Machine Learning methodology for the study of a disulfide exchange reaction

Zur Erlangung des akademischen Grades einer

DOKTORIN DER NATURWISSENSCHAFTEN

(Dr. rer. nat.)

von der KIT-Fakultät für Chemie und Biowissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M. Sc. Claudia Leticia Gómez Flores

Erster Gutachter: Prof. Marcus Elstner
Zweiter Gutachter: Prof. Wolfgang Wenzel
Tag der mündlichen Prüfung: 20.07.2022

Karlsruhe Institute of Technology
Institute of Physical Chemistry
Fritz-Haber-Weg 2
76131 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 2022

. . . . . .

(M. Sc. Claudia Leticia Gómez Flores)

Abstract

The thiol-disulfide exchange reaction is a nucleophilic substitution that occurs in a large class of proteins. It plays an important role regarding the third and fourth dimensional structure of proteins and the catalysis of biological reactions. Moreover, thiol-disulfide exchange can regulate the activity of certain proteins. In this work, the structural and environmental factors that influence this reaction will be discussed.

Due to its computational efficiency, Density-Functional based Tight-binding (DFTB) has positioned itself as a popular and reliable quantum mechanical method for condensed phase applications as it allows extensive phase space sampling and generating free-energy surfaces of complex reactions such as those occurring in biological systems. However, these savings in computational costs can come at the expense of lower accuracy. In the thiol-disulfide exchange DFTB shows inaccurate transition states. Literature review indicates that a proper description of this reaction requires high-level *ab initio* methods.

Hence, the motivation of this work was to correct the DFTB errors with a machine learning approach. To achieve this, we used a Behler–Parrinello-type Neural Network that learns the energy value differences between the *ab initio* quantum chemical potential and DFTB for a given molecular structure. The machine learned energy correction was then implemented into the DFTB+ software. With this new framework we were able to perform hybrid Quantum Mechanics/Molecular Mechanics (QM/MM) simulations of thiol-disulfide exchange with Coupled Cluster and B3LYP accuracy with a computational cost that is comparable to DFTB.

This correction algorithm is also implemented in a graphical interface pipeline that will help the user to generate and arrange training data, as well as exporting the machine learning model into DFTB+ for its further use in QM/MM simulations. The adoption of this pipeline is intended to expand the applications of the Neural Network code by prioritizing the knowledge of quantum modelling over a programming background.

Additionally, we present preliminary work on a machine learned force field to describe the disulfide-exchange reaction using Coupled Cluster reference data.

Zusammenfassung

Die Thiol-Disulfid-Austauschreaktion ist eine nukleophile Substitution, die in einer großen Klasse von Proteinen stattfindet. Sie spielt eine wichtige Rolle für die dritt- und viertdimensionale Struktur von Proteinen und die Katalyse biologischer Reaktionen. Außerdem kann der Thiol-Disulfid-Austausch die Aktivität bestimmter Proteine regulieren. In dieser Arbeit werden die strukturellen und Umgebungsfaktoren, die diese Reaktion beeinflussen, diskutiert.

Aufgrund ihrer Recheneffizienz hat sich die Density-Functional based Tight-binding (DFTB) Methode als beliebte und zuverlässige quantenmechanische Methode für Anwendungen in kondensierter Phase positioniert. Mit DFTB ist es möglich die freie Energiefläche komplexer Reaktionen zu erzeugen, da der Phasenraum ausreichen abgetastet werden kann. Diese Einsparungen bei den Rechenkosten können jedoch auf Kosten einer geringeren Genauigkeit gehen. Beim Thiol-Disulfid-Austausch zum Beispiel weisen die Übergangszustände eine fehlerhate Struktur und Energie auf. Die Literaturrecherche zeigt, dass für eine korrekte Beschreibung dieser Reaktion sehr genaue *ab initio* Methoden verwendet werden müssen.

Daher bestand die Motivation dieser Arbeit darin, die DFTB-Fehler mit einem maschinellen Lernansatz zu korrigieren. Um dies zu erreichen, haben wir ein neuronales Netzwerk vom Typ Behler-Parrinello verwendet, das die Energiewertdifferenzen zwischen der *ab initio* und DFTB Methode füreine gegebene Molekülstruktur erlernt. Die maschinell erlernte Energiekorrektur wurde dann in die DFTB+ Software implementiert. Mit diesem neuen Ansatz konnten wir hybride Quantum Mechanics/Molecular Mechanics (QM/MM)-Simulationen des Thiol-Disulfid-Austauschs mit Coupled Cluster und B3LYP-Genauigkeit mit einem Rechenaufwand durchführen, der mit DFTB vergleichbar ist.

Dieser Korrekturalgorithmus ist auch in einer Pipeline mit grafischer Schnittstelle implementiert, die dem Benutzer hilft, Trainingsdaten zu generieren und zu arrangieren sowie das maschinelle Lernmodell in DFTB+ zu exportieren, um es in QM/MM-Simulationen weiter zu verwenden. Die Einführung dieser Pipeline soll die Anwendungsmöglichkeiten des Codes für neuronale Netze erweitern, indem das Wissen über Quantenmodellierung gegenüber einem Programmierhintergrund bevorzugt wird.

Darüber hinaus stellen wir erste Arbeiten an einem maschinell erlernten Kraftfeld zur Beschreibung der Disulfid-Austauschreaktion unter Verwendung von Coupled Cluster-Referenzdaten vor.

Contents

List of Figures

1. Introduction

Proteins are the most abundant biological macromolecules. They are the last product of the central dogma of molecular biology, that explains the flow of genetic information:

$$\text{DNA} \longrightarrow \text{RNA} \longrightarrow \text{Protein}$$

They are present in all kinds of cells and all parts of cells, and have the most diversity of roles.

Proteins are constructed by a set of 20 amino acids covalently linked in characteristic linear sequences. Cysteine[1] and methionine are the only two amino acids that contain sulfur. Methionine is an essential amino acid, whereas cysteine is synthesized from methionine and therefore is nonessential.

Cysteine is the most nucleophilic of the 20 canonical amino acid residues and it is classified as a polar noncharged amino acid. Cysteines play a fundamental role in protein integrity as it is the only amino acid that can form disulfide bonds. These bonds form covalent links between parts of a protein molecule or between two different polypeptide chains, Fig.1.1[80].

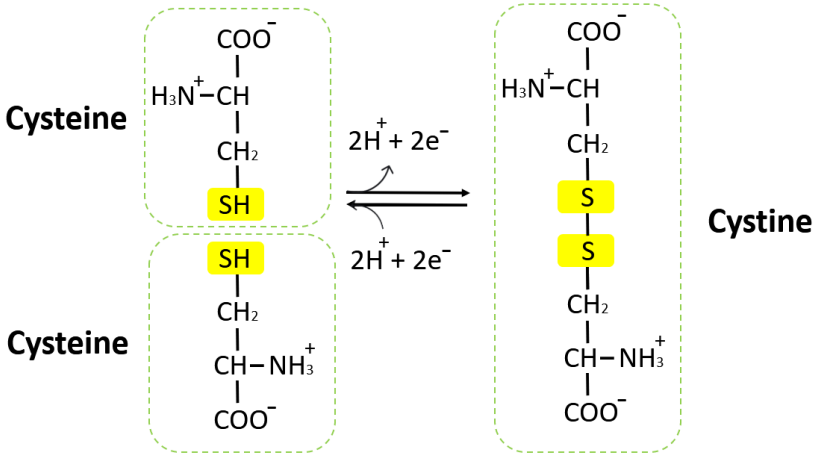


Figure 1.1.: Reversible formation of disulfide bonds by two oxidized cysteines.

Disulfide bonds can be broken and exchanged (thiol-redox reactions). Any change in these bonds may lead to partial protein folding, increasing the probability of misfolding and misassembly. Conformational changes in the proteins can lead to two outcomes: loss of function or gain of function. It is easy to think about all the catastrophic consequences due

[1]Cysteine was the first amino acid to be discovered, in 1810, by Wollaston in a urinary calculus.

to the loss of function, and how many biological pathways may be interrupted. However, gain of function can also result in aberrant signaling properties (as, for example, growth factor receptors in cancer), and severe misfolding can complicate protein degradation because of aggregation, leading to cell stress, cell death, or amyloid formation (source of neurodegenerative disease)[97]. Free cysteines may also become reactive and hold into metals, such as mercury, as the sulfhydryl side chain is a strong metal binder[3].

The role of disulfide bonds is not limited to protein function, it can also confer mechanical properties. For example, glutenin, a wheat protein rich in disulfide bonds, is responsible for the cohesive and elastic character of dough made from wheat flour. Similarly, the strong nature of hair, nails, horns and feathers is due to the extensive disulfide bonding in its α-keratin[80].

The relationship between the aminoacid sequence of a protein and its three-dimensional structure has been a query in biochemistry. An example of its complication is that the location of disulfide bonds is not coded in the DNA sequence. These bonds are formed during protein maturation in the endoplasmic reticulum of eukaryotic cells and the periplasmic space of prokaryotic cells, but are not necessarily stable. These bonds can also rearrange spontaneously (intra- or intermolecular) by mechanical stress [71]. Therefore a biochemical description is not sufficient, and a physical chemistry approach is also necessary to understand which factors influence the formation and exchange of these bonds.

Molecular dynamics (MD) are simulation methods that analyze the physical movement of atoms and molecules. In the present day, the use of molecular simulations complement the experimentalists, in Fig.1.2 we can observe the established relationship. Molecular dynamics influence the way experiments are designed and also can help understand results from experimental work. Thanks to MD simulations we can see properties that would not be unveiled empirically. Experiments can also be designed to test specific predictions from these simulations and validate the simulation results.

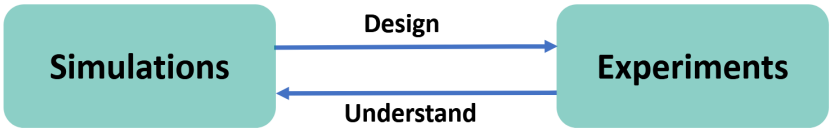


Figure 1.2.: Relationship between simulations and experiments.

For some MD simulations a microscopical explanation may not be sufficient and a quantum explanation is necessary, this is the case when studying changes in the electronic structures such as the formation and cleavage of bonds, charge transfer, or electronic excitation. Quantum Mechanics/Molecular Mechanics (QM/MM) methods are by now established as the state-of-the-art computational technique to treat these reactive and other "electronic" processes in biochemical systems[93]. An example of this methodology and the aforementioned relationship will be provided in this thesis: where experimentalists couldn't understand why some disulfide exchange reactions were preferred over others in a model protein[5], the answer to this behavior was given by simulating these experiments and analyzing the physical and chemical structure by *ps* time frames[65, 75].

Having a proper description of how electrons distribute themselves around nuclei is one of the biggest challenges in chemistry. Schrödinger provided an elegant mathematical formulation for this problem and although it is possible to solve this equation analytically for a single electron system, in practical applications when the number of electrons increases, the solution becomes basically impossible.

As a response to this problem, several approximations have been developed and the race to get closer to the exact solution to the Schrödinger equation for multiple electrons has been nonstop since its formulation. Advances in computational chemistry are not only motivated by finding an accurate method but also one that is computationally efficient. Although approaches such as Coupled cluster[11] can afford highly accurate values of the electronic energy, their computational costs increase exponentially with the number of electrons.

Semiempirical methods, such as Density-Functional based Tight-binding (DFTB)[33], have become an alternative as they require much less computer resources. DFTB has positioned itself as a reliable method for condensed phase applications, especially where an extensive sampling of the configurational space is important to the reactive process of interest such as chemical reactions in biological systems[32]. However, it can neglect important energy contributions, or show inaccurate transition states, as seen previously in thiol-disulfide exchange[87].

Machine learning has offered another way to solve quantum chemistry calculations: the algorithms can predict properties based on knowledge gained about known systems, without the need for solving the Schrödinger equation. Fig.1.3 illustrates the pipeline of information between classical quantum mechanical calculations and machine learning driven predictions.

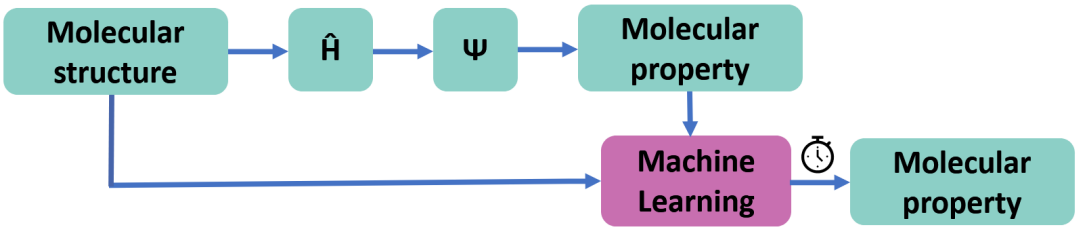


Figure 1.3.: Machine Learning can accelerate the calculation of molecular properties after a given model.

Over the last two decades, a growing amount of research has focused on defining and representing the Potential Energy Surface (PES) of systems using machine learning techniques. Since Doren and colleagues used artificial neural networks to fit a DFT PES in 1995 [18], artificial neural networks have attracted a lot of interest. Neural Network (NN) have been shown to be useful in the development of PES because of their ability to represent arbitrary functions.

In early work, NN focused on predicting the PES from the spectra of molecules, but over time it became more popular to predict the PES from structural information of the system, where the energy is based upon the relative positions of the atoms[56]. In 2007,

Behler & Parrinello presented a NN framework where the total energy of the system is constructed as a sum of atomic energies, and each atomic energy contribution is the output of an individual NN [17].

Yang's group extended Behler and Parrinello's neural-network representation into a method known as QM/MM-NN that predicted the potential energy difference between semiempirical and *ab initio* QM/MM methods. The potential energies predicted with the built neural network are then used to correct the free-energy profile obtained from the semiempirical QM/MM simulation to the *ab initio* QM/MM [98]. Applications of this methodology have again been proven to be successful, such as the Irle's group correction of intramolecular hydrogen bond energetics by learning the difference between DFTB to DFT potentials [115].

The aforementioned methodologies represent what is called a second (2nd) generation potential[2]. However, long-range interactions have being included in the latest developments. Four generation of NN potentials have been defined and can be summarized as (Fig.1.4):

- 1° generation: The total energy of a system is calculated directly by the NN.
- 2° generation: The total energy of a system is the sum of atomic energies.
  - Each atomic energy contribution is the output of an individual NN.
- 3° generation: The total energy of a system consists of an electrostatic and a short-range contribution.
  - The short-range contribution gets calculated by a 2° generation potential.
  - The electrostatic potential gets calculated by an analogous NN that learns atomic charges.
- 4° generation: The total energy of a system consists of an electrostatic and a short-range contribution.
  - The atomic charges depend on the global system and are obtained from a charge equilibration process relying on environment-dependent atomic electronegativities, which are learned by a NN.
  - The short-range contribution gets calculated by a 2° generation potential where the input vectors contain in addition the atomic charges as global descriptors for changes in the local electronic structure.

This work started from the hypothesis that, as long as the error when describing the disulfide exchange reaction is based on the local environment, a machine learning algorithm can achieve as good accuracy as an *ab initio* method to describe the reaction with the computational efficiency of a semiempirical method. A graphical illustration of the hypothesis can be seen at Fig.1.5.

[2]With the exception of early work using the spectra of molecules, which represent a 1st generation potential.

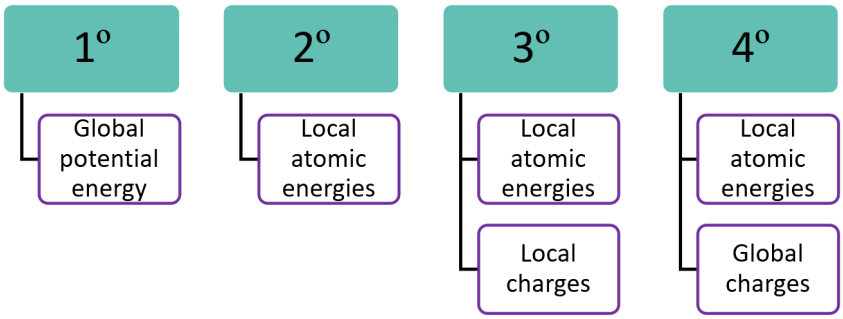


Figure 1.4.: Overview of the four generations of neural network potentials.

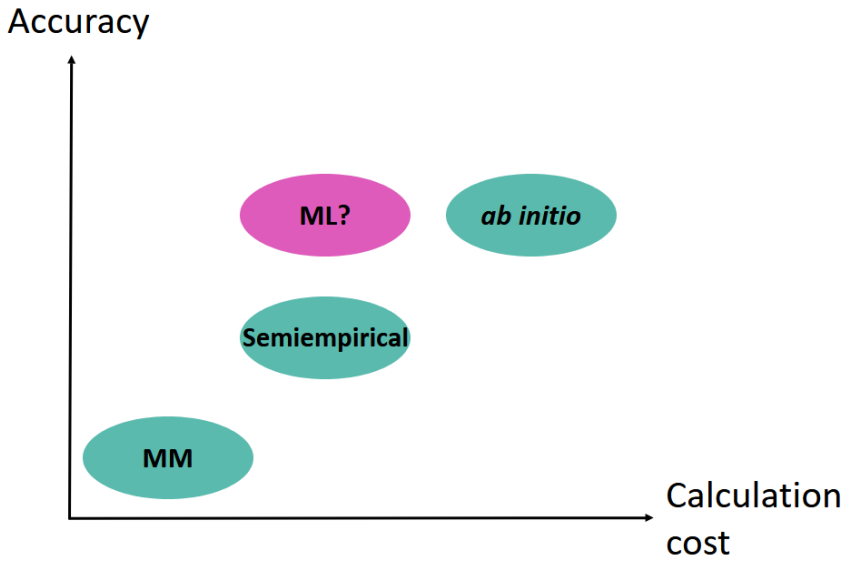


Figure 1.5.: Graphical representation of the hypothesis: The ML approach (in purple) is hypothesised to achieve the same accuracy as *ab initio* methods, while retaining the low computational cost of the semiempirical methods. MM-approaches have the lowest cost at the expense of the lowest accuracy.

For getting a better understating of the disulfide-shuffling exchange reaction, snapshots of QM/MM simulations of a mutated protein (with strategical cysteines located inside a loop), were analysed by statistical methods and a ML algorithm. The structural and environmental features were analyzed on a basis of a ratio of successful:unsuccessful exchange reactions via some preferred cysteine. The obtained results were published in Ref.[75]. Furthermore, this system would be part of the dataset used for learning the aforementioned energy correction.

Initially a Δ-ML methodology (2nd generation) was implemented as an analogy to reparametrization of the S–S repulsive potentials to correct the gas-phase DFTB-genereated PES. A Behler-Parrinello NN was used to learn the energy diffrence. The obtained results formed part of the published Ref.[53], where the ML models were used for further QM/MM

simulations. The implementation of this methodology into a GUI and its benefits are also explored.

At the end of this project we gave the initial steps for implementing global charges into the learning of an *ab inito* energy PES (4th generation). We present the results achieved as a preamble and invite new students to continue in this direction.

2. Theoretical Background

2.1. Quantum chemistry

The discovery of the Schrödinger equation (SE) in 1926 was a milestone in the development of quantum mechanics. The SE is for particles, the equivalent to the Newton's equations of motions for macroscopic systems. The main differences that distinguishes the microscopic realm from the macroscopic, is its discrete quantization and wave-particle duality, and that the classical mechanics is deterministic while quantum mechanics is probabilistic.

Any quantum-mechanical system can be described by a continuous wave function ψ. The time-dependent SE can observed in Eq.2.1, where $\hbar$ is the Planck constant, a constant that defines the amount of energy that a photon can carry, v which is an external potential, m the mass of the particle, t time and r is spatial position.

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + v\right)\psi(r,t) = i\hbar\frac{\partial\psi(r,t)}{\partial t} \tag{2.1}$$

If we consider the external potential to be independent of time and we assume no evolution on this coordinate, the time-independent SE can be written as Eq.2.2:

$$\hat{H}\psi = E\psi \tag{2.2}$$

The SE falls into the category of partial differential eigenvalue equations. The energy can be obtained by solving its Hamiltonian $\hat{H}$ operator, which includes all the possible energy contributions as seen in Eq.2.3, where $\vec{R}_i$, M_i and Z_i indicate the spatial coordinates, masses, and charges of the nuclei in the molecule respectively, and $\vec{r}_i$ are the electronic coordinates.

$$\hat{H} = \underbrace{-\sum_i \frac{\nabla^2_{\vec{R}_i}}{2_{M_i}}}_{K_N} \underbrace{-\sum_i \frac{\nabla^2_{\vec{r}_i}}{2}}_{K_{e^-}} \underbrace{-\sum_{i,j} \frac{Z_i}{|\vec{R}_i - \vec{r}_j|}}_{V_{N-e^-}} + \underbrace{\sum_{i,j>1} \frac{Z_i Z_j}{|\vec{R}_i - \vec{r}_j|}}_{V_{N-N}} + \underbrace{\sum_{i,j>1} \frac{1}{|\vec{r}_i - \vec{r}_j|}}_{V_{e^--e^-}} \tag{2.3}$$

where

K_N = Kinetic energy of nuclei
K_{e^-} = Kinetic energy of electrons
V_{N-e^-} = Nuclei-electron attraction
V_{N-N} = Nuclei-nuclei repulsion
$V_{e^--e^-}$ = Electron-electron repulsion

A solution of the time-independent electronic SE of a given atomic system provides, in principle, full access to its chemical properties. In bigger systems, with an increased

number of electrons, an exact solution quickly becomes infeasible. Therefore, the hydrogen atom is the only atom for which the SE is exactly solvable.

2.1.1. Born-Oppenheimer approximation

The Born–Oppenheimer approximation (BOA), or adiabiatic approximation, treats the nuclei as a stationary point entity. It is based on the hypothesis that electrons evolve on a much shorter time scale than the nuclei, then it makes use of a time-scale separation between fast and slow degrees of freedom and states that the electrons "adiabatically follow" the nuclei.

The Hamiltonian then can be split into a nuclear and electronic part (Eq.2.4),

$$\hat{H} = \hat{H}_{nucl}(\vec{R}) + \hat{H}_{elec}(\vec{R}, \vec{r}) \tag{2.4}$$

and consequentially the wave function gets equally separated (Eq.2.5).

$$\psi(\vec{R}, \vec{r}) = \phi_{nucl}(\vec{R})\phi_{elec}(\vec{R}, \vec{r}) \tag{2.5}$$

$$\hat{H}_{elec}\chi_{elec}(\vec{R}, \vec{r}) = E_{elec}(\vec{R})\chi_{elec}(\vec{R}, \vec{r}) \tag{2.6}$$

As a result, we can solve the electronic state of a molecule for fixed nuclear configurations. For every nuclear configurations $\vec{R}_i$, the energy would describe the electronic movement (Eq.2.6). Gradually stepping nuclear configurations and solving for the energy leads to a potential energy surface, or adiabatic state.

If the BOA is valid, ie: the atomic positions, nuclear charges and total charge are known, the potential energy of a system is fully defined by its electronic Hamiltonian (Eq.2.7) and provides a Potential Energy Surface (PES).

$$\hat{H}_{elec} = \underbrace{-\sum_i \frac{\nabla^2_{\vec{r}_i}}{2}}_{K_{e^-}} \underbrace{-\sum_{i,j} \frac{Z_i}{|\vec{R}_i - \vec{r}_j|}}_{V_{N-e^-}} + \underbrace{\sum_{i,j>1} \frac{1}{|\vec{r}_i - \vec{r}_j|}}_{V_{e^--e^-}} \tag{2.7}$$

In other words, a PES is a multidimensional real-valued function that provides the potential energy of a system as a function of the atomic coordinates, see Fig.2.1. The description of chemical reaction dynamics are presented in terms of propagation on these PES. The barriers on these surfaces are how we describe the rates of chemical reactions and transition states. The trajectories along these surfaces are used to describe mechanism.

2.1.2. *Ab initio* methods

Ab initio methods, as the name implies, require no empirical information about the molecular system but rather apply various approximations to solve the SE through the use of wavefunctions to describe atomic orbitals. The accuracy of this methods will depend on the model chosen for solving the wavefunction.

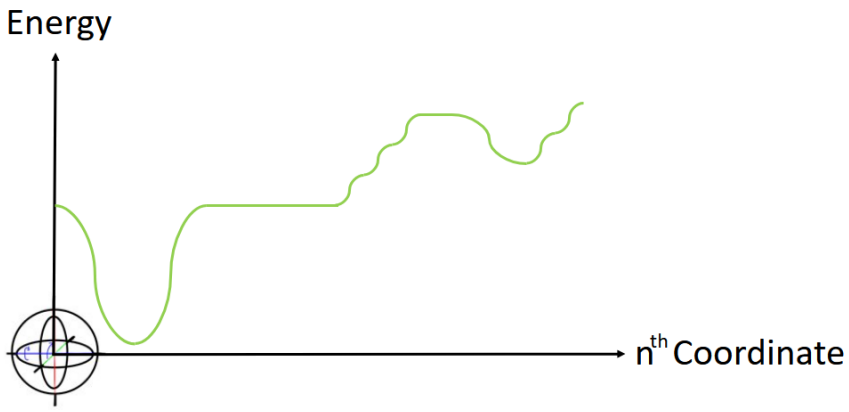


Figure 2.1.: Simplified representation of an arbitrary 1-Dimensional PES.

2.1.2.1. Hartree-Fock

The simplest type of an *ab initio* calculation is the Hartree–Fock (HF) scheme. To solve the electronic Hamiltonian, Hartree suggested that we can approximate its electron-electron repulsion ($V_{e^- - e^-}$ in Eq.2.7) as an average, denoted as $v^{HF}(x_i)$. Then, the Hamiltonian can be re-written as a sum of one-electron operators (Fock operators), see Eq.2.8.

$$\hat{H}_{elec} = \sum_{i=1}^{N} f(x_i) \tag{2.8}$$

$$f(x_i) = -\frac{\nabla^2_{\vec{r}_i}}{2} - \frac{Z_i}{|\vec{R}_i - \vec{r}_j|} + v^{HF}(x_i)$$

Then the electronic SE can be expanded into a set of one-electron eigenvalue equations:

$$f(x_i)\chi(x_i) = \varepsilon\chi(x_i) \tag{2.9}$$

$v^{HF}(x_i)$ depends on the entire system's wavefunction. The HF method used an initial-guess wavefunction, then calculates $v^{HF}(x_i)$ and solves Eq.2.9 iteratively until a convergence criteria is satisfied.

The HF approximation breaks down a multi-electron wavefunction into a set of molecular orbitals (one-electron wavefunctions). Hartree approximately considered the electrons to be uncorrelated to build a separable wavefunction. The square of the wavefunction is the probability of finding electrons in a specific volume of space, and the probability of occurring two independent events at the same time is equal to the product of their individual probabilities. Thus, the electronic wavefunction of N uncorrelated electrons must be equal to the product of the one-electron wavefunctions.

$$\psi_{HP}(x_1, x_2, ..., x_n) = \chi_i(x_1)\chi_j(x_2)...\chi_k(x_N) \tag{2.10}$$

Eq.2.10, referred as the Hartree-product, shows the one-electron wavefunctions as spin-orbitals (χ), and each of them is a function of spatial coordinates r and a spin coordinate ω,

which can be either α (↑) or β (↓). The set of space-spin coordinates will then be $x = \{r, \omega\}$. While the Hartree-product is fairly convenient, the electrons are not indistinguishable and it fails to satisfy the antisymmetry principle [1].

To satisfy the aforementioned requirements, we can use a linear combination of the Hartree products with a normalization factor (for an N-electron system, there are $N!$ possibilities for interchanging them). This will be know as the Slater determinant (Eq.2.11), where every row belongs to an electron and each column represents a spin-orbital. The antisymmetry is equivalent to the assumption that each electron moves independently of all the others except that it feels the Coulomb repulsion due to the average positions of all electrons. The Slater determinant also introduces some degree of correlation, as electrons with parallel spin can not be in the same spatial orbitals.

$$\psi(x_1, x_2, .., x_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_i(x_1) & \chi_j(x_1) & .. & \chi_k(x_1) \\ \chi_i(x_2) & \chi_j(x_2) & .. & \chi_k(x_2) \\ : & : & .. & : \\ \chi_i(x_N) & \chi_j(x_N) & .. & \chi_k(x_N) \end{vmatrix} \tag{2.11}$$

Now that the electronic wavefunction has been built from the linear combination of spin-orbitals, the task is to find spin-orbitals. Each spin-orbital $\chi_i(x)$ consists of two parts, the spatial function $\varphi_j(r)$ and the spin function $\varsigma(s)$. The spatial function is a function of spatial coordinates, and the spin function is related to electron spin.

The Molecular orbital theory (MO) describes the distribution of electrons in molecules as an analogy to electrons in atoms by the atomic orbitals. Molecular orbitals are obtained by combining the atomic orbitals of the atoms in the molecule, see Fig.2.2. The process of combining atomic orbitals to generate molecular orbitals is called the Linear Combination of Atomic Orbitals (LCAO).

