating machine learning into computational chemistry and materials science. *The Journal of Chemical Physics*, 154:230903, 6 2021. ISSN 0021-9606. doi: 10.1063/5.0047760. URL https://aip.scitation.org/doi/abs/10.1063/5.0047760.

[5] Stuart I Campbell, Daniel B Allan, and Andi M Barbour. Machine learning for the solution of the schrödinger equation. *Machine Learning: Science and Technology*, 1:013002, 4 2020. ISSN 2632-2153. doi: 10.1088/2632-2153/AB7D30. URL https://iopscience.iop.org/article/10.1088/2632-2153/ab7d30https://iopscience.iop.org/article/10.1088/2632-2153/ab7d30/meta.

[6] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[7] Julia Westermayr and Philipp Marquetand. Machine learning for electronically excited states of molecules. *Chemical Reviews*, 121:9873–9926, 8 2021. ISSN 15206890. doi: 10.1021/ACS.CHEMREV.0C00749. URL https://pubs.acs.org/doi/full/10.1021/acs.chemrev.0c00749.

[8] Jean Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48:722–730, 3 2015. ISSN 15204898. doi: 10.1021/AR500432K. URL https://pubs.acs.org/doi/full/10.1021/ar500432k.

[9] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98:146401, 4 2007. ISSN 00319007. doi: 10.1103/PHYSREVLETT.98.146401/FIGURES/4/MEDIUM. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.98.146401.

[10] Jörg Behler. Constructing high-dimensional neural network potentials: A tutorial review. *International Journal of Quantum Chemistry*, 115:1032–1050, 8 2015. ISSN 1097-461X.

doi: 10.1002/QUA.24890. URL https://onlinelibrary.wiley.com/doi/full/10.1002/qua.24890.

[11] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole Von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid dft error. *Journal of Chemical Theory and Computation*, 13:5255–5264, 11 2017. ISSN 15499626. doi: 10.1021/ACS.JCTC.7B00577. URL https://pubs.acs.org/doi/abs/10.1021/acs.jctc.7b00577.

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. 2017.

[13] Sami Kaappa, Casper Larsen, and Karsten Wedel Jacobsen. Atomic structure optimization with machine-learning enabled interpolation between chemical elements. *Physical Review Letters*, 127, 7 2021. doi: 10.1103/PhysRevLett.127.166001. URL http://arxiv.org/abs/2107.01055http://dx.doi.org/10.1103/PhysRevLett.127.166001.

[14] Jiaqi Wang, Seungha Shin, and Sangkeun Lee. Interatomic potential model development: Finite-temperature dynamics machine learning. *Advanced Theory and Simulations*, 3: 1900210, 2 2020. ISSN 2513-0390. doi: 10.1002/ADTS.201900210. URL https://onlinelibrary.wiley.com/doi/full/10.1002/adts.201900210.

[15] M Malshe, LM Raff, MG Rockley, M Hagan, Paras M Agrawal, and R Komanduri. Theoretical investigation of the dissociation dynamics of vibrationally excited vinyl bromide on an ab initio potential-energy surface obtained using modified novelty sampling and feedforward neural networks. ii. numerical application of the method. *The Journal of chemical physics*, 127(13):134105, 2007.

[16] Xiaoxiao Lu, Qingyong Meng, Xingan Wang, Bina Fu, and Dong H Zhang. Rate coefficients of the h+ h2o2→ h2+ ho2 reaction on an accurate fundamental invariant-neural network potential energy surface. *The Journal of chemical physics*, 149(17):174303, 2018.

[17] Tom A Young, Tristan Johnston-Wood, Volker L Deringer, and Fernanda Duarte. A transferable active-learning strategy for reactive molecular force fields. *Chemical science*, 12(32):10944–10955, 2021.

[18] Sergei Manzhos and Tucker Carrington Jr. Neural network potential energy surfaces for small molecules and reactions. *Chemical Reviews*, 121(16):10187–10217, 2020.

[19] Olli Pekka Koistinen, Freyja B. Dagbjartsdóttir, Vilhjálmur Ásgeirsson, Aki Vehtari, and Hannes Jónsson. Nudged elastic band calculations accelerated with gaussian process regression. *Journal of Chemical Physics*, 147, 2017. ISSN 00219606. doi: 10.1063/1.4986787.

[20] Daniel Sheppard, Rye Terrell, and Graeme Henkelman. Optimization methods for finding minimum energy paths. *The Journal of Chemical Physics*, 128:134106, 4 2008. ISSN 0021-9606. doi: 10.1063/1.2841941. URL https://aip.scitation.org/doi/abs/10.1063/1.2841941.

[21] Stefan Heinen, Max Schwilk, Guido Falk von Rudorff, and O Anatole von Lilienfeld.

Machine learning the computational cost of quantum chemistry. *Machine Learning: Science and Technology*, 1(2):025002, 2020.

[22] O. Anatole von Lilienfeld, Klaus Robert Müller, and Alexandre Tkatchenko. Exploring chemical compound space with quantum-based machine learning. *Nature Reviews Chemistry 2020 4:7*, 4:347–358, 6 2020. ISSN 2397-3358. doi: 10.1038/s41570-020-0189-9. URL https://www.nature.com/articles/s41570-020-0189-9.

[23] Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg. Ani-1, a data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific Data 2017 4:1*, 4:1–8, 12 2017. ISSN 2052-4463. doi: 10.1038/sdata.2017.193. URL https://www.nature.com/articles/sdata2017193.

[24] Justin S. Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E. Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific Data 2020 7:1*, 7:1–10, 5 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0473-z. URL https://www.nature.com/articles/s41597-020-0473-z.

[25] Tobias Fink and Jean Louis Raymond. Virtual exploration of the chemical universe up to 11 atoms of c, n, o, f: Assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *Journal of Chemical Information and Modeling*, 47:342–353, 2007. ISSN 1549960X. doi: 10.1021/CI600423U.

[26] Tobias Fink, Heinz Bruggesser, and Jean Louis Reymond. Virtual exploration of the small-molecule chemical universe below 160 daltons. *Angewandte Chemie - International Edition*, 44:1504–1508, 2 2005. ISSN 14337851. doi: 10.1002/ANIE.200462457.

[27] Gotthard Seifert and Jan Ole Joswig. Density-functional tight binding—an approximate density-functional theory method. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2:456–465, 5 2012. ISSN 1759-0884. doi: 10.1002/WCMS.1094. URL https://onlinelibrary.wiley.com/doi/full/10.1002/wcms.1094.

[28] Schreiner M., Bhowmik A., Vegge T., and Busk J.and Winther O. a Dataset for Building Generalizable Reactive Machine Learning Potentials. 6 2022. doi: 10.6084/m9.figshare.19614657.v4. URL https://figshare.com/articles/dataset/Transition1x/19614657.

[29] Graeme Henkelman, Blas P. Uberuaga, and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 113:9901, 11 2000. ISSN 0021-9606. doi: 10.1063/1.1329672. URL https://aip.scitation.org/doi/abs/10.1063/1.1329672.

[30] Søren Smidstrup, Andreas Pedersen, Kurt Stokbro, and Hannes Jónsson. Improved initial guess for minimum energy path calculations. *The Journal of Chemical Physics*, 140:214106, 6 2014. ISSN 0021-9606. doi: 10.1063/1.4878664. URL https://aip.scitation.org/doi/abs/10.1063/1.4878664.

[31] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6:76–90, 3 1970. ISSN 0272-4960. doi: 10.1093/IMAMAT/6.1.76. URL https://academic.oup.com/imamat/article/6/1/76/746016.

[32] Stela Makri, Christoph Ortner, and James R. Kermode. A preconditioning scheme for minimum energy path finding methods. *The Journal of Chemical Physics*, 150:094109, 3 2019. ISSN 0021-9606. doi: 10.1063/1.5064465. URL https://aip.scitation.org/doi/abs/10.1063/1.5064465http://creativecommons.org/licenses/by/4.0/.

[33] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data 2014 1:1*, 1:1–7, 8 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422.

[34] R. Ditchfield, W. J. Hehre, and J. A. Pople. Self-consistent molecular-orbital methods. ix. an extended gaussian-type basis for molecular-orbital studies of organic molecules. *The Journal of Chemical Physics*, 54:724, 9 2003. ISSN 0021-9606. doi: 10.1063/1.1674902. URL https://aip.scitation.org/doi/abs/10.1063/1.1674902.

[35] Jeng Da Chai and Martin Head-Gordon. Systematic optimization of long-range corrected hybrid density functionals. *The Journal of Chemical Physics*, 128:084106, 2 2008. ISSN 0021-9606. doi: 10.1063/1.2834918. URL https://aip.scitation.org/doi/abs/10.1063/1.2834918.

[36] Kevin E. Riley, Michal Pitonˇák, Petr Jurecˇka, and Pavel Hobza. Stabilization and structure calculations for noncovalent interactions in extended molecular systems based on wave function and density functional theories. *Chemical Reviews*, 110:5023–5063, 9 2010. ISSN 00092665. doi: 10.1021/CR1000173.

[37] Kanchana S. Thanthiriwatte, Edward G. Hohenstein, Lori A. Burns, and C. David Sherrill. Assessment of the performance of dft and dft-d methods for describing distance dependence of hydrogen-bonded interactions. *Journal of Chemical Theory and Computation*, 7:88–96, 1 2011. ISSN 15499618. doi: 10.1021/CT100469B.

[38] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E. Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148:241733, 5 2018. ISSN 0021-9606. doi: 10.1063/1.5023802. URL https://aip.scitation.org/doi/abs/10.1063/1.5023802.

[39] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. 2017.

[40] Kristof T Schütt, Sch¨ Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. 2021.

[41] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization.

[42] Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Sankar Raman Sathyamoorthy, and Cristofer Englund. Performance analysis of out-of-distribution

detection on various trained neural networks. 2021. URL `https://www.iso.org/deliverables-all.html`.

[43] Lily H Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. 2021.

Appendix A.

Table displaying results of the models when training and testing on various datasets. In all test set-ups the models that perform best, are models that have been trained on training data from the corresponding dataset.

		Energy [eV]		Forces [eV/Å]	
Trained on	Tested on	MAE	RMSE	MAE	RMSE
ANI1x		**0.02(0)**	**0.04(1)**	**0.01(0)**	**0.04(0)**
Transition1x	ANI1x	0.22(1)	0.35(2)	0.08(0)	0.34(2)
QM9x		2.32(1)	3.03(2)	0.56(2)	1.3(7)
ANI1x		0.28(2)	0.61(7)	0.10(5)	0.5(1)
Transition1x	Transition1x	**0.10(0)**	**0.15(1)**	**0.04(1)**	**0.09(0)**
QMx		1.42(1)	2.61(2)	0.19(0)	0.43(0)
ANI1x		0.12(0)	0.13(0)	0.02(0)	0.05(0)
Transition1x	QM9x	0.07(1)	0.12(0)	0.04(0)	0.07(0)
QM9x		**0.02(1)**	**0.04(2)**	**0.01(0)**	**0.01(0)**

Table A1. Test results of PaiNN models trained on ANI1x, QM9x, Transition1x. We report RMSE and MAE on energy and forces. Force error is the Euclidian distance between the predicted and true force vector.

Appendix B. Additional Figures

This section contains the unbounded version of Fig. 2, a scatter plot equivalent to Fig. B2, but without subsampling reactions, and additional plots of MEPs and PESs comparing PaiNN trained on Transition1x with DFT.

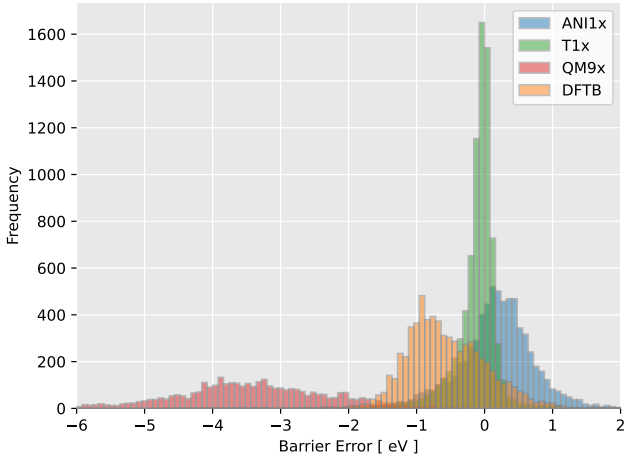


Figure B1. Histogram of barrier errors. The figure is equivalent to 2, but without a truncated x-axis. The x-axis shows errors between reaction barriers calculated using Density Functional Theory (DFT) and surrogate potentials for Nudged Elastic Band (NEB). The y-axis shows the frequency of each bin. Green, red and blue display results from PaiNN models trained on Transition1x, QM9x and ANI1x, respectively. Yellow displays results from Density Functional based Tight Binding (DFTB).

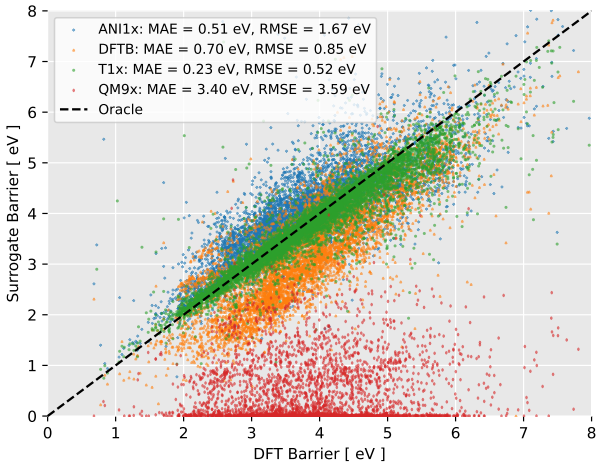


Figure B2. Comparison of reaction barriers found with Nudged Elastic Band (NEB) using Density Functional Theory (DFT) as potential on the x-axis vs. various surrogate potentials on the y-axis. Green, red and blue markers are PaiNN models trained on The Transition1x, QM9x, and ANI1x datasets respectively. Yellow is Density Functional based Tight Binding (DFTB). Points on the dashed line have been calculated perfectly.

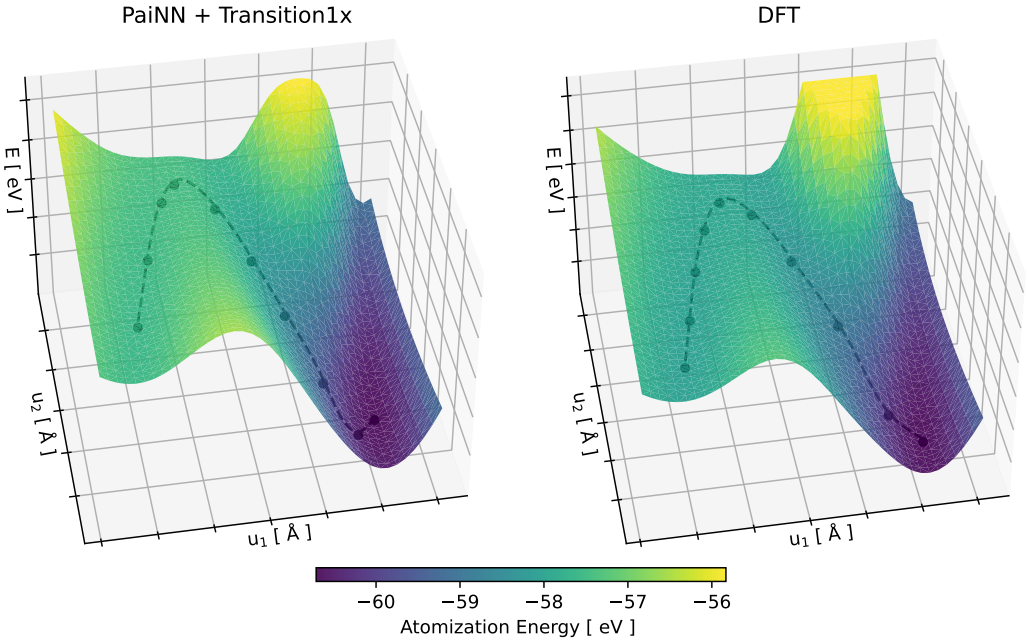


Figure B3. Reaction involving C5OH8. The reaction can be seen as a GIF by following this link.

