Development of Machine Learning Enhanced Density Functional Tight Binding Parametrization

Dissertation
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
vorgelegt dem
Fachbereich Physik / Elektrotechnik der Universität Bremen

Guozheng Fan
Bremen Center for Computational Materials Science
University of Bremen
Bremen, Germany

Primary supervisor: Dr. Bálint Aradi
Secondary supervisor: Prof. em. Thomas Frauenheim
Defense date: 06 November 2023

Reviewer: Dr. Bálint Aradi
Reviewer: Prof. Dr. Thomas A. Niehaus

Universität Bremen

Universität Bremen | Postfach 33 04 40, 28334 Bremen

An
- Herrn Guozheng Fan
- die Mitglieder des Prüfungsausschusses

Fachbereich 01
Physik / Elektrotechnik

Doris Pérez Fraga
Geschäftsstelle Promotionsausschuss Dr. rer. Nat.

S2370
Otto-Hahn-Allee 1

28359 Bremen

Tel. 218-62710

E-Mail: dperez@fb1.uni-bremen.de
www.uni-bremen.de/fb1/studium/promotion

Bremen, d. 04.10.2023

EINLADUNG

zum Dissertationskolloquium von

Herrn Guozheng Fan

am: 06.11.2023 um 14:00 Uhr st. Raum: 1020/1030, AIB Building, Hochschulring 40, 28359 Bremen

Titel der Dissertation:

Development of Machine Learning Enhanced Density Functional Tight Binding Parametrization

1. Gutachter :	Herr Dr. Bálint Aradi
2. Gutachter:	Herr Prof. Dr. Thomas Niehaus
1. Prüfer:	Herr Prof. Dr. Gordon Callsen
2. Prüfer:	Herr PD Dr. Christopher Gies
akad. Mitarbeiter:	Herr Dr. Carlos Raul Medrano
Student:	Herr Jonas Müller

Mit freundlichem Gruß
Im Auftrag

D. Pérez Fraga

Prüfungsamt

Abstract

Density functional tight binding (DFTB) theory is an approximate method derived from density functional theory (DFT). Accurate and transferable parametrization is one of the key issues of DFTB development. Over the past two decades, machine learning (ML) has expanded significantly in physics, chemistry, and materials science, which also shows a potential application in the DFTB parametrization. This thesis concentrates on the parametrization of DFTB through both traditional and machine learning based methods.

First, we have focused on parametrizing a solid-state battery system consisting of lithium, phosphorus, sulfur, and chlorine elements, which shows great potential as a solid-state electrolyte. The resulting DFTB parametrization of the electronic and repulsive components yields reasonable accuracy of band structures and optimized geometries of DFTB calculations, comparable to the results of DFT calculations. Second, we have introduced the tight-binding machine learning toolkit (TBMaLT), an open source framework designed to incorporate physical insights into machine learning to predict quantum mechanical properties. The toolkit contains the DFTB layer with flexible interfaces that allow for the generations of Hamiltonian and overlap matrices. We have comprehensively described the DFTB layer and machine learning methodologies employed in TBMaLT, and a detailed analysis of the implementation features.Third, we have explored the applications of TBMaLT in molecular systems. The DFTB-ML workflow enables the optimization of electronic properties by generating two-centre integrals, either by training the basis function parameters (compression radii) or directly optimizing diatomic integrals. The onsite energies were also tuned. All machine learning approaches have successfully improved electronic property predictions, and multiple electronic properties can be optimized simultaneously for all approaches. Training on the basis functions yielded more consistent results of different electronic properties, with the obtained Hamiltonian and overlap matrices falling within physically reasonable ranges. Finally, we have extended the DFTB-ML framework to incorporate periodic boundary conditions, including bulk systems with different lattice types, defect systems, and slab systems consisting of silicon and carbon elements, as the training and testing systems. The reference property for the machine learning was based on band structures obtained through DFT calculations using a hybrid functional. The DFTB-ML model enables the improvement of band structure calculations across various chemical environments, showcasing the capacity of the DFTB-ML framework to predict band structures with high accuracy of the hybrid functional level at an approximate method computational cost. Besides, the DFTB-ML model also exhibits excellent scaling transferability, enabling training on small systems and prediction on larger ones.

List of Figures

List of Tables

List of Abbreviations

MBSE many-body Schrödinger equation

HF Hartree-Fock

DFT density functional theory

KS Kohn-Sham

LDA local density approximation

GGA generalized gradient approximation

PBE Perdew-Burke-Ernzerhof

HSE Heyd-Scuseria-Ernzerhof

DFTB density functional tight binding

SCC-DFTB self-consistent charge density functional tight binding

SCF self-consistent field

MD molecular dynamics

PSO particle swarm optimization

CCS curvature constrained spline

PBCs periodic boundary conditions

DOS density of states

CPA charge population analysis

VBM valence band maximum

CBM conduction band minimum

NNs neural networks

MLP multilayer perceptron

CNNs convolutional neural networks

SVM support vector machine

SGD stochastic gradient descent

BP backpropagation

ACSFs atom centered symmetry functions

SOAP smooth overlap of atomic positions

MAEs mean absolute errors

MSEs mean square errors

TBMaLT tight binding machine learning toolkit

LIBs lithium-ion batteries

Contents

1 Introduction

In the last several decades, there has been an increase in the accumulation of data from computational simulations and experiments, along with high-throughput computational approaches. As a consequence, the analysis of emerging data and extraction of knowledge requires the utilization of various innovative techniques, including machine learning. Within the field of computational science, density functional tight binding (DFTB) theory is an approximate method that enables efficient simulation of large systems. To perform DFTB calculations on a specific system, a parametrization process is required to generate the so-called Slater-Koster tables for different atomic pairs. It is worth noting that the accuracy of DFTB calculations heavily depends on the quality of the chosen parametrization. With the rapid development of machine learning algorithms and hardware, machine learning applications have expanded significantly in physics, chemistry, and materials science, with potential applications in the DFTB parametrization.

This thesis concentrates on the parametrization of DFTB through traditional and machine learning-based approaches. In this chapter, we will provide a brief overview of the development of computational science. We will then discuss the development of artificial intelligence and its applications in data-driven science. Next, we will focus on DFTB and its parametrization, as well as the state-of-the-art applications of data-driven techniques in DFTB and other semiempirical methods. The successful combination of machine learning and theoretical methods has motivated our work in this thesis, which focuses on improving the performance of DFTB calculations by incorporating machine learning.