MO describes the behavior of electrons in a molecule in terms of combinations of the atomic wavefunctions. As the wavefunction describes the wavelike properties of an electron, combining waves can lead to constructive or destructive interference. In orbitals, the waves are three dimensional, and they combine with in-phase waves producing regions with a higher electron density and out-of-phase waves producing nodes, or regions of zero-electron density.

The shape of a given molecular orbital describes the probability of finding an electron, where the attraction to all the nuclei and the average repulsion to all other electrons are included. The interaction of the orbitals of the reactants can predict the reactivity of them. The formation of bonds can be anticipated through the lowering of energy of the molecule, whenever there is an enhanced probability density in the internuclear region. It is not the shift of electron density into the internuclear region that lowers the energy of the molecules but the freedom that this redistribution gives from the wave function to shrink in the vicinity of two nuclei.

The MO theory implies:

- Considers electrons delocalized throughout the entire molecule.

[1] A wavefunction describing fermions should be antisymmetric with respect to the interchange of any set of space-spin coordinates.

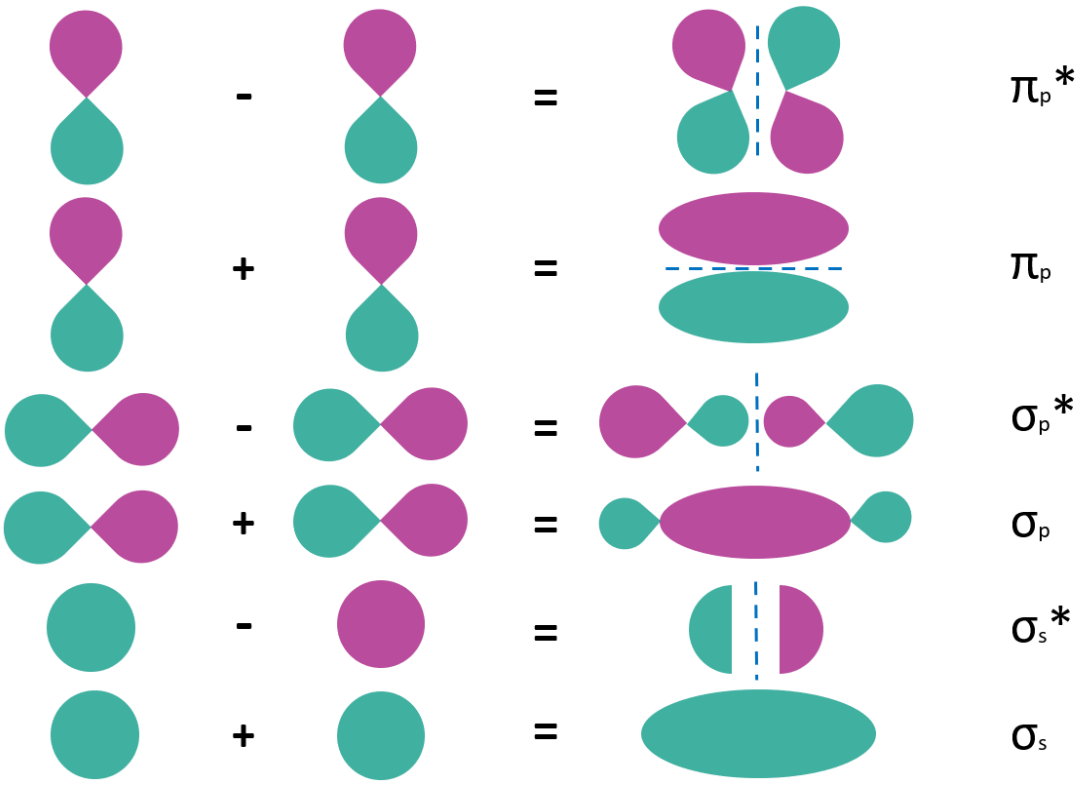


Figure 2.2.: Molecular orbital construction from the atomic orbitals superposition.

- Like an atomic orbital, a molecular orbital is full when it contains two electrons with opposite spin.
- Creates atomic orbitals to form molecular orbitals:
  - Bonding σ, π: Additive combination of orbitals, lower in energy than the original orbitals.
  - Antibonding $\sigma*, \pi*$: Subtractive combination, higher in energy than the bonding orbital and original orbitals.
- Predicts the arrangement of electrons in molecules.

Each spatial molecular orbital is expanded in terms of the basis functions φ_r, as seen in Eq.2.12. The basis set for the molecular wavefunction is represented in computer programs by stored sets of exponents and coefficients.

$$\psi = \sum_{r}^{N} c_r \varphi_r \tag{2.12}$$

Then Eq.2.9 can be rewritten as a solvable matrix equation:

$$f(r_i) \sum_{r}^{N} c_r \varphi_r = \varepsilon \sum_{r}^{N} c_r \varphi_r \tag{2.13}$$

Multiplying by φ^* and integrating:

$$\sum c_r \underbrace{\int \varphi_r^* f(r_i) \varphi_r dr_i}_{\text{Fock matrix}} = \varepsilon \sum c_r \underbrace{\int \varphi_r^* \varphi_r dr_i}_{\text{Overlap matrix } S} \tag{2.14}$$

The associated calculated integrals are then used to formulate the Hamiltonian matrix on the basis of interactions between the wavefunction of pairs of atoms (off-diagonal elements) and each atom itself (diagonal elements) via some potential that varies according to the method. The optimum values of the coefficients (c_r) of the basis set functions are found by applying the variational principle.

The variational theorem states that the energy calculated from an approximation to the true wavefunctions will always be greater than the true energy. The better the wavefunction approximation is, the lower the energy. The best wave function is when the energy is at its minimum.

The HF equations are obtained by imposing the condition on the expression for the energy, subject to the constraint that molecular orbitals remain orthonormal. The variational principle instructs that as we get closer and closer to the 'true' one electron ground state wavefunction, we will obtain lower and lower energies for our guess followng Eq.2.15. The coefficients minimize the energy for all possible linear combinations. This equation is called secular equation (Eq.2.16). In a one electron system the lowest energy molecular orbital is called ground state and the highest called excited states.

$$\frac{\partial E}{\partial c_r} = 0 \tag{2.15}$$

$$\sum_r c_r (H_{rs} - ES_{rs}) = 0 \tag{2.16}$$

The conditions for choosing a basis set is that the behaviour must agree with the physics of the problem so it can converge and that the function should go towards zero as the distance between electrons and nuclei increases.

The HF method is also called the Self-Consistent Field (SCF) and even this approximation can be of low quality due to the lack of correlation effects. The SCF wavefunction is often used as a starting point to construct a more sophisticated wavefunction ansatz.

2.1.2.2. Many body perturbation theory

Møller and Pesset proposed a way to solve the electron correlation problem. Following the Rayleigh-Schrodinger perturbation theory in which the Hamiltonian is written as a sum of a zeroth order Hamiltonian[2] with a perturbation. This zeroth order Hamiltonian is the sum of the one-electron Fock operators for the N-electrons.

$$\hat{H}_0 = \sum_{i=1}^{N} f(x_i) \tag{2.17}$$

[2] unperturbed Hamiltonian

The ground state wavefunction, ψ_0, is an eigenfunction of $\hat{H}_0$ with associated energy E_0 given by the sum of orbital energies of all occupied spin-orbitals. The perturbation is then $\hat{H}^{(1)} = \hat{H} - \hat{H}_0$. And for the i^{th} electron the perturbation will be:

$$\hat{H}^{(1)}(i) = j_o \sum_j \frac{1}{r_{ij}} - \sum_m \{2J_m(i) - K_m(i)\} \tag{2.18}$$

where the sum j omits electron i, and the sum over m is over the occupied molecular orbital. The HF energy associated with the normalized ground state ψ_0 is the expectation value of $\hat{H}$ and equiavlent to the sum of the sum of the zeroth order energy ($E_0^{(0)}$) and the first order energy ($E_0^{(1)}$), where the latter is equal to the expected value of the perturbation.

The second order correction of the energy is then:

$$E^{(2)} = \sum_{j \neq 0} \frac{\langle \psi_j | \hat{H}^{(1)} | \psi_0 \rangle \langle \psi_0 | \hat{H}^{(1)} | \psi_j \rangle}{E_0^{(0)} - E_j^{(0)}} \tag{2.19}$$

where ψ_j is a multiply excited determinant and an eigenfunction of $\hat{H}_0$ with eigenvalue $E_j^{(0)}$. The inclusion of second order correction is designated as MP2.

2.1.2.3. Coupled Cluster

Coupled cluster theory (CC) is one of the most accurate methods for quantum chemistry. The couple cluster method introduces the cluster operator C, which relates the exact electronic wavefunction ψ to the HF wavefunctions ψ_0 through Eq.2.20

$$\psi = e^C \psi_0 \tag{2.20}$$

where the exponential operator is defined by the series expansion: $e^C = 1 + C + \frac{1}{2!}C^2 + \frac{1}{3!}C^3 + ...$ The effect of the cluster operator C is the sum of the effects of a one-electron excitation operator C_1, two-electron excitation operator C_2, to N-electron excitation and so on, the t_a^p are called single excitation amplitudes and t_{ab}^{pq} double excitation amplitudes. An approximation in CC applications is to truncate the cluster operator C to include only certain types of terms. This approach is known as Coupled Cluster Singles and Doubles (CCSD), C is approximated by $C_1 + C_2$ and for triples $C = C_1 + C_2 + C_3$.

$$C_1 \psi_0 = \sum_{a,p} t_a^p \psi_a^p \tag{2.21}$$

$$C_1 \psi_0 = \sum_{a,b,p,q} t_{ab}^{pq} \psi_{ab}^{pq} \tag{2.22}$$

The CCSD augmented with perturbative triples correction [CCSD(T)] is considered the “gold standard” of quantum chemistry since CCSD(T) often provides an accuracy comparable to experiment within chemical accuracy (1 kcal/mol) for most energetic properties, such as cohesive energies.

2.1.2.4. Density Functional Theory

Density Functional Theory (DFT) is one of the most dominant procedures in computational chemistry, it uses the electron density function as a basic descriptor of the electronic system.

It can scale to larger molecules, but at the price of limited accuracy. Still, DFT allows to study dynamical processes such as structural transformations of covalently bonded materials as well as chemical reactions in condensed phases.It gives a better description of interatomic interactions such as bond redistribution.

The energy, $E[\rho]$, of an electronic system can be written in terms of an electron density functional, ρ. This functional fully depends on the spatial position of electrons, disregarding the number of electrons (Fig.2.3). For a system of N electrons, $\rho(r)$ denotes the total electron density at a particular point r in space.

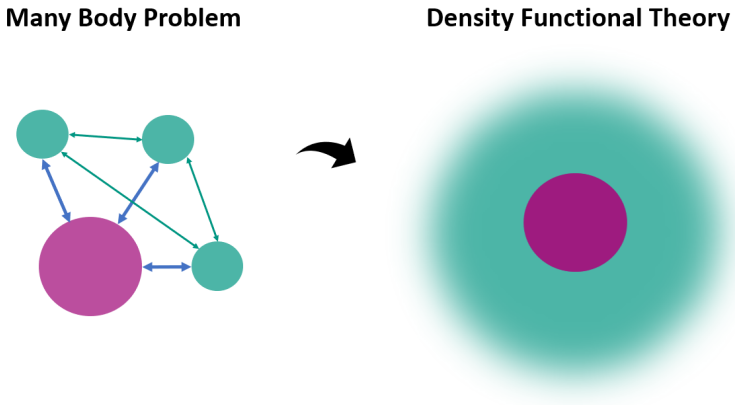


Figure 2.3.: DFT decreases the many body problem wave function form 3N variables ($\phi(\vec{r_1}, \vec{r_2}, .., \vec{r_N})$) to 3 variables ($\rho(\vec{r})$).

The Thomas-Fermi model states that the electron density determines properties of the molecule and the energy is correctly given by a variation principle. The Hohenberg-Kohn mapping theorem confirms that is possible to express the ground state energy of a molecule as a functional of the ground state electron density (Eq.2.23). It is then sufficient to know the ground state of a molecule to know the ground state electron density and to determine any property of a molecule.

$$E[\rho(r)] = \int V_{ext}(r)\rho(r)dr + F[\rho(r)] \tag{2.23}$$

The left side of the equation arises from the interaction of electrons with an external potential $V_{ext}(r)$ and $F[\rho(r)]$ is the sum of the kinetic energy of the electrons and the contributions of interelectronic interactions. The minimum value in the energy corresponds to the ground state electron density. The constraint on the electron density as the number of electrons is fixed (Eq.2.24). It is introduced as a Lagrangian multiplier μ (Eq.2.25), which can also be identified with the chemical potential of an electron cloud for its nuclei, which is in turn related to the electronegativity, χ.

$$N = \int \rho(r) dr \tag{2.24}$$

$$\left(\frac{\delta E[\rho(r)]}{\delta \rho(r)}\right)_{V_{ext}} = \mu = -\chi = \left(\frac{\partial E}{\partial N}\right)_{V_{ext}} \tag{2.25}$$

Kohn-Sham suggested later that $F[\rho(r)]$ should be an approximate sum of three terms: kinetic energy (E_{KE}), Coulomb energy (E_H)and exchange-correlation energy functional (E_{XC}):

$$F[\rho(r)] = E_{KE}[\rho(r)] + E_H[\rho(r)] + E_{XC}[\rho(r)] \tag{2.26}$$

By adding electron-nuclear interaction into Eq.2.26 we can obtain the full expression for the N-electron system within the Kohn-Sham definition:

$$\begin{aligned} E[\rho(r)] = & \underbrace{\sum_{i=1}^{N} \int \psi_i(r) \left(-\frac{\nabla^2}{2}\right) \psi_i(r) dr}_{E_{KE}[\rho(r)]} \\ & + \underbrace{\frac{1}{2} \int \int \frac{\rho(r_1)\rho(r_2)}{|r_1 - r_2|} dr_1 dr_2}_{E_H[\rho(r)]} \\ & + E_{XC}[\rho(r)] - \sum_{A=1}^{M} \int \frac{Z_A}{|r - R_A|} \rho(r) dr \end{aligned} \tag{2.27}$$

Eq.2.27 defines the exchange-correlation energy energy functional, which contains not only contributions due to exchange and correlation as the names states, but also a contribution of the energy difference between the true kinetic energy and the kinetic energy dependent of the density. Kohn and Sham rewrote the density of the system as the sum of the square moduli of a set of one-electron orthonormal orbitals (Eq.2.28).

$$\rho(r) = \sum_{i=1}^{N} |\psi_i(r)|^2 \tag{2.28}$$

To solve the Kohn-Sham equations a self-consistent approach is taken and the quality of DFT depends on the chosen exchange-correlation functional.

The simplest functional is the Local (spin-) Density Approximation (LDA), based on a model called the *uniform electron gas* in which the electron density is constant through all the space. This functional is obtained by the derivation of the total exchange-correlation energy, defined by Eq.2.29, where $\epsilon_{XC}(\rho(r))$ is the exchange-correlation energy per electron as a function of the density in the uniform electron gas. LDA gives in general an accurate description of the atomic structure, elastic, and vibrational properties for a wide range of systems. However, it fails to describe the energetics of chemical reactions, overestimating the binding energies of molecules and solids in particular.

$$E_{XC}^{LDA}[\rho(r)] = \int \rho(r)\epsilon_{XC}(\rho(r))dr \tag{2.29}$$

Following the LDA approach, we could express the exchange-only energy as:

$$E_{X}^{LDA}[\rho_\alpha(r), \rho_\beta(r)] = -\frac{3}{2}\left(\frac{3}{4\pi}\right)^{\frac{1}{3}} \int \left(\rho_\alpha^{\frac{4}{3}}(r) + \rho_\beta^{\frac{4}{3}}(r)\right) dr \tag{2.30}$$

where α and β represent up and down spins.

Recent Generalized Gradient Approximations (GGA) gives a better description of energy barriers, with an error of approx. 1 kcal/mol. They do not consider a uniform distribution of electrons as LDA, instead they depend on the local density as well as on the spatial variation of this density.

GGA gradients are typically divided into separate exchange and correlation contributions. Several gradient corrections to the exchange functional have been proposed, one of the most popular is the Becke correction, known as B88.

A combination of the standard LDA exchange (Eq.2.30) with the Becke gradient exchange correction and the correlation functional Lee-Yang-Parr correlation functional (LYP) is a popular choice, commonly abbreviated as BLYP.

Hybrid functionals are methods that include exact exchange energy. An example of these functionals is the B3LYP, constructed by the Becke 3-parameter exact exchange energy and the LYP correlation.

For an extended definition of the mentioned exchange, correlation and exchange-correlation functionals see to Appendix.A.1.

2.1.3. Semiempirical methods

Semiempirical methods are derived from HF by neglecting all the integrals involving more than two nuclei in the construction of the Fock matrix. In order to compensate for the errors caused by these approximations, semiempirical parameters are incorporated into the formalism and calibrated against reliable experimental data or calculated exactly from the corresponding analytic formulas.

2.1.3.1. Density Functional based Tight-binding

The Density-Functional based Tight-binding (DFTB) are a series of models based on DFT, they are derived from the Taylor series expansion of the Kohn-Sham total energy (Eq.2.27) with respect to charge density fluctuations ($\delta\rho$), where E is the energy and $\rho = \rho^0 + \delta\rho$. With an expansion up to the third order and the total energy can be written as:

$$
\begin{aligned}
E[\rho] = & \frac{1}{2}\sum_{a} b\frac{Z_a Z_b}{R_{ab}} \\
& - \frac{1}{2}\int\int \frac{\rho^0(r)\rho^0(r')}{|r-r]|}drdr' \\
& - \int V^{XC}[\rho^0]\rho^0(r)dr + E^{XC}[\rho^0] \\
& + \sum_i n_i \langle\psi_i|\hat{H}[\rho^0]|\psi_i\rangle \\
& + \frac{1}{2}\int\int \left(\frac{1}{|r-r'|} + \frac{\delta^2 E^{XC}[\rho]}{\delta\rho(r)\delta\rho(r')}\,|^0_\rho\right)\delta\rho(r)\delta\rho(r')drdr' \\
& + \frac{1}{6}\int\int\int \frac{\delta^3 E^{XC}[\rho]}{\delta\rho(r)\delta\rho(r')\delta\rho(r'')}\,|^0_\rho\,\delta\rho(r)\delta\rho(r')\delta\rho(r'')drdr'dr'' \\
& = E^0[\rho^0] + E^1[\rho^0,\delta\rho] + E^2[\rho^0,(\delta\rho)^2] + E^3[\rho^0,(\delta\rho)^3]
\end{aligned}
\tag{2.31}
$$

The application of tight-binding (TB) describes the Hamiltonian eigenstates with an atomic-like basis set and replaces the Hamiltonian with a parameterized Hamiltonian matrix whose elements depend only on the internuclear distances and orbital symmetries. The total energy then gets described by a sum of two contributions:

- Electronic contributions: They are the sum over the energies of all occupied orbitals obtained by diagonalization of the parameterized Hamiltonian matrix, they come from DFT and can be either LDA or GGA functionals.
- Repulsive energy contributions: They are obtained by the sum of the atomic-pair terms and are approximated as a sum of pair potentials, which are represented either by spline functions or by polynomials.

In the lowest order (DFTB1) takes only the first two contributions of Eq.2.31. E^1 gets defined as the sum of the occupied KS energies. E^0 gets approximated as superpositions of neutral atomic densities[3]. The whole Hamiltonian and overlap matrices contain one and two-centre contributions only, calculated and tabulated in advance as functions of the distance between atomic pairs. E^0 then gets redefined as the sum of pairwise repulsive terms.

$$
E^{DFTB1} = \sum_{iab}\sum_{\mu\epsilon a}\sum_{\nu\epsilon b} n_i c_{\mu i} c_{\nu i} H^0_{\mu\nu} + \frac{1}{2}\sum_{ab} V^{rep}_{ab} \tag{2.32}
$$

where H^0 are the diagonal Hamiltonian matrix elements, c the molecular orbital coefficients and n the occupation number of the KS orbital.

DFTB1 deals with systems with small and large intramolecular charge transfer, but fails for molecular systems with intermediate charge transfer.

[3]This means it will not depend on the environment, as a consequence this "stored" parameter can be applied to other molecules in different chemical environments.

In the second-order (DFTB2)[4] and third-order (DFTB3) the density fluctuations are written as a superposition of atomic contributions ($\delta\rho = \sum_a \delta\rho_a$) and the terms in the energy expansion correspond to a Self Consistent Charge (SCC) representation, where the deviation of the ground-state density from the reference density is represented by charge monopoles only.

DFTB2 is a better approach for systems with intermediate charge transfer within a molecule, its formulation is the following:

$$E^{DFTB2} = E^{DFTB1} + \frac{1}{2}\sum_{ab} \Delta q_A \Delta q_b \gamma_{ab} \tag{2.33}$$

The γ-function (γ^h for hydrogen[5]) describes the atomic chemical hardness (or Hubbard parameter as calculated from DFT), which is the second derivative of the energy with respect to the charge density, Analytically, it is defined by $\gamma_{ab} = \frac{1}{r_{ab}} - S$, where r is the distance and S is a short-range function being responsible for the correct convergence of $\gamma_{ab} = 0$

DFTB3 leads to a new degree of self-consistency, as it also describes the change of the chemical hardness of an atom with its charge state. This parameter is fitted in order to improve the performance of the model.

$$E^{DFTB3} = E^{DFTB2} + \frac{1}{3}\sum_{ab} \Delta q_a^2 \Delta q_b \Gamma_{ab} \tag{2.34}$$

where Γ_{ab} comes from the derivative of γ_{ab} w.r.t charge q.

2.1.4. Population analysis

A way to calculate the atomic charge of each atom is to partition the electron density between the nuclei so that each nucleus has a number[6] of electrons. Although atomic charges in a molecule are not experimentally observable quantities, they are fundamental and useful tools to understand and relate properties of molecules to their structures.

The charge distribution can provide information on how much of the electronic density is concentrated around an atom, which is a quantity of interest as it may be directly linked to the reactivity of an atom, often identified with its ability to be attacked by a nucleophilic or an electrophilic agent.

There is no agreed best procedure for computing the particle charge, and there are four types of models:

- Class I is not determined by quantum mechanics.
- Class II involves direct partitioning of the molecular wavefunction into atomic contributions following some arbitrary orbital based scheme.

[4] Also called SCC-DFTB

[5] This function gets modified for hydrogen, as for this atom, the atomic charge density is not proportional to the chemical hardness.

[6] not necessary integral

- Class III is computed after a physical observable based on the wavefunction.
- Class IV is calculated after a semiempirical mapping of a precursor charge.

The Mulliken atomic charges (Class II) is one of the most used distributions and we will describe them as they will be used in this work.

In the Mulliken distribution, all the electron density (P) in an orbital, is allocated to the atoms on which φ_μ is located. The remaining density is associated with the overlap population, $\varphi_\mu\varphi_\nu$. For each element $\varphi_u\varphi_v$ of the density matrix, half of the density is assigned to the atom in which φ_μ is located and half to the atom in which φ_ν is located. The net charge on atom A is then calculated by subtracting the number of electrons from the nuclear charge Z_A as:

$$q_A = Z_A - \sum_{\mu=1;\mu \text{ on A}}^{K} P_{\mu\mu} \sum_{\mu=1;\mu \text{ on A}}^{K} \sum_{\nu=1;\nu\neq\mu}^{K} P_{\mu\nu}S_{\mu\nu} \tag{2.35}$$

where

$$P_{\mu\nu} = 2\sum_{i=1}^{N/2} c_{\mu i}c_{\nu i}$$

This analysis becomes trivial when using a SCF as the elements of the density matrix have been already determined. The Mulliken charges are basis set dependent and its downside is that the equal overlap division can lead to exaggerated charge separations.

2.2. Molecular Dynamics

Molecular dynamics (MD) is a deterministic methodology that treats each atom of a system as a classical Newtonian system. This means that quantum effects are generally neglected. These simulations help in the understanding of environment effects as they can be easily controlled by an initial setup, or imposing thermodynamical constraints.

Given the positions of all of the atoms in a desired system, one can calculate the force exerted on each atom by all of the other atoms, this is achieved by the derivation of the potential energy U:

$$F(r^N) = \frac{\partial U(r^N)}{\partial r} \tag{2.36}$$

U is a parametric function of the nuclear coordinates obtained from the PES by the BOA principle. This parameters are stored in a Force Field (FF) that acts on the atoms without explicitly considering electronic degrees of freedom. This set of parameters can proceed from electronic structure calculations or empirical data.

A FF is a model of intra- and inter- molecular forces with contributions of stretching, opening-closing of bonds, and rotation of single bonds. Fig.2.4 and Eq.2.37 can exemplify the components of the model.

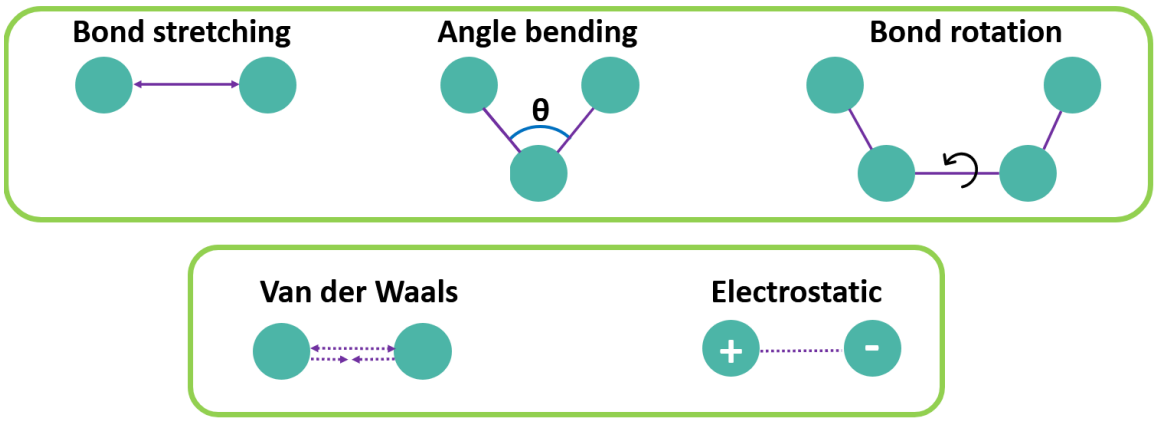


Figure 2.4.: Contributions to a force field model.
Top: Bonded contruibution.
Bottom: Non-bonded contributions.

$$U(r^N) = \underbrace{\sum_{\text{Bonds}} \frac{k_i}{2}(l_i - l_{i,0})^2}_{\text{Bond stretching}} + \underbrace{\sum_{\text{Angles}} \frac{k_i}{2}(\theta_i - \theta_{i,0})^2}_{\text{Angle bending}} + \underbrace{\sum_{\text{Torsions}} \frac{V_n}{2}(1 + cos(n\omega - \gamma))}_{\text{Bond rotation}}$$
$$+ \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\underbrace{4\varepsilon_{ij} \left[\overbrace{\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12}}^{\text{Attraction}} - \overbrace{\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}}^{\text{Repulsion}} \right]}_{\text{Van der Waals (Lennard-Jones)}} + \underbrace{\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}}_{\text{Electrostatic}} \right) \tag{2.37}$$

where

$U(r^N)$ = Potential energy as a function of the positions r of N atoms
k = Force constant[7]
l = Length
θ = Angle
V_n = Barrier of rotation
n = Multiplicity
ω = Torsional angle
γ = Phase factor
ε = Well depth
σ = Van der Waals radius
ε_0 = Dielectric constant
q = Charges

We can categorize force fields into five main groups:

- Fixed-Charge Atomistic: In this model the partial charges do not change depending on conformation and environment, this means is not polarizable.
- Polarizable: In this model the charge distribution changes on the basis of a dielectric environment.
- Coarse-grained: This model groups atoms into virtual particles. This process reduces significantly the computational costs as the number of interactions gets reduced.
- Reactive: In this model bond dynamics are calculated from interatomic distances.
- Machine learned: Ignores the formalism of Eq.2.37 and is based on ML-modelled functions that reproduce a PES from atomic coordinates.

FF in molecular modelling reproduce structural properties and are designed to predict certain properties and be parameterised accordingly. FF must be transferable. Transferability means the same set of parameters can be used to model a series of related molecules, rather than having to define a new set of parameters for each individual molecule.

[7]Both bond stretching and angle bending are modelled after Hooke's law.

The dynamics of the system is obtained by using propagation algorithms. A popular one is the Verlet integration[8] (Eq.2.38), after which one can obtain a qualitative understanding of the dynamics of a process or calculate quantitative results from correlation functions.

$$r(t + \Delta t) \approx 2r(t) - r(t - \Delta t) + \frac{F(t)}{m}\Delta t^2 \tag{2.38}$$

The result is a trajectory, a three-dimensional movie that describes the atomic-level configuration of the system at every point during the simulated time interval.

If the primary interest is in finding low-energy states[9], then the dynamics are less relevant, and one of many optimization methods is used to find the ground state of the system.