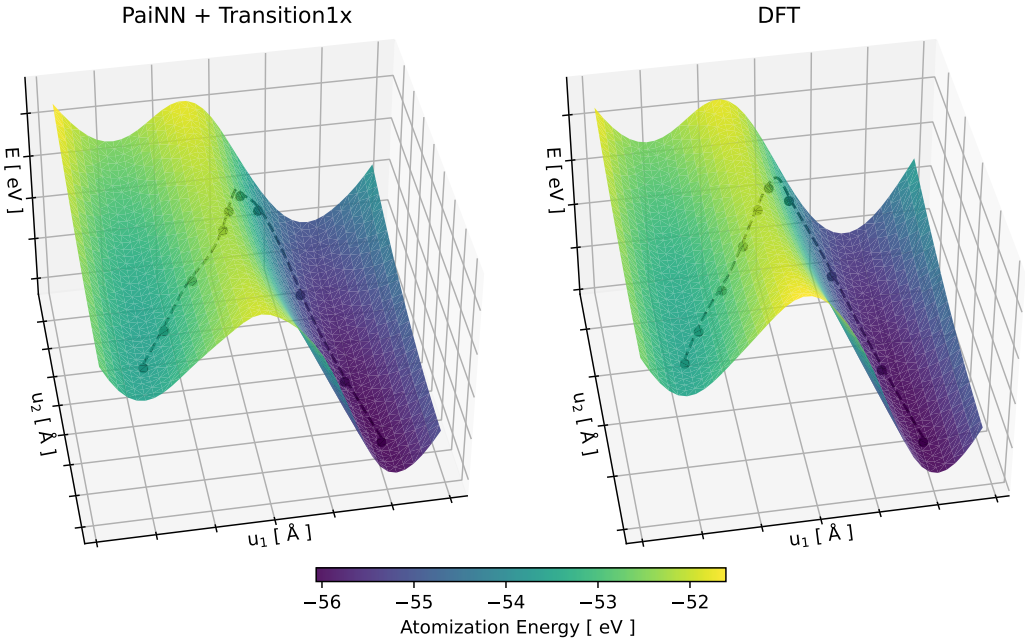


Figure B4. Reaction involving C3NCOH7. The reaction can be seen as a GIF by following this link.

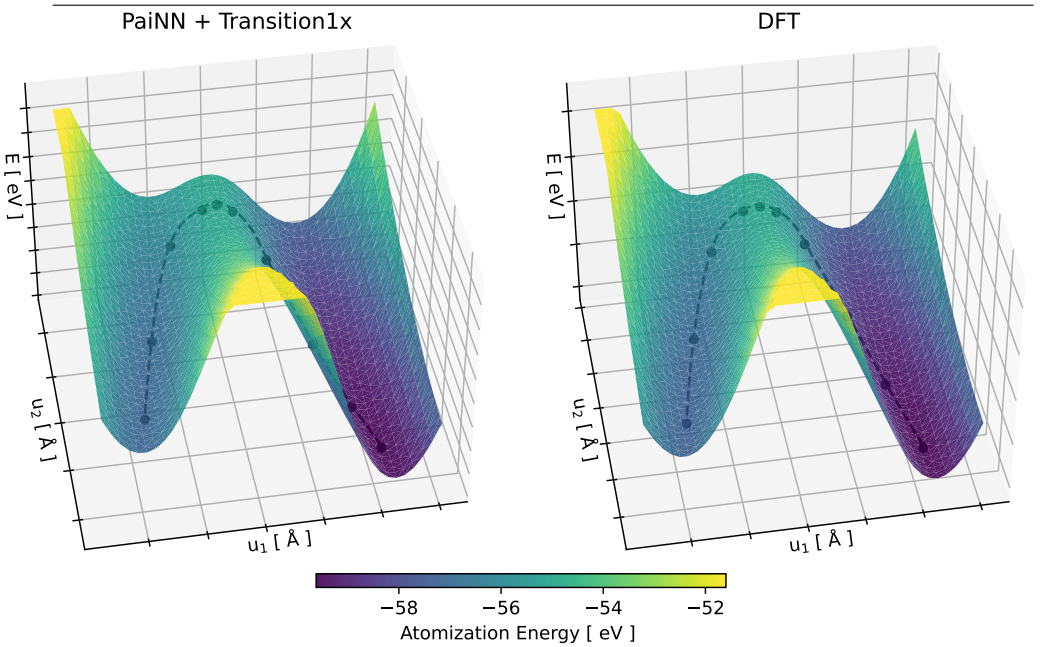


Figure B5. Reaction involving C3NCNH8. The reaction can be seen as a GIF by following this link.

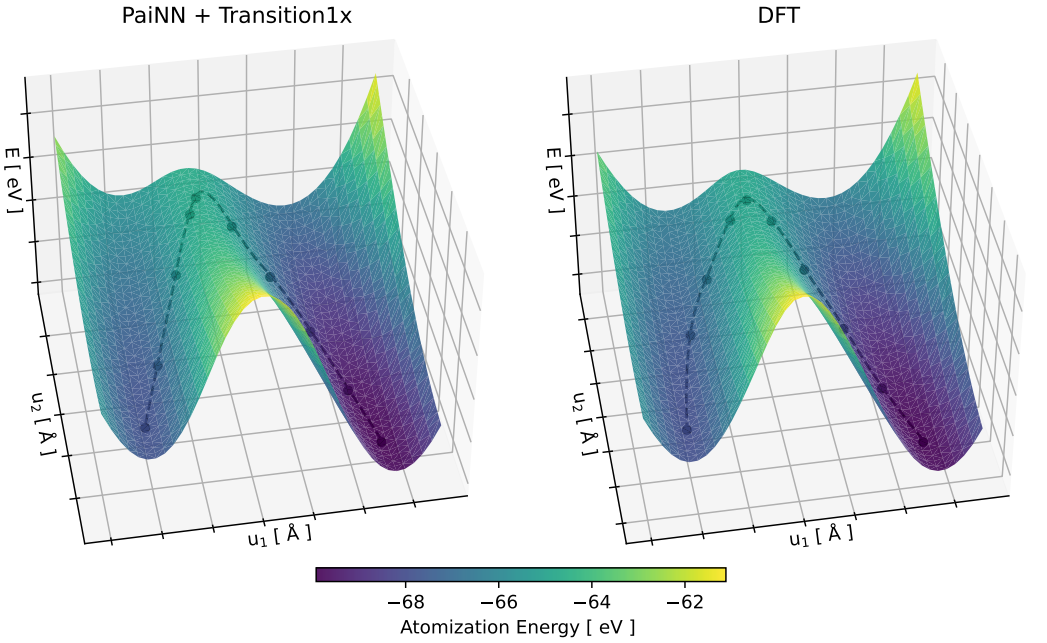


Figure B6. Reaction involving C3NC2OH9. The reaction can be seen as a GIF by following this link.

APPENDIX C

Machine Learning for Chemical Reactions

Neural Networks can solve Chemical Reactions Fast! When Given the Right Data

Mathias Schreiner, Arghya Bhowmik, Tejs Vegge
Jonas Busk, Peter Bjørn, Ole Winther
Technical University of Denmark

Abstract:

- Neural Networks have proven to be excellent emulators of DFT for calculating features of small molecules.
- We train Neural Networks as surrogate potentials for DFT and use them as forcefields for the Nudged Elastic Band Algorithm (NEB) to find Minimal Energy Paths (MEPs) fast and accurately. We call this approach NeuralNEB.
- We propose a new dataset, Tansition1x, that includes relevant configurations on and around reaction pathways - these are necessary for ML models to learn complex interactions between atoms happening during chemical reactions.

Method:

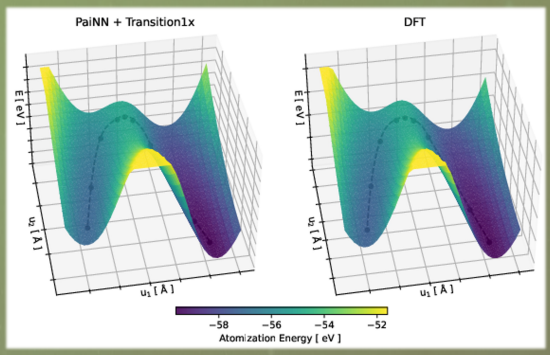
- **NEB** - a method for finding MEPs by iteratively nudging an initial guess for the reaction path using information about the potential energy surface.
- **PaiNN** - an equivariant graph neural network architecture designed for predicting molecular features was trained on the datasets listed below and used as forcefield for NEB.
- **QM9x** - a dataset of 135k molecules in equilibrium containing all configurations from the ubiquitous QM9 dataset, recalculated with a level of DFT matching ANI1x and Transition1x.
- **ANI1x** - a dataset containing 6m molecules generated with activate learning and pseudo molecular dynamics.
- **Transition1x** - a dataset containing 10m configurations on and around reaction pathways generated using NEB.

Results:

	Barrier [eV]		NEB Convergence		
	MAE	RMSE	Rate	Avg. CPU Time	Avg. Iterations
ANI1x	0.51	1.67	69.3%	37s	149
T1x	**0.23**	**0.52**	80.3%	33s	135
QM9x	3.40	3.59	35.0%	**28s**	111
DFTB	0.70	0.85	65.7%	82s	114
DFT	-	-	**84.1%**	12h14m43s	**100.74**

Convergence ratios, timings, and prediction performance on activation energies of reactions with unseen isomers, for PaiNN models trained on various datasets, compared to DFT.

Transition1x NeuralNEB



MEPs found with NEB applying PaiNN (left) and DFT (right) as potentials. The MEPs are projected onto planes in structural space, intersecting product, reactant and transition-state of the converged MEPs. The reaction involves a H-transfer coupled with a C-C bond formation on C6H8.

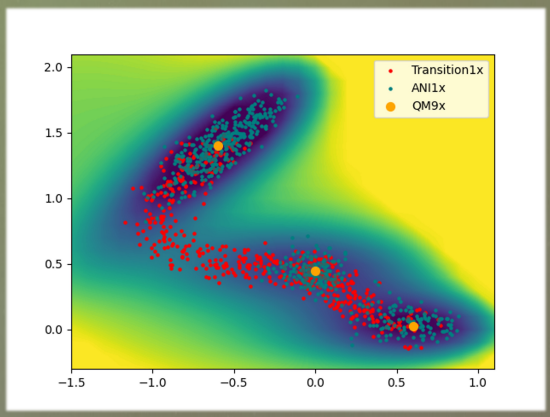
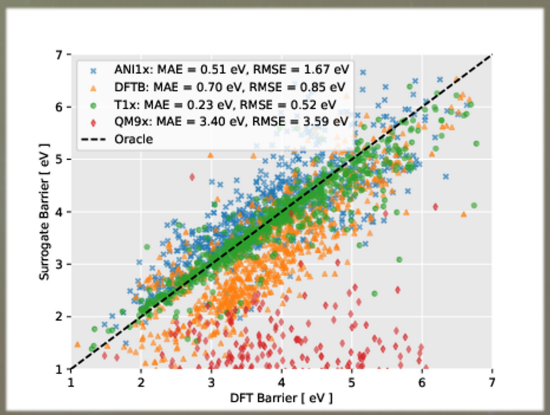


Illustration of the types of data found in ANI1x, QM9x and Transition1x.



Comparison of reaction barriers found with NEB using DFT as potential on the x-axis vs. various surrogate potentials on the y-axis. Points on the dashed line have been calculated perfectly. The figure displays a subsample of 500 reactions out of the 10k in Transition1x.

Machine Learning for Chemical Reactions A Dance of Datasets and Models

Mathias Schreiner[1] **Arghya Bhowmik**[2] **Tejs Vegge**[2]
Jonas Busk[2] **Peter B. Jørgensen**[2] **Ole Winther**[1,3,4]
[1]DTU Compute, Technical University of Denmark (DTU), 2800 Lyngby, Denmark
[2]DTU Energy, Technical University of Denmark (DTU), 2800 Lyngby, Denmark
[3]Department of Biology, University of Copenhagen (UCph), 2700 Copenhagen N, Denmark
[4]Genomic Medicine, Copenhagen University Hospital, Rigshospitalet, 2100 Copenhagen Ø, Denmark

Abstract

Machine Learning (ML) models have proved to be excellent emulators of Density Functional Theory (DFT) calculations for predicting features of small molecular systems. The activation energy is a defining feature of a chemical reaction, but, despite the success of ML in computational chemistry, an accurate, fast, and general ML-calculator for Minimal Energy Paths (MEPs) has not yet been proposed. Here, we summarize contributions from two of our recent papers, where we apply Graph Neural Network (GNN) based models, trained on various datasets, as potentials for the Nudged Elastic Band (NEB) algorithm to speed up MEP-search. We show that relevant data from reactive regions of the Potential Energy Surface (PES) in training data is paramount to success. Hitherto popular benchmark datasets primarily contain configurations in, or close to, equilibrium, and are not adequate for the task. We propose a new dataset, Transition1x, that contains force and energy calculations for 10 million molecular configurations from on and around MEPs of 10.000 organic reactions of various types. By training GNNs on Transition1x and applying the models as PES-evaluators for NEB, we achieve a Mean Average Error (MAE) of 0.23 eV on predicted activation energies of unseen reactions, compared to DFT, while running the algorithm 1200 times faster. Transition1x is a challenging dataset containing a new type of data that may serve as a benchmark for future methods for transition-state search.

Introduction

The activation energy of a chemical reaction is the difference between the reactant and transition state energies. It describes the energetic barrier of the reaction, and it dominates the reaction-rate exponentially through the Arrhenius Equation. To build a virtual laboratory, where reaction mechanisms for synthesis of drugs and materials can be studied, it is crucially important to be able to quickly and accurately predict activation energies and Minimal Energy Paths (MEPs) between equilibrium configurations. Nudged Elastic Band (NEB)[1] is an effective algorithm, designed to find MEPs, and activation energies in chemical space. It does so by iteratively nudging an initial guess for the reaction path in the direction of the force perpendicular to the path until convergence. The NEB algorithm requires a subordinate algorithm for calculating gradients and energies on the surrounding Potential Energy Surface (PES), and Density Functional Theory (DFT) is a popular choice for this. However, a single DFT calculation can take from a minute and up to several hours, depending on the size of the molecule and level of theory. Even for small molecular systems of 6-7 heavy atoms, NEB has to evaluate thousands of configurations to converge, making DFT inappropriate for large scale exploration of reaction-networks.

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

Machine Learning (ML) models, and Graph Neural Networks (GNNs)[2,3] in particular, have proved to be capable surrogate DFT potentials that can accurately evaluate molecular properties fast[4–13]. In this paper we train Polarizable Atom interaction Neural Network (PaiNN)[14] models, a GNN architecture, on various datasets and apply them as surrogate potentials for DFT to alleviate the NEB-bottleneck.

Initially, the models were trained on existing datasets in the literature. However, available datasets of significant volume that are interesting for training Neural Network (NN) models, either contain molecular configurations exclusively in equilibrium, or are generated through Molecular Dynamics (MD). Transitions between equilibriums are rare events, and datasets generated through MD-simulations do not contain sufficient samples of data around reaction paths to enable training models with accurate representations of these regions[15,16]. We created a new dataset, Transition1x, to address this problem. Transition1x leverages NEB to sample relevant configurations on and around reaction pathways for thousands of organic reactions. All intermediate configurations encountered while running the algorithm were calculated using the same level of DFT as the popular ANI1x[17,18] dataset, such that models trained on the two datasets can be compared in a meaningful way.

Method

Nudged Elastic Band

NEB[1] is a method for finding transition-states and MEPs given products and reactants of chemical reactions. It does so by iteratively nudging an initial guess for the MEP directed by the force perpendicular to the path. The path is represented by an array of molecular configurations called images connected by an artificial spring force which ensures that the images stay evenly separated and do not fall into the minimas at the reaction endpoints. Eventually, as the perpendicular force on the path converges to zero, the MEP will relax at the bottom of the local low-energy valley. At this point, NEB returns the maximal-energy configuration along the path as the transition-state. However, the true transition-state of the reaction may lie between two images representing the path. Climbing Image Nudged Elastic Band (CINEB)[19] addresses this problem by, between iterations, choosing a transition-state candidate and further maximize its energy by following the gradient on the PES, parallel to the current path. The CINEB algorithm terminates once both the maximal force perpendicular to the path, and climbing force on the transition-state candidate has converged. At this point, the maximal energy-configuration representing the path corresponds to the transition state.

Datasets

Transition1x was generated by running NEB, applying DFT as potential, on all reactions in the dataset released by Grambow et al. (2020)[20]. The original data contains products, reactants, and transition states for 11.000 organic reactions of various types. To ensure compatibility with ANI1x and QM9x, all intermediate calculations were made using the 6-31G(d)[21] basis set and ωB97x functional[22] . NEB typically converges within 200 iterations given the elements and sizes of molecules in the dataset. Calculations become gradually more similar towards convergence of NEB, as the path is nudged less between iterations. In order to reduce redundancy in data, paths are excluded from the dataset unless the cumulative maximal force, perpendicular to the path, from previous iterations exceeds 0.1 eV/Å. The data is limited to molecules with up to 7 heavy atoms, including C, N, and O. In order to train truly generalizable models we need to include all elements and sizes of molecules.

ANI1x[18,17] is a dataset based on active-learning and MD. The data generation procedure alternates between proposing new configurations using various forms of MD and pseudo-MD, and accepting or rejecting data based on the query by committee algorithm[23]. The dataset contains force and energy calculations for 6 million molecular configurations.

QM9 (QM9x) is a ubiquitous benchmark dataset for Quantum Mechanics (QM) methods that contains a multitude of QM features for 135.000 molecular equilibrium configurations. In order to make QM9 compatible with ANI1x and Transition1x we recalculated all configurations with the appropriate level of DFT, and we refer to it as QM9x.