1.1 Development of the Science Paradigms

The development of scientific paradigms has reshaped scientific research. Schleder [1] provides a historical perspective on the development of scientific paradigms, as depicted in Figure 1.1. The first paradigm was an empirical science paradigm, while the second paradigm that emerged in the last few centuries focused on theoretical science and generated many successful theoretical models, including the Schrödinger equation. The complexity of the Schrödinger equation [2] makes it challenging to generate analytical solutions for real material systems. Consequently, many efforts have been made to simplify the Schrödinger equation, and methods in computational science have become feasible solutions in various fields [3]. Hartree-Fock theory (HF) and density functional theory (DFT) have been widely used among these methods. In particular, DFT has become one of the most popular methods for electronic structure calculations due to its high accuracy and efficiency. For larger systems, approximate methods such as DFTB can provide reasonable accuracy while computationally significantly more efficient than DFT. Developing efficient and accurate computational methods has played a significant role in advancing scientific research, leading to a third paradigm. With the aid of computational methods, scientists are now able to tackle problems that were previously impossible to solve with reasonable accuracy.

Data-driven techniques, such as artificial intelligence (AI), have demonstrated their

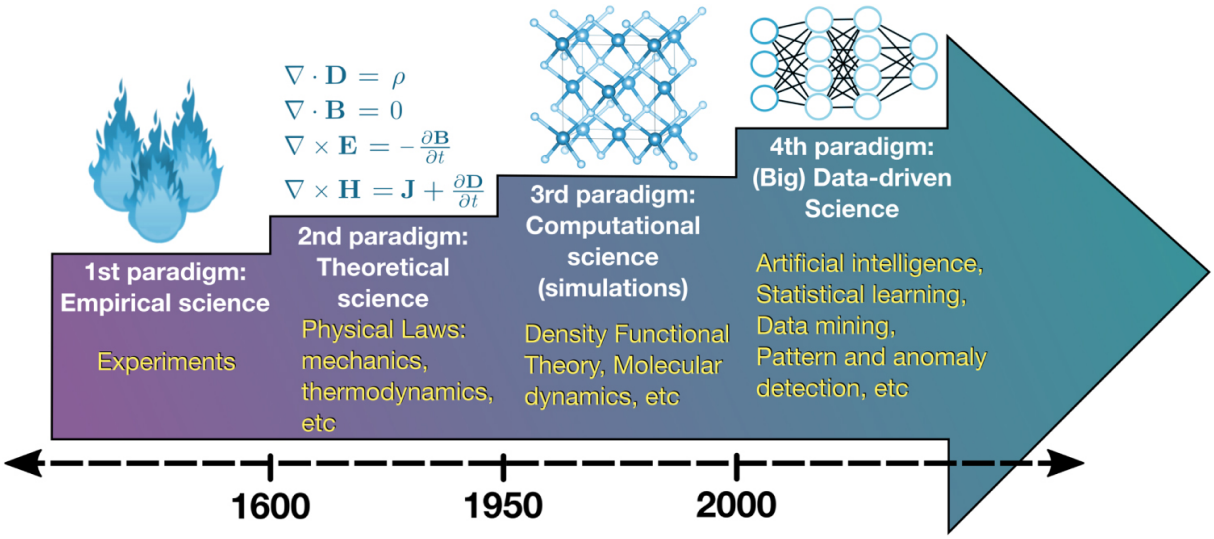


Figure 1.1: The science paradigms in human history: empirical, theoretical, computational, and data-driven. Adapted from reference [1].

power in many scientific areas [1, 4, 5]. AI is a general concept that aims at creating intelligent machines. In 1950, Alan Turing posed the question, "*Can machines think?*" which became known as the Turing test [6]. The Turing test involves determining whether a machine can exhibit human intelligence. The last several decades have witnessed the emergence of AI or machine learning as a powerful tool in various fields, including natural language processing (NLP) [7, 8] and computer vision (CV) [9]. Deep learning is a subset of the machine learning algorithm family based on neural networks (NNs). NNs are inspired by biological neurons and have been widely used in various fields, such as speech recognition, object recognition, object detection, and genomics [10]. The first generation of NNs is the multi-layer perceptron (MLP) [11], which was introduced in 1967 by Amari. In the 1980s, the backpropagation algorithm [12] was developed and applied to convolutional neural networks (CNNs), which are particularly effective for image-processing tasks. The field of deep learning has experienced substantial progress in recent times, particularly with the introduction of notable techniques such as generative adversarial networks (GANs) [13], long short-term memory (LSTM) [14], and transformer [15]. These developments have profoundly impacted the field, leading to significant improvements in various applications. Recently, transformer-based models [15] such as generative pre-trained transformer (GPT) [16], and bidirectional encoder representations from transformers (BERT) [17] have achieved state-of-the-art performance in various fields, including language processing, sentiment analysis, and science.

Data-driven science can be a pure artificial intelligence model or a combination of approaches developed in the computational science paradigm. Combining machine learning with underlying physical models has shown promising results in producing more transferable predictions of quantum mechanical properties. Previous reviews have extensively covered the application of machine learning in these fields [4, 18, 19, 20]. Machine learning based force fields show great promise by combining the accuracy of *ab initio* methods with the efficiency of force fields. Two widely used machine learning algorithms are kernel

based and neural networks based methods, both of which have succeeded in numerous applications. For example, in the field of atomic simulations, the Gaussian approximation potential (GAP) [21] based force field has reproduced DFT calculations for bulk and nanostructured phosphorus systems and has been used to calculate the transition between molecular and network liquid phases. Neural networks based force fields typically extract distance and angle information from a set of coordinates to reproduce energies, potential energy surfaces, and other properties. Examples include the Behler and Parrinello neural network (BPNN) [22] and the deep tensor neural network (DTNN) [23].

1.2 DFTB as a Computational Approach

Many efforts have been made to simplify the Schrödinger equation. DFT is a quantum mechanical method used to describe the physical properties of atoms, molecules, and solids. Hohenberg and Kohn proposed the fundamental principles in 1964 [24], and later the so-called Kohn-Sham equation [25] made DFT a more practical tool for theoretical calculations. The key idea of DFT is based on the electron density as the fundamental variable, which avoids the "exponential wall" of the computational complexity as found in wavefunction methods. By applying the self-consistent field method (SCF), we can solve the Kohn-Sham equation and generate physical properties. However, approximate methods, such as DFTB, are practical choices for large systems with more than hundreds of atoms.