The time-average of an MD simulation, according to the *ergodic* hypothesis, approaches the thermodynamic ensemble average.

A system's thermodynamical properties can be represented by averaging over all possible quantum states in accordance to the statistical mechanics[10] principles. An ensemble is assumed as an imaginary collection of a very large number of systems[11] in different quantum states with common macroscopic attributes.

Each system of the ensemble must have the same physical property as the real system it represents, which includes temperature (T), pressure (P), number of molecules (N), volume (V) and so on. The ensembles are distinguished by which thermodynamic variables are held constant over the course of the simulation:

Thermostats and barostats help in controlling these variables.

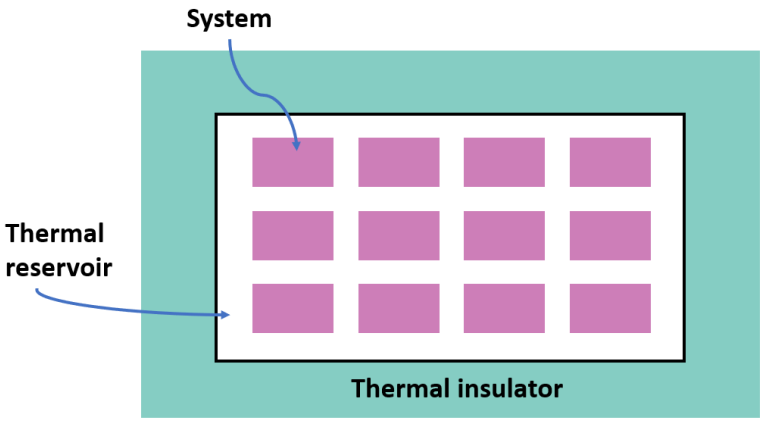


Figure 2.5.: Ensemble of systems in thermal equilibrium with a heat reservoir.

[8] As the estimation error increases with the time step interval, a variation of this algorithm –known as Velocity Verlet– introduces the particle's velocity to reduce this error to the order of t^4.

[9] If we go back to Fig.2.1, this would correspond to the structures on the well of minimum energy.

[10] Statistical mechanics can be seen as a bridge between the microscopic and macroscopic worlds.

[11] As this is an imaginary collection, we can choose the number of systems according to our convenience.

2.2.1. Quantum Mechanics/Molecular Mechanics

Some kind of simulations however require the introduction of quantum mechanical effects for a proper description. One example is the study of chemical reactions involving covalent bonds. For achieving this modelling Quantum Mechanics/Molecular Mechanics (QM/MM) simulations are required. In these systems a small part of the system is modeled using quantum mechanical calculations and the remainder by MD simulation (Fig.2.6).

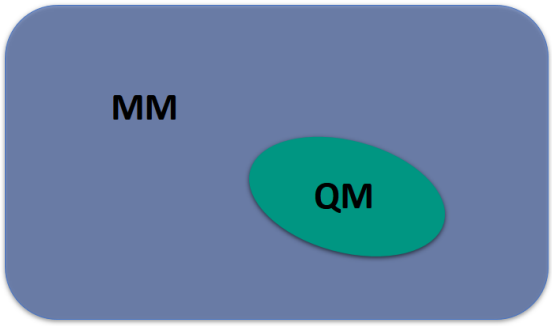


Figure 2.6.: Partitioning of the system into QM and MM subsystems. The shadowing around the QM zone represents the boundary region.

Owing to the (strong) QM–MM interactions, the total energy of the entire system cannot simply be written as the sum of the energies of the subsystems. In this case the Hamiltonian of a system is defined by Eq.2.39, where $\hat{H}_{QM(MM)}$ is the energy of the QM region, $\hat{H}_{MM(MM)}$ the energy of the MM region, and $\hat{H}_{QM/MM(QM-MM)}$ the coupling between the two regions. This last term is crucial and includes the bonded, electrostatic and van der Waals interactions between the atoms in the two regions. Special precautions need to be taken at the boundary between the subsystems, especially if it cuts through covalent bonds. Depending on the type of QM/MM scheme, the boundary region may contain additional atoms (link atoms) that cap the QM subsystem and are not part of the entire system, or it may consist of atoms with special features that appear in both the QM and the MM calculation.

$$\hat{H}_{QM/MM} = \hat{H}_{QM(MM)} + \hat{H}_{MM(MM)} + \hat{H}_{QM/MM(QM-MM)} \tag{2.39}$$

One of the major challenges inherent to all QM/MM approaches is the high computational cost needed for the repeated evaluation of the energies and forces in the QM region. Additionally, due to the large number of atoms included in these calculations, the number of local minima regions increases.

The QM code must be able to perform the SCF treatment in the presence of the external point-charge field that represents the MM charge model in the case of electronic or polarized embedding. Many current QM/MM applications use DFT or semiempirical QM methods. The DFTB method is increasingly being applied in biomolecular QM/MM studies.

2.3. Machine Learning

John McCarthy, one of the pioneers of Artificial Intelligence (AI), defines it as *"The science and engineering of making intelligent machines, especially intelligent computer programs"*.

The AI's target is to enable machines with the ability to conceptualize and create abstractions. For fulling this target, the field comprises of different concepts and tools such as Machine Learning (ML) and inside this, Deep Learning (DL) (Fig.2.7).

ML is a subset of AI and its goal is to automate analytical model building. While ML is based on the idea that machines should be able to learn and adapt through experience, AI refers to a broader idea where machines can execute tasks by simulating human capability and behavior, such as decision taking.

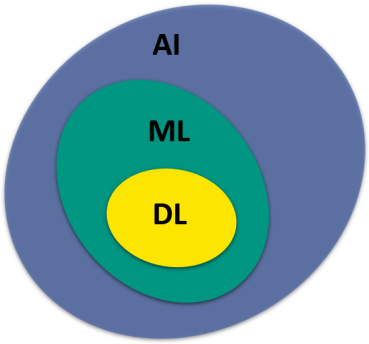


Figure 2.7.: Hierarchy of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL).

As a general rule, problems can be described by a list of formal and mathematical rules. Computers have proven their capability and helpfulness at solving logic, algebra, geometry, and optimization problems through "traditional programming", ie. when we provide then a function to solve. Yet, some of the problems have not an interpretation or function to predict a result. For those problems the exact mathematical formulation is not yet known, and ML has positioned itself as a tool to address the problems.

The main difference between ML and traditional programming is based on the inputs and desired outputs (Fig.2.8). In traditional programming we want a computer to provide a result based on an input and a program, in ML instead we provide the input and the result and expect to gain a program as an output.

The tasks that ML can solve can be divided between regression and classification (Fig.2.9). Regression tasks assume an immediate relationship between two variables x and y, which are often of deterministic nature. The objective of regression is to find a function f that yields the mapping as seen in Eq.2.40. In the tasks of classification, the goal is to assign discrete class labels to examples. In contrast to regression, we are optimizing a model to find a mapping from an input vector $\vec{x}$ to a target y, which encodes a representation of the different possible classes.

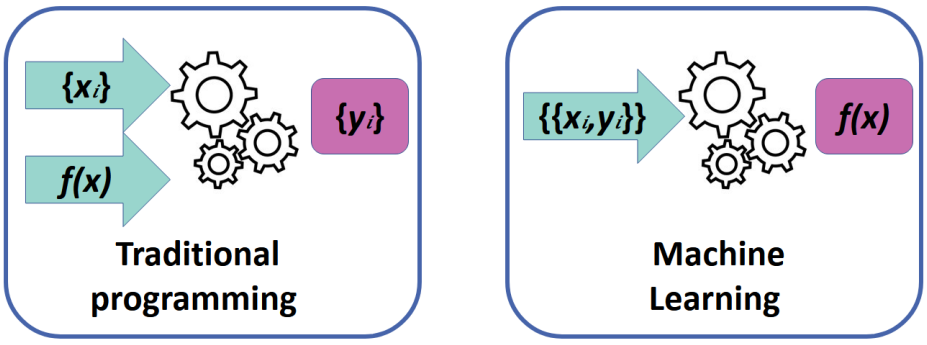


Figure 2.8.: Comparison between traditional programming and machine learning paradigms.

$$y = f(x) \tag{2.40}$$

Certain tasks can only be solved if sufficient data is available and, representative data allows the learning done with seemingly low effort. We usually refer to the data in terms of a data set $\mathcal{D}$, containing a finite amount of data instances often called data points, which may be represented as $\mathcal{D} = \{\mathbf{x}_i\}$ or may be accompanied by predefined labels, $\mathcal{D} = \{\{\mathbf{x}_i, \mathbf{y}_i\}\}$.

The field of ML leverages fundamental concepts of applied statistics, emphasizing the use of computers to estimate complicated functions and with a decreased emphasis on proving confidence intervals around them. From a practical point of view, we can neither optimize over the set of all possible functions nor over the full domain of $\vec{x}$. Instead, we resort to a finite data set for which we opt to find a model that maps every input $\vec{x}$ to its corresponding target y. Usually, the model is predefined up to some parameters.

Learning schemes can be divided in the following types and all follow the same parametrization in Eq.2.41 as a model for Eq.2.40:

$$f(x) = \theta^T x + b \tag{2.41}$$

- Supervised: Can be seen as a generalized notion of regression and classification, and learns from labeled data, $\mathcal{D} = \{\{\mathbf{x}_i, \mathbf{y}_i\}\}$.
- Unsupervised: Can be used for preliminary preprocessing steps, such as change of dimensionality, or representation learning.
- Reinforcement: The training is based on rewarding desired behaviors and/or punishing undesired ones. It is employed to find the best possible behavior or path it should take in a specific situation.
- Active: Includes selection strategies that allows an iterative construction of a data set in interaction with a human expert or environment. Its aim is to select the most informative examples and minimize the cost of labeling.

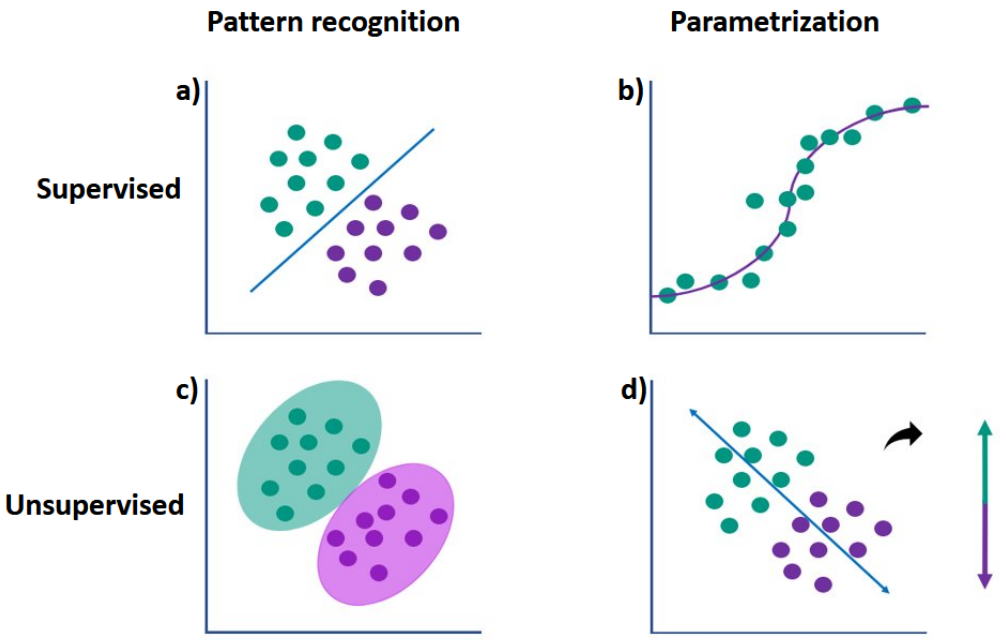


Figure 2.9.: Types of learning and tasks in ML.
a) Classification
b) Regression
c) Clustering
d) Dimensionality reduction

ML can be treated as an optimization problem– the mentioned learning schemes have the same underlying process of learning: finding an optimal model $\hat{f} \equiv f_{\theta^*}$ with optimal parameters θ^* in the hypothesis space, which minimizes a target loss function, $\mathcal{L}$, and maximizes a model's performance. A loss function compares a model predictions or a developed solution against the expected outcome.

Once we choose a loss function, we can minimize it by varying the parameters of the ML model, using any optimization method of our choice. In general, we can reach the minimum of the loss function either via analytical construction or optimization methods that can be either gradient-based or gradient-free. The optimization usually starts in a random place within the loss landscape (meaning with a model with randomly initialized parameters, $\theta = \theta_0$). Using the model with θ_0, one makes predictions over the training data and from here, computes the loss function.

The next step consists of computing the gradients of the loss function with respect to each model parameter, θ_j. This step is typically done by automatic differentiation.

Then the parameters are updated by subtracting the respective gradients multiplied by a learning rate η, see Eq.2.42.

$$\theta_j := \theta_j - \eta \frac{\partial \mathcal{L}}{\partial \theta_j} \tag{2.42}$$

These steps need to be repeated until the minimum is reached, and each repetition is called an epoch. This gradient descent updates model parameters by making steps toward the minimum of the function, and the learning rate controls the size of these steps.

These parameters that control the learning process such as quality and speed of convergence are learning–hyperparameters (number of epochs, learning rate) and are fixed by the user.

Additionally to the setup of learning, a good practice to find optimal parameters is to split the data into training, validation and test sets. A visualization of the data splitting can be seen at Fig.2.10.

Figure 2.10.: $\mathcal{D}$ splitting. Following a typical ratio of 80/10/10 of training/validation/test

The training data gives the examples from where the ML identify patterns to obtain the desired outcome.

The validation data is only used to validate the model (not used in the training) by adjusting the model parameters in a way that the error of the validation set is minimized.

The test set is composed of data points that are used neither for learning or optimization of model parameters. As the generalization of the ML relies on making accurate predictions on new data, this set will help in the evaluation of the accuracy of the model.

2.3.1. Neural networks

Neural Network (NN) are general function approximators that can be trained via labelled data. The goal is to approximate Eq.2.40 as well as possible by choosing suitable parameters in the hypothesis space θ. In other words we can view artificial neural networks as a multiparamater curve fitting.

The basic unit of an artificial neural network is the neuron, which holds a scalar value. Its value y is obtained starting from the values a_k of some other neurons that feed into it.

First, a linear function of those values gets calculated as in Eq.2.43, where w_k are weights and b is a bias. Then a nonlinear and differentiable activation function is applied to yield the neurons value, $y = f(z)$. This arrangement already constitutes a single layer neural network, also known as perceptron.

$$z = \sum_k w_k a_k + b \tag{2.43}$$

It has been demonstrated by the Arnold–Kolmogorov representation theorem[95] that any arbitrary continuous high-dimensional function can be expressed as a linear combination of a set of nonlinear functions [12]. Therefore, multiple layers of neurons are needed.

[12]This property is the base for their applicability to the construction of PESs. The BOA states that the energy of a system can be given as a parametric function of the nuclear positions.

Each neuron receives the values of all the neurons in the preceding layer, with suitable weights.

A multi-layered NN can be seen in Fig.2.11. The first layer in a fully-connected NN is the input layer, where the activations of its nodes are set according to the vector x encoding the input data. The last layer is called the output layer and the activations of its nodes constitute the output of the NN. All intermediate layers are called hidden layers. To keep track of the notation of these multi-layered cases, some extra indexes will be introduced. $a_k^{(n)}$ will be the value (also known as output activation) of the neuron k in the n^{th} layer, the weight $w_{jk}^{(n+1)}$ will show how much the neurons k in the n^{th} layer will affect neuron j in the layer $n+1$. The biases are a constant offset of values b_j^{n+1}. The output of the entire neural network is obtained by going through Eq.2.44 layer by layer, staring at the input layer ($n = 0$).

The number of layers, nodes, and their connections is known as the architecture of a NN.

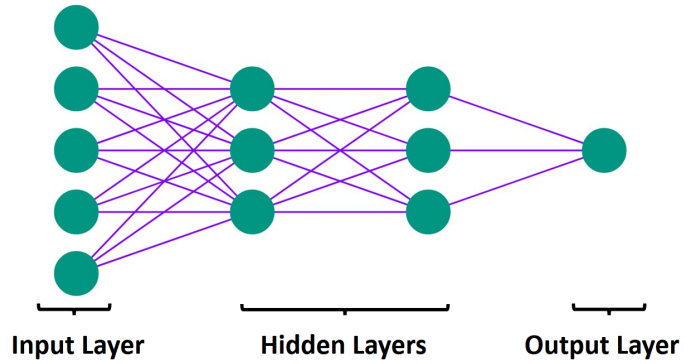


Figure 2.11.: Basic architecture of a multi-layered feed-forward Neural Network.

$$z_j^{n+1} = \sum_k w_{jk}^{(n+1)} a_k^{(n)} + b_j^{(n+1)} \tag{2.44}$$
$$a_j^{(n+1)} = f(z_j^{(n+1)})$$

The activation function, f, is usually kept fixed. NNs where each node is by default connected to all nodes in the subsequent layer are referred to as fully connected. NNs are considered deep if they are composed of many hidden layers.

All the weights w and biases b form the parameters of the network. They will be updated during training by gradient-based methods.

Backpropagation is a reverse mode of automatic differentiation to calculate these gradients, based on Eq.2.42. Backpropagation computes the partial derivatives $\frac{\partial \mathcal{L}}{\partial w}$ and $\frac{\partial \mathcal{L}}{\partial b}$ of the loss function with respect to any weight w or bias b in the network.

It starts from the output layer and propagates backwards due to the fact that the loss function is based on the outputs of a NN; i.e, $\mathcal{L} = \mathcal{L}(a^L)$.

The algorithm of backpropagation[13] for a NN states:

[13]Mathematical proof for the output errors can be found in Appendix A.2.

Algorithm 1: Backpropagation

Input Training data;
foreach x **do**
  Set the corresponding activation a^l for the input layer;
  foreach *layer* $l = 2, 3, ..L$ **do**
    $z^l = w^l a^{l-1} + b^l$;
    $a^l = f(z^l)$;
  end
  Compute Output error δ^L;
  $\delta^L = \nabla_a \mathcal{L} \odot f'(z^L)$;
  Backpropagate the error;
  foreach *layer* $l = L-1, L-2, ..2$ **do**
    $\delta^l = (\delta^{l+1}(w^{l+1})^T) \odot f'(z^l)$;
  end
end
Output ;
$\frac{\partial \mathcal{L}}{\partial w^1_{jk}} = a^{l-1}_k \delta^l_j$;
$\frac{\partial \mathcal{L}}{\partial b^l_j} = \delta^l_j$;

A summary of a NN training can be seen in Fig.2.12.

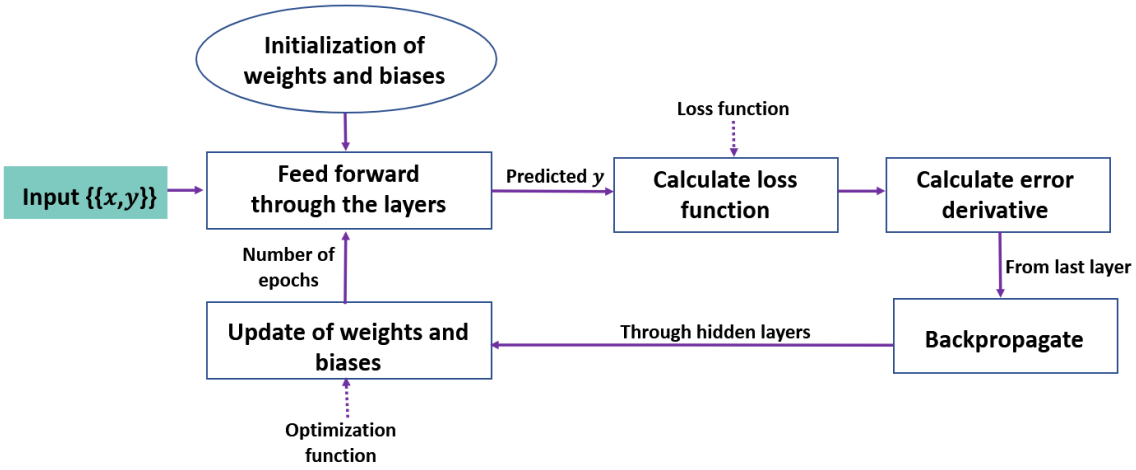


Figure 2.12.: Steps for the training of a Neural Network using the Backpropagation.

3. Case of study

Compounds containing a disulfide bridges are able to undergo disulfide exchange reactions with free thiols, this constitutes a two-electron oxidation process[1]. The thiol-disulfide exchange is an S_N2 reaction, which mechanism can be observed in Fig.3.1.

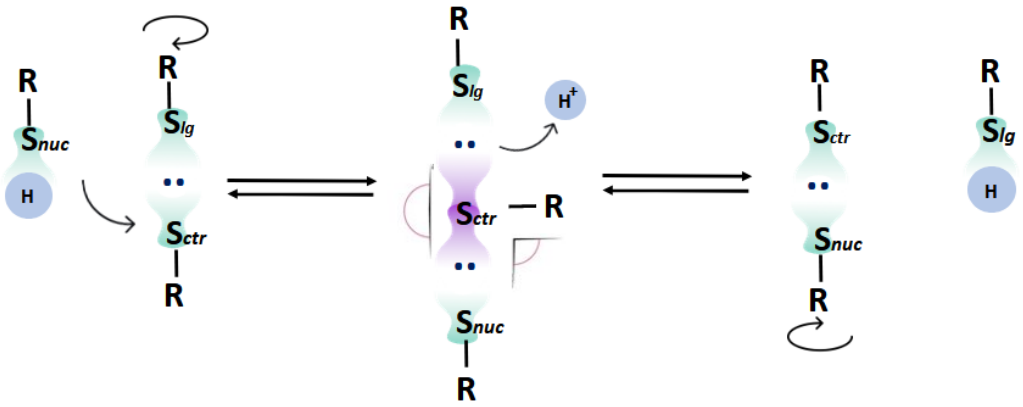


Figure 3.1.: Thiol disulfide exchange mechanism.

A nucleophile (S_{nuc}) attacks the substrate (S_{ctr}), and a leaving group departs simultaneously (S_{lg}). The substrate and the nucleophile are both present in the transition state for this step. Because two molecules are present in the Transition state (TS), the reaction is bimolecular, as indicated by the number "2" in the S_N2 symbol.

An S_N2 reaction usually requires a trigonal bipyramidal transition state with the entering and leaving groups in apical positions and substituents at the central atom in an angle of approx. 90°. As the cleavage of a disulfide bond is thought to occur without essential participation of 3d orbitals, a linear orientation *seems* to be valid for sulfur atoms.

We know that the chemistry of a disulfide exchange is directly influenced by the following key factors:

- Accessibility: Physical proximity to partner thiol groups.
- Environment:

[1]Oxidation refers to loss of electron density while reduction implies a gain on electron density.

- Increased reactivity: High pH and oxidizing environment[2].
- Decreased reactivity: Low pH and reducing environment[3].

However, new insights in quantum chemistry show with better precision how does this factors impact the bond formation and exchange. A detailed study, benchmarking 92 density functionals for the thiolate-disulfide exchange between a methylthiolate and a dimethyldisulfide, was presented by Neves et al.[82]. A thiolate is the deprotonated state of a thiol (Fig.3.2).

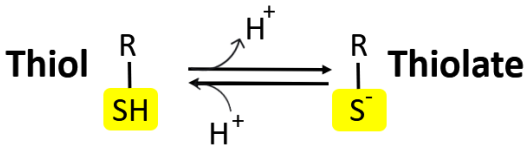


Figure 3.2.: Deprotonation of a thiol forms a thiolate.

In a polar environment, the charge is more localized. Consequently, the thiolate and the disulfide states are stabilized whereas the trisulfide state is the transition state. While the reference method MP2/aug-cc-pVTZ yielded a linear structure with S–S bond lengths of 2.40 and 2.42 Å, both LDA and GGA functionals showed deviations from linearity and significantly longer bonds.

The symmetric conformation becomes a transition state with a significant barrier: CCSD(T)/aug-cc-pVTZ shows a barrier of 9.28 kcal/mol, which decreases with decreasing basis set size and level of theory to 6.24 kcal/mol with MP2/aug-cc-pVDZ.

These energy estimates are based on continuum solvent models, which yield substantial contributions to the thermal free energy and zero-point energy of ≈8 kcal/mol. Such free energy contributions in solution can be taken into account in a straightforward manner by sampling the configurational space using MD simulations.

In gas phase, theoretical evidence also sustain this linear trisulfide-like transition state, with the negative charge being delocalized, but most abundant on the attacking and leaving sulfurs[10, 37].

It has also been observed that potential energy differences (energy minimas) depend sensitively on the level of theory and the basis set size. CCSD(T)/aug-cc-pVTZ yields an energy difference of $\Delta E = -2.13$ kcal/mol in the gas phase, while $\Delta E = -5.05$ kcal/mol with MP2/aug-cc-pVDZ, and $\Delta E = -5.65$ kcal/mol with B3LYP/TZVP. An error of up to 10 kcal/mol was reported for the LDA functionals, while the GGAs still show errors of more than 5 kcal/mol (e.g. 6.7 kcal/mol with PBE). Including the exact exchange improves the situation, however the error of the widely used B3LYP functional is still ≈3 kcal/mol.

A proper inclusion of electron correlation is crucial for this reaction. Fernandes' work on the description of Thiol–Disulfide Exchange states that the quality of the CCSD(T)/CBS

[2]This enviornment is found outside the cell. Exogenous factors such as smoke, xenobiotics, radiation and other sources may generate Reactive oxygen species (ROS) to diffuse into the cell or induce endogenous ROS production aftecting the natural homeostasis of the cell (oxidative damage).

[3]Glutathione is the main antioxidant inside the cells and helps in buffering this reducing environment.

energy is the most adequate to be taken as a reference for describing its energetic landscape [82, 10].

To study the reaction, QM/MM approaches can be applied. However, when using *ab initio* approaches for the QM region, sampling is usually restricted and mostly done by searching reaction pathways. As an alternative, semiempirical methods have been attempted for describing the reaction and it has been observed that DFTB overestimated distances in the transition state[87], this correlates with the behavior of the DFT-LDA and DFT-GGA approaches [82]. Being based on the PBE functional, DFTB thus seems to reproduce the DFT-PBE errors.

4. Methodology

4.1. Molecular descriptor

A ML algorithm only processes numbers, any change on the input will lead to a different output. When positioning ourselves on a cartesian coordinate system, translating or rotating a rigid molecule in vacuum does not change the relative atomic positions and consequently the energy remains invariant although the numerical values of the cartesian coordinates change.

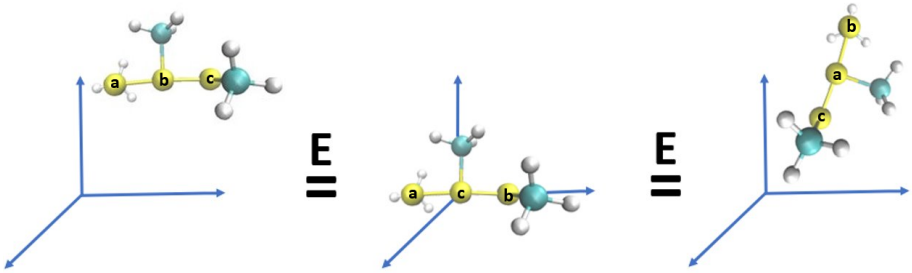


Figure 4.1.: The three configurations are energetically equivalent independent of permutation, translation or rotation.

Cartesian coordinates are not suitable as the $\{x_i\}$ input for the ML, then is important to transform the chemical structural information into a a set of parameters invariant to the aforementioned changes.

The Atom-Centered Symmetry Functions (ACSF) were the first type of descriptors developed for construction of high-dimensional ML potentials [17, 15, 14]. As mentioned previously, the PES is an N-dimensional space function, and the ACSF can give a correct description of it as they are based on a transformation of nuclear coordinates. They are continuous, therefore they can be analytically differentiable to calculate forces, as stated in Eq.2.36.

The ACSF follow the hypothesis that large part of the atomic interactions of the atoms are described by interactions of the atoms with their local chemical environments, these describe the positions of atoms in the environment of a given central atom based on distances R_{ij} and angles θ_{ijk} of the neighboring atoms up to a cut-off distance R_c.

R_c represents a convergence parameter that include all energetically relevant interactions needed to reach the desired level of accuracy. The use of a cut-off effectively reduces the dimensionality of the atomic environments to the positions of the close atoms enabling the use of the energy expression in Eq. 4.1, that states that the total energy of a system is the sum of all the energetic contributions of each atom.

$$E = \sum_{i=1}^{N_{\text{atoms}}} E_i \tag{4.1}$$

ACSF include radial and angular contributions.