PaiNN

PaiNN[14] is a message-passing GNN architecture designed for predicting molecular properties of atomic systems represented as graphs. The molecular graph is generated by a neighborhood function

	Barrier			NEB Convergence		
	MAE [eV]	RMSE [eV]	RMSD [Å]	Rate	Avg. CPU Time	Avg. Iterations
ANI1x	0.51(1)	1.67(3)	0.28(2)	69.3%	37s	149
T1x	**0.23(3)**	**0.52(1)**	**0.21(1)**	80.3%	33s	135
QM9x	3.4(1)	3.59(8)	0.59(2)	35.0%	**28s**	111
DFTB	0.70	0.85	0.22	65.7%	82s	114
DFT	-	-	-	**84.1%**	12h14m43s	**101**

Table 1: Comparison of potentials for Nudged Elastic Band (NEB). ANI1x, Transition1x and QM9x indicate PaiNN models trained on the respective dataset. The barrier column shows Mean Average Error (MAE) and Root Mean Squared Error (RMSE) of activation energies, and the Root Mean Square Deviation of atomic positions (RMSD) between transition states found with DFT and surrogate potentials. The Convergence column displays the convergence rate, average CPU time to compute a reaction, and average iterations before convergence.

that assigns edges between atoms if they are sufficiently close. The network calculates molecular features by letting neighboring atoms exchange messages calculated from their internal representations. Conservative forces can be calculated by the back-propagation algorithm as the negative gradient of energy with respect to positions.

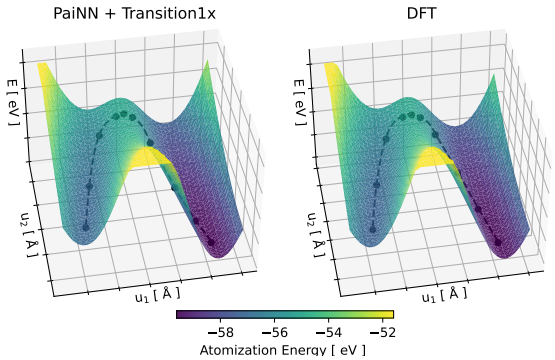
Five hundred reactions are set aside from Transition1x for evaluating various models' capabilities to find reaction pathways. Five equivalent PaiNN models are trained on each of ANI1x, QM9x, and the remaining data from Transition1x. The models have three hidden layers with 256 neurons in each. The molecular graph is generated with a cutoff radius of 5 Å. The models are trained with the ADAM optimizer, a batch size of 75, an initial learning rate of 10^{-3}, and a scheduler that scales the learning rate with a factor of 0.8 if the model has not improved for 5000 training steps.

Results

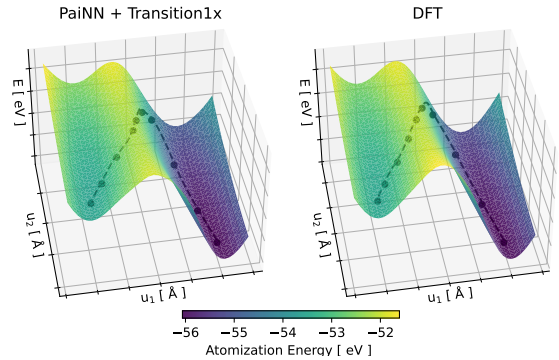
We trained five PaiNN models on each of Transition1x, ANI1x, and QM9x, and evaluated their capability to predict transition states on test reactions set aside from Transition1x. In Table 1 we report convergence rate, timings, Mean Average Error (MAE), Root Mean Squared Error (RMSE) and Root Mean Square Deviation of atomic positions (RMSD) of the various models' predictions. Datasets in the leftmost column represent PaiNN models trained on the respective dataset. Density Functional based Tight Binding (DFTB), is a fast and cheap approximation to DFT, used as a benchmark. We do not report standard deviation of MAE and RMSE of DFTB predictions as the algorithm is deterministic, whereas the performance of the PaiNN models depends on the training seed. The predicted activation energy is the difference between energies of reactant and transition state, while the error is the difference between the activation energies predicted by the surrogate potential and DFT. ML potentials trained on Transition1x outperform all other tested surrogate potentials in terms of MAE, RMSE and RMSD. QM9x contains only equilibrium configurations, and its models have not learned the intricacies of the PES around reaction pathways. This is reflected in the results through low convergence rates and high errors.

Figure 1 compares cross sections of PESs, spanned by reactant, product, and transition state of the reactions, calculated using DFT, and PaiNN trained on Transition1x, for two different reactions. The x and y axes are basis vectors describing the plane in units of Å, and the z-axis and color-coding show the atomization energy of configurations in the plane in eV. Not only does PaiNN trained on the Transition1x accurately calculate the barrier energy for the reaction, it also predicts an almost identical PES in the vicinity of the MEP and correctly identifies the plane spanned by the configurations defining the reaction.

Figure 2 displays the performance of each surrogate potential. Panel *a* is the distribution of RMSDs between transition states predicted by surrogate potentials and DFT. The distributions are unnormalized to reflect convergence ratios. Panel *b* compares activation energies found by DFT on the x-axis with energies found by surrogate potentials on the y-axis. Models trained on ANI1x tend to overestimate activation energies as they have not seen the low energy valleys connecting equilibrium configurations, while DFTB tends to underestimate activation energies. Correcting ANI1x and DFTB

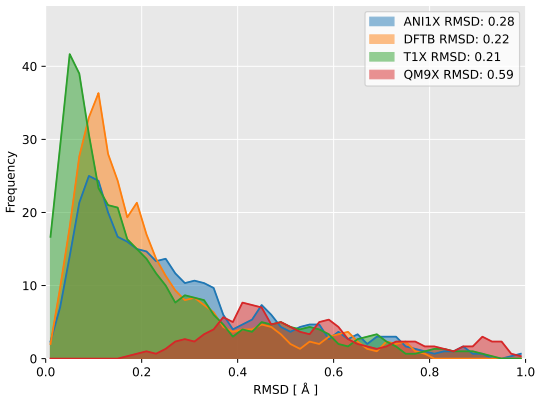


(a) Reaction involving C4N2H8. Follow this link to see GIF of reaction.

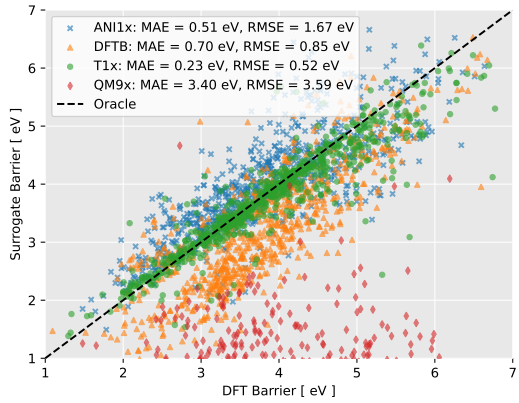


(b) Reaction involving C4NOH7. Follow this link to see GIF of reaction.

Figure 1: Minimal Energy Paths (MEPs) for two different reactions, found with the Nudged Elastic Band (NEB) algorithm, applying the Graph Neural Network (GNN) architecture PaiNN, trained on Transition1x, and DFT as potentials. The MEPs are projected onto intersections of the Potential Energy Surfaces (PESs) spanned by product, reactant and transition-state of the converged MEPs. The x- and y-axes are basis vectors describing the plane. The PESs have been calculated in the vicinity of the MEPs with the respective potential and is shown with colors and on the z-axis.



(a)



(b)

Figure 2: Performance of surrogate potentials compared to DFT. Panel *(a)* displays the unnormalized distribution of Root Mean Square Deviation of atomic positions (RMSD) between transition-states found by DFT and surrogate potentials. Panel *(b)* displays activation energies found by DFT on the x-axis, and found by surrogate potentials on the y-axis. Points that lie on the dashed line have been calculated perfectly.

for systematic error leads to a MAE of 0.48 eV and 0.48 eV and RMSE of 1.66 eV and 0.62 eV for ANI1x models and DFTB, respectively.

Conclusion

Relevant data is as important as expressive models for solving higher-order tasks in computational chemistry. We have presented the Transition1x dataset which contain force and energy calculations for 10 million molecular configurations on and around reaction pathways, and used it to train a fast

and accurate ML calculator that can predict reaction pathways for general organic reactions. We achieved a MAE of 0.23 eV on activation energies on unseen reactions when compared to evaluating the PES with DFT, while simultaneously speeding up the MEP-search significantly.

Impact Statement

Transition1x and QM9x

We believe that Transition1x [24] is an important contribution to the completeness of available data in the literature for ML for molecular science. It provides a different type of data, facilitating new downstream tasks for ML models. It is calculated with the 6-31G(d) [21] basis set and ωB97x [22] functional, which makes it compatible with the ANI1x [17,18], and permits training models with rich representations by leveraging strengths from both datasets. The ubiquitous QM9 [25], dataset was recalculated with the appropriate level of theory and released under the name QM9x [24]. Data loaders, examples, and scripts for Transition1x and QM9x are available in their respective repositories - Transition1x: https://gitlab.com/matschreiner/Transition1x, and QM9x: https://gitlab.com/matschreiner/QM9x. The data collection procedure for Transition1x is scalable and can easily be extended to include new elements and reactions.

NeuralNEB

NeuralNEB is an accurate, inexpensive, and general ML calculator for transition-state and MEP search that outperforms NEB with DFTB [26] as potential, both in terms of accuracy and computational cost. It yields a new and better trade-off between speed and accuracy for screening of large reaction-networks.

References

[1] Daniel Sheppard, Rye Terrell, and Graeme Henkelman. Optimization methods for finding minimum energy paths. *The Journal of Chemical Physics*, 128:134106, 4 2008. ISSN 0021-9606. doi: 10.1063/1.2841941. URL https://aip.scitation.org/doi/abs/10.1063/1.2841941.

[2] Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 12 2019. doi: 10.1016/j.neunet.2020.06.006. URL http://arxiv.org/abs/1912.12693http://dx.doi.org/10.1016/j.neunet.2020.06.006.

[3] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. 2021. doi: 10.1016/j.aiopen.2021.01.001. URL https://doi.org/10.1016/j.aiopen.2021.01.001.

[4] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole Von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid dft error. *Journal of Chemical Theory and Computation*, 13:5255–5264, 11 2017. ISSN 15499626. doi: 10.1021/ACS.JCTC.7B00577. URL https://pubs.acs.org/doi/abs/10.1021/acs.jctc.7b00577.

[5] Julia Westermayr, Michael Gastegger, Kristof T. Schütt, and Reinhard J. Maurer. Perspective on integrating machine learning into computational chemistry and materials science. *The Journal of Chemical Physics*, 154:230903, 6 2021. ISSN 0021-9606. doi: 10.1063/5.0047760. URL https://aip.scitation.org/doi/abs/10.1063/5.0047760.

[6] Stuart I Campbell, Daniel B Allan, and Andi M Barbour. Machine learning for the solution of the schrödinger equation. *Machine Learning: Science and Technology*, 1:013002, 4 2020. ISSN 2632-2153. doi: 10.1088/2632-2153/AB7D30. URL https://iopscience.iop.org/article/10.1088/2632-2153/ab7d30https://iopscience.iop.org/article/10.1088/2632-2153/ab7d30/meta.

[7] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[8] Julia Westermayr and Philipp Marquetand. Machine learning for electronically excited states of molecules. *Chemical Reviews*, 121:9873–9926, 8 2021. ISSN 15206890. doi: 10.1021/ACS.CHEMREV.0C00749. URL https://pubs.acs.org/doi/full/10.1021/acs.chemrev.0c00749.

[9] Jean Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48:722–730, 3 2015. ISSN 15204898. doi: 10.1021/AR500432K. URL https://pubs.acs.org/doi/full/10.1021/ar500432k.

[10] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98:146401, 4 2007. ISSN 00319007. doi: 10.1103/PHYSREVLETT.98.146401/FIGURES/4/MEDIUM. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.98.146401.

[11] Jörg Behler. Constructing high-dimensional neural network potentials: A tutorial review. *International Journal of Quantum Chemistry*, 115:1032–1050, 8 2015. ISSN 1097-461X. doi: 10.1002/QUA.24890. URL https://onlinelibrary.wiley.com/doi/full/10.1002/qua.24890.

[12] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole Von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid dft error. *Journal of Chemical Theory and Computation*, 13:5255–5264, 11 2017. ISSN 15499626. doi: 10.1021/ACS.JCTC.7B00577. URL https://pubs.acs.org/doi/abs/10.1021/acs.jctc.7b00577.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. 2017.

[14] Kristof T Schütt, Sch¨ Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. 2021.

[15] Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Sankar Raman Sathyamoorthy, and Cristofer Englund. Performance analysis of out-of-distribution detection on various trained neural networks. 2021. URL https://www.iso.org/deliverables-all.html.

[16] Lily H Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. 2021.

[17] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E. Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148: 241733, 5 2018. ISSN 0021-9606. doi: 10.1063/1.5023802. URL https://aip.scitation.org/doi/abs/10.1063/1.5023802.

[18] Justin S. Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E. Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific Data 2020 7:1*, 7:1–10, 5 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0473-z. URL https://www.nature.com/articles/s41597-020-0473-z.

[19] Graeme Henkelman, Blas P. Uberuaga, and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 113:9901, 11 2000. ISSN 0021-9606. doi: 10.1063/1.1329672. URL https://aip.scitation.org/doi/abs/10.1063/1.1329672.

[20] Colin A. Grambow, Lagnajit Pattanaik, and William H. Green. Reactants, products, and transition states of elementary chemical reactions based on quantum chemistry. *Scientific Data*, 7, 12 2020. ISSN 20524463. doi: 10.1038/s41597-020-0460-4.

[21] R. Ditchfield, W. J. Hehre, and J. A. Pople. Self-consistent molecular-orbital methods. ix. an extended gaussian-type basis for molecular-orbital studies of organic molecules. *The Journal of Chemical Physics*, 54:724, 9 2003. ISSN 0021-9606. doi: 10.1063/1.1674902. URL https://aip.scitation.org/doi/abs/10.1063/1.1674902.

[22] Jeng Da Chai and Martin Head-Gordon. Systematic optimization of long-range corrected hybrid density functionals. *The Journal of Chemical Physics*, 128:084106, 2 2008. ISSN 0021-9606. doi: 10.1063/1.2834918. URL https://aip.scitation.org/doi/abs/10.1063/1.2834918.

[23] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 287–294, 1992. doi: 10.1145/130385.130417. URL https://www.researchgate.net/publication/221497539_Query_by_Committee.

[24] Schreiner M., Bhowmik A., Vegge T., and Busk J.and Winther O. Transition1x – a Dataset for Building Generalizable Reactive Machine Learning Potentials. 6 2022. doi: 10.6084/m9.figshare.19614657.v4. URL https://figshare.com/articles/dataset/Transition1x/19614657.

[25] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data 2014 1:1*, 1:1–7, 8 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422.

[26] Gotthard Seifert and Jan Ole Joswig. Density-functional tight binding—an approximate density-functional theory method. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2:456–465, 5 2012. ISSN 1759-0884. doi: 10.1002/WCMS.1094. URL https://onlinelibrary.wiley.com/doi/full/10.1002/wcms.1094.

APPENDIX D

Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics

Implicit Transfer Operator Learning: Multiple Time-Resolution Surrogates for Molecular Dynamics

Mathias Schreiner[*]
DTU[†]
matschreiner@gmail.com

Ole Winther
DTU

Simon Olsson[‡]
Chalmers University of Technology
simonols@chalmers.se

Abstract

Computing properties of molecular systems rely on estimating expectations of the (unnormalized) Boltzmann distribution. Molecular dynamics (MD) is a broadly adopted technique to approximate such quantities. However, stable simulations rely on very small integration time-steps (10^{-15} s), whereas convergence of some moments, e.g. binding free energy or rates, might rely on sampling processes on time-scales as long as 10^{-1} s, and these simulations must be repeated for every molecular system independently. Here, we present Implict Transfer Operator (ITO) Learning, a framework to learn surrogates of the simulation process with multiple time-resolutions. We implement ITO with denoising diffusion probabilistic models with a new SE(3) equivariant architecture and show the resulting models can generate self-consistent stochastic dynamics across multiple time-scales, even when the system is only partially observed. Finally, we present a coarse-grained CG-SE3-ITO model which can quantitatively model all-atom molecular dynamics using only coarse molecular representations. As such, ITO provides an important step towards multiple time- and space-resolution acceleration of MD.

1 Introduction

Numerical simulation of stochastic differential equations (SDE) is critical in the sciences, including statistics, physics, chemistry, and biology applications [1]. Molecular dynamics (MD) simulations are an important example of such simulations [2]. These simulations prescribe a set of mechanistic rules governing the time evolution of a molecular system through numerical integration of, for example, the Langevin equation [3]. MD grants mechanistic insights into experimental observables. These observables are expectations, including time-correlations, of observable functions (e.g., pairwise distances or angles) computed for the Boltzmann distribution $\hat{\mu}(\mathbf{x}) \propto \exp[-\beta U(\mathbf{x})]$ corresponding to the *potential* $U(\cdot) : \Omega \to \mathbb{R}$ of a M-particle molecular system, $\mathbf{x} \in \Omega \subset \mathbb{R}^{3M}$ kept at the inverse temperature $\beta = 1/kT$. However, stable numerical integration relies on time steps, τ, which are strictly smaller than the fastest characteristic time-scales of the molecular system (10^{-15} s, e.g., bond vibrations), yet many molecular systems are characterized by processes on

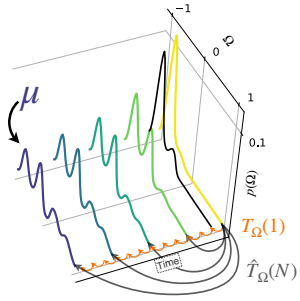


Figure 1: **Implicit Transfer Operator:** A multiple time-scale surrogate of stochastic molecular dynamics.