The non-self-consistent-charge DFTB (non-SCC DFTB) is an approximate method derived from DFT and is based on a linear combination of pseudo-atomic orbitals (LCAO) basis set [26]. The self-consistent-charge DFTB (SCC-DFTB) method has been derived [27] by applying the second-order expansion of the Kohn-Sham total energy in DFT with respect to the charge fluctuations. Currently, Hamiltonian and overlap matrices for DFTB calculations are built based on pre-calculated parameter sets, and no integrals need to be calculated during runtime. This feature makes the computational speed comparable to other semiempirical simulation methods. The accuracy of DFTB calculations depends on the pre-calculated Slater-Koster tables obtained from the DFTB parametrization. The parametrization of DFTB can be classified into two parts: electronic parametrization and repulsive parametrization. Numerous approaches have been developed to generate both sets of parameters. Electronic parametrization usually involves tuning the basis parameters to minimize the errors of band structures between DFT and DFTB calculations. On the other hand, the repulsive parametrization is usually based on energies from a series of geometries. With the electronic and repulsive parametrization, Slater-Koster tables can be generated and are ready for DFTB calculations. The good balance between efficiency and accuracy makes DFTB suitable for large systems and large time scales when compared with DFT. DFTB has been widely applied in various fields, including physics, chemistry, materials science, and biology [28, 29, 30]. These applications include large organic molecules [28, 31], metal-organic frameworks [32], and proteins [33, 34].

Electronic Parametrization

The parameters to be tuned in electronic parametrization are responsible for confining the atomic orbitals and the electronic densities [27, 35]. The confinement of orbitals and electronic densities can improve the performance of DFTB calculations since the orbitals and densities of free atoms are usually too diffuse and unsuitable for calculations of accurate two-centre integrals [35]. The parameters to be tuned in electronic parametrization are so-called compression radii.

Regarding electronic parametrization, previous studies suggest that reasonably accurate results can be obtained by employing a compression radius that is 1.85 times the covalent radius of the corresponding elemental species [35]. The empirical values based on covalent radius can not satisfy the electronic parametrization in many systems, and the compression radii need further tuning in many cases. Many methods and toolkits [36, 37, 38] have been developed to generate and optimize the parameters in electronic parametrization. Markov et al. developed a toolkit `SKPAR` [39] using particle swarm optimization (PSO), which can be used to optimize electronic parameters in DFTB, such as compression radii. Jenness et al. [40] developed an automatic parametrization which optimises scaling factors for the covalent radii in the confinement term without the need for empirical fitting. Then the optimized scaling factor can be used for generating diatomic Hamiltonian and overlap integrals.

With the electronic parametrization of DFTB, many parameter sets can be used for DFTB electronic calculations. Wahiduzzaman et al. [38] proposed a parametrization scheme to generate electronic parameter sets of DFTB that covers the periodic table. The band structures were tested on over 100 systems using the generated parameter set that outperformed band structures from previous parameters [38]. Markov et al. [41, 42] developed a parameter set `siband-1-1` which contains DFTB electronic parametrization for Si, O and H systems and this parameter set can reproduce band structures of Si/SiO_2 from experimental values.

Repulsive Parametrization

Repulsive parametrization enables DFTB to predict total energies and forces. The repulsive parametrization aims to achieve results close to reference values, usually DFT. The physical properties used in the repulsive parametrization can be energies, forces or Hessians [43]. Many toolkits have been developed to generate repulsive terms. The methods used to generate repulsive term includes genetic algorithms (GAs) [44], PSO [36], curvature constrained splines (CCS) [45, 46] and Chebyshev interaction model for efficient simulation (ChIMES) [47].

In the early stage of the repulsive parametrization, the fitting process has been hand-constructed [27] for diatomic element pairs. Knaup et al. [48] tried the initial step towards automating the fitting of the repulsive potential using splines combined with a genetic algorithm. Bossche et al. [44] developed TANGO (Tight-binding Approximation eNhanced

Global Optimization), which can realize an automatic parametrization for the pairwise repulsive potentials. Pairwise splines have been frequently used in repulsive parametrization [48]. The common problems of splines are oscillatory behaviour and nonmonotonocity. Krishna et al. [46] proposed constraints on the curvature of the repulsive potential to solve the aforementioned problems. Two-body potentials lack flexibility for large data sets and complex chemical environments. A many-body Chebyshev polynomials [47, 49] has been applied for repulsive potential fitting and achieved promising results in various systems. With parameters from repulsive parametrization, DFTB can be used to calculate energies and forces. One popular parameter set `mio-1-1` [27] has been developed and widely used in energy calculations and geometry optimizations for organic molecules. The organic and biological systems (`3ob-3-1`) parameter set [50] has an overall improved performance compared with the `mio-1-1` set, especially for non-covalent systems, and hydrogen bond in the water dimer.

1.3 Combining Machine Learning and Approximate Methods

Approximate methods derived from HF and DFT [51, 52] are often used for systems that can not afford the use of more accurate methods like DFT and HF. Further efforts are required to enhance the formalism and parametrization of these approximate methods, in order to achieve a reasonable level of accuracy for diverse and intricate chemical environments. Despite the efforts, the approximated formalism and parameters limitation restricts the applications of the approximate methods.

The machine learning technique in the fourth paradigm can be a promising solution to the aforementioned problems. Machine learning combined with approximate methods have been applied and achieved significant improvements [53]. The combination of machine learning and approximate methods aims to leverage the strengths of both approaches while mitigating their respective weaknesses. Approximate methods are known for their transferability and quantum insights, while machine learning excels in predicting accurate physical properties and extracting patterns from complex chemical environments. The most straightforward method of combing machine learning and approximate methods (or low-level methods) is Δ-machine learning [54]. Δ-machine learning trained the difference between approximate methods and more accurate reference data. Δ-machine learning has been applied in various applications, such as improving semiempirical method PM7 model performance [55], improving solution-phase molecular properties predictions [56], etc.

Another popular strategy [57, 58] is incorporating machine learning with static parameters in semiempirical methods. Zhou et al. [57] have developed an interpretable Hamiltonian-based model by incorporating a quantum chemistry framework into a deep neural network. This was achieved by replacing static parameters with machine-learned values inferred from the local environment. The strategy of incorporating machine learning with parameters in semiempirical methods has been accelerated by the development of machine learning frameworks, such as PyTorch, which enables easy parameter optimization with the help of backward gradient updates [59].

When combining machine learning and DFTB, most of the previous work focuses on repulsive fitting [60, 61, 62, 63, 64]. Kranz [60] used unsupervised machine learning to learn two-body repulsive potentials. Panosetti et al. [64] proposed kernel based methods and displayed significantly improved accuracy for force predictions compared with `3ob-3-1` set. Stöhr [61] used a deep tensor neural network to learn many-body repulsive potentials. Bissuel and co-workers [62] investigated the machine-learned repulsive potentials for the pure silicon system and achieved good performance on energetic, vibrational, and structural properties. Besides, Δ-machine learning [63, 65] has emerged as a promising solution for accurate DFTB calculations. This approach utilizes machine learning techniques to predict the energy difference between DFTB and reference energies. These previous investigations show that machine learning based repulsive potential can improve the accuracy and transferability significantly compared with previous pairwise repulsive potential.