The radial function is described by Eq.4.2 and is a sum of Gaussians that depend on interatomic distances R_{ij}. The center of the Gaussians can be shifted by the parameter $R_{s,m}$ that describe a spherical shell around the reference atom. The width of the Gaussians is controlled by the parameter η. Typically, a set of η values is used to obtain a complete radial fingerprint of the environment.

$$G_i^{\text{rad}} = \sum_{\text{atoms } j \neq i}^{\text{within } R_c} e^{-\eta(R_{ij}-R_{s,m})^2} \tag{4.2}$$

Due to the summation over all neighbors in the cutoff spheres, the number of ACSF is independent of the coordination of each atom i, which is important since NNs require a constant number of input descriptors irrespective of the coordination number, which might change frequently in MD simulations.

The angular function (Eq.4.3) provides an angular fingerprint of the environment using the angles θ_{ijk} formed by atoms i, j and k. The angular resolution can be controlled by a set of ζ values, while $\lambda = \pm 1$ defines the positions of the extrema of the cosine function. The angular functions must be constructed for all possible element combinations involved in the angles θ_{ijk}.

$$G_i^{\text{ang}} = \sum_{\text{atoms } j,k \neq i}^{\text{within } R_c} \frac{2^{1-\zeta}(1+\lambda\cos\theta_{ijk})^{\zeta}}{e^{\eta(R_{ij}^2+R_{jk}^2+R_{ik}^2)}} \tag{4.3}$$

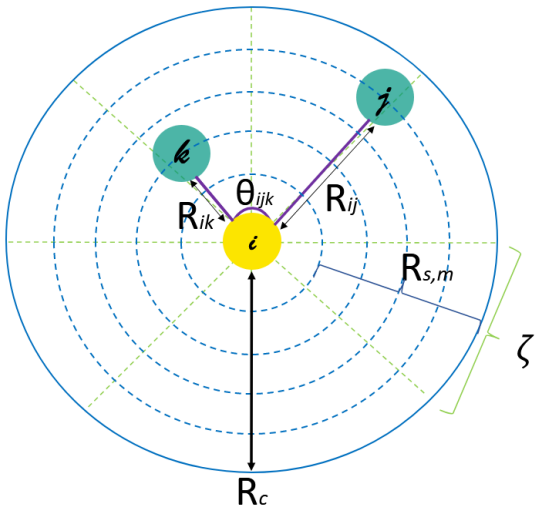


Figure 4.2.: Toy model centered around atom i, showing the ACSF parameters.

The set of function values is different for each atomic environment and the number of symmetry functions is larger than the number of degrees of freedom of the system.

4.2. Optimization of the molecular descriptor

It is important that the ACSF include all the possible energetic contributions of the system, therefore a proper selection of the ζ, $R_{s,m}$, R_c and η values for the symmetry functions will be determinant for the accuracy of the NN.

A Geneteic Algorithm (GA) is a search-based algorithm used for solving optimization problems in ML. They operate on sets of strings which evolve through generations according to the rule of natural selection. This evolution is the method for reaching the optimal solution of a problem by excluding the worse set of strings based on a fitness function[77].

The initial strings are known as "population", each set of solutions is called "chromosome", the chromosome is composed of "genes", and the "genes" stand for the parameters to be optimized, a graphical explanation can be seen at Fig.4.3. In every generation, a new set of strings is created, using parts of the best members of the old set. This step, known as crossover, carries a mutation which is a random alteration of the value of a string's position, see Fig.4.4.

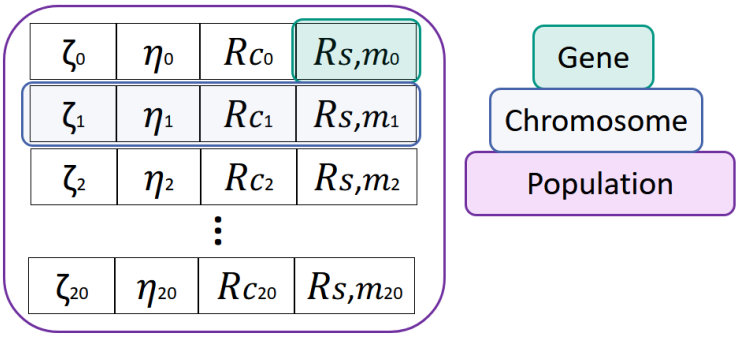


Figure 4.3.: ACSF hyperparameters as the genes of a GA.

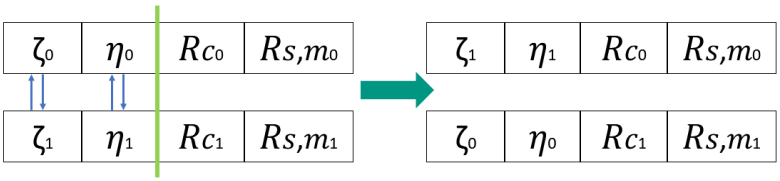


Figure 4.4.: Example of crossover in the chromosomes to find the optimal parameters for the ACSF.

4.3. Artificial neural network for Δ-learning

This work uses the Δ-ML approach by Ramakrishnan et al.[89], following also the work by Zhu et al.[115] and Shen et al.[98] on learning the energy difference between two levels of theory of quantum chemical calculations,

$$\Delta E = E_{\text{ab initio}} - E_{\text{DFTB}} \tag{4.4}$$

with a Behler–Parrinello NN[17].

The aim was to correct the DFTB level of theory to *ab initio* accuracy, and implement the correction into the DFTB+ software[60], following the principle of Eq. 4.4.

As seen on the 2nd generation potentials, when using the Behler–Parrinello NN, the total energy of a molecular system is expressed as the sum of energy contributions E_i. Each energy contribution gets assigned to all of the atoms separately, as seen in Eq.4.1, where the E_i are predicted by individual NNs using the concept of Eq. 2.44.

In the Δ-learning, the quantity to be predicted by the NN is the difference of energies obtained with two quantum chemical methods, as in Eq. 4.4. The training set consisted of a set of molecular structures, for each of which the energy is calculated with both DFTB and a reference *ab initio* method.

The implemented feed-forward NN consists of a two-layer sub-network for each atom with the *tanh* activation function. The descriptors are defined by Eq. 4.2 and Eq. 4.3 as the input parameter of Eq. 2.43. Each hidden layer consists of 15 neurons whose weights have been initialized by the Nguyen–Widrow initialization procedure[6] (see Appendix.A.3).

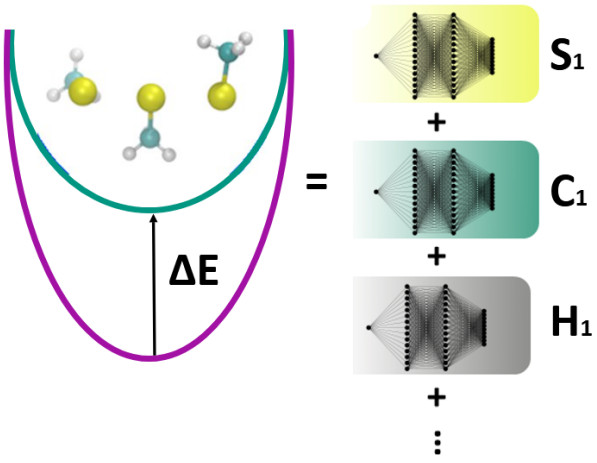


Figure 4.5.: Representation of the Behler-Parrinello NN.

4.3.1. Implementation into a workflow

The main goal of all major workflow frameworks is to capture the elements of a complex protocol and automate its execution, saving time by directing properly the information flow.

The present workflow idea is to have a semiautomatic process where the end-user under his expertise criteria can select the best sampling chemical structures for the Δ-ML dataset. The end-user will then provide the structures to the workflow and since this point the process of data preparation and training can be automated.

The purpose of using a GUI is facilitating the use of the Δ-ML code for non-Python experts directly from an interfaced local machine to a High Performance Computing (HPC) environment. The selected platform is SimStack[91], developed in a joint project by Nanomatch GmbH and KIT.

SimStack provides a highly flexible drag-and-drop environment that allows quick adaption of existing workflows, as well as a GUI for the end-user to automatically execute the workflows on remote computational resources. Multiple modules are connected into complex workflows via dragdrop and relevant simulation parameters are set for each model using automatically generated outputs. The SimStack Client connects to a SimStack Server installed on remote computational resources and handles file transfer, submission and monitoring of workflows and retrieval of results to the end-user's personal computer.

A Workflow Active Node (WaNo) is a module that SimStack interprets and uses to render the information of the node from the local machine to the available HPC resources. The WaNos are specified on an Extensivle Markup Language (XML) and use as templating engine Jinja language.

In Fig.4.6 the HPC parameters that can be adjusted can be seen.

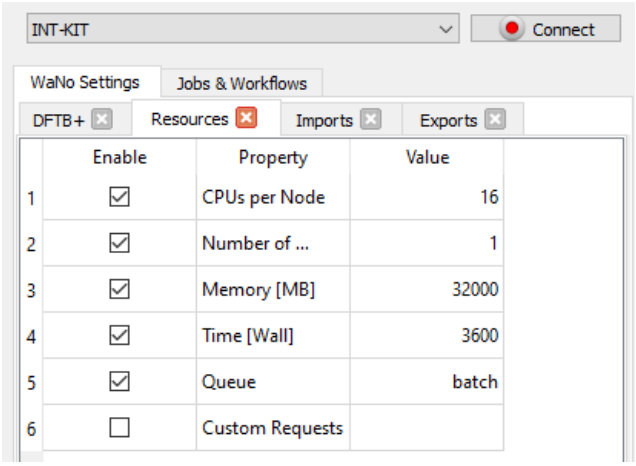


Figure 4.6.: The user can select HPC resources according to availability. resources can be easily optimized by the user.

The developed workflow allows to run DFTB+[8] and ORCA[79] calculations on parallel computing resources[1].

The WaNo modules were designed to expose charge and multiplicity as they were not arbitrary parameters in our system. The ORCA module can be seen at Fig.4.7, the geometry

[1]There is also the possibility to run the *ab initio* calculations on a previously developed Turbomole[44] WaNo, but we will not discuss that software in this application.

file can come from local files or parsed form another WaNo, all available functionals[2] and basis-sets from ORCA were implemented on a scroll-down menu. Charge and multiplicity can be selected manually, however a warning for selecting wrong multiplicity has been implemented and automatically redefines it.

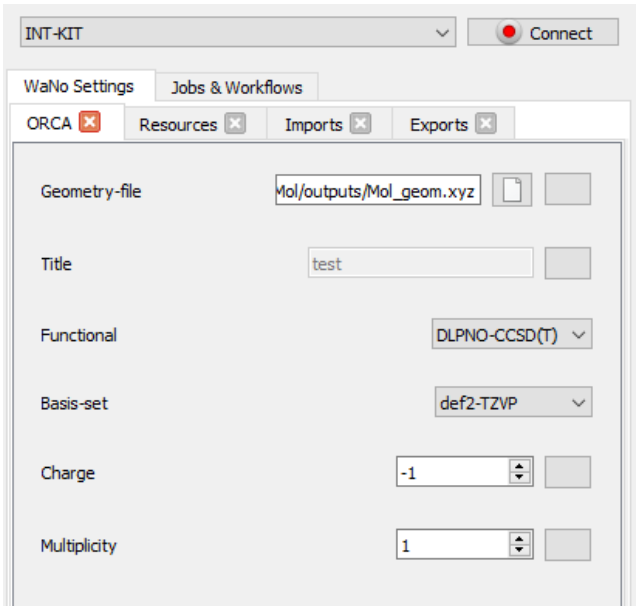


Figure 4.7.: WaNo for ORCA software

In the case of DFTB+ WaNo, Fig.4.8, just as the ORCA module, the geometry file can come from local files or parsed from another WaNo, charges and multiplicity are defined analogously to the ones in the ORCA WaNo. Other exposed parameters are the selection of Slater-Kloster parameters from a scroll-down menu and number of SCC iterations. Different types of calculations were also implemented:

- Single shot calculation
- Structure optimization
  - Steepest Descent
  - Conjugate Gradient
  - gDIIS
- Molecular Dynamics with the thermostats:
  - Berendsen
  - Andersen
  - Nosé-Hoover

[2]The DLPNO-CCSD(T) approximation[72] was implemented too and uses the same basis-set as its complimentary.

  - No thermostat

- Single shot calculation with the machine learning correction[3].
- Molecular dynamics with the machine learning correction.

For the reference values, we use the energy of infinitely separated reactants. These energies are calculated by a small independent workflow, which can be seen in Fig.4.9. The output of this workflow will give a chart (suitable for its use in the "NN-Delta-ML" WaNo) with the energy values of each structure

The "NN-Delta-ML" WaNo trains and creates a model suitable for DFTB+. The exposed parameters can be seen in Fig.4.10. The cut-off radius, the number of neurons, layers and learning rate take positive integers while the activation function can be selected from a scroll-down menu. After completion of the training, this module will export a graphical report (Fig.4.11) with the histogram of the ΔE, information about the range of correction, a correlation plot of the accuracy on the test data and report of RMSE–MAE of the test data. The graphical output as well as the datasets for its creation are available to be saved locally. Information such as reference energy and error information are saved into a text file.

The workflow for the Δ-correction can be seen at Fig.4.12.

1. The geometries selected under the criterion of the user are given to the workflow as a compressed file (tar, tar.gz...).
2. The geometries get extracted and go into a parallel calculation loop, for a semiempirical method by DFTB and an *ab intio* by ORCA.
3. The calculation outputs get passed to "Table-generator" that cleans the data and creates a table with the associated energy of each structure.
4. "Super XYZ" rearrange the table ensuring that the index between structure, DFTB+ and ORCA energies match, this step will be important for the Neural Network module.
5. "NN-Delta-ML" trains and creates a model suitable for DFTB+.
6. For verifying the accuracy of forces, the DFTB+ WaNo can perform an NVE simulation importing the model and creating automatically the input file without the need of the user intervention outside the MD exposed parameters.

This workflow is expected to expand the systems where the Δ-ML can be used, and accelerate its application by acting as a bridge enabling non-experts to access information without the complication of selecting computing resources or numerical parameters, and saving time from data preparation and plot generation. The code and documentation for the workflow be found in Ref.[90].

[3]The use of this requires the DFTB+ branch for machine learning correction[67].

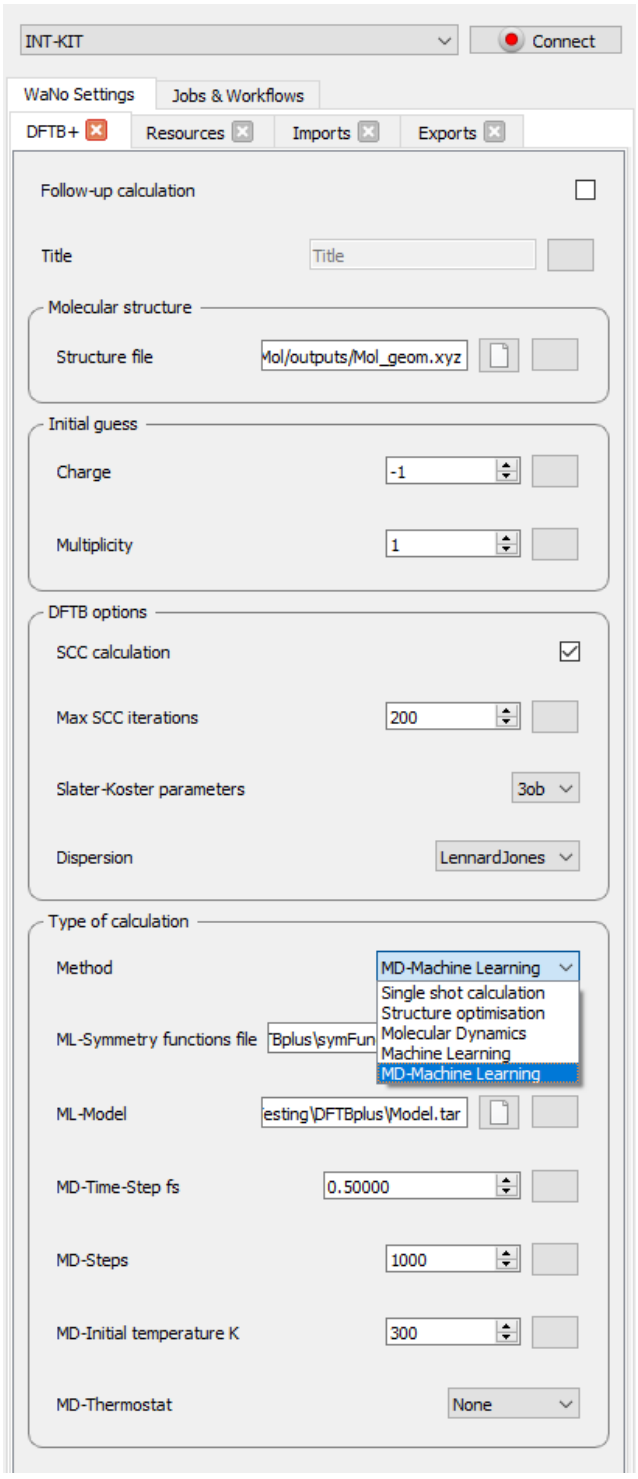


Figure 4.8.: WaNo for DFTB+

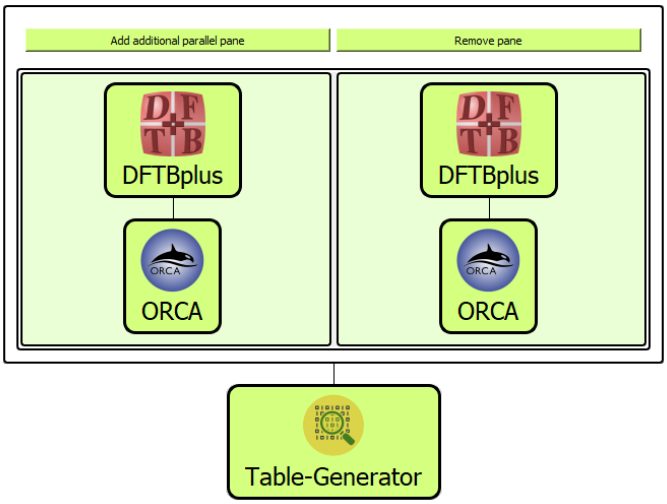


Figure 4.9.: Refence energies workflow.

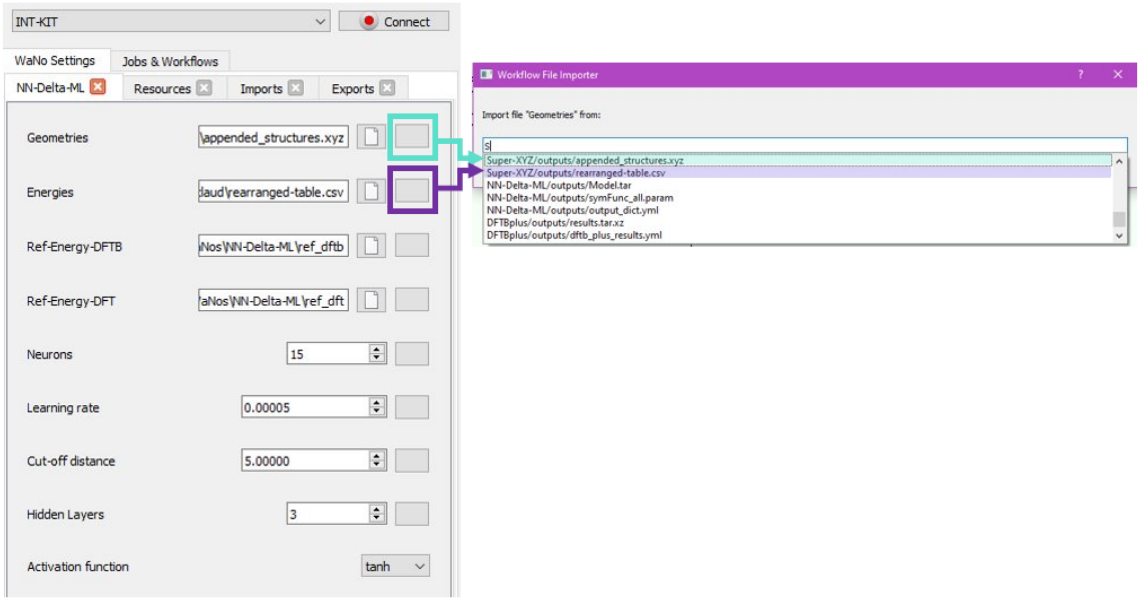


Figure 4.10.: WaNo for the Δ-ML.

python

Learning report

The data distribution can be observed at Fig.1, the training data had a range from 122.27087 to 193.00804 kcal/mol and the test data had ranged on 129.22800 to 192.61630 kcal/mol.:

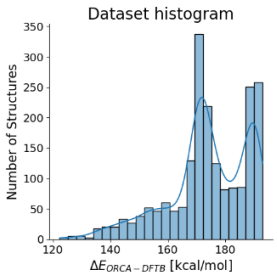
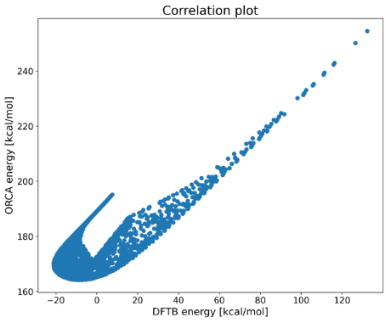


Fig.1 - Data distribution

The corelation between the ORCA generated data against the DFTB+ generated data can be seen in Fig. 2, while the delta correlation can be seen at Fig. 3. The energies used as reference were:
-9930.99694 kcal/mol for DFTB
-824842.30734 kcal/mol for DFT



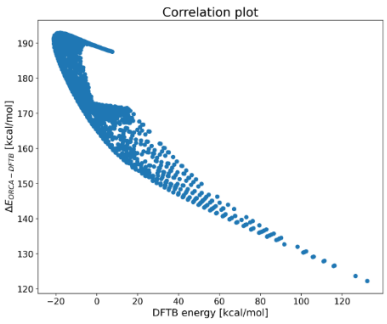


Fig.3 - Delta correlation

The used neural network consisted on 15 neurons distributed in 3 hidden layers for each element of the system. The activation function was tanh. The input layer consisted on 99 symmetry vectors for each one of the atoms. The data set was divided in Training, Validation and Test with its respective number of structures: 1600, 200, 200. The RMSE of the training was 0.8144 kcal/mol and the MAE was of 0.6321 kcal/mol. The accuarcy plot of the test set can be seen at Fig.4.

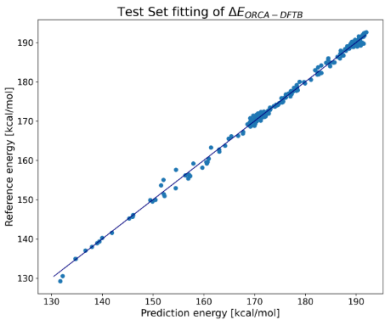


Fig. 4 - Test Set

Figure 4.11.: Workflow learning report.

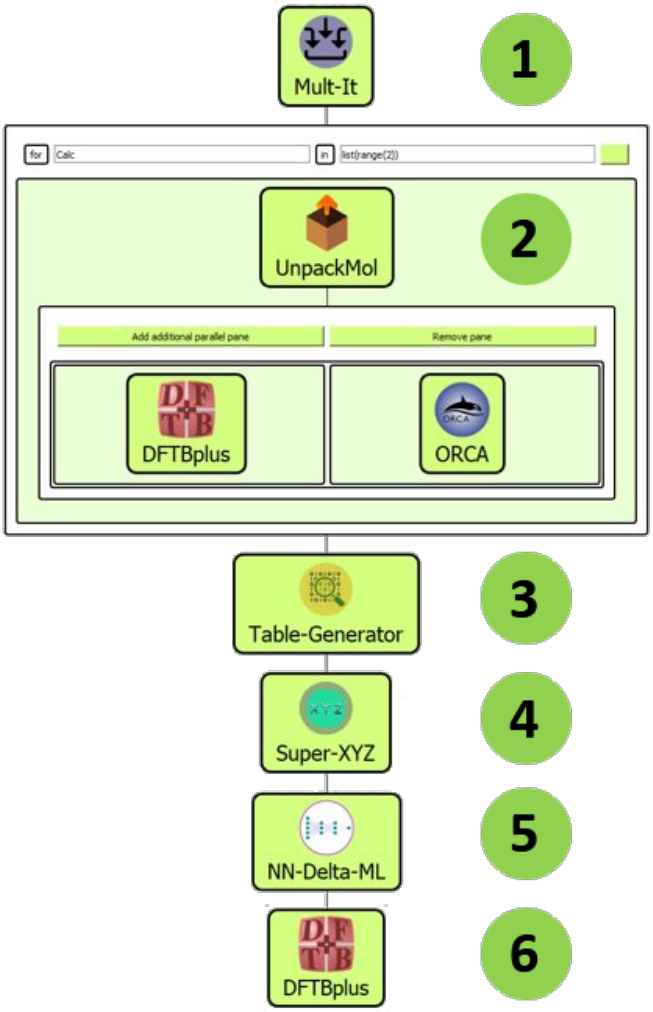


Figure 4.12.: Δ-ML workflow.

4.4. Artificial neural network for learning a fourth generation potential

Instead of limiting ourselves at the Δ-learning, the total $E_{\text{ab initio}}$ gets fitted by the same principles of the Behler-Parrinello NN.

To overcome the locality of the atomic energies, long-range charge transfer is obtained by a Charge Equilibration via Neural network Technique (CENT)[34] by employing environment-dependent atomic electronegativities. The total energy of the system gets defined by Eq.4.5, where the electrostatic part depends on a set of atomic charges $Q = \{Q_i\}$ and the positions of the atoms $\vec{r} = \{\vec{r}_i\}$.

$$E(\vec{r}, Q) = E_{\text{elec}}(\vec{r}, Q) + E_{\text{short}}(\vec{r}, Q) \tag{4.5}$$

Following the 4th generation potentials proposed by Behler-Parrinello[104]. Short-range atomic energies and electronegativities are expressed by NNs as a function of the chemical environments, employing the ACSF. The architecture is a binary system of two NNs, see Fig.4.13, the first one calculates the charges by the CENT and and gives provides them as an additional input to the short-energy NNs.

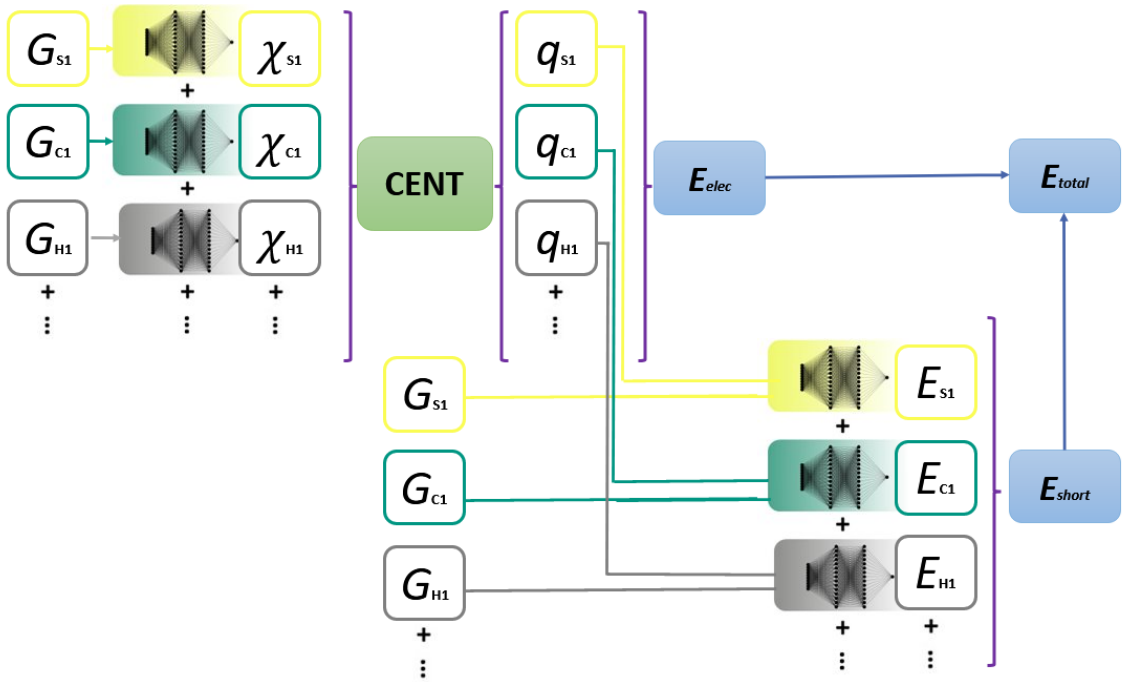


Figure 4.13.: Representation of the binary system of a fourth generation potential.