[*]Contributions to this work were done while visiting Chalmers University of Technology
[†]Technical University of Denmark
[‡]Corresponding author

Preprint. Under review.

much longer time-scales ($10^{-3} - 10^{-1}$ s, e.g. protein-folding,
protein-ligand unbinding, regulation). Consequently, we need infeasibly long simulations to characterize many important processes quantitatively due to the slow mixing in Ω.

In this work, we present the implicit Transfer Operator (ITO, Fig. 1) as an effective way to learn multiple time-step surrogate models of the stochastic generating distribution of MD. To our knowledge, this is the first surrogate modeling approach that allows for the simultaneous generation of stochastic dynamics at multiple different time resolutions. By adopting an SE(3)-equivariant generative model, we further demonstrate stable long-time-scale dynamics in increasingly difficult settings where an increasing number of degrees of freedom are marginalized. Our approach can be several orders of magnitude more efficient than direct MD simulations and can be made asymptotically unbiased if the generative model permits exact likelihood evaluation.

Our contributions are

1. the **Implicit Transfer Operator (ITO)** framework for learning generative models for multiple time resolution molecular dynamics simulations,
2. **ChiroPaiNN** an efficient SE(3) equivariant message passing neural network,
3. implementation of ITO using a denoising diffusion probabilistic model (DDPM) [4] with strong empirical results across resolutions: **SE(3)-equivariant ITO model (SE3-ITO)** gives stable long time-scale simulations and self-consistent dynamics across multiple time-scales for molecular benchmarks and **Coarse-grained SE3-ITO model (CG-SE3-ITO)** trained on large-scale protein folding data sets shows quantitative agreement with major dynamic and stationary observables of interest.

2 Background and Preliminaries

Notation Throughout this work, diffusion time, related to Diffusion Models (see Sec. 2), and physical time are represented using superscripts and subscripts, respectively.

Molecular dynamics and observables Molecular dynamics (MD) is a wide-spread simulation strategy in computational chemistry and physics. In this approach, the time-evolution of N particles configuration in Euclidean space $\mathbf{x} \in \Omega \subset \mathbb{R}^{3M}$, is modeled via a stochastic differential equation (SDE) with a drift term based on a potential energy model $U(\mathbf{x}) : \Omega \to \mathbb{R}$. An important aim of MD is to compute:

1. **Stationary observables:** $O_f = \mathbb{E}_\mu[f(\mathbf{x})]$
2. **Dynamic observables:** $O_{f(t)h(t+\Delta t)} = \mathbb{E}_{\mathbf{x}_t \sim \mu}[\mathbb{E}_{\mathbf{x}_{t+\Delta t} \sim p_\tau(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)}[f(\mathbf{x}_t)h(\mathbf{x}_{t+\Delta t})]]$

where μ is the normalized Gibbs measure, and $p_\tau(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t)$ is a conditional probability density function encoding the time-discrete evolution of the molecular system $\mathbf{x}$, with time-step $\Delta t = N\tau$ as prescribed by a dynamic model, e.g. *Langevin dynamics* [3], integrated with time-step, τ. N is typically a large integer. The functions $f, h : \Omega \to \mathbb{R}$ are observable functions or '*forward models*' describing the microscopic observation process, e.g. computing a distance or an angle. The observables, $O_{f(t)}$, and $O_{f(t)h(t+\Delta t)}$, include binding affinities and binding rates of a drug to a protein, respectively. Conventionally, these observables are estimated from simulation trajectories using naive Monte Carlo estimators.

For illustrative purposes, we assume the temporal behavior of a state, $\mathbf{x}$, follows the Brownian dynamics SDE (Itô form)

$$\mathrm{d}\mathbf{x}_t = -\nabla U(\mathbf{x}_t)\gamma^{-1}\,\mathrm{d}t + \sqrt{2D}\mathrm{d}W, \tag{1}$$

where $D = \gamma^{-1}\beta^{-1}$ is a diffusion constant, with friction γ and inverse-temperature β, and $\mathrm{d}W$ is a Wiener process. Using the Euler–Maruyama time-discretization, with time-step τ, simulating the SDE corresponds to simulating a Markov chain with the transition probability density

$$p(\mathbf{x}_{t+\tau} \mid \mathbf{x}_t, \tau) = \mathcal{N}(\mathbf{x}_{t+\tau}|\mathbf{x}_t - \tau\nabla U(\mathbf{x}_t)\gamma^{-1}, \tau\sqrt{2D}\mathbb{I}_{3M}) \tag{2}$$

where $\mathcal{N}$ specifies the multi-variate Normal distribution, and $\mathbb{I}_{3M}$ is the $3M$-dimensional identity matrix. If τ is sufficiently small to allow stable simulation, the *invariant measure*, of the Markov chain

(eq. 2), is the Boltzmann distribution (normalized Gibbs measure) corresponding to the potential energy model $U(\mathbf{x})$ at β. Consequently, by simulating a large number of steps we can draw samples from μ to compute stationary observables and compute dynamic observables by simulating $\Delta t = N\tau$ steps enough times with initial states distributed according to μ. Explicit simulation make such computations extremely costly, and consequently, there's much interest in speeding up the calculations of these quantities.

Transfer Operators Let ρ specify an initial condition, a probability density function on Ω. We can define a Markov operator $T_\Omega : L^1(\Omega) \to L^1(\Omega)$ using a transition density (e.g., 2):

$$\left[T_\Omega \circ \rho\right](\mathbf{x}_{t+\tau}) \triangleq \frac{1}{\mu(\mathbf{x}_{t+\tau})} \int_{x_t} \mu(\mathbf{x}_t)\rho(\mathbf{x}_t)p(\mathbf{x}_{t+\tau} \mid \mathbf{x}_t)\mathrm{d}\mathbf{x}_t \tag{3}$$

which then describes the μ-weighed evolution of absolutely convergent probability density functions on Ω according to eq. 1, with time-step, τ. Such an operator is called the (Ruelle) Transfer Operator [5, 6]. We can express the operator using a spectral form

$$T_\Omega(\tau) = \sum_{i=0}^{\infty} \lambda_i(\tau)|\psi_i\rangle\langle\phi_i| \tag{4}$$

where only eigenvalues $\lambda_i(\tau) = \exp(-\tau\kappa_i)$ depend on the time-step, τ. κ_i are characteristic 'relaxation' rates associated the left and right eigenfunction pair, ϕ_i and ψ_i [7]. We can compute the operator with time-lag $N\tau$ via the Chapman-Kolmogorov equation (see Sec. A.1, for details)

$$T_\Omega(N\tau) = \sum_{i=0}^{\infty} \lambda_i(\tau)^N|\psi_i\rangle\langle\phi_i|. \tag{5}$$

Equivariant Message Passing Neural Networks In this work, we are concerned with MD, where the time-evolution of a molecule is governed by a force field $\mathcal{F}(\cdot) \triangleq -\nabla U(\cdot)$ derived from a central potential $U(\cdot)$. While $U(\cdot)$ is *invariant* to group-actions of the Euclidean group in three dimensions (E(3)), its corresponding force field is E(3)-*equivariant*. We call a function, f '*invariant*' under a group-action g iif $f(\mathbf{x}) = f(S_g\mathbf{x})$ and '*equivariant*' iff $T_g f(\mathbf{x}) = f(S_g\mathbf{x})$, where S_g and T_g are linear representations of the group element g [8].

While the force field $\mathcal{F}(\cdot)$ is equivariant under E(3) group-actions, in practice, classical molecular dynamics simulations do not change parity during simulation, and consequently, our data generating distribution — molecular dynamics — is equivariant under actions of the special E(3) (SE(3)) group, also called the group of rigid body motions, which excludes reflections.

As we aim to distinguish mirror images of molecules, we extend PaiNN [9], an E(3)-equivariant message passing neural network (MPNN), making it SE(3) equivariant by breaking its symmetry with respect to parity. Briefly, PaiNN embeds a graph $G = (V, E)$, where nodes, V, exchange equivariant messages through edges within a local neighborhood defined as $\mathcal{N}(i) = \{j \mid \|r_{ij}\| \leq r_{\text{cutoff}}\}$, where r_{ij} is the distance between nodes denoted i and j, and r_{cutoff} is the maximal distance at which nodes are allowed to exchange messages. Messages are pooled and subsequently used to update node features, thereby enabling exchange of equivariant information. We achieve parity symmetry-breaking by constructing the equivariant messages in a manner that depends on cross-products between equivariant node features and direction vectors between interacting nodes. The cross-product is an axial vector (i.e., does not change sign under parity). Consequently, by combining these vectors with polar vectors (change sign under parity), the model can learn to distinguish chiral molecules. We refer to this modified PaiNN architecture as ChiroPaiNN (CPaiNN). Further details are in the Appendix D.

Diffusion Models The diffusion model (DM) formalism is a powerful generative modeling framework that learns distributions by modeling a gradual denoising process [4, 10, 11]. In DMs, we pre-specify a *forward diffusion process* (noising process), which gradually transforms the data distribution $p(\mathbf{x}^0)$ to a simple prior distribution $p(\mathbf{x}^T)$, e.g., a standard Gaussian, through a time-inhomogenous Markov process, described by the following SDE (Itô form)

$$\mathrm{d}\mathbf{x}^t = f(\mathbf{x}^t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}W. \tag{6}$$

where $0 < t < T$ is the *diffusion time*, f and g are chosen functions, and $\mathrm{d}W$ is a Wiener process. We can generate samples from the data distribution $p(\mathbf{x}^0)$ by sampling from $p(\mathbf{x}^T)$ and solving the *backward diffusion process* (denoising process)

$$\mathrm{d}\mathbf{x}^t = \left[f(\mathbf{x}^t, t) - g^2(t)\nabla_{\mathbf{x}^t} \log p(\mathbf{x}^t \mid t)\right] \mathrm{d}t + g(t)\,\mathrm{d}W \tag{7}$$

by approximating the *score field* $\nabla_{\mathbf{x}^t} \log p(\mathbf{x}^t \mid t)$ — or equivalently a time-dependent Gaussian transition kernel [4] — with a deep neural network surrogate $\nabla_{\mathbf{x}^t} \log \hat{p}(\mathbf{x}^t \mid t, \boldsymbol{\theta})$. We can use the learned score field to define a neural ordinary differential equation (ODE) [12, 13], or probability flow ODE [14] — eq. 7 less the term $g(t)\,\mathrm{d}W$ and scaling $g^2(t)$ by $1/2$ — which we can leverage for efficient sampling and sample likelihood evaluation.

Here, we are concerned with building equivariant probability density functions under SE(3) group actions. Consequently, we parameterize the DM using a learned Gaussian transition kernel of a time-inhomogenous diffusion process. By restricting the transition kernels $p(\mathbf{x}^{t+1} \mid \mathbf{x}^t)$ to be equivariant under SE(3) group-actions, the marginal of $\mathbf{x}^{t+1}$ is always invariant [15]. Combining the equivariant transition kernel with an invariant prior density [16] ensures the whole Markov process is invariant to SE(3) group actions. Consequently, combining an isotropic mean-free Gaussian as prior with ChiroPaiNN-parameterized transition kernels, we can construct an SE(3) equivariant diffusion model.

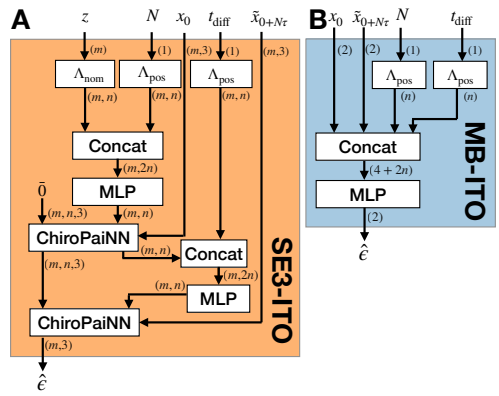


Figure 2: **ITO $\hat{\epsilon}$ networks** (A) SE3-ITO used for molecular application (B) MB-ITO, used for experiments with the Müller-Brown potential. Λ_{pos} and Λ_{nom} are positional and nominal embedding respectively, Concat is a concatenation, and MLP is a multi-layer perceptron. Arrows are annotated with input and output shapes.

3 Implicit Transfer Operator

We here consider simulation trajectories of a system as ancestral samples from a conditional probability density: $\mathbf{X} = \{\mathbf{x}_\tau, \ldots, \mathbf{x}_{N\tau}\} \sim p(\mathbf{x}_{n\tau} \mid \mathbf{x}_{(n-1)\tau})$, with $n = \{1, \ldots, N\}$, generated by explicit simulation by time-discretization of, for example, the Langevin equation. τ is a '*physical time-step*' determined by the integration scheme used to carry out MD. In general, the state variable $\mathbf{x}$, contains both position and velocity information of the particles, along with other details such as box dimensions, depending on the simulation scheme and target ensemble. Throughout this study, we only consider the position information.

If our MD simulation is performed with time-invariant potential energy (drift), we can express the generating transition probability as a decomposition of time-variant and -invariant parts (Proof, see Sec. A.2)

$$p(x_{N\tau} \mid x_0) = \sum_{i=1}^{\infty} \underbrace{\lambda_i^N(\tau)}_{\text{time-variant}} \underbrace{\alpha_i(x_{N\tau})\beta_i(x_0)}_{\text{time-invariant}} \tag{8}$$

where α_i and β_i are *time-invariant* projection coefficients of the state variables on-to the left and right eigenfunctions ϕ_i and ψ_i, of the *Transfer operator* $T_\Omega(\tau)$ [5] and $|\lambda_i(\tau)| \leq 1$ is its i'th eigenvalue.

We build a surrogate of the conditional transition probability distribution (eq. 8) from MD data. In practice, we learn a generative model $\mathbf{x}_{t+N\tau} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{t+N\tau} \mid \mathbf{x}_t, N)$ with a conditional denoising diffusion probabilistic model (cDDPM) of the form

$$p(\mathbf{x}^0_{t+N\tau} \mid \mathbf{x}_t, N) \triangleq \int p(\mathbf{x}^{0:T}_{t+N\tau} \mid \mathbf{x}_t, N)\,\mathrm{d}\mathbf{x}^{1:T} \tag{9}$$

where $\mathbf{x}^{1:T}$ are *latent variables* of the same dimension as our output, and follow a joint density describing the backward diffusion process (eq. 7) and $\mathbf{x}^T \sim \mathcal{N}(0, \mathbb{I})$. We define a conditional sample likelihood as

$$\ell(\mathbf{I}; \theta) \triangleq \prod_{i \in \mathbf{I}} p_{\boldsymbol{\theta}}(\mathbf{x}^0_{t_i + N_i \tau} \mid \mathbf{x}_{t_i}, N_i) \tag{10}$$

where $\mathbf{I}$ is a list of generated indices i specifying a time t_i and a time-lag (τ) integer multiple N_i, associating two time-points in the trajectory, $\mathbf{X}$. Following Ho et al., we train the cDDPM by optimizing a simplified form of the variational bound of the log-likelihood [4],

$$\mathcal{L}(\theta) = \mathbb{E}_{i \sim \mathbf{I}, \epsilon \sim \mathcal{N}(0,\mathbb{I}), t_{\text{diff}} \sim \mathcal{U}(0,T)} \left[\| \epsilon - \hat{\epsilon}_{\boldsymbol{\theta}}(\widetilde{\mathbf{x}}^{t_{\text{diff}}}_{t_i + N_i \tau}, \mathbf{x}_{t_i}, N_i, t_{\text{diff}}) \|_2 \right], \tag{11}$$

where $\widetilde{\mathbf{x}}^{t_{\text{diff}}}_t \triangleq \sqrt{\bar{\alpha}^{t_{\text{diff}}}} \mathbf{x}_t + \sqrt{1 - \bar{\alpha}^{t_{\text{diff}}}} \epsilon$, with $\bar{\alpha}^{t_{\text{diff}}} = \prod_i^{t_{\text{diff}}} (1 - \beta_i)$ and β_i is the variance of the forward diffusion process at diffusion time, i. $\hat{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ is one of the two ITO neural network model architectures shown in Fig. 2, and is directly related to the score [4].

As outlined in Algorithm 1, we generate the indices $i \in \mathbf{I}$, in a manner such that the model is exposed to multiple time-lags, sampled uniformly across orders of magnitude, used for gradient-based optimization with Adam [17]. As a result, the model will be exposed to multiple different linear combinations of the eigenfunctions of $T_\Omega(\tau)$ in each batch during training. We conjecture that this data augmentation procedure will enable better learning of implicit representations of these eigenfunctions and, consequently, better generalization across time scales and yield more stable sampling.