For incorporating DFTB electronic parametrization and machine learning, Li [66] used spline or machine-learned models to generate the diatomic Hamiltonian integrals for DFTB calculations, leading to improved accuracy of electronic properties. The lack of well-defined basis functions in machine learning methods has limited their extensibility and transferability, rendering them unable for molecular orbital calculations. To address this limitation, an interesting research topic is to optimize and predict chemical environment adaptive basis function parameters to improve extensibility and transferability. This research topic uses a similar strategy in previous work [57, 58], incorporating machine learning with static parameters in semiempirical methods. This thesis explores this strategy by optimizing the compression radii to ensure a well-defined basis. We compare the extensibility and transferability of this method to the method of generating Hamiltonian integrals directly (without a well-defined basis) obtained from machine learning.

For electronic properties in periodic boundary conditions, if we extend from DFTB to more general tight binding methods, many work using machine learning to optimize tight binding parameters to improve band structure performance [67, 68, 69]. Previous work includes using pure machine learning models or incorporating machine learning with physical models to predict the density of states (DOS) or band structures. Peano and coworkers [67] combined tight binding and deep learning to rapidly explore and optimize band structures and classify their topological characteristics. Knøsgaard and coworkers [68] reproduced GW band structures of 2D materials using machine learning-based DFT calculations. Schattauer and coworkers [69] employed machine learning to derive tight-binding parametrizations for the electronic structure of defects with DFT accuracy level. These applications show that machine learning can be applied in various chemical environments, including molecules and solids.

1.4 Outline

The field of data-driven science has many successful examples, with the combination of physical models and machine learning often leading to accurate and transferable results. This thesis presents a novel approach of combining DFTB with machine learning to achieve

more accurate DFTB calculations.

Our work begins by developing parameters for electronic and repulsive parametrization in lithium-ion batteries (LIBs), enabling accurate predictions of electronic band structures and geometry optimization. We demonstrate the accuracy of our DFTB parametrization by comparing band structures and optimized geometries with DFT results.

Motivated by the successes of data-driven science, we have implemented the tight binding machine learning toolkit (TBMaLT), an open source framework that allows for standard DFTB calculations, high throughput DFTB calculations, and DFTB-ML framework for DFTB parametrization.

We employ this DFTB-ML framework in TBMaLT for molecular systems, utilizing three distinct approaches to construct Hamiltonian and overlap matrices for DFTB calculations. These approaches involve optimizing basis function parameters, which can be done through global or local optimization. Additionally, we also explored the direct optimization of the Hamiltonian and overlap integrals.

Finally, we have extended our framework to periodic boundary conditions. We have optimized band structures of bulk, slab, and defect systems together with hybrid functional accuracy, enabling accurate band structure calculations with low computational cost.

2 Theoretical Review

This chapter presents the theoretical foundations used in this work, primarily focusing on DFT and DFTB theory. The chapter begins with an overview of solutions and approximations to the many-body Schrödinger equation (MBSE) and traces the development of modern DFT theory. We introduce key concepts such as the Hohenberg-Kohn theorem, the Kohn-Sham method, and the development of exchange-correlation functionals. Additionally, we introduce DFTB theory, beginning with the approximation from standard DFT and progressing to DFTB with second-order and third-order corrections. As the performance of DFTB is heavily dependent on its parametrization, we also describe some widely used DFTB parametrization methods.

2.1 Many-Body Schrödinger Equation

For a stationary MBSE, with nuclei positions $\mathbf{R}$ and electron positions $\mathbf{r}$, the equation can be expressed as:

$$\hat{H}\Psi(\{\mathbf{r}_i\},\{\mathbf{R}_i\}) = E\Psi(\{\mathbf{r}_i\},\{\mathbf{R}_i\}), \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian operator, $\Psi(\{\mathbf{r}_i\},\{\mathbf{R}_i\})$ is the wavefunction of nuclei and electrons, E is the energy. Hamiltonian operator consists of the kinetic energy term of the electrons $\hat{T}_e$, the kinetic energy term of the nuclei $\hat{T}_I$, the nuclei-nuclei interaction $V_{\text{I–I}}$, the electron-electron interaction V_{e-e} and the electron-nuclei interaction V_{e-I}. The Hamiltonian in atomic units can be written as:

$$\hat{H} = -\frac{1}{2}\sum_{i}^{n}\nabla_i^2 - \sum_{I}^{N}\frac{1}{2}\nabla_I^2 + \frac{1}{2}\sum_{i\neq j}^{n,n}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|} + \frac{1}{2}\sum_{I\neq J}^{N,N}\frac{Z_I Z_J}{|\mathbf{R}_I-\mathbf{R}_J|} - \sum_{i}^{n}\sum_{I}^{N}\frac{Z_I}{|\mathbf{r}_i-\mathbf{R}_I|}, \tag{2.2}$$

where n and N are the numbers of electrons and nuclei respectively, and Z_I is the charge of nuclei I. Many efforts have been made to simplify the MBSE [70, 71, 72, 73, 74, 75, 76, 77], with the key issue being to reduce the computational expense while still preserving an acceptable level of accuracy.

Because nuclei are much heavier than electrons, Born and Oppenheimer [70] assumed that the electrons and nuclei can be treated separately, and the electrons are assumed to respond instantaneously (adiabatically) to changes in the nuclei. The electronic Hamiltonian in the Born-Oppenheimer (BO) approximation is expressed as:

$$\hat{H}_{\text{elect}} = -\frac{1}{2}\sum_{i}^{n}\nabla_i^2 - \sum_{I}^{N}\sum_{i}^{n}\frac{Z_I}{|\mathbf{R}_I-\mathbf{r}_i|} + \frac{1}{2}\sum_{i\neq j}^{n,n}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{2.3}$$