For the CENT part, a selected charge population distribution gets converted into electronegativities, χ_i, following the principle of Eq.4.6.

$$\sum_{j=1}^{N_{at}} A_{ij} Q_j + \chi_i = 0 \tag{4.6}$$

where elements of matrix **A** are given by Eq.4.7. J_i is an element-specific hardness, defined by the Hubbard parameters, λ is defined by Eq.4.8 and σ represents the covalent radii of the respective elements.

$$[A]_{ij} = \begin{cases} J_i + \frac{1}{\sigma_i \sqrt{\pi}} & \text{if } i = j \\ \frac{\operatorname{erf}(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}})}{r_{ij}} & \text{otherwise} \end{cases} \tag{4.7}$$

with

$$\gamma_{ij} = \sqrt{\sigma_i^2 + \sigma_j^2} \tag{4.8}$$

The electronegativities then get calculated by a feed-forward NN consisting of a five-layer sub-network for each atom with the *tanh* activation function. The descriptors are defined by Eq. 4.2, Eq. 4.3 and atomic number as the input parameter of Eq. 2.43. Each hidden layer consists of 45 neurons whose weights have been initialized by the Nguyen–Widrow initialization procedure.

After the loss minimization of the electronegativities, the calculation of charges is computed following the principle of Eq.4.6, considering that the sum of all charges must be equal to the total charge of the system, in the disulfide case $Q_{tot} = -1$ a constrain is added via Lagrange multipliers and solved as a system of linear equations:

$$\left(\begin{array}{ccc|c} & \mathbf{A} & & \begin{matrix}1\\:\\1\end{matrix} \\ \hline 1 & .. & 1 & 0 \end{array}\right)\left(\begin{array}{c} \begin{matrix}Q_1\\:\\Q_{N_{at}}\end{matrix} \\ \hline \lambda \end{array}\right) = \left(\begin{array}{c} \begin{matrix}-\chi_1\\:\\-\chi_{N_{at}}\end{matrix} \\ \hline Q_{tot} \end{array}\right) \tag{4.9}$$

The $E_{\text{ab initio}}$ gets calculated by a feed-forward NN consisting of a five-layer sub-network for each atom with the *tanh* activation function. The descriptors are defined by Eq. 4.2, Eq. 4.3 and the partial charge of each atoms as the input parameter of Eq. 2.43. Each hidden layer consists of 35 neurons also initialized by the Nguyen–Widrow initialization procedure.

5. Results

5.1. Analysis of a disulfide reaction

Titin is one of the most abundant and force-bearing proteins in muscle, which connects the Z line to the M line in the sarcomere. Titin functions as a molecular spring that absorbs the accumulated tension during the sarcomere contraction–relaxation cycles and determines the elasticity of muscle[30, 31]. It has been proposed that the titin's function as a molecular spring not only depend on the elastic extension of the unstructured regions, but also in the unfolding and refolding of the force-bearing immunoglobulin-like (Ig) domains[49, 101].

It has been hypothesized that modulatory effects on titin stiffness may arise from disulfide bonding under oxidant stress, Ig domains in titin's spring region have a potential for S-S formation. Titin's I27 domain is the most studied Ig domain due its high mechanical stability and low unfolding rate[74].

QM/MM force-clamp simulations were performed on a mutated I27*, that had two oxidized cysteines at the residue positions 24 and 55. The QM/MM simulations were set up in order to cover a possible nucleophilic attack of the deprotonated reduced Cys32, located on a flexible loop, on both Cys24 and Cys55. These simulations were aiming to explain the regioselectivity of the disulfide shuffling in proteins.

The QM/MM simulation setup was prepared by Dr. Marina Putzu, the starting structures were snapshots from the force-clamp swapping simulations of Gräter's group[65]. Due to an applied external pulling force on the termini, the protein was already unfolded up to the disulfide bond between S24 and S55. There were 334 QM/MM setups, in 160 of the selected structures S32 was closer to S24, and in 174 structures closer to S55.

The QM region comprised the side chains of Cys24, Cys55 and Cys32 up to Cβ. Bonds between Cα and Cβ were treated with the link atom approach. In total, the QM region consisted of 15 atoms described with DFTB3 and 3OB parameters, see Fig. 5.1. The rest of the system was described with the AMBER99SB-ILDN forcefield [73] and TIP3P water. Temperature and pressure were kept at 300 K and 1 bar with the Nosé-Hoover thermostat and the Parrinello–Rahman barostat, respectively.

The QM/MM simulations were extended up to 20 ns to observe disulfide shuffling in the I27* domain. Snapshots of the trajectories were saved every 0.5 ps. The simulations where a disulfide exchange occurred were stopped due to the protein termini leaving the simulation box at both sides.

A reaction occurred 66 times from the 334 initial setups, with a preference for Cys32 attacking Cys55 (48 reactions) over Cys24 (18 reactions). A disulfide exchange reaction was possible by means of an attack of Cys32, present in the deprotonated thiolate form, on either Cys24 or Cys55. The preference for Cys32 agreed with experimental observations, and the Cys55/Cys24 ratio of 2.7 agreed with an observed experimental ratio of 3.8[5].

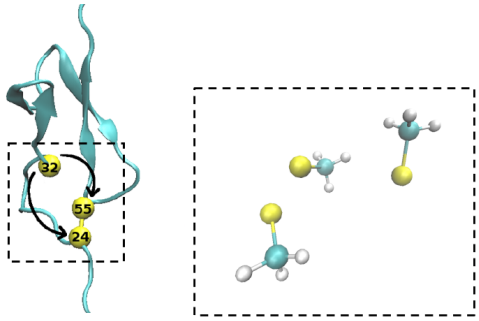


Figure 5.1.: QM zone from the I27* domain consisting in Cys24, Cys32 and Cys55.

In order to predict disulfide shuffling with dependency on a preferred cysteine, a decision tree classifier algorithm based only on structural data was attempted. To establish the features that determine a successful exchange reaction, the initial, 4 equitably distributed and the snapshot 10 ps prior the ending of the simulation were taken from (see Fig.5.2) from both successful and unsuccessful reactions. Distances and angles between the sulfurs were measured with Plumed[19] in all trajectories, charges of the QM atoms were calculated with DFTB+[60].

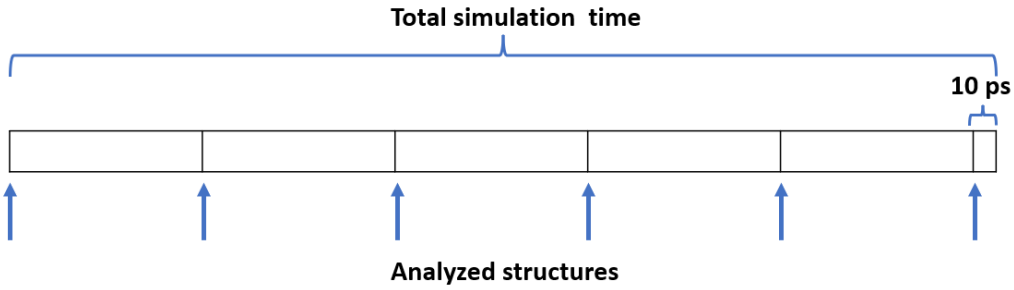


Figure 5.2.: Trajectory splitting for recording structural information.

In Fig.5.3, it can be seen that the first decision criteria is the distance between the sulfurs atoms of Cys32 and Cys24. Then we ca make the following statements:

- Cys32-Cys24 < 4.38 Å will lead to a exchange towards Cys24.
- Cys32-Cys24 ≥ 4.38 Å and Cys32-Cys55 ≤ 4.21 Åwill lead exchange towards Cys55.
- When both distances are greater than 4.3 Å, an exchange will not take place.

Although angle information was also given as an input, the decision tree did not consider it a fundamental value for determining shuffling.

Additional structural analysis concluded that the distance between S_{nuc} and S_{ctr}, (S24 or S55) has a range between 3–5 Å. Whereas S_{nuc} is further away from the leaving sulfur, S_{lg}, at 4.5–7 Å. The TS is formed as soon as $|S_{nuc} - S_{ctr}|$ has decreased to ~2.75 Å and $|S_{ctr} - S_{lg}|$ has increased to ~2.75 Å, while $|S_{nuc} - S_{lg}|$ ~5.4 Å indicating a linear arrangement[87].

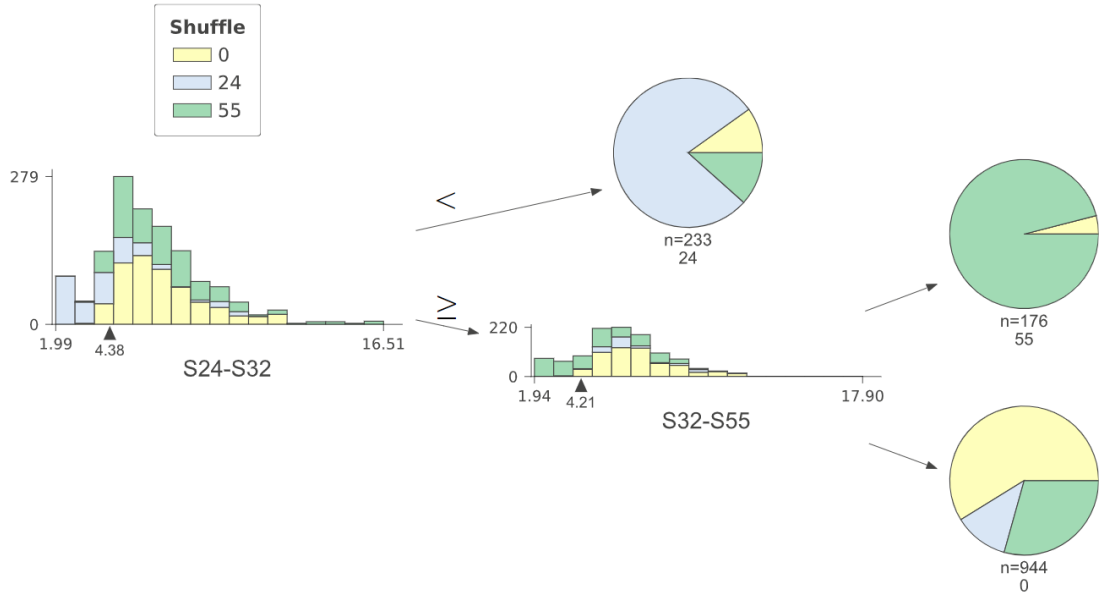


Figure 5.3.: Exchange towards a preferred cysteine is indicated by its index: 24 or 55. No exchange is denoted by "0"

The angle $S_{nuc} - S_{ctr} - S_{lg}$ oscillates between 80–180° and this range narrows down to 120–170° right before the formation of the TS.

Simulations where the reaction succeeded were re-run to get the electrostatic potential. S55 in the role of nucleophilic target carried a more positive charge than S24, and S24 carried a more negative charge as a leaving group than S55 did. This means that S55 is the better nucleophilic target, and S24 the better leaving group of the two[1]. Consequently, it is the electrostatic effects of the molecular environment that support the reaction S32→S55 more than S32→S24. This is an additional explanation of the outcome of the force-clamp experiments on I27*.

5.2. Dataset for machine learning

The selection of structures came from two systems. The first one came from the analysed I27* domain where disulfide shuffling between three cysteines in the reactive center occurs. The reaction was driven with QM/MM metadynamics (performed by Dr. Denis Maag). Snapshots of the QM region were extracted from the trajectories and binned regarding the three sulfur-sulfur distance combinations. Only distances with a minimum length of 1.85 Å, a maximum bond length of 6.95 Å and a binsize of 0.15 Å were considered. This yielded in 8,436 bins, less than the theoretical amount since not all distance combinations are chemically feasible or present in the QM/MM metadynamics. From each bin one structure was added to the training data. Important structures which lie very high in

[1]For a more detailed explanation see Ref.[75] and Dr. Denis Maag PhD thesis.

energy with DFTB/3OB, e.g. structures around the “true” transition state, are usually not sampled during QM/MM metadynamics.

Thus, a second system was used to compensate for this. A dimethyl disulfide and a methylthiolate anion were energy minimized with DFTB/3OB and subsequently an unrelaxed potential energy scan was performed, resulting in 5,112 additional structures which were added to the dataset.

The reference DFTB energies for the selected structures were calculated in gas-phase with the 3OB parametrization by the DFTB+ software. The threshold value taken as reference for normalizing the DFTB3 energy values was of -9931.149 kcal/mol, this value corresponds to the energy of infinitely separated reactants. The B3LYP functional energies were calculated by TurboMole 6.5 using the aug-cc-pVTZ basis set, the threshold value is -824854.937 kcal/mol. For the CCSD(T) the DLPNO approximation[72] was employed by the ORCA 4.2.1 software.

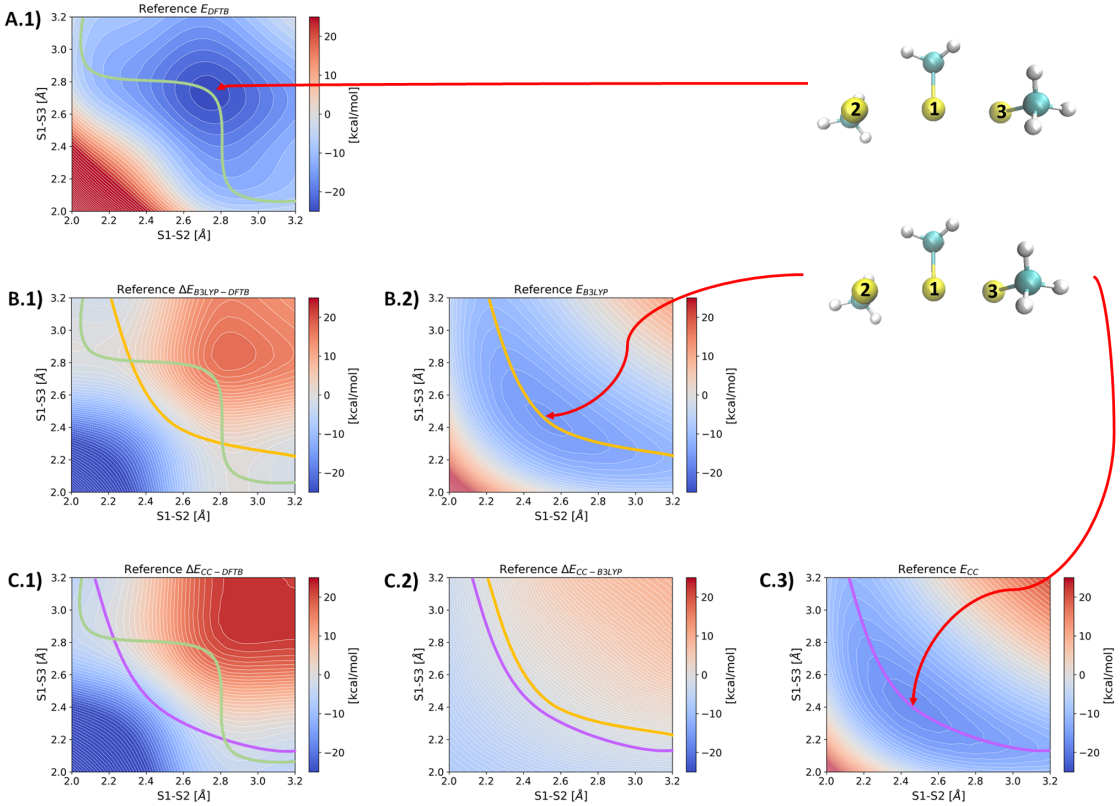


Figure 5.4.: Gas-phase potential energy surfaces, representing the total energy as a function of the S1–S2 and S1–S3 bond length in a linear S3–S1–S2 configuration. Contour lines are drawn every 0.5 kcal/mol. Pathways of minimum energy are drawn as green lines for DFTB/3OB, yellow for B3LYP/aug-cc-pVTZ and purple for DLPNO-CCSD(T)/aug-cc-pVTZ. – (A.1) DFTB3 with 3OB parameters, (B.1) $\Delta E_{\text{B3LYP–DFTB}}$, (B.2) B3LYP/aug-cc-pVTZ, (C.1) $\Delta E_{\text{CC–DFTB}}$, (C.2) $\Delta E_{\text{CC–B3LYP}}$, (C.3) DLPNO-CCSD(T)/aug-cc-pVTZ. The minimum energy details can be seen at Table 5.1.

Table 5.1.: Minimum energy on gas-phase potential energy surfaces.

Method	Minimum energy [kcal/mol]
DFTB/3OB	−25.0
B3LYP	−15.2
CC	−16.7

Visual inspection of the PES in Fig.5.4 reveals that DFTB3/3OB predicts much longer S–S bonds than the higher-level references. B3LYP performs much better but still slightly overestimates the bond lengths in the minimum, with an error of 3 kcal/mol. The B3LYP PES deviates from the DLPNO-CCSD(T) PES, therefore, energy differences of 3 kcal/mol may be due to a comparison of single point energies for static structures as done in Ref.[82], where a 1-dimensional energy scan was performed.

Following the respective minimum energy pathways for dissociation, the differences between DLPNO-CCSD(T) and B3LYP seem to be less pronounced. Even though the overall PES shows some differences definitely, the largest qualitative difference are apparent for high-energy structures (see Fig. 5.4 C.2), which are hardly relevant in typical applications.

5.3. Artificial neural network for Δ-learning

From the selection of structures, two datasets were formed: $\Delta E_{\mathrm{CC-DFTB}}$ and $\Delta E_{\mathrm{B3LYP-DFTB}}$. Due to the energy distribution being skewed towards high ΔE values, and having a higher density of sampling over structures with no energy difference, lower delta values structure-energy pairs were duplicated achieving a more Gaussian-like distribution (Fig. 5.5). The distribution of the training structures as a function of the S1–S2 and S1–S3 are shown in Fig. 5.6[2].

5.3.1. Construction of the symmetry functions

A GA was employed to find the hyperparameters that construct the symmetry function vectors described in Eq. 4.2 and Eq. 4.3 for each element of the considered system: carbon, hydrogen and sulfur. The fitness was been defined as the RMSE of the test set, aiming for a value of 0.75 kcal/mol.

The GA ran on "Batch-NNs", using the $\Delta E_{\mathrm{B3LYP-DFTB}}$ dataset with training, validation and test set split in the ratios of 60/20/20. The initial population was composed of 20 random strings. The training hyperparameters remained invariant at every training generation. The implemented workflow is shown in Fig. 5.7

The GA took 12 generations to find the best descriptor features for the system, the evolution after each generation can be observed in Fig.5.8. The best genes constructed 144 symmetry function vectors for each atom, following Eq. 4.2 12 $R_{s,m}$ shells were considered

[2]This image is reproduced with permission form Ref.[53] Copyright 2022 American Chemical Society and re-used from Dr. Denis Maag PhD thesis.

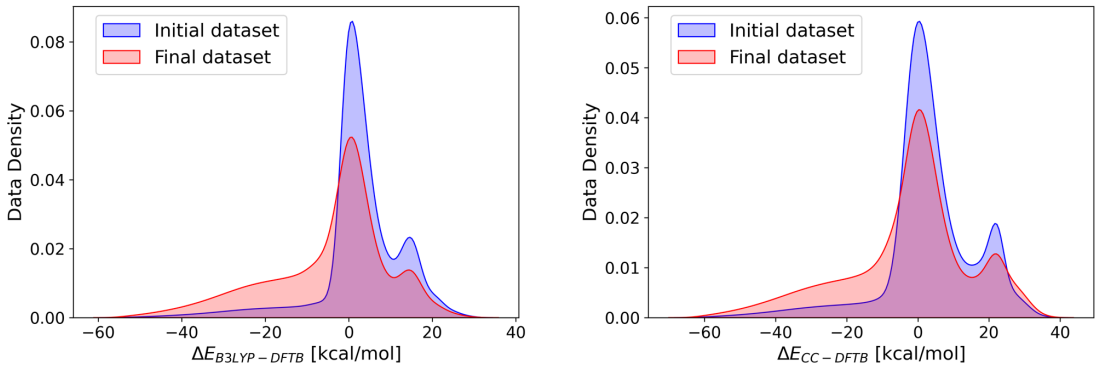


Figure 5.5.: Data density from $\Delta E_{\text{CC-DFTB}}$ and $\Delta E_{\text{B3LYP-DFTB}}$ initial dataset distribution seen in green color. Red color shows the new data density after the data augmentation

taking into account a minimal distance of 0.2 Å to a maximum of 5 Å and an $\eta = \frac{5^3}{R_{s,m}^2}$ was used. The angular parameters were set as $\eta = [0.001, 0.01, 0.05]$, $\lambda = \pm 1$ and $\zeta = [1, 4, 16]$ as in Eq. 4.3.

5.3.2. Training of the ΔE

With the optimized molecular descriptor from the GA, two new NNs were trained with 80% of the dataset, one for $\Delta E_{\text{B3LYP-DFTB}}$ and the other for $\Delta E_{\text{CC-DFTB}}$, the training times on a GeForce GTX 1080 Ti GPU took 52 min and 58 minutes respectively. The NNs were trained during six sessions using 80 % of the structures,the Adam optimizer[62] (see Appendix A.4), a descending learning rate of 0.01, 150 epochs and a decay rate of 0.001.

The loss function, $\mathcal{L}$, used to minimize the error after every session was defined by Eq. 5.1. L refers to the number of geometries, ΔE^{ref} the reference values of $E_{\text{high}} - E_{\text{low}}$ and E^{pred} the energy difference predicted by the NN. This step was performed on the Validation Set. After the six sessions a final model was exported and evaluated on the Test Set.

$$\mathcal{L} = \frac{1}{N}\sqrt{\sum_{X=1}^{L}\left(\Delta E^{\text{pred}} - \Delta E^{\text{ref}}\right)^2} \tag{5.1}$$

The accuracy of the NNs was evaluated with the test set showing that they are able to predict this difference between *ab initio* methods and DFTB. The B3LYP training achieved a RMSE of 0.5 kcal/mol from a range of values between −29.41 kcal/mol and 22.21 kcal/mol. The CC training achieved a RMSE of 0.62 kcal/mol from a range of values between −62.78 kcal/mol and 36.39 kcal/mol. The accuracy of the predictions when compared with the reference values on the test set can be seen in Fig. 5.9.

The model of each training was saved as a 27 kB text file containing the values for weights and biases for each element.

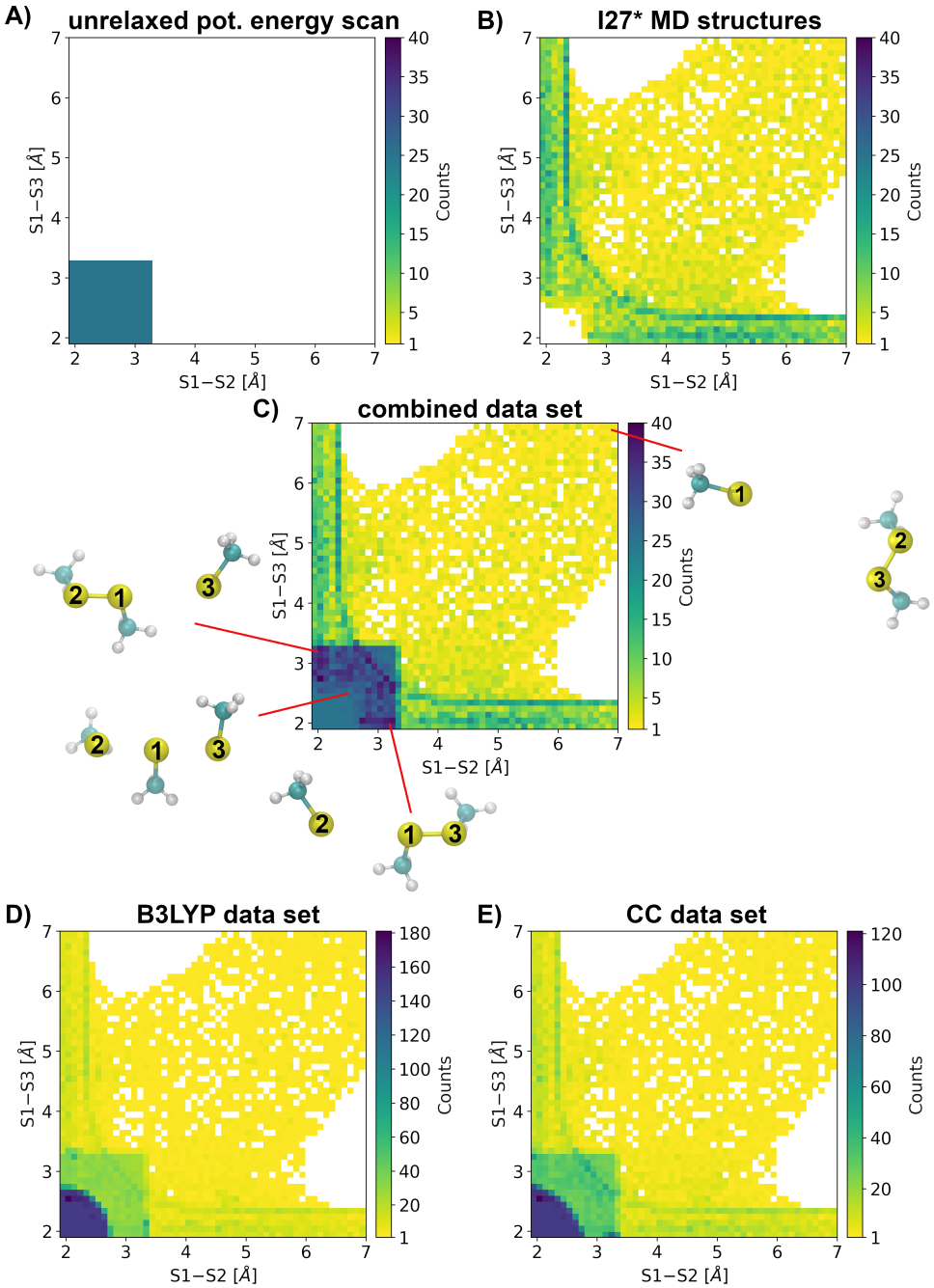


Figure 5.6.: Distribution of the training structures as a function of the S1–S2 and S1–S3 of the (A) unrelaxed potential energy scan (5112 structures), (B) QM/MM simulations of I27* (8436 structures), (C) combined data set (13,548 structures), (D) B3LYP data (21,238 structures), (E) DLPNO-CCSD(T) data set (18,357 structures). A minimum bond length of 1.9 Å, a maximum bond length of 7.0 Å and a bin width of 0.1 Å was considered for the histograms.

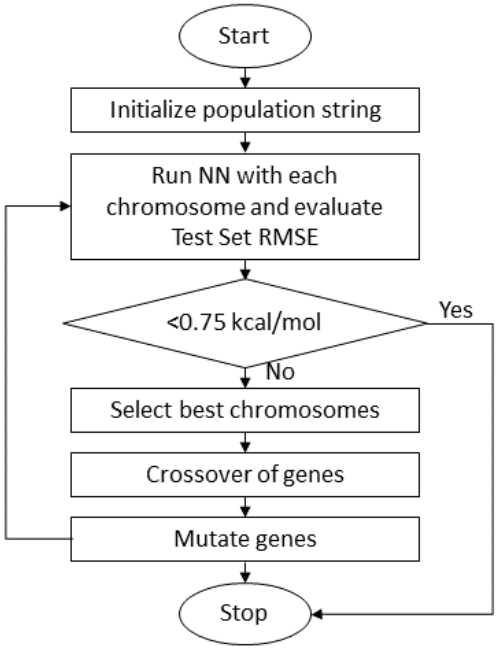


Figure 5.7.: The workflow of the genetic algorithm. The fitness is defined as the RMSE on the test set; the best chromosomes will be the ones having a fitness towards 0.75 kcal/mol. If the stop criterion of a RMSE < 0.75 kcal/mol is achieved, the GA terminates, otherwise a new iteration is started.

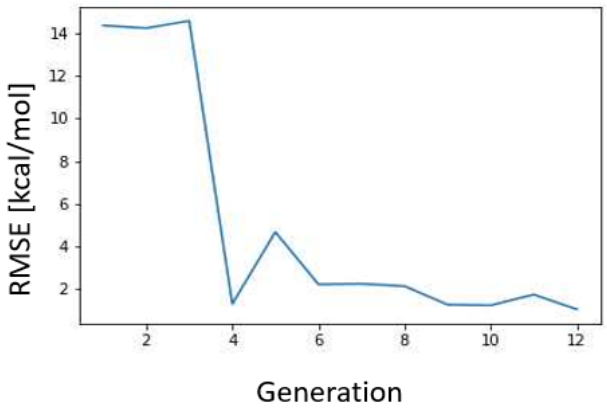


Figure 5.8.: RMSE evolution through evolving generations with 60% of training samples.

5.3.3. Evaluation of the model Δ-ML

The Behler-Parrinello NN scheme of calculation of a correction for quantum chemical calculations was implemented in the DFTB+ software package[8, 60] from scratch by Dr. Tomáš Kubař [67].