3.1 ITO Architectures

We present two architectures for learning cDDPMs encoding ITO models, one for molecular applications SE3-ITO and one for the Müller-Brown benchmark system (Fig. 2). The SE3-ITO architecture uses our new SE(3) equivariant MPNN (ChiroPaiNN, described in sec. 2) to encode $\mathbf{x}_t$, N, and atom-types, z, to invariant features, s, and equivariant features, v. We concatenate s with an encoding of the diffusion-time t_{diff} and process them through a MLP (multi-layer perceptron). The output from the MLP are passed along with v and $\widetilde{\mathbf{x}}^{t_{\text{diff}}}_t$ as input to a second ChiroPaiNN module which predicts $\hat{\epsilon}$. More details on the architecture and hyperparameters are available in Appendices D and E.

Algorithm 1 Training. DisExp is defined in Appendix E

Input: n MD-trajectories; $\mathcal{X} = \{\mathbf{x}^j_0, \ldots, \mathbf{x}^j_{t_j}\}^n_{j=0}$, ITO score-model; $\hat{\epsilon}_\theta$, max lag; N_{max}
$\mathcal{X}' = \text{Concatenate}(\{\mathbf{x}^j_0, \ldots, \mathbf{x}^j_{t_j - N_{\text{max}}}\}^n_{j=0})$
while not converged **do**
  $\mathbf{x}_t \sim \text{Choice}(\mathcal{X}')$
  $N \sim \text{DisExp}(N_{\text{max}})$
  $t_{\text{diff}} \sim \text{Uniform}(0, T)$
  Take gradient step on:
  $\nabla_{\boldsymbol{\theta}} \left[\| \epsilon - \hat{\epsilon}_{\boldsymbol{\theta}}(\widetilde{\mathbf{x}}^{t_{\text{diff}}}_{t+N\tau}, \mathbf{x}_t, N, t_{\text{diff}}) \|_2 \right]$
end while
return $\hat{\epsilon}_{\boldsymbol{\theta}}$

Algorithm 2 Ancestral sampling. Sampling from $p_{\boldsymbol{\theta}}$ is defined in Appendix E, Algorithm 4

Input: initial condition $\mathbf{x}_0$, lag N, nesting steps n.
Allocate $\mathcal{T} \in \mathbb{R}^{(n+1) \times \dim(\mathbf{x}_0)}$
$\mathcal{T}[0] = \mathbf{x}_0$
for $i = 1 \ldots n$ **do**
  $\mathbf{x}_i \sim \hat{p}_{\boldsymbol{\theta}}(\mathcal{T}[i-1], N)$
  $\mathcal{T}[i] = \mathbf{x}_i$
end for
return $\mathcal{T}$

4 Long time-step stochastic dynamics with Implicit Transfer Operators

4.1 Datasets and test-systems

To evaluate how robustly ITO models can model long time-scale dynamics, we conducted three classes of experiments, ranging from fully observed, high time-resolution, to sparsely observed and low time resolution. Details on training and computational resources are available in Appendices E and F, respectively.

Table 1: **VAMP2 score-gaps.** Difference in VAMP2-scores of ancestral sampling from ITO models with fixed lag and stochastic lags, compared to baseline Langevin simulations. Perfect match is 0, negative and negative values correspond to under and over estimation of meta-stability, respectively. Standard deviations on last decimal place are given in parentheses.

system ╲ lag	10	100
Müller-Brown (fixed)	-0.0351 (5)	-0.1189 (2)
Müller-Brown (stochastic)	**-0.0312** (4)	**-0.0970** (5)

Müller-Brown is a 2D potential commonly used for benchmarking molecular dynamics sampling methods. We generate a training data-set by integrating eq. 1 with the Müller-Brown potential energy as $U(\mathbf{x})$ (For details, see Appendix B.1). This dataset corresponds to a fully observed case.

Alanine dipeptide We use publicly available data from MDshare [18]. Simulation is performed with 2 fs integration time-steps and data is saved at 1 ps intervals. The simulations are performed in explicit solvation, but we only model the 22 atoms of the solute, without considering velocities. Consequently, this dataset is only partially observed.

Fast-folding proteins We use molecular dynamics data previously reported by Lindorff-Larsen et al. on the fast-folding proteins Chignolin, Trp-Cage, BBA, and Villin [19]. The data is proprietary but available upon request for research purposes. The simulations were performed in explicit solvent with a 2.5 fs time-step and the positions was saved at 200 ps intervals. We coarse-grain the simulation by representing each amino-acid by the Euclidean coordinate of their $C\alpha$ atom as done previously [20], leading to 10, 20, 28, and 35 particles in each system respectively. Consequently, these data correspond to a mostly unobserved case.

4.2 Stochastic lag improves meta-stability prediction

In sec. 2, we conjecture that exposing an ITO model to multiple lag times during training leads to better and more robust models. To test this, we trained a set of models on the Müller-Brown dataset with fixed constant lags $N = \{10, 100, 1000\}$ (fixed lag) and a single model with $N \sim \mathrm{DisExp}(1000)$ (stochastic lag) using the MB-ITO model (Fig. 2).

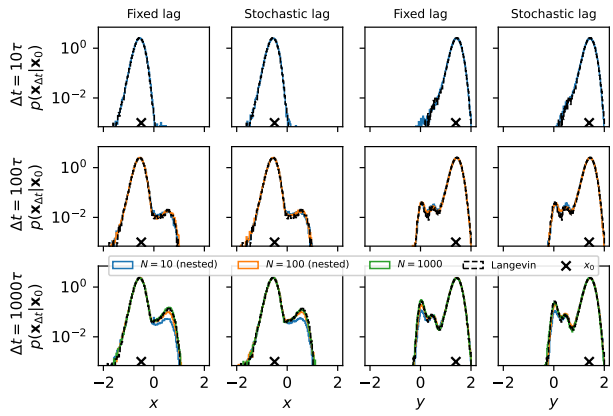


Figure 3: **Müller-Brown potential.** Conditional Probability Densities starting in x_0 indicated by cross, in ITO models trained with fixed or stochastic lag. Comparison of histograms of direct and ancestral sampling to direct simulation (Langevin). $N_{\mathrm{samples}=250k}$

We find that the model trained with a stochastic lag systematically outperforms models trained with fixed lag (Table 1). We gauge the agreement by comparing Variational Approach to Markov Processes (VAMP)-2 scores [21] (for details, see Appendix G), between model samples and training data and find that both models tend to underestimate meta-stability compared to training data slightly. However, the model trained with stochastic lag is marginally closer to the reference values. We note that the difference in the ability of fixed and

stochastic lag ITO models to capture long-time-scale dynamics is also reflected in the learned transition densities (Fig. 3). Together, these results suggest that lag-time augmentation during training leads to better implicit learning of the Transfer operator's eigenfunctions than training with a fixed lag.

4.3 Efficient and accurate self-consistent long time-scale dynamics

We evaluate the ITO models trained with stochastic lags to capture long time-scale dynamics in a self-consistent manner, in the Chapman-Kolmogorov sense, i.e., $p(\mathbf{x}_{\Delta t} \mid \mathbf{x}_0) \triangleq p(\mathbf{x}_{N\tau} \mid \mathbf{x}_0) = \prod_{i=1}^{N} p(\mathbf{x}_{i\tau} \mid \mathbf{x}_{(i-1)\tau})$, or if samples generated by direct sampling with time-step $\Delta t = N\tau$ are distributed similarly to samples generated by performing ancestral sampling N times, each with time-step τ.

For the fully-observed Müller-Brown case, we find that the ITO model is self-consistent by the strong overlap in transition densities sampled in a direct and ancestral manner (Algorithm 2). These results generalize to molecular systems and partially observed systems. Sampling an SE3-ITO model (Fig. 2) trained with alanine dipeptide data, we find strong agreement between the ancestrally and directly sampled transition densities (Fig. 4) and we again have a strong consistency with corresponding transition densities computed from molecular dynamics simulations. Note here, that the time-step of the ITO-sampled transition densities varies from 10^4 to 10^6 times the MD integration time-step.

Next, we consider four fast-folding proteins [19] where only the $C\alpha$ atoms are visible during model training. In this sparsely observed case (CG-SE3-ITO), we find strong model self-consistency, as shown by the comparison between conditional densities from the folded and unfolded states (Fig. 5) projected onto a linear subspace determined using *time-lagged independent component analysis* (time-lagged independent components, tIC) [22] (see Appendix B.3). Further, the long time-scale transition density gradually converges to the data distribution as expected.

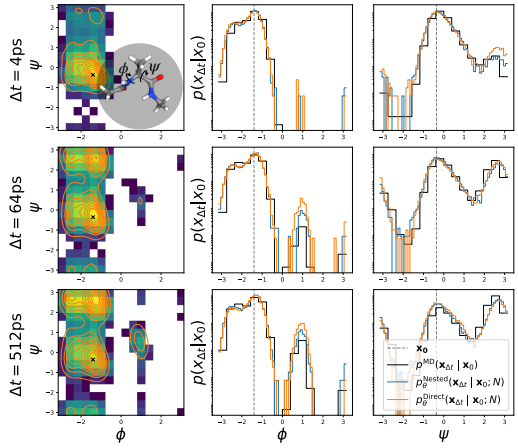


Figure 4: **Alanine dipeptide dynamics with SE3-ITO model**; Rows of increasing time-lag (from top to bottom). Contours are samples from SE3-ITO model, and 2D histograms show estimates from MD data. The first column shows conditional transition densities projected onto the torsion angles ϕ and ψ (inset). The black cross indicates the initial condition. The second and third columns show marginal distributions of ϕ and ψ, respectively, with direct sampling in orange, ancestral sampling in blue, and MD data in black.

Finally, by ancestral sampling (Algorithm 2), we perform a simulation of Chignolin with the same length as the training trajectory ($106\,\mu s$), using a CG-SE3-ITO model, and compare with MD. The CG-SE3-ITO simulation is 2120 steps with $\Delta t = 5\,\mathrm{ns}$. Running in parallel, on a single Titan X GPU we can simulate the CG-SE3-ITO model at a rate of $363\,\mathrm{ns}/(\mathrm{s}_w M^2)$ where s_w denotes seconds wall-time (Appendix C.2). Remarkably, these trajectories are virtually indistinguishable in the slowly relaxing TICA coordinates, illustrating stability of ITO. These conclusions extend to the proteins Villin, BBA, and Trp-Cage (See Appendix, Figs. 6,7 and 8)

Together these results suggest that ITO models accurately and self-consistently capture the slow dynamics of molecular systems and are robust to situations where the system is only partially observed. In general, we expect robustness to sparsely observed representations as long as the input representations are sufficient to span the eigenfunctions of T_Ω [23, 24]. Approximation errors will translate into systematic under-estimation of relaxation time-scales [7], consistent with our slight

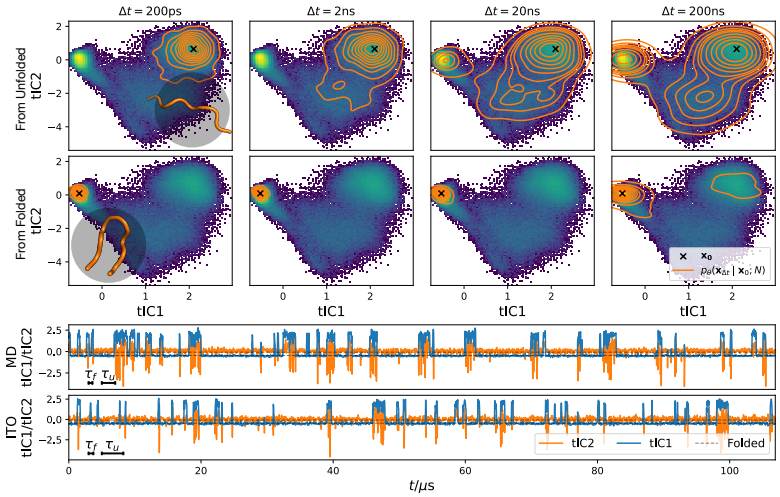


Figure 5: **Reversible protein folding-unfolding of Chignolin with CG-SE3-ITO** Conditional probability densities (orange contours) starting from unfolded (upper panels) and folded (lower panels) protein states, at increasing time-lag (left to right), shown on top of data distribution. Below: time-traces of 106 microsecond MD simulations and ITO simulations on tICs 1 and 2.

under-estimation of VAMP-2 scores (Table 1). In future work, combining the learning of SE3-ITO models with a systematic scheme for coarse-graining [25, 26], could be an avenue for scaling to large-scale molecular systems at a low computational cost.

5 Prediction of dynamic and stationary observables of using CG-SE3-ITO

As outlined in section 2, an important aim of MD simulations is to compute stationary and dynamic observables, which involves intractable integrals typically approximated via Monte Carlo estimators. Using the trained ITO models we can efficiently sample i.i.d. from the transition density needed for computing dynamic observables, and by choosing a time-step which is sufficiently large we can also sample i.i.d from the Boltzmann distribution μ, the latter akin to *Boltzmann generators* [27] (See Appendix A.1). We note that, the ITO models are surrogates and as such without reweighing we cannot expect unbiased samples from the Boltzmann and dynamic transition densities. Nevertheless, we gauge how accurately ITO models we can compute these observables of interest in the context of protein folding without reweighing:

- **Free Energy of Folding**, $\Delta G = -\log\left[\frac{p_f}{1-p_f}\right]$
- **Mean first passage time, folding**, $\langle \tau_f \rangle = \int_{x_0 \in \neg f} \int_0^\infty \delta(x_t \in f) p(x_t \mid x_0, t) t \, \mathrm{d}t \, \mathrm{d}x_0$
- **Mean first passage time, unfolding**,$\langle \tau_u \rangle = \int_{x_0 \in f} \int_0^\infty \delta(x_t \in \neg f) p(x_t \mid x_0, t) t \, \mathrm{d}t \, \mathrm{d}x_0$

where $\{f, \neg f\} \subset \Omega$ are disjoint subsets corresponding to the folded and unfolded states of a protein, $p_f = \int_{x \in f} \mu(x) \, \mathrm{d}x$, is the folded state probability and $\delta(\cdot)$ is the Dirac delta.

We compute these observables using the reference molecular simulation data [19] and samples statistics from the CG-SE3-ITO models of each of the four fast-folding proteins (details in Appendix C.1). Strikingly, the observables computed using CG-SE3-ITO models agree well with those computed from long all-atom MD simulations (Table 2).

We implemented all experiments using PyTorch[28], PyTorch Lightning[29], JAX[30], and used DPM-Solver[31] for probability flow ODE Sampling.

Table 2: **Molecular observables** Standard deviations on last decimal place are given in parentheses. Stationary and dynamic observables are denoted **s** and **d**, respectively.

	$\Delta G_{\text{fold}}/kT$ (**s**)	$\langle \tau_f \rangle / \mu s$ (**d**)	$\langle \tau_u \rangle / \mu s$ (**d**)
Chignolin (MD/ITO)	-1.28(1)/-1.53(2)	0.565(4)/0.700(8)	2.01(2)/3.24(4)
Trp-Cage (MD/ITO)	1.47(6)/2.84(6)	13.6(4)/37(2)	3.4(2)/2.85(9)
BBA (MD/ITO)	0.97(3)/1.52(3)	11.7(2)/8.6(2)	5.1(1)/1.75(4)
Villin (MD/ITO)	1.21(2)/2.22(3)	2.41(3)/3.27(7)	0.68(1)/0.354(5)

6 Related Work

Molecular sampling Sampling molecular configurations is a broad field and can broadly be divided into two main areas: physically motivated sampling of the Boltzmann distribution and conformer generation. The first area includes algorithmic approaches to sample the Boltzmann distribution including Molecular Dynamics simulations [2], Markov Chain Monte Carlo, extended ensemble methods [32, 33, 34], including analysis methods involving deep generative nets [35], and surrogate models which directly approximate the Boltzmann distribution and allow for recovery of unbiased statistics, including Boltzmann generators [36, 16]. Conformer generation concerns generating physically plausible conformers without explicitly trying to follow the Boltzmann distribution. The latter approaches can be split into ML [15, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46] and chemoinformatic [47, 48] approaches. Finally, speeding up molecular simulations by reducing the effective number of particles to simulate through coarse-graining with special purpose forcefield models [49] including machine learned variants [50, 51, 20, 52] and learned coarse-graining maps [25, 26] is an orthogonal approach to sample conformation space. Further, several methods to recover all-atom models from coarse-grained representations through ML [53, 54] and rule-based approaches [55] are available.

Transfer Operator surrogates Building transfer operator surrogates is commonly used in molecular modeling including (Deep Generative) Markov state models (MSM) [56, 7, 57, 58], dynamic graphical models,[59] VAMPnets[60, 21], observable operator models[61], however, primarily for analysis of molecular dynamics data. Markov state models are time-space discrete approximations of the transfer operator and Deep Generative MSM [62] and VAMPnets [60] are deep learning infused versions, where state discretization is learned by deep nets. Dynamic graphical models reparameterize MSMs as kinetic Markov random fields allowing for scaling to larger systems [59]. Klein et al. recently introduced *timewarp* which is a flow-based generative model to simulate molecular systems with a large (up to $0.5\,\text{ns}$), fixed, time-lag, [63] providing asymptotically unbiased equilibrium samples through a Metropolis-Hastings correction [64]. While timewarp generates conformers with realistic local structure, it has limitations in capturing long time-scale dynamics, which is reflected in the predicted transition probability densities. In contrast, our approach captures long time-scale dynamics efficiently allowing for accurate prediction of dynamic observables.