The BO approximated electronic Schrödinger equation is still too complex to be solved, and many approaches [71, 72, 73, 74] have been developed to generate feasible solutions to simplify the MBSE further. One of the approaches was introduced by Hartree in 1927 [71, 72]. The wavefunction in the Hartree approximation is written as a product of

single-particle wavefunctions $\Psi(\{\mathbf{r}_i\}) = \psi(\mathbf{r}_1)\psi(\mathbf{r}_2)\dots\psi(\mathbf{r}_N)$. This leads to the following eigenvalue problem of the single-particle wavefunction:

$$\left[-\frac{1}{2}\nabla^2 + V_{\text{ext}}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}'\right]\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}), \tag{2.4}$$

where the first part is the kinetic term, and the second term is the external potential which is the potential for attraction to the nuclei. $n(\mathbf{r})$ is the electron density and ϵ_i are the eigenvalues. Later, Slater and Fock [73, 74] noted that the Hartree *ansatz* does not yield antisymmetric wavefunctions. One solution to this problem is using Slater determinants so that the corresponding wavefunction in the Hartree-Fock (HF) method is expressed as $\Psi(\{\mathbf{r}_i\}) = \mathcal{A}\prod_{j=1}^{N}\psi(\mathbf{r}_j)$, where $\mathcal{A}$ is the antisymmetrizing operator. The spin-unpolarized formalism for the closed-shell HF equation is

$$E_{\text{HF}} = \sum_i H_i + \sum_{ij}(J_{ij} - K_{ij}) \tag{2.5}$$

$$H_i = f_i \int d\mathbf{r}\psi_i^*(\mathbf{r})\left[-\frac{1}{2}\nabla^2 + V_{\text{ext}}(\mathbf{r})\right]\psi_i(\mathbf{r}) \tag{2.6}$$

$$J_{ij} = \frac{1}{2}f_i f_j \iint d\mathbf{r}_1 d\mathbf{r}_2 \psi_i^*(\mathbf{r}_1)\psi_i(\mathbf{r}_1)\frac{1}{\mathbf{r}_{12}}\psi_j^*(\mathbf{r}_2)\psi_j(\mathbf{r}_2) \tag{2.7}$$

$$K_{ij} = \frac{1}{4}f_i f_j \iint d\mathbf{r}_1 d\mathbf{r}_2 \psi_i^*(\mathbf{r}_1)\psi_j(\mathbf{r}_1)\frac{1}{\mathbf{r}_{12}}\psi_j^*(\mathbf{r}_2)\psi_i(\mathbf{r}_2), \tag{2.8}$$

where H_i is the single-particle contribution describing kinetic energy and potential energy of electron i, J_{ij} is the Coulomb term, and K_{ij} is the exchange term. The occupation of eigenstate i is denoted as $f_i \in [0, 2]$. The solution of the HF equation is obtained through the self-consistent field (SCF) method. The HF equation includes the correct exchange interactions but does not include the correlation effects. In order to account for the electron correlation effects, some post-HF methods such as configuration interaction (CI) [78] have been developed. However, highly accurate post-HF methods are computationally very demanding and can only be applied to small systems. Therefore, semiempirical methods simplified from HF have been developed and can be used for large systems. The most frequently used semiempirical methods include modified neglect of diatomic overlap (MNDO) [79], Austin model 1 (AM1) [80], and parametric model number 7 (PM7) [81, 82].

2.2 Density Functional Theory

The 1998 Nobel Prize in Chemistry was awarded to Walter Kohn for his development of the DFT [83]. This section introduces the two main contributions of Kohn: the Hohenberg-Kohn theorem [24] and the Kohn-Sham equation [25]. Subsequently, the development of exchange-correlation functionals is briefly introduced.

2.2.1 Hohenberg-Kohn Theorem

In 1964, the publication from Hohenberg and Kohn [24] was a milestone of modern DFT, and two main theorems were introduced in the paper:

Theorem 1 *The ground state electron density of a bound system in an external potential, except for an additive constant, determines this potential uniquely.*

The first theorem can be proved by contradiction [24, 83]. Hohenberg and Kohn [24] further defined the universal functional of density $n(\mathbf{r})$

$$F[n(\mathbf{r})] = \langle \Psi | \hat{T} + \hat{U} | \Psi \rangle, \tag{2.9}$$

where Ψ is a functional of $n(\mathbf{r})$, and $\hat{T}$ and $\hat{U}$ are the kinetic energy operator and the Coulomb repulsion of the electron-electron operator, respectively. This universal functional is valid for any number of particles and any external potential. For a given external potential, the energy functional is

$$E[n(\mathbf{r})] = \int d\mathbf{r} V_{\text{ext}}(\mathbf{r}) n(\mathbf{r}) + F[n(\mathbf{r})], \tag{2.10}$$

where $V_{\text{ext}}(\mathbf{r})$ is the external potential. With the definition of the energy functional and the constrained condition $\int d\mathbf{r} n(\mathbf{r}) = N$, where N is the number of electrons in the system, the second theorem is:

Theorem 2 *The energy functional in Eq.* (2.10) *is guaranteed to deliver the lowest energy only when the input density is the true ground state density* $n_0(\boldsymbol{r})$.

Based on Theorem 1, the external potential uniquely determines the electron density for the ground state wavefunction Ψ. Theorem 2 states that for the ground state of the system, which corresponds to the lowest energy state, the input density must be the ground state density $n_0(\mathbf{r})$.

2.2.2 Kohn-Sham Method

The Thomas-Fermi (TF) theory, which dates back to 1927 and was proposed by Thomas [75] and Fermi [76], is a rudimentary form of modern DFT. However, Kohn demonstrated that the HF theory provides better descriptions of the atomic ground states compared to the TF method [83]. In 1965, Kohn proposed a Hartree-like formulation based on the Hartree equations [25], which also follows the Hohenberg-Kohn principle. This equation is the so-called Kohn-Sham (KS) equation:

$$E_{\text{KS}}[n(\mathbf{r})] = T[n(\mathbf{r})] + \int d\mathbf{r} V_{\text{ext}}(\mathbf{r}) n(\mathbf{r}) + \frac{1}{2} \iint d\mathbf{r} d\mathbf{r}' \frac{n(\mathbf{r}) n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + E_{\text{XC}}[n(\mathbf{r})]. \tag{2.11}$$

The first term is the kinetic energy functional, while the second component denotes the energy term due to the external potential. The third term is the Hartree term containing the Coulomb repulsion, and the last term refers to the exchange-correlation energy term. Notably, the last term remains a challenge for the development of DFT.