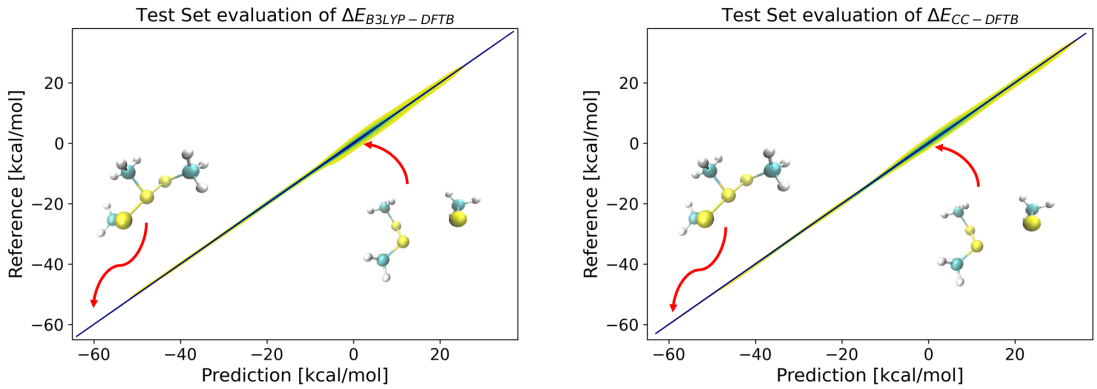


Figure 5.9.: Test set fitting of the Neural Network with 10% of the dataset.

A machine learning module is necessary inside the Hamiltonian specifications in the input file for the DFTB+, with the explicit symmetry functions parameters and the path for the model files. The input file can be seen at the Appendix A.5.

The obtained models: $\Delta E_{\mathrm{CC-DFTB}}$ and $\Delta E_{\mathrm{B3LYP-DFTB}}$ were evaluated using this DFTB+/ML implementations, see Fig. 5.10 (top). Single point energies and forces were calculated for 2000 structures randomly selected from the dataset of linear structures.

Comparison with the differential PES in the data sets (Fig. 5.4 B.1 and C.1) as well as the comparison of the entire potential energies (Fig. 5.10 (center) versus Figs. 5.4 B.2 and C.3) shows that the NN correction captures the energy differences well. Overall, the *ab initio* and DFT surfaces are reproduced very well, as illustrated by the small deviation of predicted differential PES from those in the data sets, see Fig. 5.10 (bottom).

Interestingly, the error of the respective fit is smaller than the difference between the reference methods B3LYP and CCSD(T). Therefore, it indeed makes sense to target high-level reference methods, i.e. one can try to fit semiempirical methods to reproduce high level *ab initio* results. Therefore, such fitted semiempirical methods in principle can outperform standard DFT methods not only in computational efficiency, but also in accuracy within the limits they have been parametrized for.

To evaluate the accuracy of ML corrected forces, we followed the methodology from Zhu et al. [115] by testing the energy conservation on microcanonical (NVE) simulations of a random structure from the linear dataset. The simulations ran for 100 ps on DFTB+/ML with a timestep of 0.5 fs and an initial temperature of 300 K. The energy fluctuation of <0.25 kcal/mol surrounding an average energy of zero can be seen in Fig.5.11, indicating a good accuracy of the MD simulation and therefore an accurate calculation of forces by the DFTB+/ML implementation.

On Fig.5.12 we can position the Δ-ML algorithm, following the proposed hypothesis in Fig.1.5. The computational cost of the energy correction does not depend on the level-of-theory accuracy, as $\Delta E_{\mathrm{CC-DFTB}}$ and $\Delta E_{\mathrm{B3LYP-DFTB}}$ are described by the same number of symmetry functions. A single point calculation including the ΔE correction and force estimation took 41 ± 2 ms on average for both CC and B3LYP corrections, from this average 7 % (≈3 ms) corresponds to the addition of the ΔE while the calculation of forces

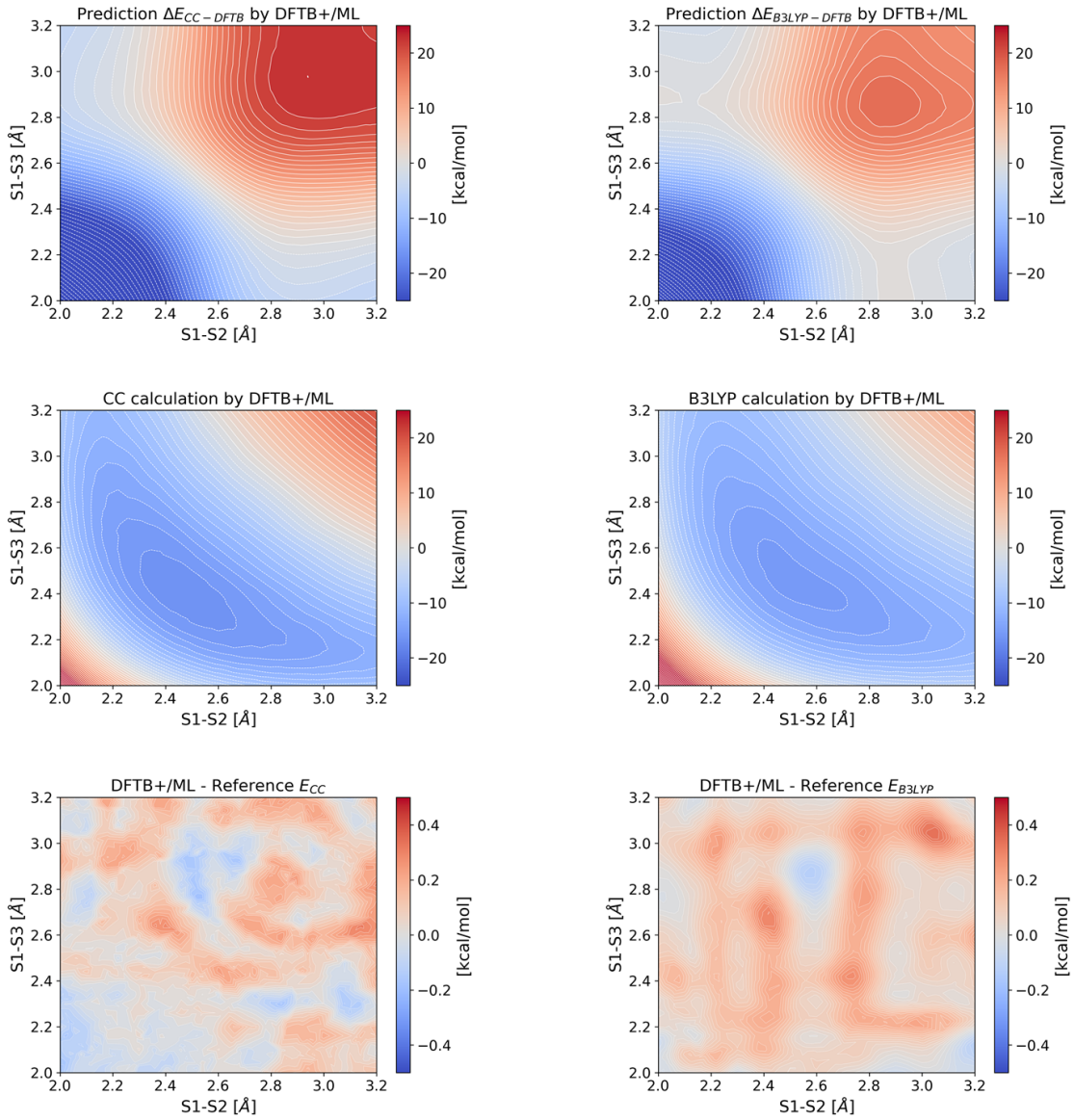


Figure 5.10.: Top: Magnitude of the ML correction $\Delta E_{\text{B3LYP–DFTB}}$ and $\Delta E_{\text{CC–DFTB}}$, to the B3LYP and CCSD(T)–DLPNO levels respectively. Energy correction as a function of the S1–S2 and S1–S3 bond length in a linear S3–S1–S2 configuration. Contour lines are drawn every 0.5 kcal/mol. Center: The PES calculated with DFTB+/ML correction, i.e., $E_{\text{DFTB}} + \Delta E$) Contour lines are drawn every 0.5 kcal/mol. Bottom: Deviation of the PES calculated by DFTB+/ML from the true PES obtained with the reference method. The error remains deep under 1 kcal/mol.

accounts for 42 % (≈17 ms) of the time[3]. Obviously, the calculation of gradients makes

[3]If we go back to Eq.2.36, forces get calculated by calculating the gradients of the potential energy w.r.t. each coordinate. To calculate the forces of this correction, is necessary to calculate the gradients of the predicted ΔE w.r.t each symmetry vector and the gradients of each vector w.r.t each Cartesian coordinate.

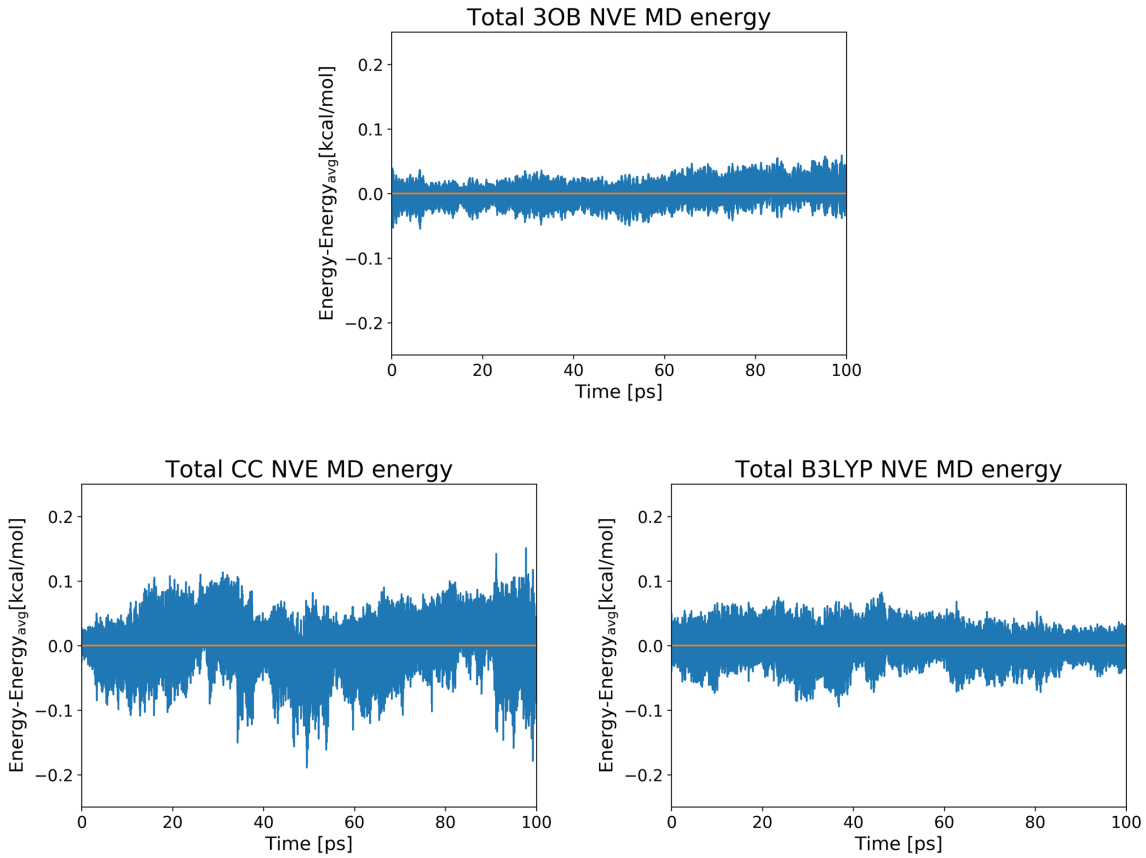


Figure 5.11.: Total energy during MD simulations in a microcanonical ensemble using the force derivatives from the ML corrected energy up to B3LYP and CC.

DFTB+/ML calculations take twice as long as a corresponding DFTB-only calculation. When compared with CCSD(T) calculations – even the extremely efficient DLPNO variant, which approaches the efficiency of DFT, DFTB+/ML remains three orders of magnitude faster.

Table 5.2.: Δ-ML correction times for single point calculations.

Method	Average Time [ms]
DFTB+	20
DFTB+/ML	24
DFTB+/ML with forces	41

It is interesting to see, however, that the computational cost of the NN is similar to DFTB3. Once again, this demonstrates that semiempirical methods represent chemical information in a very efficient way, which is hard to outperform by data-driven approaches

If we have 144 vectors per atom and 15 atoms on a 3D space, those are 12960 operations! Reducing the symmetry vectors would make this process faster

in terms of efficiency. Due to the similar computing times involved, they may represent a very good combination of computational approaches in terms of speed and accuracy.

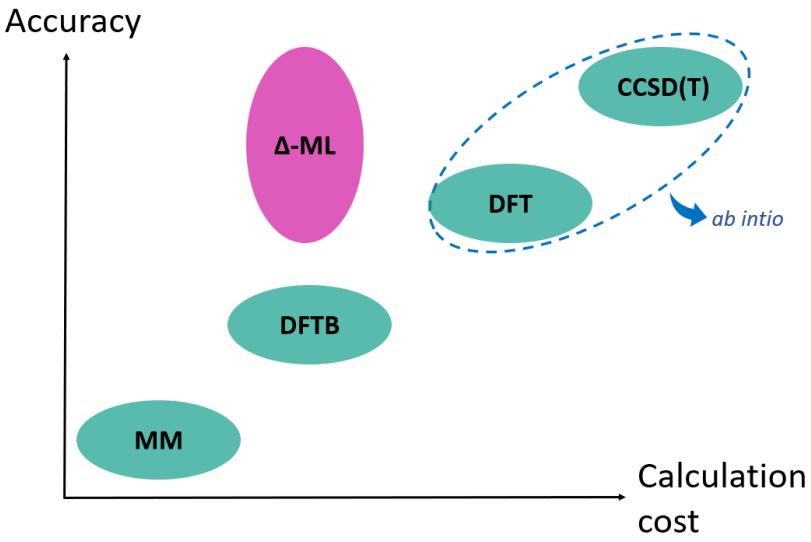


Figure 5.12.: Real positioning of the Δ-ML implementation costs with respect of DFTB.

5.4. Artificial neural network for learning a fourth generation potential

From the selection of structures, the E_{CC} dataset was used and the ACSF remained as the chosen molecular descriptor (Eq.4.3 and Eq.4.2).

5.4.1. Training of the electronegativities

For the Charge Equilibration via Neural network Technique (CENT), the atomic number was appended to the ACSF descriptor. We used the DFTB2 Mulliken charges obtained by DFTB+ as reference data. We converted these charges into the electronegativities by Eq.4.6 and fed them as the target input for the NN.

The radial parameters for this learning were defined with by a cut-off radius=10 Å, 12 $R_{s,m}$ shells from 0.8 Å to 12 Å and an $\eta = \frac{5^3}{R_{s,m}^2}$ was used. The angular parameters were set as $\eta = [0.001, 0.01, 0.05]$, $\lambda = \pm 1$ and $\zeta = [1, 4, 16]$. The entire distribution of features was normalized to ensure a better fitting, the normalization followed Eq.5.2, where μ is the mean of the feature values and σ the standard deviation.

$$X' = \frac{X - \mu}{\sigma} \tag{5.2}$$

The NN was trained during four sessions using 80 % of the structures, the Adam optimizer, a descending learning rate of $5e-5$, 1500 epochs and a decay rate of $1e-4$. We used as

control the "EarlyStopping", which stops the training when the loss function is no longer decreased. The loss function, $\mathcal{L}$, used to minimize the error after every session was defined by Eq. 5.3. L refers to the number of geometries, χ^{ref} the obtained elecronegativities from reference partial charges and χ^{pred} the predicted electronegativities by the NN.

$$\mathcal{L} = \frac{1}{N}\sqrt{\sum_{X=1}^{L}\left(\chi^{\text{pred}} - \chi^{\text{ref}}\right)^2} \tag{5.3}$$

The accuracy of the NN was evaluated with the test set by transforming the predicted electronegativities into charges via Eq.4.9. The training achieved a RMSE of 0.08 A.U. when predicting electronegativities, and RMSE of 0.06 A.U. w.r.t to charges from a range of values between −2.33 A.U. and 1.14 A.U (Fig. 5.13).

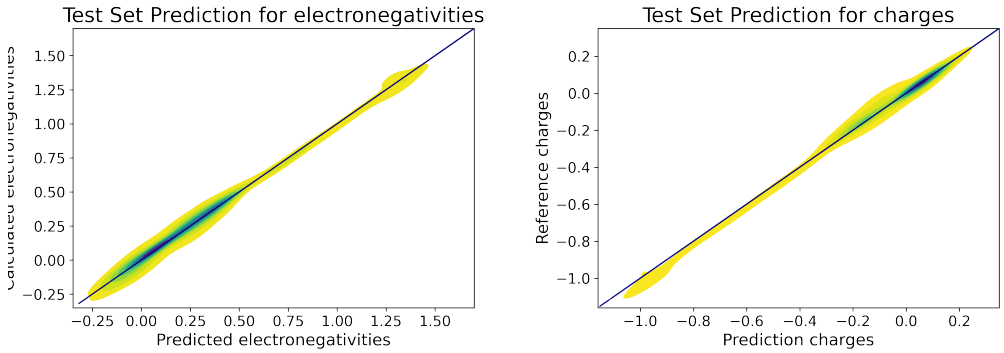


Figure 5.13.: Test Set evaluation of elecronegativity and charge.

For getting a better visualization of the predicted charges, they were separated into the trained elements, see Fig. 5.14. Some clustering and deviations can be observed by calculating the carbon charge distributions. This could be caused by the selection of reference data. Even when DFTB2 includes charge transfer and polarization effects, DFTB3 has a better accuracy of the charge distribution.

In DFTB2 the Hubbard parameters remain constant for every element (Eq.2.33), this parameter strongly influences charge transfer, which is supposed to occur at C-H bonds. DFTB3 uses a charge dependent Hubbard parameters (Eq.2.34), which is particularly important for the description of systems with localized charges.

In the current CENT methodology, the Hubbard parameters are provided as an input constant. Integrating the Hubbard parameters into the NN, i,e. by making them adjustable as the weights and biases, would lead to better accuracy, independent of the reference charge model.

It is crucial to have good accuracy during training. As the atomic energy calculation depends on the charges, the system of Fig.4.13 is prone to error propagation.

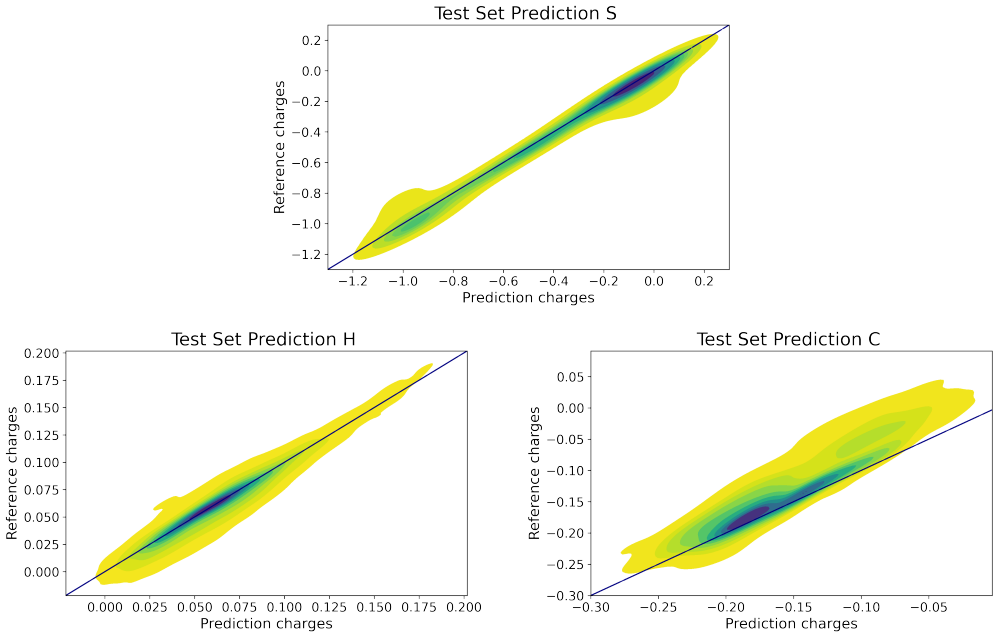


Figure 5.14.: Test Set charges per element.

5.4.2. Training of the atomic energies

We investigated a second (2nd) and fourth (4th) generation potential for fitting the atomic E_{CC}. The 2nd generation potential is analogous to the Δ-ML ($E_{CC} - \cancel{E_{DFTB}}^{0}$).

The 4th generation has as an additional descriptor to the ACSF, the partial charges of each atom. For congruence, we used as reference the Mulliken charges obtained by DFTB2.

The radial parameters for both 2nd and 4th generation NN were defined with by a cut-off radius=12 Å, 24 $R_{s,m}$ shells from 0.8 Å to 5 Å and an $\eta = \frac{5^3}{R_{s,m}^2}$ was used. The angular parameters were set as $\eta = [0.001, 0.01, 0.05]$, $\lambda = \pm 1$ and $\zeta = [1, 4, 16]$.

A slight data augmentation was performed based on the values of E_{CC}, the new dataset can be observed in Fig.5.15

Both NNs were trained during six sessions using 80 % of the structures, the Adam optimizer, a descending learning rate of $5e-6$, 1500 epochs and a decay rate of $1e-5$. The loss function, $\mathcal{L}$, used to minimize the prediction error after every session was analogous to Eq. 5.1.

The accuracy of the NNs was evaluated with the test set. The 2nd generation E_{CC} training achieved a RMSE of 1.98 kcal/mol, while the 4th generation E_{CC} training had a RMSE of 2.09 kcal/mol. The dataset had a range of values between −16.69 kcal/mol and 71.58 kcal/mol. The accuracy of the predictions when compared with the reference values on the test set can be seen in Fig. 5.16.

It is important to notice in Fig.5.16 that the RMSE increases when giving reference charges as an additional descriptor to calculate the potential, even when the ACSF and training parameters are conserved.

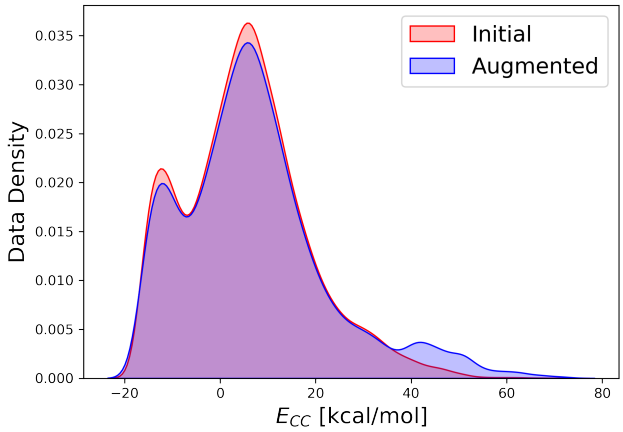


Figure 5.15.: Data augmentation for the 2nd and 3rd generation potentials.

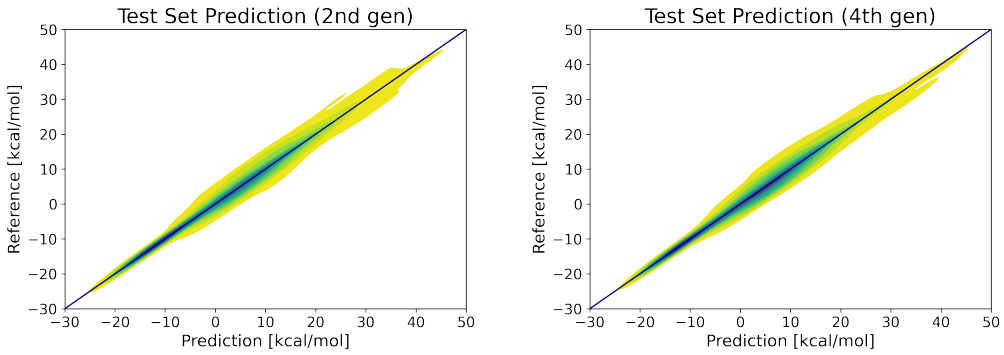


Figure 5.16.: Test set fitting for the 2nd and 4th generation potentials.

In the 2nd generation potential, the charges are implicitly fitted into the energy. The drop of accuracy in the 4th generation could be an indication of an inconsistent charge assigned to each structure[4]. Therefore, we would be using as reference an inaccurate charge distribution. As mentioned previously, DFTB3 charges could provide better reference data.

5.4.3. Evaluation of the models

As primary test to evaluate a connected system between the CENT and the 4th generation potential (as seen in Fig.4.13), 2000 structures were randomly selected from the dataset of linear structures.

By giving as the only input the molecular structures and the global charge of the systems (-1), on a first step, CENT calculated the partial charges. On a second step, these predicted charges became part of the descriptor to the 4th generation potential.

[4]These charges contain information about the global electronic structure

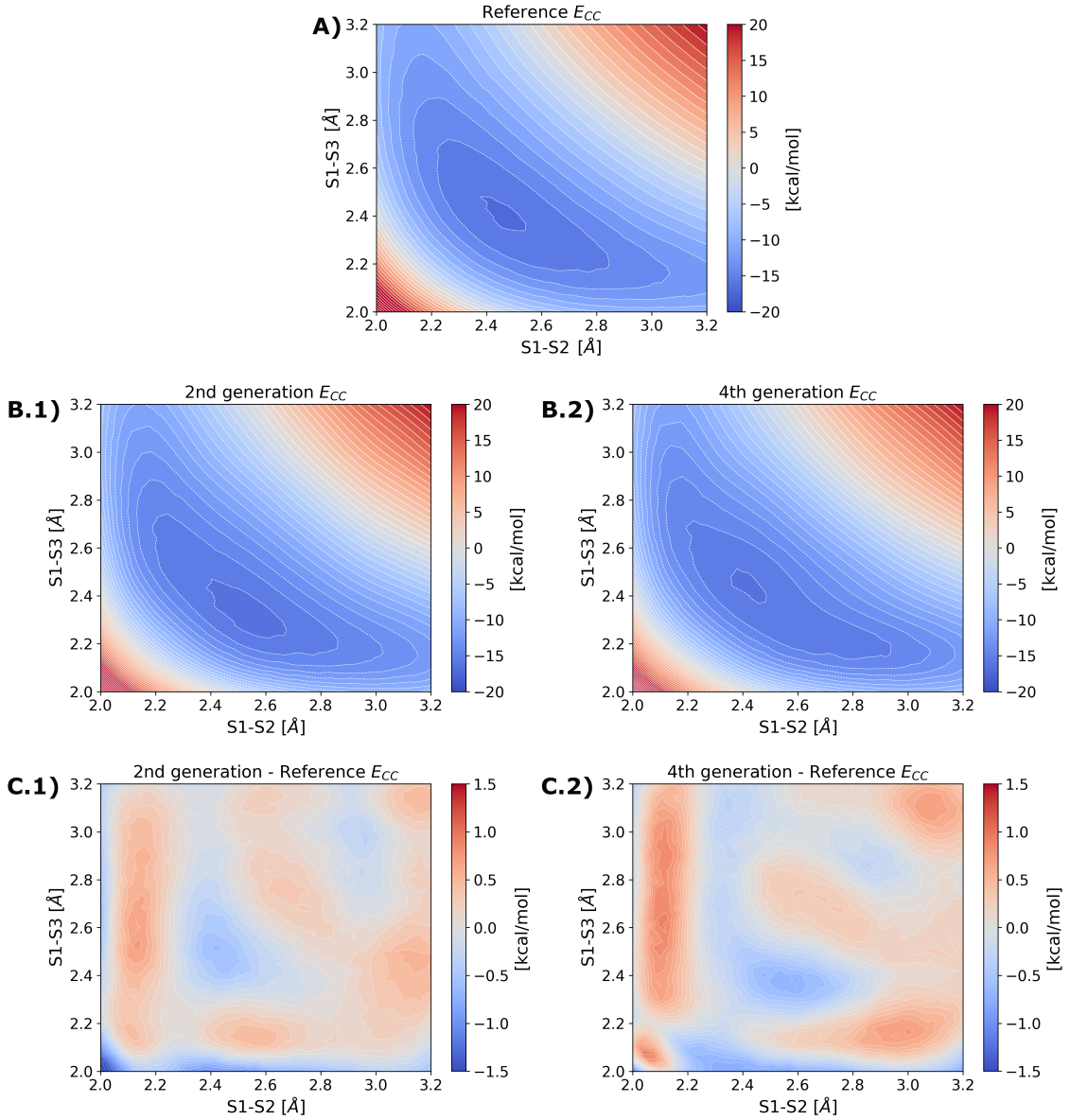


Figure 5.17.: A) Reference E_{CC} as a function of the S1–S2 and S1–S3 bond length in a linear S3–S1–S2 configuration. Contour lines are drawn every 0.5 kcal/mol.
B1) PES by a 2nd generation potential. B2) PES by a 4th generation potential using predicted charges by the CENT. Contour lines are drawn every 0.5 kcal/mol.
C.1) Deviation of the PES calculated by the 2nd generation with reference data.
C.2) Deviation of the PES calculated by the 4th generation with reference data.

Comparison between the 2nd- and 4th- generation NN PES (Fig. 5.17 B.1 and B.2) shows that the NN captures the pathways of minimum energy, something that DFTB fails to

do (Fig.4.9 A.1). However, these results still need improvement (Fig. 5.17 C.1 and C.2). Additional to evident energetic deviations, the PES do not conserve symmetry.