7 Limitations

Surrogate model Implicit Transfer Operators are surrogate models of stochastic dynamics' conditional transition probability densities. We cannot guarantee unbiased sampling of dynamics and the stationary distribution due to aleatoric (e.g., finite data) and epistemic (e.g., model misspecification) uncertainty. We can overcome the latter by reweighing against a Markov Chain Monte Carlo acceptance criterion as proposed previously [63], to ensure unbiased dynamics path-reweighing is necessary, which in turn requires closed-form expressions for the target path probabilities [65].

Transferability and scalability Currently, ITO does not generalize across chemical space and thermodynamic variables. In future work, we anticipate that generalization across chemical space limitations can be overcome by appropriate data set curation and parameter-sharing schemes. Generalization across thermodynamic variables such as temperature and pressure would require using a surrogate model which is steerable under these changes, e.g., temperature steerable flows [66]. Currently, we assume a fully connected graph that scales $\mathcal{O}(M^2)$ in system size, which limits what systems are practically accessible. Devising new surrogate models which use mean-field approxima-

tion approaches from e.g., computational physics [67] or chemistry to truncate the graphs and treat long-range as an additive term [68] could yield more favorable scaling [69].

8 Conclusions

This paper introduces Implicit Transfer Operators (ITO), an approach to building multiple time-scale surrogate models of stochastic molecular dynamics. We implement ITO models with a conditional DDPM using a new time-augmentation scheme and show how ITO models capture fast and slow dynamics on benchmarks and molecular systems. We show ITO models are self-consistent over multiple time scales and highly robust to the marginalization of degrees of freedom in the system, which are unimportant to capture the long-time-scale dynamics. Combined with a new SE(3) equivariant MPNN architecture (ChiroPaiNN), we further show strong empirical evidence of scaling to applications, such as the folding of coarse-grained proteins. As such, we are confident that ITO is a stepping-stone toward general-purpose surrogates of molecular dynamics.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation and The Novo Nordisk Foundation (SURE, NNF19OC0057822).

References

[1] Bernt Øksendal. *Stochastic Differential Equations*. Springer Berlin Heidelberg, 2003. DOI: 10.1007/978-3-642-14394-6. URL: https://doi.org/10.1007/978-3-642-14394-6.

[2] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation*. Elsevier, 2002. DOI: 10.1016/b978-0-12-267351-1.x5000-7. URL: https://doi.org/10.1016/b978-0-12-267351-1.x5000-7.

[3] Paul Langevin. "Sur la théorie du mouvement brownien". In: *C. R. Acad. Sci. (Paris)* 146 (1908), 530—533.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. DOI: 10.48550/ARXIV.2006.11239. URL: https://arxiv.org/abs/2006.11239.

[5] David Ruelle. *Thermodynamic Formalism*. en. Encyclopedia of mathematics and its applications. Harlow, England: Longman Higher Education, Nov. 1978.

[6] Christof Schütte et al. "Conformation dynamics". In: *Proc. Int. Congr. ICIAM* (2009), pp. 297–336.

[7] Jan-Hendrik Prinz et al. "Markov models of molecular kinetics: Generation and validation". In: *The Journal of Chemical Physics* 134.17 (May 2011), p. 174105. DOI: 10.1063/1.3565032. URL: https://doi.org/10.1063/1.3565032.

[8] Jean-Pierre Serre. *Linear Representations of Finite Groups*. Springer New York, 1977. DOI: 10.1007/978-1-4684-9458-7. URL: https://doi.org/10.1007/978-1-4684-9458-7.

[9] Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. *Equivariant message passing for the prediction of tensorial properties and molecular spectra*. 2021. eprint: arXiv:2102.03150.

[10] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2020. DOI: 10.48550/ARXIV.2011.13456. URL: https://arxiv.org/abs/2011.13456.

[11] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 2256–2265. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[12] Ricky T. Q. Chen et al. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

[13] Will Grathwohl et al. "Scalable Reversible Generative Models with Free-form Continuous Dynamics". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJxgknCcK7.

[14] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

[15] Minkai Xu et al. *GeoDiff: a Geometric Diffusion Model for Molecular Conformation Generation*. 2022. eprint: arXiv:2203.02923.

[16] Jonas Köhler, Leon Klein, and Frank Noe. "Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5361–5370. URL: https://proceedings.mlr.press/v119/kohler20a.html.

[17] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. eprint: arXiv:1412.6980.

[18] URL: https://markovmodel.github.io/mdshare/ALA2/#alanine-dipeptide.

[19] K. Lindorff-Larsen et al. "How Fast-Folding Proteins Fold". In: *Science* 334.6055 (Oct. 2011), pp. 517–520. DOI: 10.1126/science.1208351. URL: https://doi.org/10.1126/science.1208351.

[20] Marloes Arts et al. *Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics*. 2023. eprint: arXiv:2302.00600.

[21] Hao Wu and Frank Noé. "Variational Approach for Learning Markov Processes from Time Series Data". In: *Journal of Nonlinear Science* 30.1 (2019), pp. 23–66. DOI: 10.1007/s00332-019-09567-y. URL: https://doi.org/10.1007%2Fs00332-019-09567-y.

[22] Guillermo Pérez-Hernández et al. "Identification of slow molecular order parameters for Markov model construction". In: *The Journal of Chemical Physics* 139.1 (July 2013), p. 015102. DOI: 10.1063/1.4811489. URL: https://doi.org/10.1063/1.4811489.

[23] Natasa Djurdjevac, Marco Sarich, and Christof Schütte. "Estimating the Eigenvalue Error of Markov State Models". In: *Multiscale Modeling & Simulation* 10.1 (Jan. 2012), pp. 61–81. DOI: 10.1137/100798910. URL: https://doi.org/10.1137/100798910.

[24] Marco Sarich, Frank Noé, and Christof Schütte. "On the Approximation Quality of Markov State Models". In: *Multiscale Modeling & Simulation* 8.4 (Jan. 2010), pp. 1154–1177. DOI: 10.1137/090764049. URL: https://doi.org/10.1137/090764049.

[25] Andreas Krämer et al. "Statistically Optimal Force Aggregation for Coarse-Graining Molecular Dynamics". In: *The Journal of Physical Chemistry Letters* 14.17 (Apr. 2023), pp. 3970–3979. DOI: 10.1021/acs.jpclett.3c00444. URL: https://doi.org/10.1021/acs.jpclett.3c00444.

[26] Wangfei Yang et al. "Slicing and Dicing: Optimal Coarse-Grained Representation to Preserve Molecular Kinetics". In: *ACS Central Science* 9.2 (Jan. 2023), pp. 186–196. DOI: 10.1021/acscentsci.2c01200. URL: https://doi.org/10.1021/acscentsci.2c01200.

[27] Frank Noé et al. "Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations". In: *Proceedings of the National Academy of Sciences* 106.45 (Nov. 2009), pp. 19011–19016. DOI: 10.1073/pnas.0905466106. URL: https://doi.org/10.1073/pnas.0905466106.

[28] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: 32 (2019). Ed. by H Wallach et al. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[29] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. Mar. 2019. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

[30] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: http://github.com/google/jax.

[31] Cheng Lu et al. "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps". In: *arXiv preprint arXiv:2206.00927* (2022).

[32] Jes Frellsen et al. "Bayesian Generalised Ensemble Markov Chain Monte Carlo". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, 2016, pp. 408–416. URL: http://proceedings.mlr.press/v51/frellsen16.html.

[33] Alessandro Laio and Michele Parrinello. "Escaping free-energy minima". In: *Proceedings of the National Academy of Sciences* 99.20 (2002), pp. 12562–12566. ISSN: 0027-8424. DOI: 10.1073/pnas.202427399. eprint: https://www.pnas.org/content/99/20/12562.full.pdf. URL: https://www.pnas.org/content/99/20/12562.

[34] D. P. Landau, Shan-Ho Tsai, and M. Exler. "A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling". In: *American Journal of Physics* 72.10 (2004), pp. 1294–1302. DOI: 10.1119/1.1707017. eprint: https://doi.org/10.1119/1.1707017. URL: https://doi.org/10.1119/1.1707017.

[35] Yihang Wang, Lukas Herron, and Pratyush Tiwary. "From data to noise to data for mixing physics across temperatures with generative artificial intelligence". In: *Proceedings of the National Academy of Sciences* 119.32 (Aug. 2022). DOI: 10.1073/pnas.2203656119. URL: https://doi.org/10.1073/pnas.2203656119.

[36] Frank Noé et al. "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning". In: *Science* 365.6457 (Sept. 2019). DOI: 10.1126/science.aaw1147. URL: https://doi.org/10.1126/science.aaw1147.

[37] Bowen Jing et al. *Torsional Diffusion for Molecular Conformer Generation*. 2022. eprint: arXiv:2206.01729.

[38] Elman Mansimov et al. "Molecular geometry prediction using a deep generative graph neural network". In: *Scientific Reports* 9.1 (2019), pp. 1–13.

[39] Chence Shi et al. *Learning Gradient Fields for Molecular Conformation Generation*. 2021. eprint: arXiv:2105.03902. URL: http://arxiv.org/abs/2105.03902.

[40] Gregor Simm and Jose Miguel Hernandez-Lobato. "A Generative Model for Molecular Distance Geometry". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 8949–8958. URL: http://proceedings.mlr.press/v119/simm20a.html.

[41] Chence Shi et al. "Learning Gradient Fields for Molecular Conformation Generation". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 9558–9568. URL: https://proceedings.mlr.press/v139/shi21b.html.

[42] Tarun Gogineni et al. "TorsionNet: A Reinforcement Learning Approach to Sequential Conformer Search". In: *CoRR* abs/2006.07078 (2020). arXiv: 2006.07078. URL: https://arxiv.org/abs/2006.07078.

[43] Robin Winter, Frank Noé, and Djork-Arné Clevert. *Auto-Encoding Molecular Conformations*. 2021. URL: http://arxiv.org/abs/2101.01618.

[44] Victor Garcia Satorras et al. *E(n) Equivariant Normalizing Flows for Molecule Generation in 3D*. 2021. eprint: 2105.09016. URL: http://arxiv.org/abs/2105.09016.

[45] Niklas Gebauer, Michael Gastegger, and Kristof Schütt. "Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/a4d8e2a7e0d0c102339f97716d2fdfb6-Paper.pdf.

[46] Minkai Xu et al. *An End-to-End Framework for Molecular Conformation Generation via Bilevel Programming*. 2021. eprint: arXiv:2105.07246.

[47] Sereina Riniker and Gregory A. Landrum. "Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation". In: *Journal of Chemical Information and Modeling* 55.12 (Nov. 2015), pp. 2562–2574. DOI: 10.1021/acs.jcim.5b00654. URL: https://doi.org/10.1021/acs.jcim.5b00654.

[48] Paul C. D. Hawkins et al. "Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and Cambridge Structural Database". In: *Journal of Chemical Information and Modeling* 50.4 (Mar. 2010), pp. 572–584. DOI: 10.1021/ci100031x. URL: https://doi.org/10.1021/ci100031x.

[49] Paulo C. T. Souza et al. "Martini 3: a general purpose force field for coarse-grained molecular dynamics". In: *Nature Methods* 18.4 (Mar. 2021), pp. 382–388. DOI: 10.1038/s41592-021-01098-3. URL: https://doi.org/10.1038/s41592-021-01098-3.

[50] Jiang Wang et al. "Machine Learning of Coarse-Grained Molecular Dynamics Force Fields". In: *ACS Central Science* 5.5 (Apr. 2019), pp. 755–767. DOI: 10.1021/acscentsci.8b00913. URL: https://doi.org/10.1021/acscentsci.8b00913.

[51] Brooke E. Husic et al. "Coarse graining molecular dynamics with graph neural networks". In: *The Journal of Chemical Physics* 153.19 (Nov. 2020), p. 194101. DOI: 10.1063/5.0026133. URL: https://doi.org/10.1063/5.0026133.

[52] Jonas Köhler et al. "Flow-Matching: Efficient Coarse-Graining of Molecular Dynamics without Forces". In: *Journal of Chemical Theory and Computation* 19.3 (Jan. 2023), pp. 942–952. DOI: 10.1021/acs.jctc.3c00016. URL: https://doi.org/10.1021/acs.jctc.3c00016.

[53] Soojung Yang and Rafael Gómez-Bombarelli. *Chemically Transferable Generative Backmapping of Coarse-Grained Proteins*. 2023. eprint: arXiv:2303.01569.

[54] Yaxin An and Sanket A. Deshmukh. "Machine learning approach for accurate backmapping of coarse-grained models to all-atom models". In: *Chemical Communications* 56.65 (2020), pp. 9312–9315. DOI: 10.1039/d0cc02651d. URL: https://doi.org/10.1039/d0cc02651d.

[55] Leandro E. Lombardi, Marcelo A. Martí, and Luciana Capece. "CG2AA: backmapping protein coarse-grained structures". In: *Bioinformatics* 32.8 (Dec. 2015), pp. 1235–1237. DOI: 10.1093/bioinformatics/btv740. URL: https://doi.org/10.1093/bioinformatics/btv740.

[56] Christof Schütte et al. *A Hybrid Monte Carlo Method for Essential Molecular Dynamics*. eng. Tech. rep. SC-98-04. Takustr. 7, 14195 Berlin: ZIB, 1998.

[57] William C. Swope, Jed W. Pitera, and Frank Suits. "Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory". In: *The Journal of Physical Chemistry B* 108.21 (Apr. 2004), pp. 6571–6581. DOI: 10.1021/jp037421y. URL: https://doi.org/10.1021/jp037421y.

[58] Brooke E. Husic and Vijay S. Pande. "Markov State Models: From an Art to a Science". In: *Journal of the American Chemical Society* 140.7 (Feb. 2018), pp. 2386–2396. DOI: 10.1021/jacs.7b12191. URL: https://doi.org/10.1021/jacs.7b12191.

[59] Simon Olsson and Frank Noé. "Dynamic graphical models of molecular kinetics". In: *Proceedings of the National Academy of Sciences* 116.30 (July 2019), pp. 15001–15006. DOI: 10.1073/pnas.1901692116. URL: https://doi.org/10.1073/pnas.1901692116.

[60] Andreas Mardt et al. "VAMPnets for deep learning of molecular kinetics". In: *Nature Communications* 9.1 (2018). DOI: 10.1038/s41467-017-02388-1. URL: https://doi.org/10.1038%2Fs41467-017-02388-1.

[61] Hao Wu, Jan-Hendrik Prinz, and Frank Noé. "Projected metastable Markov processes and their estimation with observable operator models". In: *The Journal of Chemical Physics* 143.14 (Oct. 2015), p. 144101. DOI: 10.1063/1.4932406. URL: https://doi.org/10.1063/1.4932406.

[62] Hao Wu et al. *Deep Generative Markov State Models*. 2018. eprint: arXiv:1805.07601.

[63] Leon Klein et al. *Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics*. 2023. eprint: arXiv:2302.01170.

[64] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. DOI: 10.1093/biomet/57.1.97. URL: https://doi.org/10.1093/biomet/57.1.97.

[65] S. Kieninger and B. G. Keller. "Path probability ratios for Langevin dynamics—Exact and approximate". In: *The Journal of Chemical Physics* 154.9 (Mar. 2021), p. 094102. DOI: 10.1063/5.0038408. URL: https://doi.org/10.1063/5.0038408.

[66] Manuel Dibak et al. "Temperature steerable flows and Boltzmann generators". In: *Physical Review Research* 4.4 (Oct. 2022). DOI: 10.1103/physrevresearch.4.l042005. URL: https://doi.org/10.1103/physrevresearch.4.l042005.

[67] V Rokhlin. “Rapid solution of integral equations of classical potential theory”. In: *Journal of Computational Physics* 60.2 (Sept. 1985), pp. 187–207. DOI: 10.1016/0021-9991(85)90002-6. URL: https://doi.org/10.1016/0021-9991(85)90002-6.

[68] P. P. Ewald. “Die Berechnung optischer und elektrostatischer Gitterpotentiale”. In: *Annalen der Physik* 369.3 (1921), pp. 253–287. DOI: 10.1002/andp.19213690304. URL: https://doi.org/10.1002/andp.19213690304.

[69] Arthur Kosmala et al. *Ewald-based Long-Range Message Passing for Molecular Graphs*. 2023. eprint: arXiv:2303.04791.

[70] Benjamin Trendelkamp-Schroer et al. “Estimation and uncertainty of reversible Markov models”. In: *The Journal of Chemical Physics* 143.17 (Nov. 2015), p. 174101. DOI: 10.1063/1.4934536. URL: https://doi.org/10.1063/1.4934536.

[71] Susanna Röblitz and Marcus Weber. “Fuzzy spectral clustering by PCCA+ application to Markov state models and data classification”. In: *Advances in Data Analysis and Classification* 7.2 (May 2013), pp. 147–179. DOI: 10.1007/s11634-013-0134-6. URL: https://doi.org/10.1007/s11634-013-0134-6.