2.2.3 Solutions of the Kohn-Sham Method

The minimization of the energy functional is a variational problem. The electron density of single particle wavefunction $\Phi_i(\mathbf{r})$ must satisfy the following conditions:

$$\langle\Phi_i(\mathbf{r})|\Phi_i(\mathbf{r})\rangle = 1; n(\mathbf{r}) = \sum_i^{occ} f_i|\Phi_i(\mathbf{r})|^2, \tag{2.12}$$

where f_i is the occupation number. With the Lagrange parameters ϵ_i and applying the variational principle at the ground state energy, we get

$$\frac{\delta}{\delta\Phi_i^*(\mathbf{r})}\Big\{E_{\mathrm{KS}}[n(\mathbf{r})] + \sum_i^{occ} f_i\epsilon_i[1 - \int d\mathbf{r}|\Phi_i(\mathbf{r})|^2]\Big\} = 0. \tag{2.13}$$

Using the relation $\frac{\delta n(\mathbf{r})}{\delta\Phi_i^*(\mathbf{r})} = 2\Phi_i(\mathbf{r})$ and Eq. (2.13), we get

$$\Big[-\frac{1}{2}\nabla^2 + V_{\mathrm{ext}}(\mathbf{r}) + V_{\mathrm{H}}([n(\mathbf{r})],\mathbf{r}) + V_{\mathrm{XC}}([n(\mathbf{r})],\mathbf{r}) - \epsilon_i\Big]\Phi_i(\mathbf{r}) = 0. \tag{2.14}$$

The above set of equations are the so-called KS equations. Collecting V_{ext}, V_{H} and V_{XC} into the effective potential V_{eff}

$$V_{\mathrm{eff}}([n(\mathbf{r})],\mathbf{r}) = V_{\mathrm{ext}}(\mathbf{r}) + V_{\mathrm{H}}([n(\mathbf{r})],\mathbf{r}) + V_{\mathrm{XC}}([n(\mathbf{r})],\mathbf{r}), \tag{2.15}$$

the KS equations can be written as

$$\Big[-\frac{1}{2}\nabla^2 + V_{\mathrm{eff}}([n(\mathbf{r})],\mathbf{r}) - \epsilon_i\Big]\Phi_i(\mathbf{r}) = 0. \tag{2.16}$$

In real calculations, the single particle basis is usually expanded with a complete basis set $|\varphi_\nu\rangle$ and coefficients $C_{i\nu}$ as $|\Phi_i\rangle = \sum_\nu C_{i\nu}|\varphi_\nu\rangle$. The KS equations become

$$\hat{H}|\Phi_i(\mathbf{r})\rangle = \sum_\nu \hat{H}|C_{i\nu}\varphi_\nu(\mathbf{r})\rangle = \epsilon_i|\Phi_i(\mathbf{r})\rangle. \tag{2.17}$$

With the multiplication of $\langle\varphi_\mu|$ on the left side, we get the generalized eigenvalue problem $\mathbf{HC} = \epsilon\mathbf{SC}$:

$$\sum_\nu C_{i\nu}(H_{\mu\nu} - \epsilon_i S_{\mu\nu}) = 0, \tag{2.18}$$

with $H_{\mu\nu}$ and $S_{\mu\nu}$ being:

$$\begin{aligned} H_{\mu\nu} &= \langle\varphi_\mu|\hat{H}|\varphi_\nu\rangle = \int d\mathbf{r}\varphi_\mu^*(\mathbf{r})\Big[-\frac{1}{2}\nabla^2 + V_{\mathrm{eff}}([n(\mathbf{r})],\mathbf{r})\Big]\varphi_\nu(\mathbf{r}) \\ S_{\mu\nu} &= \langle\varphi_\mu|\varphi_\nu\rangle = \int d\mathbf{r}\varphi_\mu^*(\mathbf{r})\varphi_\nu(\mathbf{r}). \end{aligned} \tag{2.19}$$

In the KS method, similar to the HF method, the solution of the KS equations involves a self-consistent problem, necessitating the use of iterative techniques to obtain a solution. Initially, an electron density $n_0(\mathbf{r})$ is guessed and used to solve the Kohn-Sham equation, which will generate a newly updated electron density. The calculation reaches convergence when the electron density difference between two iterative steps becomes smaller than a tolerance value.

2.2.4 Exchange-Correlation Functionals

Reasonable approximations of the exchange-correlation functional are crucial for DFT. Perdew and Schmidt [84] introduced Jacob's ladder for the exchange-correlation functional approximation. This ladder consists of five different levels, with higher rungs potentially offering more accurate calculations but also more complicated exchange-correlation functional constructions. The five levels are:

1 local-density approximation (LDA)

2 generalized gradient approximation (GGA)

3 meta-GGA

4 hybrid functional

5 random phase approximation (RPA) like functional.

In the following part, rungs 1, 2, and 4 will be briefly introduced since these methods will be used later in this thesis.

We proceed from simple to complex. LDA has been derived from the homogeneous electron gas model [85], and the LDA exchange-correlation energy term is written as:

$$E_{\mathrm{XC}}[n(\mathbf{r})] = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\mathrm{XC}}[n(\mathbf{r})], \tag{2.20}$$

where $\varepsilon_{\mathrm{XC}}$ is the exchange-correlation energy per electron of a homogeneous electron gas. Previous work [83] shows that LDA achieves a $10-20\%$ error in ionization energies of atoms and dissociation energies of molecules and a 1% accuracy level in bond lengths.

LDA-based methods have been widely used in the 1970s. Later GGA-based calculations have shown that GGA, which considers the gradient of the density for non-homogeneous electron density situations, can significantly reduce the error of atomization energies and total energies [86, 87].

$$E_{\mathrm{XC}}^{\mathrm{GGA}}[n(\mathbf{r})] = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\mathrm{XC}}[n(\mathbf{r}), \nabla n(\mathbf{r})], \tag{2.21}$$

In 1991, Perdew and Wang developed the PW91 functional [88] with second-order gradient expansion exchange-correlation energy, which works well in many different systems, contributing to the improvements of energies in atoms and molecules. In 1996, the Perdew-Burke-Ernzerhof functional (PBE) [89] was developed to address many of the problems in PW91, and it has been widely used since then.