For getting a robust system as seen in Fig.4.13, errors in both trainings must be decreased but also be taken care of not falling into overfitting. Other reference charge distributions are encouraged to explore. Preliminary data surrounding these distributions can be seen in Appendix.A.6.

6. Outlook

The Δ-ML approach provided very satisfying results on different levels of theory. A possible avenue of improvement could be decreasing the size of the ACSF descriptor as it could provide large improvements in regards to computational cost.

Integrating the Δ-ML into the GUI framework allows for more flexibility as it can be adapted to use cases where the error is contained within the atomic environment. The GUI also allows users with no ML background to use the system which could greatly increase its adoption. We look forward to seeing the framework applied to a large variety of uses cases in the near future.

The 4th generation potential, although accurate (error around 2% of the range of the data), still proved to be less accurate than the Δ-ML. To improve accuracy, we can start by looking at the 2nd generation potential (learning of E_{CC}).

By the representation theorem (see Section 2.3.1), any arbitrary function can be fitted by a NN. We can treat the 2nd generation potential as a "black-box" where the weights and biases are adjusted without physical interpretability, just like in the Δ-ML approach. Therefore, we would expect an accuracy comparable to the Δ-ML with an error of less than 1% of the range of the data. The 2nd generation inaccuracy should be addresses as a learning problem, in which case we should start focusing on the training parameters, NN architecture, and the ACSF parameters.

In Fig.5.17 C.1 we can see a uniform distribution of the error w.r.t the structures. Augmenting the sampling of these structures on the dataset could improve the general accuracy as well.

When the accuracy of the 2nd generation potential is improved, we can reuse the same training setup for learning the short-range energy of the 4th generation and observe if the accuracy improves on the same scale.

Depending on the observed accuracy, we can explore other charge distributions. We can propose DFTB3 for obtaining reference charges as the charge distribution is better represented than in DFTB2 (see Section 2.1.3.1).

For improving the character of the CENT we suggest giving flexibility to the hardness value and make it an adjustable parameter during the training sessions.

We look forward for the application of the 4th generation model into a QM/MM scheme, acting as a machine-learned force field. The local bonding would be described by the atomic short-range energy and the CENT would ensure long-range electrostatic interactions based on charge distribution dependent on the environment in which the quantum system is embedded.

7. Conclusion

Disulfide bonds have an important role for the function of many proteins. Therefore, addressing these reactions accurately by applying low computational cost approaches is of great importance in the simulation community. This work emphasized the importance of a correct description of the QM zone energies and how the QM/MM interaction is fundamental for the description of these reactions.

Electrostatic and steric factors play a vital role in controlling the disulfide exchange, as observed with the QM/MM simulations in the 127* domain. The rate of QM/MM successful reactions was consistent with experimental force-clamp observations suggesting a trustful set up to be used as an explanatory model.

Successful exchanged reactions were dependent on electrostatic effects of the molecular environment. The charges of the sulfurs, and consequently the free energies, changed in the presence of an external electric potential. A negative applied ESP results in a more positive charge, which makes the touched atom a better nucleophilic target but a worse leaving group; and a positive applied ESP results in a more negative charge, making the atom a better leaving group but a worse nucleophilic target.

In nature, an electrostatic potential may come from the protein itself or from the water environment. This potential would lead to a slight polarization of the disulfide bond, letting the nucleophile attack one of the sulfur atoms preferentially. The energies from this electric field would result in a variation of energy barriers of some few kJ/mol, modulating the reaction rates by a small factor.

Structural factors are as important but can be misleading if there is an inaccurate description of the potential energies. A sampling of these variables on the range of the reaction can require very costly computational approaches. Semiempirical methods are from 3 to 4 orders of magnitude faster than DFT-GGA using moderately sized basis sets; however, they may run into even greater difficulties for challenging reactions than DFT-GGA does. In the disulfide reaction the error is rooted in local bonding effects, DFTB/3OB can predict much longer S–S bonds than the higher-level references, B3LYP performs much better but still slightly overestimates the bond lengths in its minimum energy, with an error of 3 kcal/mol when compared with CC.

The Δ-ML methodology proved to be adequate to correct the qualitative and quantitative errors of density based functionals such as DFT and DFTB. Its further application in a QM/MM scheme as shown in Ref.[53] involved two molecular complexes: a solvated model system and a blood protein, and proved to be transferable into complex environments without loss of accuracy. However as the correction is based on an addition over a previously calculated energy (DFTB), it brings an additional cost in time. As long as the architecture and descriptor system remain constant, the aftermentioned cost of the energy correction does not depend on the level-of-theory.

In the presented work, the Δ-ML computing time was comparable with a DFTB calculation itself. Force calculation was the pitfall on computational resources as the gradient energy calculation with respect of each symmetry vector (144 vector for representing an atom), is the most consuming operation. Even when the GA acted as an efficient method to ensure the best descriptor it did not took into account its dimensionality, as a future application it is suggested to add a constraint of length at the moment of declaring the fitness function.

Even if the Δ-ML correction times are comparable to a DFTB calculation, it saved up to 3 orders of magnitude on computing resources. We believe other systems, where the energetic errors are grounded inside the QM local effects, can benefit from this methodology.

The workflow was designed to accelerate this process by automating the data generation and preparation, particularly helpful when handling big-data, as well saving time on the Python-code interpretation and installation of hidden repositories or libraries. Due to the GUI's nature of drag-and-drop, the workflow can be adapted to each application by adding loops or enabling parallelization computing, the pipeline design can also be saved, repurposed and shared among users, ensuring reproducibility of the protocols. This functionality can also be used as a teaching framework for computational chemistry.

This correction algorithm is also implemented on a graphical interface pipeline that will help the user with the production and cleaning of training data, as well exporting the machine learning model into DFTB+ for its further use on QM/MM simulations. The adoption of this pipeline is intended to expand the applications of the Neural Network code prioritizing the knowledge of quantum modelling over a programming background.

By the end of this work, we began the development of a machine learned force field based on the disulfide exchange dynamics. As mentioned previously, electrostatic effects play an important role in the reaction, therefore is crucial to address them properly. Preliminary results showed a qualitative description of the PES based on Coupled cluster theory data, however a better quantitative accuracy is expected. We look forward for the continuation of this work, improving the accuracy of the training and testing other charge distributions, as well integrating this scheme into the description of QM–MM electrostatic interactions.

A. Appendix

A.1. Exchange-correlation functionals

The LDA exchange-correlation functional can be defined by Eq.A.1, where $\epsilon_{XC}(\rho(r))$ is the exchange-correlation energy per electron as a function of the density in the uniform electron gas.

$$E_{XC}^{LDA}[\rho(r)] = \int \rho(r)\epsilon_{XC}(\rho(r))dr \tag{A.1}$$

Following the LDA approach, we could express the exchange-only energy as:

$$E_{X}^{LDA}[\rho_\alpha(r), \rho_\beta(r)] = -\frac{3}{2}\left(\frac{3}{4\pi}\right)^{\frac{1}{3}} \int \left(\rho_\alpha^{\frac{4}{3}}(r) + \rho_\beta^{\frac{4}{3}}(r)\right) dr \tag{A.2}$$

where α and β represent up and down spins.
From the GGA gradients one of the most popular is the Becke correction, known as B88.

$$E_{XC}^{B88}[\rho(r)] = E_{X}^{LDA}[\rho(r)] - b \sum_{\sigma=\alpha,\beta} \int \rho_\sigma^{\frac{4}{3}} \frac{x_\sigma^2}{1 + 6bx_\sigma sinh^{-1}(x_\sigma)} dr \tag{A.3}$$

where $x_\sigma = \frac{|\nabla\rho_\sigma|}{\rho_\sigma^{\frac{4}{3}}}$ is a dimensionless parameter and b a constant=0.0042 a.u.
A widely used correlation functional is the Lee, Yang and Parr expressed as:

$$E_{C}^{LYP}[\rho(r)] = -a \int \frac{1}{1 + dp^{-\frac{1}{3}}} \{r + b\rho^{-\frac{2}{3}}[C_F\rho^{\frac{5}{3}} - 2t_W + \left(\frac{1}{9}t_W + \frac{1}{18}\nabla^2\rho\right)e^{-cr^{\frac{1}{3}}}]\} dr \tag{A.4}$$

where

$$t_W(r) = \sum_{i=1}^{N} \frac{|\nabla\rho_i(r)|^2}{\rho_i(r)} - \frac{1}{8}\nabla^2\rho$$

and

$$C_F = \frac{3}{10}\left(3\pi^2\right)^{\frac{2}{3}}$$

using as constants a=0.049, b=0.132, c=0.2533 and d=0.349.

A combination of the standard LDA exchange (Eq.A.2) with the Becke gradient exchange correction (Eq.A.3) and the LYP (Eq.A.4) is a popular choice, commonly abbreviated as BLYP.

The Perdew-Burke-Ernzerhof functional (PBE) is another example of these combinations. In this functional, the exchange part is written as an enhanced factor multiplied onto the LDA functional, where the dimensionless gradient variable x gets defined by:

$$E_X^{PBE} = E_X^{LDA}\left(a - \frac{a}{1+bx^2}\right) \tag{A.5}$$

The PBE correlation is also an enhancement of the LDA, where t is related to x on means on a spin-polarization function.

$$E_C^{PBE} = E_C^{LDA}\left(cf_3^3 ln\left[1 + dt^2\left(\frac{1+At^2}{1+At^2+A^2t^4}\right]\right)\right) \tag{A.6}$$

$$A = d\left[exp\left(\frac{E_c^{LDA}}{cf_3^3}\right) - 1\right]^{-1}$$

$$f_3(\zeta) = \frac{1}{2}\left[(1+\zeta)^{2/3} + (1-\zeta)^{2/3}\right]$$

$$t = \left[2(3\pi^3)^{1/3} f_3\right]^{-1} x$$

Hybrid functionals are methods that include exact exchange energy. An example of these functionals is the B3LYP, expressed by Eq. A.7, constructed by the Becke 3-parameter exact exchange energy and the LYP correlation.

$$E_X C^{B3LYP} = (1-A)E_X^{LDA} + aE_X^{exact} + bE_X^{B88} + (1-c)E_C^{LDA} + cE_C^{LYP} \tag{A.7}$$

A.2. Backpropagation

The output error δ^L is computed using the chain rule from multivariable calculus by re-expressing its partial derivative in terms of partial derivatives with respect to the output activation.

$$\begin{aligned}
\delta^L &= \frac{\partial \mathcal{L}}{\partial z^L} \\
&= \frac{\partial \mathcal{L}}{\partial a^L}\frac{\partial a^L}{\partial z^L} \\
&= \frac{\partial \mathcal{L}}{\partial a^L}\frac{\partial f(z^L)}{\partial z^L} \\
&= \frac{\partial \mathcal{L}}{\partial a^L} f'(z^L) \\
&= \nabla_a \mathcal{L} \odot f'(z^L)
\end{aligned}$$

The error δ^l is calculated in terms of the next layer δ^{l+1}

$$\begin{aligned}
\delta^l &= \frac{\partial \mathcal{L}}{\partial z^l} \\
&= \frac{\partial \mathcal{L}}{\partial z^{l+1}}\frac{\partial z^{l+1}}{\partial z^l} \\
&= \delta^{l+1}\frac{\partial(w^{l+1}a^l + \cancelto{0}{b^{l+1}})}{\partial z^l} \\
&= \delta^{l+1} w^{l+1}\frac{\partial f(z^l)}{\partial z^l} \\
&= \delta^{l+1}(w^{l+1})^T \odot f'(z^l)
\end{aligned}$$

For the output gradients w.r.t to weights and biases:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w^l_{jk}} &= \frac{\partial \mathcal{L}}{\partial z^l_j}\frac{\partial z^l_j}{\partial w^l_{jk}} \\
&= \delta^l_j \frac{\partial(w^l_{jk}a^{l-1}_k + \cancelto{0}{b^l_j})}{\partial w^l_{jk}} \\
&= \delta^l_j a^{l-1}_k \cancelto{1}{\frac{\partial w^l_{jk}}{\partial w^l_{jk}}} \\
&= \delta^l_j a^{l-1}_k
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial b^l_j} &= \frac{\partial \mathcal{L}}{\partial z^l_j}\frac{\partial z^l_j}{\partial b^l_j} \\
&= \delta^l_j \frac{\partial(\cancel{w^l_{jk}}\cancelto{0}{a^{l-1}_k} + b^l_j)}{\partial b^l_j} \\
&= \delta^l_j \cancelto{1}{\frac{\partial b^l_j}{\partial b^l_j}} \\
&= \delta^l_j
\end{aligned}$$

A.3. Nguyen–Widrow initialization

Nguyen-Widrow is a method for initialization of the weights of a NN to reduce training time[6].

Algorithm 2: Nguyen–Widrow initialization

Requires n: Number of input units ;
Requires p: Number of hidden units ;
Set Scale factor $\beta = 0.7p^{1/n}$;
Set Random $w_j \exists(-0.5, 0.5)$;
Set Random $b_j \exists(-1, 1)$;
foreach *hidden unit* $j = 1, 2, ..p$ **do**
    $w_j = \beta \frac{w_j(random)}{\|w_j(random)\|}$;
    $\| w_j \| = \sqrt{w_{1j}^2 + w_{2j}^2 + .. + w_{nj}^2}$;
    $b_j = \beta \cdot b_j(random)$
end

A.4. Adam optimizer

The Adaptive Moment Estimation (Adam) optimizer is an extension to the Stochastic Gradient Descent (SGD) based algorithm, as seen in Eq.2.42. Adam uses Momentum and Adaptive Learning Rates to converge faster. The momentum is a time parameter that accelerates SGD into a relevant direction. Adaptive learning rates allow the training algorithm to monitor the performance of the model and automatically adjust the learning rate for the best performance[62].

The Adam optimizer uses estimations of the first (mean) and second (uncentered variance) moments of the gradient to adapt the learning rate for each weight of a NN. As it only requires first-order gradients it memory requirement is too little. It uses the average of the second moments of the gradients and calculates the exponential moving average of gradients (m_t) and square gradients (v_t). The parameters β_1, β_2 are the control the decay rates of these moving averages.

The algorithm is the following:

Algorithm 3: Adam optimizer

Requires v: Stepsize;
Requires $\beta_1, \beta_2 \exists [0, 1)$: Exponential decay rates for the moment estimates;
Requires $f(\theta)$: Stochastic objective function with parameters θ;
θ_0: Initial parameter vector;
$m_0 \leftarrow 0$: Initialize 1st moment vector;
$v_0 \leftarrow 0$: Initialize 2nd moment vector;
$t \leftarrow 0$: Initialize timestep;
while θ_t *not converged* **do**
  $t \leftarrow t + 1$;
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ Get gradients at timestep t;
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t)$ Update biased first moment estimate;
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2)$ Update second raw moment estimate;
  $\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$ Compute bias-corrected first moment estimate;
  $\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$ Compute bias-corrected second raw moment estimate;
  $\theta_t \leftarrow \theta_{t-1} - \frac{\eta \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$ Update parameters;
end
Return θ_t Resulting parameters

where

$\epsilon \rightarrow 0$ is a term preventing division by zero

η is the learning rate

g_t^2 $= g_t \odot g_t$

A.5. dftb_in.hsd file for DFTB+

The module for ML at the dftb_in.hsd file used for the single point calculations:

```
Geometry = GenFormat {
    <<< "Structure"
}
Hamiltonian = DFTB{
   MaxAngularMomentum = {
      C = "p"
      H = "s"
      S = "d"
   }
   Charge = -1
   SCC = Yes
   SCCTolerance = 1e-05
   SlaterKosterFiles = Type2FileNames {
      Prefix = /usr/local/src/dftbplus-19.1/3ob-3-1/
      Separator = "-"
      Suffix = ".skf"
   }
   ThirdOrderFull = Yes
   HubbardDerivs {
      C = -0.1492
      S = -0.11
      H = -0.1857
   }
   HCorrection = Damping {
      Exponent = 4.0
   }
   MachineLearning = NeuralNet {
      SymmetryFunctions {
         Neighboursearching = Yes
         AtomicNumber = {
             S = 16
             C = 6
             H = 1
         }
         RadialCutoff = 10.
         RadialParameters {
         2.00000000e-01 3.12500000e+03
         6.36363636e-01 3.08673469e+02
         1.07272727e+00 1.08625395e+02
         1.50909091e+00 5.48882276e+01
         1.94545455e+00 3.30269019e+01
```

```
         2.38181818e+00 2.20339724e+01
         2.81818182e+00 1.57388137e+01
         3.25454545e+00 1.18012859e+01
         3.69090909e+00 9.17578684e+00
         4.12727273e+00 7.33810088e+00
         4.56363636e+00 6.00188886e+00
         5.00000000e+00 5.00000000e+00
         }
         AngularCutoff = 5.
         AngularParameters {
         1.0e-03  1.0e+00 -1.0e+00
         1.0e-03  1.0e+00  1.0e+00
         1.0e-03  4.0e+00 -1.0e+00
         1.0e-03  4.0e+00  1.0e+00
         1.0e-03  1.6e+01 -1.0e+00
         1.0e-03  1.6e+01  1.0e+00
         1.0e-02  1.0e+00 -1.0e+00
         1.0e-02  1.0e+00  1.0e+00
         1.0e-02  4.0e+00 -1.0e+00
         1.0e-02  4.0e+00  1.0e+00
         1.0e-02  1.6e+01 -1.0e+00
         1.0e-02  1.6e+01  1.0e+00
         5.0e-02  1.0e+00 -1.0e+00
         5.0e-02  1.0e+00  1.0e+00
         5.0e-02  4.0e+00 -1.0e+00
         5.0e-02  4.0e+00  1.0e+00
         5.0e-02  1.6e+01 -1.0e+00
         5.0e-02  1.6e+01  1.0e+00
         }
      }
      NeuralNetworkFiles = Type2Filenames {
         Prefix = $PATH/data_model/
         Suffix = "-subnet.param"
      }
   }
}
ParserOptions {
   IgnoreUnprocessedNodes = Yes
}
Options = {
  TimingVerbosity = -1
}
Analysis = {
  CalculateForces = Yes
}
```

A.6. Charge analysis

Different charge density populations were evaluated to determine the best model to fit. A histogram of all the charge distribution showed no greater difference between the models, as seen in Fig.A.1, a further analysis plotting the models against each other showed that the Löwdin model had a biomodal distribution as observed on the histogram, having a peak around 0.20 and 0.00. This model was discharged. The Mulliken charges were evaluated by two different softwares, ORCA and DFTB+, its plot showed that DFTB+ did overestimations on the charges. The Mulliken and Hirshfeld charges showed consistency between each other having a relation of almost 1:1.

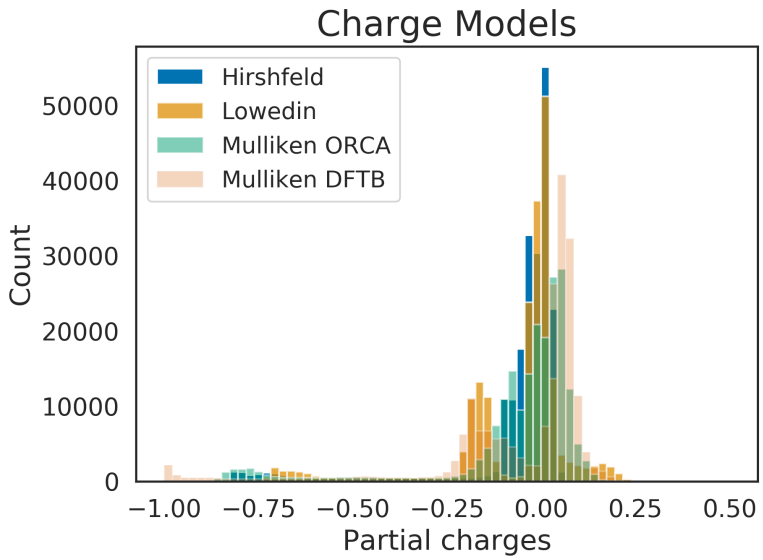


Figure A.1.: Histogram of charge distribution profiles of different charge population models.

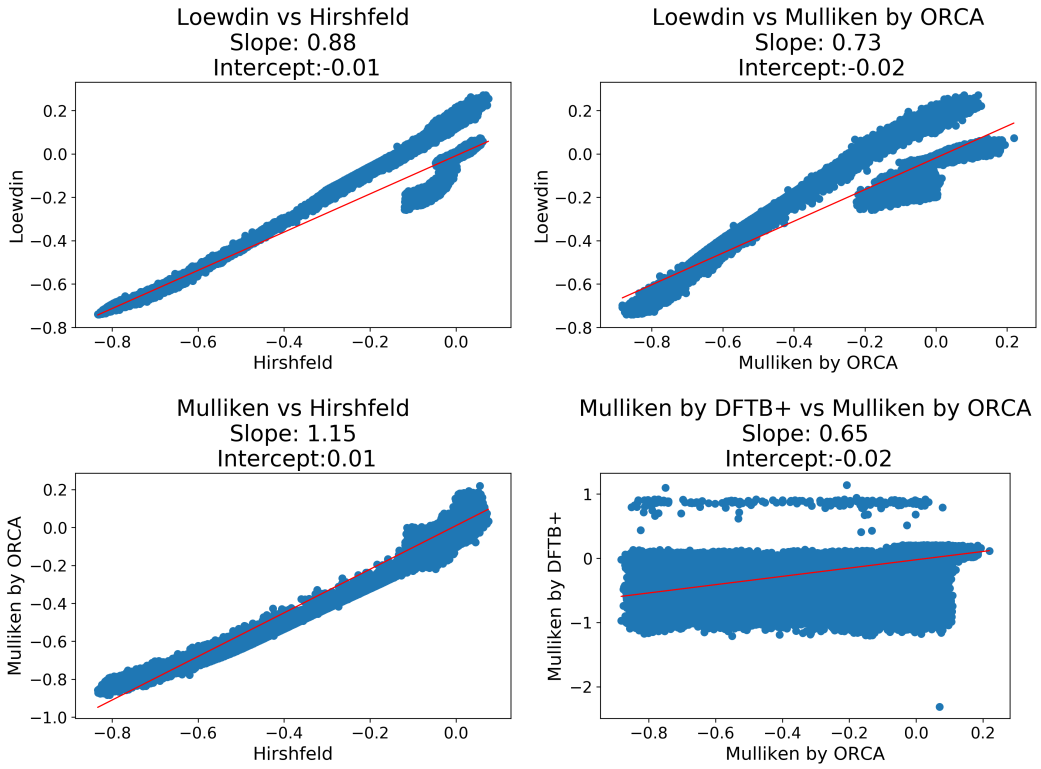


Figure A.2.: Correlation plots between different charge models of the dataset.

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor Prof. Marcus Elstner, who gave me the opportunity to make my PhD in his group and expand my knowledge into new areas.

I would like to pay my special regards to Dr. Tomáš Kubař for his patience in supervision and constant advice in several topics of my PhD research. As well to Dr. Celso R. Caldeira for his interest in the implementation of the ML codes into the workflows, friendly conversations and feedback on my work.

Thank you, to the Karlsruhe Institute of Technology and the Research Training Group 2450: "Tailored Scale-Bridging Approaches to Computational Nanoscience" for providing infrastructure and funding for this research.

Special thanks to Dr. Ali Thabet, Dr. Denis Maag and Dr. Mila Krämer for the professional and friendly discussions. Our conversations were vital for my professional growth as you were my guides into the topics of programming, QM/MM simulations and the Machine Learning field.

I extend my gratitude to my previous advisor Prof. Jose G. Sampedro and former colleagues of the Protein Biophysics Laboratory, you contributed to my personal and scientific formation.

I am profoundly grateful with my family, my parents Leticia and Ladislao, David and the little one for their love, patience and unconditional support. This endeavor would not have been possible without you.

To conclude, I cannot forget to thank my friends for the great moments. Rooby, Samaneh and Sara, you made the PhD such an enjoyable experience. And Ioana, for the cheering and smiles while writing this dissertation.

Bibliography

[1] Mark James Abraham et al. "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers". In: *SoftwareX* 1–2 (Sept. 2015), pp. 19–25. ISSN: 23527110. DOI: 10.1016/j.softx.2015.06.001. arXiv: arXiv:1503.05249v1. URL: https://www.sciencedirect.com/science/article/pii/S2352711015000059.

[2] Reinhart Ahlrichs et al. "Electronic structure calculations on workstation computers: The program system TURBOMOLE". In: *Chem. Phys. Lett.* 162.3 (1989), pp. 165–169. DOI: 10.1016/0009-2614(89)85118-8.

[3] Olga P Ajsuvakova et al. "Sulfhydryl groups as targets of mercury toxicity". In: *Coordination chemistry reviews* 417 (2020), p. 213343.

[4] B Alberts et al. *Molecular cell biology*. New York: Garland Science, 2008.

[5] Jorge Alegre-Cebollada et al. "Direct observation of disulfide isomerization in a single protein". In: *Nat. Chem.* 3 (2011), pp. 882–887.

[6] Ulfi Andayani et al. "Optimization backpropagation algorithm based on Nguyen-Widrom adaptive weight and adaptive learning rate". In: *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*. IEEE. 2017, pp. 363–367.

[7] W. Andreoni, A. Curioni, and T. Mordasini. "DFT-based molecular dynamics as a new tool for computational biology: First applications and perspective". In: *IBM Journal of Research and Development* 45.3.4 (May 2001), pp. 397–407. ISSN: 0018-8646, 0018-8646. DOI: 10.1147/rd.453.0397. URL: http://ieeexplore.ieee.org/document/5389048/ (visited on 04/23/2020).

[8] B. Aradi, B. Hourahine, and Th. Frauenheim. "DFTB+, a Sparse Matrix-Based Implementation of the DFTB Method †". en. In: *The Journal of Physical Chemistry A* 111.26 (July 2007), pp. 5678–5684. ISSN: 1089-5639, 1520-5215. DOI: 10.1021/jp070186p. URL: https://pubs.acs.org/doi/10.1021/jp070186p (visited on 04/23/2020).

[9] Peter W Atkins and Ronald S Friedman. *Molecular quantum mechanics*. Oxford university press, 2011.

[10] Robert D. Bach, Olga Dmitrenko, and Colin Thorpe. "Mechanism of Thiolate-Disulfide Interchange Reactions in Biochemistry". en. In: *The Journal of Organic Chemistry* 73.1 (Jan. 2008), pp. 12–21. ISSN: 0022-3263, 1520-6904. DOI: 10.1021/jo702051f. URL: https://pubs.acs.org/doi/10.1021/jo702051f (visited on 03/29/2021).

[11] Rodney J. Bartlett and Monika Musiał. "Coupled-cluster theory in quantum chemistry". en. In: *Reviews of Modern Physics* 79.1 (Feb. 2007), pp. 291–352. ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.79.291. URL: https://link.aps.org/doi/10.1103/RevModPhys.79.291 (visited on 03/29/2021).

[12] Axel D. Becke. "A new mixing of Hartree-Fock and local density-functional theories". In: *J. Chem. Phys.* 98.2 (1993), pp. 1372–1377.

[13] J Behler. "Representing potential energy surfaces by high-dimensional neural network potentials". In: *Journal of Physics: Condensed Matter* 26.18 (May 2014), p. 183001. ISSN: 0953-8984, 1361-648X. DOI: 10.1088/0953-8984/26/18/183001. URL: http://stacks.iop.org/0953-8984/26/i=18/a=183001?key=crossref.e2110f4e5f0e5ce5600d2eb9e27e4391 (visited on 01/27/2020).

[14] Jörg Behler. "Atom-centered symmetry functions for constructing high-dimensional neural network potentials". In: *The Journal of Chemical Physics* 134.7 (2011), p. 074106. DOI: 10.1063/1.3553717. eprint: https://doi.org/10.1063/1.3553717. URL: https://doi.org/10.1063/1.3553717.

[15] Jörg Behler. "Constructing high-dimensional neural network potentials: A tutorial review". en. In: *International Journal of Quantum Chemistry* 115.16 (Aug. 2015), pp. 1032–1050. ISSN: 00207608. DOI: 10.1002/qua.24890. URL: http://doi.wiley.com/10.1002/qua.24890 (visited on 01/27/2020).

[16] Jörg Behler. "Perspective: Machine learning potentials for atomistic simulations". en. In: *The Journal of Chemical Physics* 145.17 (Nov. 2016), p. 170901. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.4966192. URL: http://aip.scitation.org/doi/10.1063/1.4966192 (visited on 11/22/2019).

[17] Jörg Behler and Michele Parrinello. "Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces". en. In: *Physical Review Letters* 98.14 (Apr. 2007), p. 146401. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.98.146401. URL: https://link.aps.org/doi/10.1103/PhysRevLett.98.146401 (visited on 01/29/2020).

[18] Thomas B. Blank et al. "Neural network models of potential energy surfaces". en. In: *The Journal of Chemical Physics* 103.10 (Sept. 1995), pp. 4129–4137. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.469597. URL: http://aip.scitation.org/doi/10.1063/1.469597 (visited on 03/29/2021).