[72] Igor Mezić. “Spectral Properties of Dynamical Systems, Model Reduction and Decompositions”. In: *Nonlinear Dynamics* 41.1-3 (2005), pp. 309–325. DOI: 10.1007/s11071-005-2824-x. URL: https://doi.org/10.1007%2Fs11071-005-2824-x.

[73] Moritz Hoffmann et al. “Deeptime: a Python library for machine learning dynamical models from time series data”. In: *Machine Learning: Science and Technology* 3.1 (2021), p. 015009. DOI: 10.1088/2632-2153/ac3de0. URL: https://doi.org/10.1088%2F2632-2153%2Fac3de0.

A Properties of the Transfer operator

A.1 Relaxation of T_Ω spectrum

In this section, we outline the 'relaxation' or 'decay' of the spectral components of T_Ω as a function of time-step, τ. We use that $\langle\phi_i|\psi_i\rangle_\mu = \int \phi_i(x)\psi_i(x)\,\mathrm{d}\mu(x) = 1$ if $i = j$ and 0 if $i \neq j$, e.g. the eigenfunctions are orthonormal under the μ-weighed inner-product. Since $T_\Omega(\tau)$ is Markov, composing $T_\Omega(\tau)$ with itself N times we get,

$$
\begin{aligned}
[T_\Omega(\tau)]^N &= T_\Omega(\tau) \circ \cdots \circ T_\Omega(\tau) && (12)\\
&= \sum_{i=0}^{\infty} \lambda_i(\tau)|\psi_i\rangle\langle\phi_i|\lambda_i(\tau)|\psi_i\rangle\langle\phi_i| \ldots \lambda_i(\tau)|\psi_i\rangle\langle\phi_i| && (13)\\
&= \sum_{i=0}^{\infty} \lambda_i(\tau)^N|\psi_i\rangle\langle\phi_i|\psi_i\rangle_\mu\langle\phi_i| \ldots |\psi_i\rangle\langle\phi_i| && (14)\\
&= \sum_{i=0}^{\infty} \lambda_i(\tau)^N|\psi_i\rangle\langle\phi_i| && (15)
\end{aligned}
$$

We assume the dynamics governed by T_Ω are

1. reversible $\lambda_i \in \mathbb{R}$
2. measure-preserving $0 \leq |\lambda_i| \leq 1$
3. ergodic, $\lambda_0 = 1$ and $|\lambda_{i>0}| < 1$

where we have sorted the eigenvalues eigenfunction pairs in descending order. Consequently, for $N \to \infty$ we have $T_\Omega(N\tau) \to |\mathbb{1}\rangle\langle\mu|$, where $\mathbb{1}$ is the constant function.

A.2 Decomposition of transition density

In this section, we detail the decomposition of the transition density, $p(\mathbf{x}_{N\tau} \mid \mathbf{x}_0)$.

Let ρ specify an initial condition, an absolutely convergent probability density function on Ω. We can define a Transfer operator T_Ω using a transition probability density [6]:

$$
[T_\Omega \circ \rho]\,(\mathbf{x}_{N\tau}) \triangleq \frac{1}{\mu(\mathbf{x}_{N\tau})} \int_{\mathbf{x}_0} \mu(\mathbf{x}_0)\rho(\mathbf{x}_0)p(\mathbf{x}_{N\tau} \mid \mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0, \qquad T_\Omega : L^1(\Omega) \to L^1(\Omega) \tag{16}
$$

which then describes the μ-weighed evolution of densities on Ω according to MD discretized in time by a step-size of τ. μ is a normalized Gibbs measure, or the Boltzmann distribution.

Since we only consider MD with time-invariant drift, only the eigenvalues $\lambda_i(\tau)$ of $T_\Omega(\tau)$ depend on τ. We can express arbitrary transition probabilities through a bilinear form

$$
p(\mathbf{x}_{N\tau} \mid \mathbf{x}_0) = \langle\delta_{\mathbf{x}_{N\tau}}|T_\Omega^N(\tau)|\delta_{\mathbf{x}_0}\rangle = \sum_{i=1}^{\infty} \lambda_i^N(\tau)\langle\delta_{\mathbf{x}_{N\tau}}|\phi_i\rangle\langle\psi_i|\delta_{\mathbf{x}_0}\rangle = \sum_{i=1}^{\infty} \lambda_i^N(\tau)\alpha_i(\mathbf{x}_{N\tau})\beta_i(\mathbf{x}_0) \tag{17}
$$

where α_i and β_i are *time-invariant* projections coefficients of the state variables on-to the eigenfunctions ϕ_i and ψ_i, and $\delta_{\mathbf{x}}$ is the Dirac delta centered at $\mathbf{x}$. $T_\Omega^N(\tau)$ means $T_\Omega(\tau)$ acting N times (See A.1).

B Datasets

Throughout we train on all available data, as it is often sparse and difficult to split in an appropriate manner due to rare events e.g. folding and unfolding.

Table 3: Details about the Alanine dipeptide data (taken verbatim from mdshare)

Property	Value
Code	ACEMD
Forcefield	AMBER ff-99SB-ILDN
Integrator	Langevin
Integrator time step	2 fs
Simulation time	250 ns
Frame spacing	1 ps
Temperature	300 K
Volume	$(2.3222nm)^3$ periodic box
Solvation	651 TIP3P waters
Electrostatics	PME
PME real-space cutoff	0.9 nm
PME grid spacing	0.1 nm
PME updates	every two time steps
Constraints	all bonds between hydrogens and heavy atoms

B.1 Müller Brown

We generate the Müller Brown data set used for training by integrating the 2D potential energy model:

$$U(x,y) = \sum_{i=1}^{4} A_i \exp\left[a_i(x-\bar{x}_i)^2 + b_i(x-\bar{x}_i)(y-\bar{y}_i) + c_i(y-\bar{y}_i)^2\right] \tag{18}$$

using simulating overdamped Langevin or Brownian dynamics SDE, through a Euler-Mayurama time-discretization, and where

$$\begin{aligned} A &= (-200, -100, -170, 15) \\ a &= (-1, -1, -6.5, 0.7) \\ b &= (0, 0, 11, 0.6) \\ c &= (-10, -10, -6.5, 0.7) \\ \bar{x} &= (1, 0, -0.5, -1) \\ \bar{y} &= (0, 0.5, 1.5, 1). \end{aligned} \tag{19}$$

We we generate 32 trajectories with random initial conditions in the ranges

$$\begin{aligned} x &= [-1.5, 1.2] \\ y &= [-0.2, 2.0], \end{aligned} \tag{20}$$

and save every 10th step after a burn-in of 1000 steps. Each trajectory is simulated for 100000 steps.

A separate testing set was generated in an identical manner but with a different random seed. The values in Table. 1 are computed compared to this test set.

B.2 Alanine dipeptide

We use the data from MDShare (Table 3) which consists of three independent trajectories of 250 ns each.

Pre-processing The atomic coordinates are standardized before model training, each atom has a unique nominal embedding as atom type.

B.3 Fast folding proteins

The original data was obtained upon request from DE Shaw Research, and details about the simulations are available in the original publication [19]. All configurations were preprocessed by centering them at the origin. Furthermore, all configurations were scaled to ensure a standard deviation of one across the dataset.

Table 4: MSM hyperparameters. All models used 100 cluster centers, and clustered in the 5 first TICs.

	TICA lag	MSM lag	ITO Δt
Chignolin	1ns	100ns	200ns
Trp-Cage	1ns	100ns	200ns
BBA	1ns	800ns	200ns
Villin	1ns	200ns	200ns

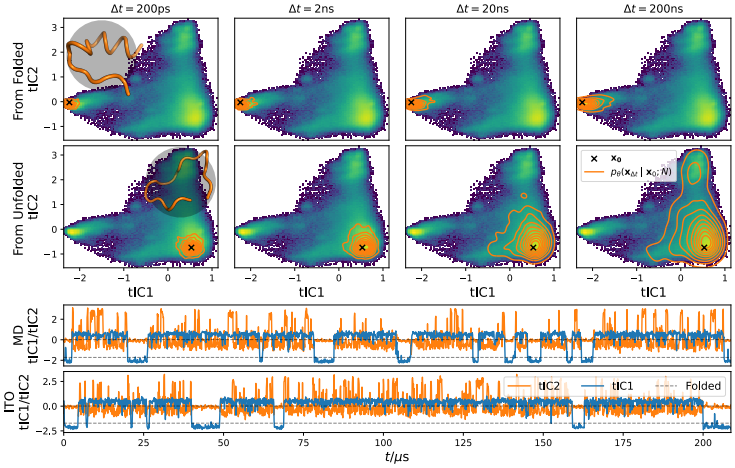


Figure 6: **Reversible protein folding-unfolding of Trp-Cage with CG-SE3-ITO** Conditional probability densities (orange contours) starting from folded (upper panels) and unfolded (lower panels) protein states, at increasing time-lag (left to right), shown on top of data distribution. Below: time-traces of 208 microsecond MD simulations and ITO simulations on tICs 1 and 2.

C Additional results

C.1 Fast folding proteins

Figures 6, 7 and 8, show conditional distributions generated by CG-SE3-ITO models and comparisons of MD with ITO simulations on the fast folders Trp-Cage, BBA, and Villin, respectively.

Reference value and observables We compute observables using Markov state models. First, we estimate a reference model for each system (see hyper-parameters in Table 4). Briefly, non-redundant and non-trivial pair-wise Cα distances were used as input for TICA dimension reduction, the reduced space was clustered using k-means. MSMs were sampled from a Bayesian posterior as previously described [70], using cluster assignments as state assignments. We identified folded and unfolded states using PCCA (Perron Cluster-Cluster analysis) [71], which in turn enabled the calculation of mean first passage times (MFPT) of folding $\langle \tau_f \rangle$ and unfolding $\langle \tau_u \rangle$ and the free energy of folding ΔG_{fold}.

Observables computed from ITO simulations were computed by processing the simulation data by projecting them onto the TICA space and the cluster centers determined on the MD data. MSMs were sampled as for MD data and observables were computed in the same way.

The reported uncertainties are standard deviations from Bayesian posterior sampling.

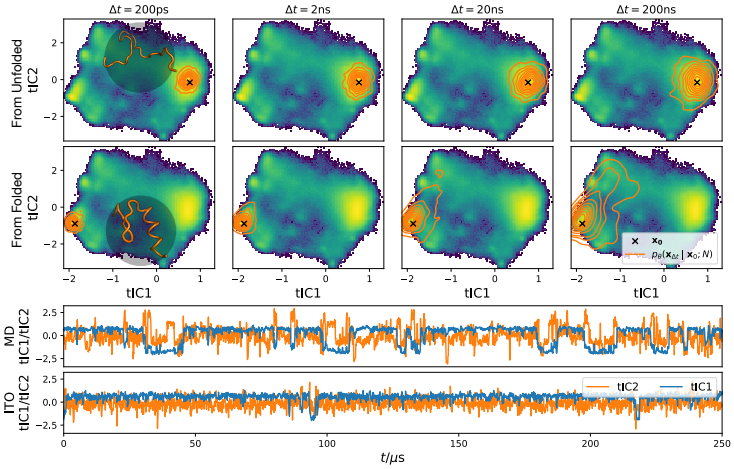


Figure 7: **Reversible protein folding-unfolding of BBA with CG-SE3-ITO** Conditional probability densities (orange contours) starting from unfolded (upper panels) and folded (lower panels) protein states, at increasing time-lag (left to right), shown on top of data distribution. Below: time-traces of 250 microsecond MD simulations and ITO simulations on tICs 1 and 2.

C.2 Sample timings

Running on a single device of a NVIDIA TITAN V node, using all memory, we can concurrently generate

- 253 simulation-steps/s for Chignolin
- 61 simulation-steps/s for Trp-Cage
- 35 simulation-steps/s for BBA
- 21 simulation-steps/s for Villin
- 48 simulation-steps/s for Alanine-Dipeptide

Note that all samples presented in this paper have been calculated equivalently using 50 ODE-steps. Depending on simulated lag, arbitrarily long trajectories can be sampled efficiently. Our models were trained on lags of up to 200 ns, but our findings suggest no constraints on extending the framework to much longer time scale.

D Architectural details

Positional embedding, Λ_{pos} , maps diffusion time t_{diff}, physical time Δt, and interatomic distances r_{ij} to n-dimensional features-vectors with the n'th dimension defined as:

$$\Lambda^n_{\text{pos}}(x) = \begin{cases} \cos\left(\left(1+\frac{n}{2}\right) x \frac{\pi}{l_0}\right) & \text{for even } n \\ \sin\left(\left(1+\frac{n-1}{2}\right) x \frac{\pi}{l_0}\right) & \text{for odd } n, \end{cases} \tag{21}$$

where l_0 is a hyperparameter.

Nominal embedding Λ_{nom} , maps atomic elements or residue types to continuous n-dimensional feature vectors, $f : C \to R^n$, where C is the set of all categorical values and n is the dimension of the embedded vector.

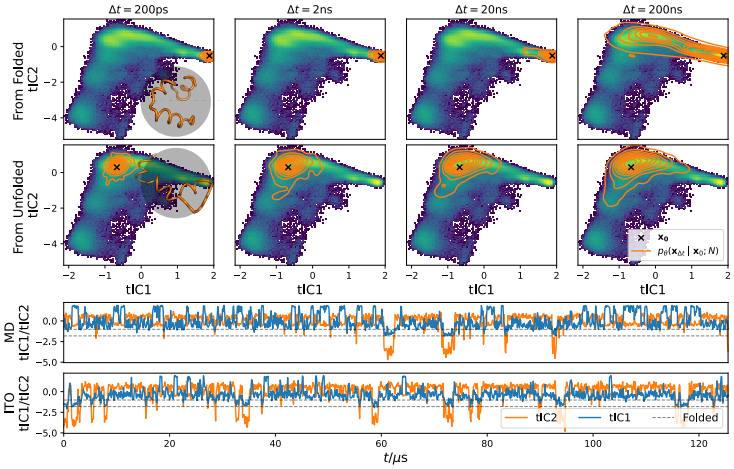


Figure 8: **Reversible protein folding-unfolding of Villin with CG-SE3-ITO** Conditional probability densities (orange contours) starting from folded (upper panels) and unfolded (lower panels) protein states, at increasing time-lag (left to right), shown on top of data distribution. Below: time-traces of 125 microsecond MD simulations and ITO simulations on tICs 1 and 2.

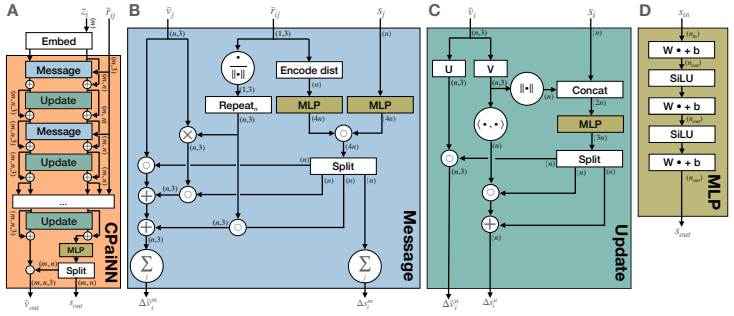


Figure 9: **ChiroPaiNN architecture** utilized in SE3-ITO and CG-SE3-ITO models (Fig. 2) for the embedding of conditional configuration and score prediction. Arrows are annotated with input and output shapes. $\times$ indicates cross product operations between all vectors along the first dimension, and $\circ$ indicates element-wise multiplication along the first dimension.

E Training details

E.1 Sampling of configurations

The last $N_{\max}$ frames were truncated from each trajectory such that $\mathbf{x}_t$ could be sampled uniformly while keeping $\mathbf{x}_{t+N_{\max}}$ in bounds. N is sampled discretely from DisExp($N_{\max}$) following;

Algorithm 3 Sampling from DisExp

$N_{\log} \sim \text{Uniform}(0, \log(N_{\max}))$
Return: $\text{floor}(\exp(N_{\log}))$

Algorithm 4 Sampling from $\hat{p}_{\boldsymbol{\theta}}(\mathbf{x}_0, N)$

Input: initial condition $\mathbf{x}_0$, lag; N, diffusion steps; T_{diff}, ITO score-model; $\hat{\epsilon}_\theta$
$\mathbf{x}_N^{T_{\text{diff}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
for $t_{\text{diff}} = T_{\text{diff}} \dots 1$ **do**
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
  $\mathbf{x}_N^{t_{\text{diff}}-1} = \frac{1}{\sqrt{\alpha^{t_{\text{diff}}}}} \left(\mathbf{x}_N^{t_{\text{diff}}} - \frac{1-\alpha^{t_{\text{diff}}}}{\sqrt{1-\bar{\alpha}^{t_{\text{diff}}}}} \hat{\epsilon}_\theta(\mathbf{x}_N^{t_{\text{diff}}}, \mathbf{x}_0, N, t_{\text{diff}}) \right) + \sigma_t \epsilon$
end for
return $\mathbf{x}_N^0$

E.2 Data splits

All available data was used for training with no test/validation set. Reference MFPT values are already coarse estimates and cannot be accurately calculated from a subset of the data due to slow time scales compared to the length of available trajectories.