Hybrid functionals which incorporate some HF-like exchange can provide accurate atomic energies and bond length calculations [90]. In 1993, Becke [91] has given an approach to construct DFT with HF exchange energy, where the exchange energy is written as:

$$E_{\mathrm{X}}^{\mathrm{B3LYP}} = 0.8 E_{\mathrm{X}}^{\mathrm{LDA}} + 0.2 E_{\mathrm{X}}^{\mathrm{HF}} + 0.72 E_{\mathrm{X}}^{\mathrm{B88}}. \tag{2.22}$$

The terms $E_{\mathrm{X}}^{\mathrm{LDA}}$, $E_{\mathrm{X}}^{\mathrm{HF}}$ and $E_{\mathrm{X}}^{\mathrm{B88}}$ are the LDA exchange energy, the HF exchange energy and the Becke88 exchange energy [92], respectively. The correlation energy is written as

$$E_{\mathrm{C}}^{\mathrm{B3LYP}} = 0.19E_{\mathrm{C}}^{\mathrm{VWN3}} + 0.81E_{\mathrm{C}}^{\mathrm{LYP}}, \tag{2.23}$$

where $E_{\mathrm{C}}^{\mathrm{VWN3}}$ and $E_{\mathrm{C}}^{\mathrm{LYP}}$ are the Vosko–Wilk–Nusair III [93] and the Lee–Yang–Parr [87] correlation energies, respectively. Another popular hybrid functional is the Heyd-Scuseria-Ernzerhof (HSE) functional [94, 95], which is given by

$$E_{\mathrm{XC}}^{\mathrm{HSE}} = \alpha E_{\mathrm{X}}^{\mathrm{HF,\,SR}}(\mu) + (1-\alpha)E_{\mathrm{X}}^{\mathrm{PBE,\,SR}}(\mu) + E_{\mathrm{X}}^{\mathrm{PBE,\,LR}}(\mu) + E_{\mathrm{C}}^{\mathrm{PBE}}, \tag{2.24}$$

where SR and LR denote the short-range exchange and the long-range exchange parts, respectively. The parameter μ determines the range separation of the Coulomb term according to

$$\frac{1}{r} = \frac{\mathrm{erfc}(\mu r)}{r} + \frac{\mathrm{erf}(\mu r)}{r}, \tag{2.25}$$

where the first and second terms are the SR and the LR contributions, respectively. In the HSE06 method [95], α and μ are chosen as 0.25 and 0.2, respectively.

2.3 Density Functional based Tight Binding Theory

2.3.1 Introduction to DFTB

The Kohn-Sham equation without approximation is

$$E = \sum_i f_i \left\langle \Phi_i \middle| -\frac{1}{2}\nabla^2 + V_{\mathrm{ext}} \middle| \Phi_i \right\rangle + \frac{1}{2}\iint d\mathbf{r}d\mathbf{r}' \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + E_{\mathrm{XC}}[n(\mathbf{r})] + E_{\mathrm{II}}, \tag{2.26}$$

where f_i is the occupation of state Φ_i and E_{II} is the energy term covers the nuclei-nuclei repulsion. When considering an approximation, one possibility is to use a reference density $n_0 = n_0(\mathbf{r})$ subject to a density fluctuation $\delta n = \delta n(\mathbf{r})$ so that with this approximation, we expand the total energy up to the second order:

$$\begin{aligned} E[n_0+\delta n] = &\sum_i f_i \left\langle \Phi_i \middle| -\frac{1}{2}\nabla^2 + V_{\mathrm{ext}} + V_{\mathrm{H}}[n_0] + V_{\mathrm{XC}}[n_0] \middle| \Phi_i \right\rangle \\ &+ \frac{1}{2}\iint' \Big(\frac{1}{|\mathbf{r}-\mathbf{r}'|} + \frac{\delta^2 E_{\mathrm{XC}}[n_0]}{\delta n \delta n'} \Big) \delta n \delta n' \\ &- \frac{1}{2}\iint' \frac{n_0 n_0'}{|\mathbf{r}-\mathbf{r}'|} + E_{\mathrm{XC}}[n_0] - \int V_{\mathrm{XC}}[n_0]n_0 + E_{\mathrm{II}}. \end{aligned} \tag{2.27}$$

here, $\delta n(\mathbf{r})$ had been substituted by δn and $\int d\mathbf{r}$ by $\int$. The first line of Eq. (2.27) is the so-called band-structure energy. The second line is the second-order correction term, and the third line is the so-called repulsive term. In DFTB, the input density is composed of a sum of compressed densities of neutral atoms, and there is no consideration of charge transfer in this band-structure term. The single particle wavefunction can be expanded

into a set of atomic orbitals φ_μ using a linear combination of atomic orbitals (LCAO) *ansatz* with coefficients $C_{i\mu}$:

$$\Phi_i(\mathbf{r}) = \sum_\mu C_{i\mu}\varphi_\mu(\mathbf{r}). \tag{2.28}$$

To calculate two-centre integrals, the effective potential is decomposed into atomic-like contributions. There are two ways to decompose the effective potential: the potential superposition and density superposition. In the potential superposition, the potential is approximated as $V_{\text{eff}}([n_0, \mathbf{r}]) \approx \sum_i V_i^{\text{eff}}([n_i^0(\mathbf{r}_i)], \mathbf{r}_i)$ while in the density superposition $V_{\text{eff}}([n_0, \mathbf{r}]) \approx V_{\text{eff}}(\sum_i n_i^0(\mathbf{r}_i))$. The Hamiltonian can then be written as:

$$H_{\mu_A\nu_A} = \overbrace{\langle\varphi_{\mu_A}| -\frac{1}{2}\nabla^2 + V_{\text{eff}}^A|\varphi_{\nu_A}\rangle}^{\text{on-site}} + \overbrace{\sum_{B\neq A}\langle\varphi_{\mu_A}|V_{\text{eff}}^B|\varphi_{\nu_A}\rangle}^{\text{crystal-field}} \tag{2.29}$$

$$H_{\mu_A\nu_B} = \overbrace{\langle\varphi_{\mu_A}| -\frac{1}{2}\nabla^2 + \left\{ \begin{array}{c} V_{\text{eff}}^A(\mathbf{r}_A) + V_{\text{eff}}^B(\mathbf{r}_B) \\ V_{\text{eff}}([n_A^0 + n_B^0], \mathbf{r}) \end{array} \right\} |\varphi_{\nu_B}\rangle}^{\text{two-centre}} + \overbrace{\sum_{C\neq B\neq A}\langle\varphi_{\mu_A}|V_{\text{eff}}^C|\varphi_{\nu_B}\rangle}^{\text{three-centre}}, \tag{2.30}$$

where A, B, and C denote the orbital centres. When the crystal-field terms are neglected, only the on-site terms are retained in $H_{\mu_A\nu_A}$. In DFTB, the on-site terms are the eigenvalues ϵ of free atoms instead of using compressed atoms, which ensures the correct DFT dissociation limits. The on-site terms are

$$H_{\mu_A\nu_A} = \epsilon_{\nu_A}, \quad \mu_A = \nu_A. \tag{2.31}$$

In Eq. (2.30), the three-centre terms are neglected while the two-centre terms are retained in $H_{\mu_A\nu_B}$, the resulting equation is

$$H_{\mu_A\nu_B} = \langle\varphi_{\mu_A}| -\frac{1}{2}\nabla^2 + \left\{ \begin{array}{c} V_{\text{eff}}^A(\mathbf{r}_A) + V_{\text{eff}}^B(\mathbf{r}_B) \\ V_{\text{eff}}([n_A^0 + n_B^0], \mathbf{r}) \end{array} \right\} |\varphi_{\nu_B}\rangle. \tag{2.32}$$