[19] Massimiliano Bonomi et al. "PLUMED: A portable plugin for free-energy calculations with molecular dynamics". In: *Comput. Phys. Commun.* 180.10 (2009), pp. 1961–1972.

[20] M. Born and R. Oppenheimer. "Zur Quantentheorie der Molekeln". In: *Annalen der Physik* 389.20 (1927), pp. 457–484.

[21] Lennard Böselt, Moritz Thürlemann, and Sereina Riniker. "Machine Learning in QM/MM Molecular Dynamics Simulations of Condensed-Phase Systems". In: *J. Chem. Theory Comput.* 17.5 (2021), pp. 2641–2658. DOI: 10.1021/acs.jctc.0c01112.

[22] P. J. Britto, Leslie Knipling, and J. Wolff. "The Local Electrostatic Environment Determines Cysteine Reactivity of Tubulin". In: *J. Biol. Chem.* 277.32 (2002), pp. 29018–29027. ISSN: 0021-9258. DOI: 10.1074/jbc.M204263200.

[23] Diego Butera et al. "Control of blood proteins by functional disulfide bonds". en. In: *Blood* 123.13 (Mar. 2014), pp. 2000–2007. ISSN: 0006-4971, 1528-0020. DOI: 10.1182/blood-2014-01-549816. URL: https://ashpublications.org/blood/article/123/13/2000/32588/Control-of-blood-proteins-by-functional-disulfide (visited on 04/23/2020).

[24] Yudong Cao et al. "Quantum Chemistry in the Age of Quantum Computing". In: *Chemical Reviews* 119.19 (2019). PMID: 31469277, pp. 10856–10915. DOI: 10.1021/acs.chemrev.8b00803. eprint: https://doi.org/10.1021/acs.chemrev.8b00803. URL: https://doi.org/10.1021/acs.chemrev.8b00803.

[25] Philippe Carl et al. "Forced unfolding modulated by disulfide bonds in the Ig domains of a cell adhesion molecule". In: *Proceedings of the National Academy of Sciences* 98.4 (2001), pp. 1565–1570.

[26] Samuel Chackalamannil, David P Rotella, and Simon E Ward. *Comprehensive medicinal chemistry III*. English. OCLC: 1065261603. 2017. ISBN: 978-0-12-803200-8 978-0-08-102240-5 978-0-08-102241-2 978-0-08-102242-9 978-0-08-102243-6 978-0-08-102244-3 978-0-08-102245-0 978-0-08-102246-7 978-0-08-102247-4.

[27] Christopher J Cramer. *Essential of computational chemistry: theories*. Oxford University Press, USA, 1994.

[28] Anna Dawid et al. *Modern applications of machine learning in quantum sciences*. 2022. DOI: 10.48550/ARXIV.2204.04198. URL: https://arxiv.org/abs/2204.04198.

[29] Marcel Deponte. "Glutathione catalysis and the reaction mechanisms of glutathione-dependent enzymes". In: *Biochimica et Biophysica Acta (BBA)-General Subjects* 1830.5 (2013), pp. 3217–3266.

[30] Edward C Eckels et al. "The mechanical power of titin folding". In: *Cell reports* 27.6 (2019), pp. 1836–1847.

[31] Edward C Eckels et al. "The work of titin protein folding as a major driver in muscle contraction". In: *Annual review of physiology* 80 (2018), pp. 327–351.

[32] Marcus Elstner. "The SCC-DFTB method and its application to biological systems". In: *Theoretical Chemistry Accounts* 116.1 (2006), pp. 316–325.

[33] Marcus Elstner and Gotthard Seifert. "Density functional tight binding". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 372.2011 (2014), p. 20120483.

[34] Somayeh Faraji et al. "High accuracy and transferability of a neural network potential through charge equilibration for calcium fluoride". In: *Physical Review B* 95.10 (2017), p. 104105.

[35] Matthias J. Feige, Ineke Braakman, and Linda M. Hendershot. "CHAPTER 1.1 Disulfide Bonds in Protein Folding and Stability". In: *Oxidative Folding of Proteins: Basic Principles, Cellular Regulation and Engineering*. The Royal Society of Chemistry, 2018, pp. 1–33. ISBN: 978-1-78262-990-0. DOI: 10.1039/9781788013253-00001. URL: http://dx.doi.org/10.1039/9781788013253-00001.

[36] E. Fermi. "Eine statistische Methode zur Bestimmung einiger Eigenschaften des Atoms und ihre Anwendung auf die Theorie des periodischen Systems der Elemente". de. In: *Zeitschrift fr Physik* 48.1-2 (Jan. 1928), pp. 73–79. ISSN: 1434-6001, 1434-601X. DOI: 10.1007/BF01351576. URL: http://link.springer.com/10.1007/BF01351576 (visited on 04/23/2020).

[37] Pedro Alexandrino Fernandes and Maria João Ramos. "Theoretical Insights into the Mechanism for Thiol/Disulfide Exchange". en. In: *Chemistry - A European Journal* 10.1 (Jan. 2004), pp. 257–266. ISSN: 0947-6539, 1521-3765. DOI: 10.1002/chem.200305343. URL: http://doi.wiley.com/10.1002/chem.200305343 (visited on 03/29/2021).

[38] Julio M. Fernandez and Hongbin Li. "Force-Clamp Spectroscopy Monitors the Folding Trajectory of a Single Protein". In: *Science* 303.5664 (2004), pp. 1674–1678. DOI: 10.1126/science.1092497.

[39] Max Feughelman. *Mechanical properties and structure of alpha-keratin fibres : wool, human hair and related fibres*. Sydney : UNSW Press, 1997. ISBN: 0868403598.

[40] Thomas E Fisher et al. "The study of protein mechanics with the atomic force microscope". In: *Trends in Biochemical Sciences* 24.10 (1999), pp. 379–384. DOI: https://doi.org/10.1016/S0968-0004(99)01453-X.

[41] Julia R Forman and Jane Clarke. "Mechanical unfolding of proteins: insights into biology, structure and folding". In: *Current Opinion in Structural Biology* 17.1 (2007), pp. 58–66. DOI: https://doi.org/10.1016/j.sbi.2007.01.006.

[42] Daan Frenkel et al. *Understanding molecular simulation*. Vol. 11. 4. American Institute of Physics, 1997, pp. 351–354.

[43] Pascal Friederich et al. "Machine-Learned Potentials for Next-Generation Matter Simulations". en. In: *Nat. Mater.* 20.6 (June 2021), pp. 750–761. ISSN: 1476-1122, 1476-4660. DOI: 10.1038/s41563-020-0777-6. URL: http://www.nature.com/articles/s41563-020-0777-6 (visited on 10/12/2021).

[44] Filipp Furche et al. "Turbomole". In: *WIREs Comput. Mol. Sci.* 4.2 (2014), pp. 91–100. DOI: 10.1002/wcms.1162.

[45] Yarin Gal and Zoubin Ghahramani. "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks". In: *arXiv:1512.05287 [stat]* (Oct. 2016). arXiv: 1512.05287. URL: http://arxiv.org/abs/1512.05287 (visited on 04/23/2020).

[46] Michael Gastegger and Philipp Marquetand. "Molecular Dynamics with Neural-Network Potentials". In: *arXiv:1812.07676 [physics, stat]* (Dec. 2018). arXiv: 1812.07676. URL: http://arxiv.org/abs/1812.07676 (visited on 11/22/2019).

[47] Michael Gaus, Albrecht Goez, and Marcus Elstner. “Parametrization and Benchmark of DFTB3 for Organic Molecules”. en. In: *Journal of Chemical Theory and Computation* 9.1 (Jan. 2013), pp. 338–354. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/ct300849w. URL: https://pubs.acs.org/doi/10.1021/ct300849w (visited on 04/23/2020).

[48] Michael Gaus et al. “Parameterization of DFTB3/3OB for Sulfur and Phosphorus for Chemical and Biological Applications”. en. In: *Journal of Chemical Theory and Computation* 10.4 (Apr. 2014), pp. 1518–1537. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/ct401002w. URL: https://pubs.acs.org/doi/10.1021/ct401002w (visited on 04/23/2020).

[49] David Giganti et al. “Disulfide isomerization reactions in titin immunoglobulin domains enable a mode of protein elasticity”. In: *Nature communications* 9.1 (2018), pp. 1–11.

[50] Hiram F. Gilbert. “Molecular and Cellular Aspects of Thiol-Disulfide Exchange”. In: *Advances in Enzymology - and Related Areas of Molecular Biology*. Ed. by Alton Meister. Hoboken, NJ, USA: John Wiley & Sons, Inc., Nov. 2006, pp. 69–172. DOI: 10.1002/9780470123096.ch2. URL: http://doi.wiley.com/10.1002/9780470123096.ch2 (visited on 04/06/2021).

[51] Garrett B. Goh, Nathan O. Hodas, and Abhinav Vishnu. “Deep Learning for Computational Chemistry”. In: *arXiv:1701.04503 [physics, stat]* (Jan. 2017). arXiv: 1701.04503. URL: http://arxiv.org/abs/1701.04503 (visited on 11/22/2019).

[52] Garrett B. Goh, Nathan O. Hodas, and Abhinav Vishnu. “Deep learning for computational chemistry”. In: *J. Comput. Chem.* 38.16 (2017), pp. 1291–1307. DOI: https://doi.org/10.1002/jcc.24764.

[53] Claudia L. Gómez-Flores et al. “Accurate Free Energies for Complex Condensed-Phase Reactions Using an Artificial Neural Network Corrected DFTB/MM Methodology”. In: *Journal of Chemical Theory and Computation* 18.2 (2022). PMID: 34978438, pp. 1213–1226. DOI: 10.1021/acs.jctc.1c00811. eprint: https://doi.org/10.1021/acs.jctc.1c00811. URL: https://doi.org/10.1021/acs.jctc.1c00811.

[54] Maja Gruden et al. “Benchmarking density functional tight binding models for barrier heights and reaction energetics of organic molecules”. en. In: *Journal of Computational Chemistry* 38.25 (Sept. 2017), pp. 2171–2185. ISSN: 01928651. DOI: 10.1002/jcc.24866. URL: http://doi.wiley.com/10.1002/jcc.24866 (visited on 04/23/2020).

[55] László Gyevi-Nagy, Mihály Kállay, and Péter R. Nagy. “Integral-Direct and Parallel Implementation of the CCSD(T) Method: Algorithmic Developments and Large-Scale Applications”. en. In: *Journal of Chemical Theory and Computation* 16.1 (Jan. 2020), pp. 366–384. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.9b00957. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.9b00957 (visited on 03/29/2021).

[56] Chris M. Handley and Paul L. A. Popelier. "Potential Energy Surfaces Fitted by Artificial Neural Networks". en. In: *The Journal of Physical Chemistry A* 114.10 (Mar. 2010), pp. 3371–3383. ISSN: 1089-5639, 1520-5215. DOI: 10.1021/jp9105585. URL: https://pubs.acs.org/doi/10.1021/jp9105585 (visited on 03/29/2021).

[57] Ask Hjorth Larsen et al. "The atomic simulation environment—a Python library for working with atoms". In: *Journal of Physics: Condensed Matter* 29.27 (July 2017), p. 273002. ISSN: 0953-8984, 1361-648X. DOI: 10.1088/1361-648X/aa680e. URL: https://iopscience.iop.org/article/10.1088/1361-648X/aa680e (visited on 04/23/2020).

[58] P. Hohenberg and W. Kohn. "Inhomogeneous Electron Gas". In: *Phys. Rev.* 136 (3B Nov. 1964), B864–B871.

[59] Scott A Hollingsworth and Ron O Dror. "Molecular dynamics simulation for all". In: *Neuron* 99.6 (2018), pp. 1129–1143.

[60] B. Hourahine et al. "DFTB+, a software package for efficient approximate density functional theory based atomistic simulations". In: 152.12 (2020), p. 124101. DOI: 10.1063/1.5143190.

[61] Frank Jensen. *Introduction to computational chemistry*. John wiley & sons, 2017.

[62] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[63] W. Kohn and L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". en. In: *Physical Review* 140.4A (Nov. 1965), A1133–A1138. ISSN: 0031-899X. DOI: 10.1103/PhysRev.140.A1133. URL: https://link.aps.org/doi/10.1103/PhysRev.140.A1133 (visited on 04/23/2020).

[64] W. Kohn and L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". In: *Phys. Rev.* 140 (4A 1965), A1133–A1138.

[65] Katra Kolšek, Camilo Aponte-Santamarıa, and Frauke Gräter. "Accessibility explains preferred thiol-disulfide isomerization in a protein domain". In: *Scientific reports* 7.1 (2017), pp. 1–10.

[66] Pallav Kosuri et al. "Protein folding drives disulfide formation". In: *Cell* 151.4 (2012), pp. 794–806.

[67] T. Kubař. https://github.com/tomaskubar/dftbplus/tree/machine-learning. last accessed on 18 February 2021. 2021.

[68] Manoj Kumar et al. "Genetic algorithm: Review and application". In: *Available at SSRN 3529843* (2010).

[69] Fabian Kutzki et al. "Force-Propelled Reduction and Exchange of Disulfide Bonds in Von Willebrand Factor's C4 Domain". In: *submitted* (2021).

[70] Andrew R Leach and Andrew R Leach. *Molecular modelling: principles and applications*. Pearson education, 2001.

[71] Wenjin Li and Frauke Gräter. “Atomistic evidence of how force dynamically regulates thiol/disulfide exchange”. In: *Journal of the American Chemical Society* 132.47 (2010), pp. 16790–16795.

[72] Dimitrios G. Liakos and Frank Neese. “Is It Possible To Obtain Coupled Cluster Quality Energies at near Density Functional Theory Cost? Domain-Based Local Pair Natural Orbital Coupled Cluster vs Modern Density Functional Theory”. In: *Journal of Chemical Theory and Computation* 11.9 (2015). PMID: 26575901, pp. 4054–4063. DOI: 10.1021/acs.jctc.5b00359. eprint: https://doi.org/10.1021/acs.jctc.5b00359. URL: https://doi.org/10.1021/acs.jctc.5b00359.

[73] Kresten Lindorff-Larsen et al. “Improved side-chain torsion potentials for the Amber ff99SB protein force field”. In: *Proteins* 78.8 (2010), pp. 1950–1958. DOI: 10.1002/prot.22711.

[74] Hui Lu et al. “Unfolding of Titin Immunoglobulin Domains by Steered Molecular Dynamics Simulation”. en. In: *Biophysical Journal* 75.2 (Aug. 1998), pp. 662–671. ISSN: 00063495. DOI: 10.1016/S0006-3495(98)77556-3. URL: https://linkinghub.elsevier.com/retrieve/pii/S0006349598775563 (visited on 04/23/2020).

[75] Denis Maag et al. “Electrostatic interactions contribute to the control of intramolecular thiol–disulfide isomerization in a protein”. In: *submitted* (2021).

[76] Donald Allan McQuarrie and John Douglas Simon. *Physical chemistry: a molecular approach*. Vol. 1. University science books Sausalito, CA, 1997.

[77] Seyedali Mirjalili. “Genetic algorithm”. In: *Evolutionary algorithms and neural networks*. Springer, 2019, pp. 43–55.

[78] Frank Neese. “Software Update: the ORCA Program System, Version 4.0”. In: *WIREs Comput. Mol. Sci.* 8.1 (2018), e1327. DOI: https://doi.org/10.1002/wcms.1327. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1327.

[79] Frank Neese. “The ORCA Program System”. In: *WIREs Comput. Mol. Sci.* 2.1 (2012), pp. 73–78. DOI: https://doi.org/10.1002/wcms.81. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.81.

[80] David L Nelson, Albert L Lehninger, and Michael M Cox. *Lehninger principles of biochemistry*. Macmillan, 2008.

[81] Rui P. P. Neves, Pedro Alexandrino Fernandes, and Maria João Ramos. “Mechanistic Insights on the Reduction of Glutathione Disulfide by Protein Disulfide Isomerase”. In: *Proc. Natl. Acad. Sci. U.S.A.* 114.24 (June 2017), E4724–E4733. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1618985114. (Visited on 02/11/2021).

[82] Rui P. P. Neves et al. “Benchmarking of Density Functionals for the Accurate Description of Thiol–Disulfide Exchange”. en. In: *Journal of Chemical Theory and Computation* 10.11 (Nov. 2014), pp. 4842–4856. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/ct500840f. URL: https://pubs.acs.org/doi/10.1021/ct500840f (visited on 03/29/2021).

[83] Michael A Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, USA, 2015.

[84] Augusto F. Oliveira et al. "Density-functional based tight-binding: an approximate DFT method". en. In: *Journal of the Brazilian Chemical Society* 20.7 (2009), pp. 1193–1205. ISSN: 0103-5053. DOI: 10.1590/S0103-50532009000700002. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-50532009000700002&lng=en&nrm=iso&tlng=en (visited on 04/23/2020).

[85] J. R. Oppenheimer. "Zur Quantentheorie kontinuierlicher Spektren". de. In: *Zeitschrift für Physik A Hadrons and nuclei* 41.4-5 (Aug. 1927), pp. 268–293. ISSN: 0939-7922. DOI: 10.1007/BF01391242. URL: http://link.springer.com/10.1007/BF01391242 (visited on 03/29/2021).

[86] Leslie B Poole. "The basics of thiols and cysteines in redox biology and chemistry". In: *Free Radical Biology and Medicine* 80 (2015), pp. 148–157.

[87] Marina Putzu et al. "On the mechanism of spontaneous thiol–disulfide exchange in proteins". en. In: *Physical Chemistry Chemical Physics* 20.23 (2018), pp. 16222–16230. ISSN: 1463-9076, 1463-9084. DOI: 10.1039/C8CP01325J. URL: http://xlink.rsc.org/?DOI=C8CP01325J (visited on 08/06/2019).

[88] Meng Qin, Wei Wang, and D. Thirumalai. "Protein folding guides disulfide bond formation". In: *Proc. Natl. Acad. Sci. USA* 112.36 (2015), pp. 11241–11246. DOI: 10.1073/pnas.1503909112.

[89] Raghunathan Ramakrishnan et al. "Big Data Meets Quantum Chemistry Approximations: The -Machine Learning Approach". en. In: *Journal of Chemical Theory and Computation* 11.5 (May 2015), pp. 2087–2096. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.5b00099. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.5b00099 (visited on 01/29/2020).

[90] Celso RC Rêgo and Claudia L. Gomez-Flores. https://github.com/KIT-Workflows/DFTB-Neural-Net.git. 2022.

[91] Celso RC Rêgo et al. "SimStack: An Intuitive workflow framework". In: *Frontiers in Materials* (), p. 283.

[92] Wilfried Roetzel, Xing Luo, and Dezhen Chen. "Optimal design of heat exchanger networks". en. In: *Design and Operation of Heat Exchangers and their Networks*. Elsevier, 2020, pp. 231–317. ISBN: 978-0-12-817894-2. DOI: 10.1016/B978-0-12-817894-2.00006-6. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780128178942000066 (visited on 02/28/2021).

[93] Tamar Schlick. *Molecular modeling and simulation: an interdisciplinary guide*. Vol. 2. Springer, 2010.

[94] Tamar Schlick and Stephanie Portillo-Ledesma. "Biomolecular modeling thrives in the age of technology". In: *Nature computational science* 1.5 (2021), pp. 321–331.

[95] Johannes Schmidt-Hieber. "The Kolmogorov–Arnold representation theorem revisited". In: *Neural Networks* 137 (2021), pp. 119–126. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2021.01.020. URL: https://www.sciencedirect.com/science/article/pii/S0893608021000289.

[96] K. T. Schütt et al. “SchNet – A Deep Learning Architecture for Molecules and Materials”. In: *J. Chem. Phys.* 148.24 (2018), p. 241722. DOI: 10.1063/1.5019779.

[97] Carolyn S Sevier and Chris A Kaiser. “Formation and transfer of disulphide bonds in living cells”. In: *Nature reviews Molecular cell biology* 3.11 (2002), pp. 836–847.

[98] Lin Shen and Weitao Yang. “Molecular Dynamics Simulations with Quantum Mechanics/Molecular Mechanics and Adaptive Neural Networks”. en. In: *Journal of Chemical Theory and Computation* 14.3 (Mar. 2018), pp. 1442–1455. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.7b01195. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.7b01195 (visited on 03/19/2021).

[99] Grayson H. Snyder et al. “Electrostatic influence of local cysteine environments on disulfide exchange kinetics”. In: *Biochemistry* 20.23 (1981), pp. 6509–6519. DOI: 10.1021/bi00526a001.

[100] Grayson H. Snyder et al. “Use of local electrostatic environments of cysteines to enhance formation of a desired species in a reversible disulfide exchange reaction”. In: *Biochim. Biophys. Acta, Protein Struct. Mol. Enzymol.* 749.3 (1983), pp. 219–226. DOI: 10.1016/0167-4838(83)90228-5.

[101] Wolfram Stacklies et al. “Mechanical Network in Titin Immunoglobulin from Force Distribution Analysis”. In: *PLoS Comput. Biol.* 5.3 (2009), e1000306. DOI: 10.1371/journal.pcbi.1000306.

[102] L. H. Thomas. “The calculation of atomic fields”. en. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 23.5 (Jan. 1927), pp. 542–548. ISSN: 0305-0041, 1469-8064. DOI: 10.1017/S0305004100011683. URL: https://www.cambridge.org/core/product/identifier/S0305004100011683/type/journal_article (visited on 04/23/2020).

[103] Gareth A. Tribello et al. “PLUMED 2: New feathers for an old bird”. In: *Comput. Phys. Commun.* 185.2 (2014), pp. 604–613.

[104] Ko Tsz Wai et al. “A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer”. In: *Nature Communications* 12 (Jan. 2021). DOI: 10.1038/s41467-020-20427-2.

[105] Christina Eleftheria Tzeliou, Markella Aliki Mermigki, and Demeter Tzeli. “Review on the QM/MM Methodologies and Their Application to Metalloproteins”. In: *Molecules* 27.9 (2022), p. 2660.

[106] Oliver T. Unke et al. “Machine Learning Force Fields”. In: *Chemical Reviews* 121.16 (2021). PMID: 33705118, pp. 10142–10186. DOI: 10.1021/acs.chemrev.0c01111. eprint: https://doi.org/10.1021/acs.chemrev.0c01111. URL: https://doi.org/10.1021/acs.chemrev.0c01111.

[107] Anthony Yu-Tung Wang et al. “Machine Learning for Materials Scientists: An Introductory Guide toward Best Practices”. en. In: *Chemistry of Materials* 32.12 (June 2020), pp. 4954–4965. ISSN: 0897-4756, 1520-5002. DOI: 10.1021/acs.chemmater.0c01907. URL: https://pubs.acs.org/doi/10.1021/acs.chemmater.0c01907 (visited on 02/28/2021).

[108] Chih-Hsien Wang et al. "The effect of disulfide bonds on protein folding, unfolding, and misfolding investigated by FT-Raman spectroscopy: Disulfide bonds on protein folding, unfolding, and misfolding". en. In: *Journal of Raman Spectroscopy* 47.8 (Aug. 2016), pp. 940–947. ISSN: 03770486. DOI: 10.1002/jrs.4935. URL: http://doi.wiley.com/10.1002/jrs.4935 (visited on 05/12/2020).

[109] Julia Westermayr et al. "Perspective on integrating machine learning into computational chemistry and materials science". In: *The Journal of Chemical Physics* 154.23 (2021), p. 230903.

[110] Arun P. Wiita et al. "Force-dependent chemical kinetics of disulfide bond reduction observed with single-molecule techniques". In: *Proc. Natl. Acad. Sci. USA* 103.19 (2006), pp. 7222–7227. DOI: 10.1073/pnas.0511035103.

[111] Janet M. Wilson, Robert J. Bayer, and D. J. Hupe. "Structure-Reactivity Correlations for the Thiol-Disulfide Interchange Reaction". In: *J. Am. Chem. Soc.* 99.24 (Nov. 1977), pp. 7922–7926. ISSN: 0002-7863. DOI: 10.1021/ja00466a027. URL: https://doi.org/10.1021/ja00466a027 (visited on 02/23/2021).

[112] Emma-Ruoqi Xu et al. "Structure and Dynamics of the Platelet Integrin-Binding C4 Domain of Von Willebrand Factor". In: *Blood* 133.4 (Jan. 2019), pp. 366–376. ISSN: 0006-4971. DOI: 10.1182/blood-2018-04-843615. URL: https://doi.org/10.1182/blood-2018-04-843615 (visited on 04/30/2021).

[113] Jinzhe Zeng et al. "Development of Range-Corrected Deep Learning Potentials for Fast, Accurate Quantum Mechanical/Molecular Mechanical Simulations of Chemical Reactions in Solution". In: *J. Chem. Theory Comput.* published (2021), DOI 10.1021/acs.jctc.1c00201. DOI: 10.1021/acs.jctc.1c00201.

[114] Igor Ying Zhang and Andreas Grüneis. "Coupled Cluster Theory in Materials Science". In: *Frontiers in Materials* 6 (June 2019), p. 123. ISSN: 2296-8016. DOI: 10.3389/fmats.2019.00123. URL: https://www.frontiersin.org/article/10.3389/fmats.2019.00123/full (visited on 03/29/2021).

[115] Junmian Zhu et al. "Artificial neural network correction for density-functional tight-binding molecular dynamics simulations". In: *MRS Communications* 9.3 (Sept. 2019), pp. 867–873. ISSN: 2159-6859, 2159-6867. DOI: 10.1557/mrc.2019.80. URL: https://www.cambridge.org/core/product/identifier/S2159685919000806/type/journal_article (visited on 01/29/2020).

Acronyms

ACSF Atom-Centered Symmetry Functions. 37–39, 48, 62, 64, 69

Adam Adaptive Moment Estimation. 56, 62, 64, 77

AI Artificial Intelligence. 26

B3LYP Becke 3-parameter exact exchange energy and the LYP correlation. 18, 34, 54–57, 59–61, 71, 74

B88 Becke gradient exchange correction. 18, 73

BLYP Becke-Lee-Yang-Parr exchange-correlation functional. 18, 73

BOA Born–Oppenheimer approximation. 10, 22

CBS Complete Basis Set. 34

CC Coupled cluster theory. 15, 55, 56, 59, 61, 71, 72

CCSD Coupled Cluster Singles and Doubles. 15

CCSD(T) Coupled Cluster Singles and Doubles with perturbative triples correction. 15, 34, 42, 54, 55, 57, 59–61

CENT Charge Equilibration via Neural network Technique. 48, 62, 63, 65, 66, 69

DFT Density Functional Theory. vii, 3, 4, 16–19, 25, 35, 59, 61, 71

DFTB Density-Functional based Tight-binding. i, iii, 3–5, 18–20, 25, 40, 51, 54–56, 61, 66, 71, 72

DL Deep Learning. 26

DLPNO Domain-based Local Par Orbitals. 42, 54, 55, 57, 60, 61

DNA Deoxyribonucleic Acid. 1, 2

ESP Electrostatic Potential. 71

FF Force Field. 22, 23

GA Geneteic Algorithm. 39, 55, 56, 58, 72

gDIIS Geometry optimisation using Direct Inversion in the Iterative Subspace. 42

GGA Generalized Gradient Approximations. 18, 19, 34, 35, 71, 73

GPU Graphical Processor Unit. 56

GUI Graphical User Interface. 6, 41, 69, 72

HF Hartree–Fock. 11, 14, 15, 18

HPC High Performance Computing. 41

KS Kohn-Sham. 19

LCAO Linear Combination of Atomic Orbitals. 12

LDA Local (spin-) Density Approximation. 17–19, 34, 35, 73, 74

LYP Lee-Yang-Parr correlation functional. 18, 73

MAE Mean Absolute Error. 43

MD Molecular dynamics. 2, 22, 24, 25, 38

ML Machine Learning. v, viii, 5, 23, 26–29, 37, 39–41, 43, 45, 47, 58–61, 69, 71, 72

MM Molecular mechanics. 5, 25, 72

MO Molecular orbital theory. vii, 12

MP2 Møller and Pesset second order correction. 15, 34

NN Neural Network. vii, 3–5, 29, 30, 38–40, 48, 49, 55, 56, 58, 59, 61–64, 66, 69, 76, 77

PBE Perdew-Burke-Ernzerhof functional. 35, 74

PES Potential Energy Surface. viii, 3, 5, 6, 10, 22, 23, 37, 55, 59, 60, 66, 67, 72

pH Potential of hydrogen. 34

QM Quantum mechanics. 25, 35, 51–53, 71, 72

QM/MM Quantum Mechanics/Molecular Mechanics. i, iii, v, 2, 4, 5, 25, 35, 51, 53, 54, 57, 69, 71, 72, 83

RMSE Root Mean Sqaured Error. 43, 55, 56, 58, 63, 64

RNA Ribonucleic Acid. 1

ROS Reactive oxygen species. 34

SCC Self Consistent Charge. 20, 42

SCF Self-Consistent Field. 14, 21, 25

SE Schrödinger equation. 9–11

SGD Stochastic Gradient Descent. 77

TIP3P Transferable Intermolecular Potential with 3 Points. 51

TS Transition state. 33, 52, 53

WaNo Workflow Active Node. 41–45

XML Extensivle Markup Language. 41