E.3 Hyper Parameters

Müller-Brown For the Müller-Brown results we trained with the MLP in MB-ITO architecture with 32 dimensional positional embeddings for t_{phys} and N and the MLP had 32 hidden nodes and 5 layers. We used a cosine learning rate scheduler and a sigmoidal β-scheduler with parameters as listed for alanine dipeptide and the fast folders. The model with stochastic lag was trained with $N_{\max} = 1000$ and for fixed lag models N was fixed during data generation and the positional embeddings of N were removed from the model.

Alanine dipeptide and Fast folders Hyperparameters employed for experiments on the fast folding proteins and Alanine Dipeptide are outlined below:

```
n_features: 64
n_message_passing_blocks_cpainn_embed: 2
n_message_passing_blocks_cpainn_score: 5

N_max: 1000
length_scale: 3.
beta_scheduler: sigmoidal(-8,-4)
diffusion_steps: 1000

batch_size: 128
learning_rate: 1e-3
optimizer: Adam
```

`n_message_passing_blocks_cpainn_{embed/score}` refers to the number of message passing and update blocks in the CPaiNN networks shown in Figure 2. A *message passing block* refers to a message block followed by an update block as shown in Figure 9. Where `sigmoidal(t_0,T)` $= \frac{1}{1+e^{-x}}\big|_{x\in(t_0,T)}$. `n_features` and `batch_size` corresponds to n and m in Figure 9. `length_scale` correspond to the value of l_0 in (Eq. 21) and defines the radial resolution of the embedding. `n_features` was chosen such that equivalent models could fit in memory of available hardware while maintaining a consistent `batch_size` across all systems. The remaining hyperparameters were fixed and were not systematically optimized.

E.4 Bond lengths Alanine Dipeptide

We evaluate how well the fast vibrational degrees of freedom are captured by the SE3-ITO model on Alanine dipeptide by inspecting the bondlength distributions of model samples (Fig. 10). The variances are generally over estimated slightly, but it does not appear to significantly our ability to predict slow dynamics. However, it would impact importance sampling as many configurations would have unfavorable physical energies. We leave it for future work to improve.

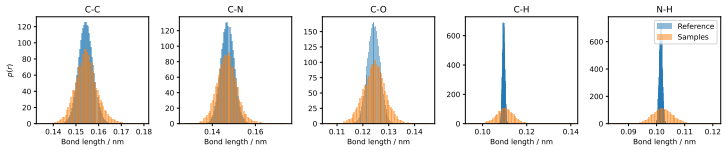


Figure 10: Bond lengths of samples Alanine Dipeptide

F Compute resources

F.1 Training

All reported experiments have been conducted on NVIDIA TITAN V, NVIDIA TITAN X (Pascal), and NVIDIA GeForce GTX TITAN X's. All GPUs have $\sim$ 12GB memory and range from 3000-5000 CUDA cores. Given the hyperparameters specified above, the SE3-ITO models converge within 2-4 days of training depending on system size.

Throughout the project, $\sim$ 250 models were trained for an average duration of $\sim$ 12 hours pr model on single GPU devices, resulting in a total of $\sim$ 3000 GPU hours spent on training.

F.2 Sampling

In total 589 GPU hours have been spent on sampling throughout the entire project.

G Variational Approach to Markov Processes (VAMP)

The Variational Approach to Markov Processes (VAMP) is a recent result in non-linear dynamics theory, its key contribution is a family of VAMP-scores [21]. The VAMP-scores are devised based upon the insight that the best (smallest prediction error) linear model can be expressed in terms of the top singular components of the Koopman operator, $\mathcal{K}$ [72]. The scores measure sum of the singular values of $\mathcal{K}$ multiplied by overlap coefficients between a set of (ortho-normalized) feature-maps f and g and the singular components of $\mathcal{K}$. We can optimize VAMP-scores to learn optimal feature mappings and Markovian models of the dynamics from time-series data [60] or for model comparison [21]. We here use the VAMP-score for the latter and assume $f = g$.

VAMP-r score is computed via the singular values of the Koopman matrix $\mathbf{K}$ estimated from data using the feature maps f and g [73],

$$\mathrm{VAMP}{-}r = \sum_{i=0}^{k} \sigma_i^r \tag{22}$$

where $r \in \mathbb{N}_+$.

G.1 VAMP gap

Informally, the VAMP-r scores quantify the meta-stability of a Koopman matrix. We define the VAMP-gap ΔV between two Koopman matrices, $\mathbf{K}$ and $\mathbf{K}'$, as the difference between their VAMP-2 scores:

$$\Delta V = \mathrm{VAMP}{-}2(\mathbf{K}) - \mathrm{VAMP}{-}2(\mathbf{K}'), \tag{23}$$

where $\mathbf{K}'$ is a reference and $\mathbf{K}$ is a query matrix, respectively. In this context, $\Delta V = 0$ means meta-stability in $\mathbf{K}$ and $\mathbf{K}'$ is indistinguishable, $\Delta V < 0$ means $\mathbf{K}$ underestimates meta-stability, and *vice versa* for $\Delta V > 0$.

APPENDIX E

Mors Vejledning Til Min Afhandling

(Selvfølgelig også til andre interesserede - tak for at læse med)

I begyndelsen af det 20. århundrede bevidnede vi, gennem fødslen af kvantemekanik, en banebrydende udvikling i vores forståelse af fysikken og den mikroskopiske verden, der omgiver os. Med kvantemekanikken kunne vi svare på fundamentale spørgsmål, omdiskuteret i tusindvis af år, om hvordan atomer og molekyler fungerer. Det er yderst vigtigt at have en dybdegående forståelse af molekyler da de styrer alt fra biologiske processer til egenskaberne af de materialer, vi bygger verden omkring os med. Selvom kvantemekanikken rent matematisk er blevet beskrevet fyldestgørende, har det vist sig at være en formidabel udfordring at anvende den på molekyler. Intensiv forskning har gennem mange år resulteret i en række sofistikerede værktøjer, kendt som elektronstruktur metoder, der kan give visse indsigter i molekylers kvantemekaniske egenskaber. Elektronstruktur metoder er, ikke overraskende, meget beregningstunge, hvilket forhindrer dem i at blive brugt på en stor skala. Udfordringerne forbundet med at bruge disse metoder udgør en flaskehals for udviklingen af nye teknologier og metoder, der potentielt kunne have afgørende indflydelse på mange af verdens problemer.

I de senere år har vi været vidne til en ny revolution i form af kunstig intelligens, der måske er lige så vigtig som kvantemekanikkens revolution. Neurale netværk, der er en specifik form for kunstig intelligens model, har vist sig at være i stand til at løse problemer så komplekse at det virkede utænkeligt for bare få år siden at det skulle kunne lade sig gøre. Neurale netværk kan generere realistiske og kreative billeder og have uhyggeligt menneskelige samtaler. Det er nemt at vurdere hvor langt vi er nået med kunstig intelligens i disse 'menneskelige regi', men faktisk er der ikke meget forskel, set fra et neuralt netværks perspektiv, på at bestemme et molekyles elektroniske struktur eller generere et sjovt billede af hunde der fester på skateboards. Desuden opererer neurale netværk mange gange hurtigere end de omtalte elektronstruktur metoder, og det er en oplagt mulighed at integrere dem i kvantekemi for at løse udfordringer forbundet med elektronstruktur beregninger. I min afhandling udforsker jeg nogle af de muligheder, der åbner sig op, når vi forener de to felter.

Molekyler er små systemer af atomer forbundet i en specifik konfiguration af kemiske bånd. Grundlæggende kan en kemisk reaktion forstås som en omorganisering af de bånd, der forbinder atomerne i et eller flere molekyler. På sin vis er dette meget lig at skille en LEGO-figur ad, og samle klodserne igen på en anderledes måde. Klodserne svarer til atomer, og figurerne svarer til molekyler. Der er dog en væsentlig ekstra udfordring i atom-LEGO, da klodserne her agerer som små magneter, der er forbundet med fjedre på kryds og tværs, og man kan ikke skille den ene figur helt ad, før man samler den anden. En kemisk reaktion kan beskrives ved tre vigtige molekylære konfigurationer - reaktanten, overgangstilstanden, og produktet. Reaktanten beskriver, hvilke atomer der er forbundet, via kemiske bånd før reaktionen, og tilsvarende beskriver produktet, hvilke atomer der er forbundet efter. Overgangstilstanden beskriver den konfiguration af molekylet, der udgør 'vendepunktet' i reaktionen. Tilsvarende det øjeblik, hvor en bold der skubbes op ad en bakke når topper og pludselig falder ned på den anden side af sig selv. Overgangstilstande er særligt svære at beskrive kvantemekanisk, fordi de på samme tid involverer brud på gamle bånd og dannelsen af nye. I LEGO analogien kan overgangstilstande på sin vis ses som en kritisk figur vi skal have bygget undervejs for nemmest at komme fra den gamle til den nye LEGO figur. Man kan udlede meget vigtig information om en reaktion ved at kende til dens overgangstilstand. For eksempel, hvor hurtigt den vil foregå, eller hvordan den bliver påvirket af temperatur. Gennem en dyb forståelse af overgangstilstande for kemiske reaktioner kan vi designe procedurer for, hvordan vi kan syntetisere molekyler og materialer med specifikke egenskaber i laboratorier i den virkelige verden. Dette kunne for eksempel være nye former for medicin, nedbrydelige alternativer til plastic, eller nye måder at lagre energi på i batterier. Kun fantasien sætter grænser. Problemet er, at overgangstilstande er ekstremt svære at finde, da det kræver at vi udregner komplekse bevægelser for mange atomer på en gang, der alle påvirker hinanden på kryds og tværs. Som regel betyder det at vi er nødt til at lave tusindvis af tunge elektronstruktur beregninger for at finde overgangstilstanden for bare en enkelt reaktion. Dette betyder, at hvis vi for eksempel har flere ideer til, hvordan en reaktion kan foregå, kan det være udfordrende at prøve dem alle af.

E.1 Transition1x

For at bruge et neuralt netværk skal vi igennem en såkaldt træningsfase. I denne fase har vi brug for adskillige eksempler på korrekte elektronstruktur beregninger som vi kan vise modellen. I løbet af trænings fasen lærer modellen de grundlæggende mønstre for, hvordan man laver disse beregninger, ved at se utallige eksempler. Samlingen af vores træningseksempler udgør det vi kalder vores træningssæt. Hvis vi har mange eksempler kan vi lægge nogle af dem til side under træningsfasen, så modellen aldrig får lov at se dem. Disse eksempler kan så bruges, når træningsfasen er overstået, til en slags afsluttende eksamen for modellen. Udfra hvor godt modellen klarer sin eksamen får vi en ide om, hvor meget den har lært under træningsfasen og i hvilken udstrækn-

ing vi kan stole på dens beregninger, når den skal lave elektronstruktur beregninger senere. Modeller som ChatGPT og de populære billedgenereringsmodeller skylder en stor del af deres succes til de enorme mængder af billede- og tekst data der ligger frit tilgængeligt på internettet. Træningsdata er lige så vigtige for neurale netværk der anvendes i kvantekemi som modeller der virker på tekst og billeder. I den første artikel i min Ph.D. udgav jeg et stort dataset som jeg kaldte Transition1x. Dette datasæt indeholder elektronstruktur beregninger for 10 millioner molekyler, der er i færd med at indgå i kemiske reaktioner, på forskellige tidspunkter i processen. For at samle data til Transition1x fandt jeg et sæt af 10.000 forskellige kemiske reaktioner. Efterfølgende brugte jeg tunge elektronstruktur metoder til at beregne overgangstilstande for alle reaktionerne, og opsamlede alle udregninger der blev lavet undervejs. Disse beregninger blev udført på DTUs supercomputer og tog flere uger. Hvis jeg skulle have gjort det på min egen bærbare ville det have taget ca. 25 år. Indtil nu har der ikke været meget data tilgængeligt for kemiske reaktioner i litteraturen, og derfor har det været vanskeligt at anvende neurale netværk til forskning i den retning. Forhåbentligt vil mit datasæt vise sig være et værdifuldt bidrag til feltet, der åbner op for nye muligheder og metoder for at bruge neurale netværk til at finde overgangstilstande og studere kemiske reaktioner.

E.2 NeuralNEB

I det næste projekt ville jeg forsøge at bruge det datasæt, som jeg havde skabt for at se, om det faktisk var muligt at bruge neurale netværk til at finde overgangstilstande for kemiske reaktioner. Jeg kaldte min procedure for NeuralNEB da den er baseret på neurale netværk og en metode fra kvantekemi der hedder Nudged Elastic Band (NEB). Forestil dig et ujævnt landskab med høje bakker og dybe dale. Bakker i dette landskab svarer til molekylære konfigurationer med høj energi, og tilsvarende svarer dale til konfigurationer med lav energi. Vi kalder dette landskab en energi-overflade. Et molekyle i bunden af en dal er stabilt. Det er det fordi at det kræver en væsentlig mængde energi at 'sparke' det ud af dalen. Hvis molekylet får et lille spark, utilstrækkeligt til at det kan undslippe ud af dalen, vil det simpelthen vende tilbage til bunden hvor det kom fra. Forestil dig nu at reaktanten og produktet (start og slut konfigurationerne i reaktionen) ligger i bunden af to tilstødende dale. Overgangstilstanden for reaktionen svarer til den passage, hvor man kan komme fra den ene dal til den anden med den mindst mulige mængde energi. Se for eksempel forsiden af min afhandling - her ser du en illustration af en kemisk reaktion på en energioverflade. Konfigurationerne i dalene svarer til reaktionens produkt og reaktant, og konfigurationen i midten er overgangstilstanden. Idéen med NEB er grundlæggende at forbinde de to nabodale med en meget tung og slap elastik. Vægten af elastikken vil trække den ned, og den vil automatisk finde den laveste passage mellem de to dale. Normalt beregnes denne energioverflade ved hjælp af elektronstruktur beregninger, og det er derfor at det er så tungt at finde overgangstilstande. I NeuralNEB er idéen i stedet at bruge neurale netværk, der er trænet på Transition1x datasættet til at udregne denne

overflade. Dette gjorde algoritmen i stand til at finde overgangstilstande for kemiske reaktioner 1350 gange hurtigere. Desværre var de overgangstilstande metoden fandt ikke helt inden for det vi kalder 'kemisk nøjagtigthed'. Kemisk nøjagtighed er utroligt svært at opnå, og resultaterne er stadig gode. Metoden er ny, og der er mange ting der kan pudses af for at gøre den bedre. Det vigtige resultat var at vise nytten af Transition1x og at konceptet fungerer.

E.3 Implicit Transition Operator

I det sidste projekt udviklede jeg en ny metode til at simulere molekylære systemer. I de foregående projekter fokuserede jeg primært på at finde overgangstilstande for reaktioner, da vigtig information om reaktionen kan blive udledt fra disse tilstande. En anden tilgang til at studere molekyler er simpelthen at vælge et molekyle og holde øje med, hvordan det bevæger sig over en længere periode, og så drage konklusioner baseret på ens observationer. Dette gør man gennem computersimulation - det er ikke muligt at holde øje med rigtige molekyler, de er for små og for kaotiske. Man kan bruge en metode der hedder molekylær dynamik til at lave disse simuleringer. Her har man et 'billede' af et molekyle. Ud fra dette billede er det muligt at udregne alle kvantemekaniske og magnetiske kræfter, der virker på hvert enkelt atom i molekylet. Ud fra disse beregninger kan man forudsige, hvordan et billede af molekylet vil se ud et splitsekund senere. Ved at gentage denne procedure mange gange kan man beregne en hel film af molekylets bevægelse. Dette skal gøres i ekstremt små skridt, da kræfterne på atomerne ændrer sig hurtigt, når atomerne bevæger sig den mindste smule, og derfor er sådanne simuleringer også meget vanskelige. I dette projekt trænede jeg et neuralt netværk til at se 'forud' i filmen. Modellen kunne da gætte hvordan molekylet ville have bevæget sig i løbet af 10, 100 eller 1000 skridt, uden at skulle lave mellemregningerne. Modellerne var i stand til at simulere op til en million skridt ad gangen. Jeg testede det ikke yderligere. Det giver os en mulighed for at 'spole' når vi simulerer molekyler, så vi både kan studere fænomener, der sker ekstremt hurtigt eller ekstremt langsomt. I princippet svarer det til, at vi med et kamera kan studere, hvordan en sommerfugl gennemgår sin transformation fra larve, og samtidig kan fange detaljer, såsom hvordan den bevæger sine vinger, mens den flyver.

E.4 Konlusion

Grundlæggende er de metoder jeg har udviklet i løbet af min Ph.D. en del af et verdensomspændende samarbejde, hvor vi forsøger at udvikle et virtuelt laboratorium, hvor vi effektivt kan udforske løsninger til en lang række problemer i verden. Kunstig intelligens og kvantekemi er et utroligt spændende felt, der bevæger sig meget hrugtigt, og det er fedt at være en del af det.

{2.7182818284}ϑφερτυθιοπσδφγηξκλ