The wavefunctions and the atomic densities in the two-centre terms are calculated from pseudo-atoms within a confinement potential $V_{\text{conf}}(r)$:

$$\Big[-\frac{1}{2}\nabla^2 + V_{\text{eff}}([n_A^0(\mathbf{r})]) + V_{\text{conf}}(r)\Big]\varphi_{\nu_A} = \epsilon_{\nu_A}\varphi_{\nu_A}. \tag{2.33}$$

The confining term is usually a harmonic potential. The pseudo-atoms can offer a better initial guess of densities in compound systems than free atoms [35]. The confinement term will be discussed later in detail in the electronic parametrization section. If we represent the overlap $\langle\varphi_\mu|\varphi_\nu\rangle$ as $S_{\mu\nu}$, the Hamiltonian and overlap lead to a generalized eigenvalue problem

$$\sum_\nu C_{i\nu}(H_{\mu\nu} - \epsilon_i S_{\mu\nu}) = 0. \tag{2.34}$$

Solving this generalized eigenvalue problem yields eigenvalues and eigenvectors, and physical properties can be calculated based on the eigenvalues and eigenvectors.

The second-order correction in Eq. (2.27) becomes significant for systems with chemical bonding between different types of atoms. The second-order term in Eq. (2.27) becomes [27]

$$E^2 = \frac{1}{2}\sum_{AB}\iint' \Gamma[\mathbf{r},\mathbf{r}',n_0]\delta n_A(\mathbf{r})\delta n_B(\mathbf{r}'), \tag{2.35}$$

where Γ represents the Hartree and XC terms. The density variation δn_A of atom A can be expanded as a series of radial and angular functions:

$$\delta n_A(\mathbf{r}) = \sum_{lm} K_{ml}F^A_{ml}(\mathbf{r}-\mathbf{R}_A)Y_{lm}\Big(\frac{\mathbf{r}-\mathbf{R}_A}{|\mathbf{r}-\mathbf{R}_A|}\Big) \approx \Delta q_A F^A_{00}(|\mathbf{r}-\mathbf{R}_A|)Y_{00}, \tag{2.36}$$

where F^A_{ml} denotes the corresponding radial dependency on atom A, Y_{lm} gives the angular dependency, K_{ml} are expansion coefficients, and atomic charge fluctuations are Mulliken charges q_A with respect to the neutral atoms Z_A: $\Delta q_A = q_A - Z_A$. Taking charge conservation into consideration, we obtain $\sum_A \Delta q_A = \int \delta n(\mathbf{r})$. Substituting Eq. (2.36) into Eq. (2.35), we obtain the second-order energy term:

$$E^2 = \frac{1}{2}\sum_{AB}\Delta q_A \Delta q_B \gamma_{AB} \tag{2.37}$$

$$\gamma_{AB} = \iint' \Gamma[\mathbf{r},\mathbf{r}',n_0]\frac{F^A_{00}(|\mathbf{r}-\mathbf{R}_A|)F^B_{00}(|\mathbf{r}-\mathbf{R}_B|)}{4\pi} \tag{2.38}$$

Assuming an exponential decay for the charge fluctuations, the normalized spherical charge densities are [27]

$$\delta n_A(\mathbf{r}) = \frac{\tau_A^3}{8\pi}e^{-\tau_A|\mathbf{r}-\mathbf{R}_A|}, \tag{2.39}$$

where the new parameter τ_A has been introduced. Neglecting the second-order exchange-correlation term in the second line of Eq. (2.27) and only considering the second-order Coulomb term, we get

$$\gamma_{AB} = \iint' \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\tau_A^3}{8\pi}e^{-\tau_A|\mathbf{r}-\mathbf{R}_A|}\frac{\tau_B^3}{8\pi}e^{-\tau_B|\mathbf{r}-\mathbf{R}_B|}. \tag{2.40}$$

Setting $R = |\mathbf{R}_A - \mathbf{R}_B|$ and following the transformations in a previous work [27], we obtain

$$\gamma_{AB} = \frac{1}{R} - S(R,\tau_A,\tau_B), \tag{2.41}$$

where S is a short-range function. When A equals to B and $R = 0$, we get

$$S(R,\tau_A,\tau_A) \overset{R\to 0}{=} \frac{5}{16}\tau_A + \frac{1}{R}. \tag{2.42}$$

If we assume $R \to 0$, the second-order contribution can be approximately expressed using so-called chemical hardness (Hubbard parameters). The Hubbard parameters U_A are the second derivatives of free atomic energies E_A from DFT ($U_A = \frac{\delta^2 E_A}{\delta q_A^2}$), and we request that

$$\frac{1}{2}\Delta q_A^2 \gamma_{AA} = \frac{1}{2}\Delta q_A^2 U_A. \tag{2.43}$$

From Eq. (2.42) and Eq. (2.43) we get the following:

$$\tau_A = \frac{16}{5} U_A. \tag{2.44}$$

Finally, adding the band structure term and second-order term in Eq. (2.27), we obtain

$$E^1 + E^2 = \sum_i f_i \langle \Phi_i | \hat{H}_0 | \Phi_i \rangle + \frac{1}{2} \sum_{A,B}^{N} \Delta q_A \Delta q_B \gamma_{AB}. \tag{2.45}$$

The SCC-DFTB has been derived [27] by applying the second-order expansion, and the energy term depends on the Mulliken charge fluctuations. In SCC-DFTB, Hubbard parameters are constant and derived from DFT calculations of the free atoms. However, the Hubbard parameters of positively charged atoms are larger than neutral atoms while the Hubbard parameters of negatively charged atoms are smaller [96]. Therefore, in the third-order DFTB approach, the atomic charge-dependent Hubbard U parameters have been introduced [96]. Additionally, the so-called DFTB3 scheme also introduces a modified γ^h parameter for H-X pairs to improve the electrostatic treatment within the second-order terms, where H denotes hydrogen, and X represents another heavy atom. The modified second-order Hamiltonian and third-order DFTB Hamiltonian can be written as

$$\begin{aligned} H^2_{\mu\nu} &= \frac{S_{\mu\nu}}{2} \sum_C (\gamma^h_{AC} + \gamma^h_{BC}) \Delta q_C \\ H^3_{\mu\nu} &= S_{\mu\nu} \sum_C \left(\frac{\Delta q_A \Gamma_{AC}}{3} + \frac{\Delta q_B \Gamma_{BC}}{3} + (\Gamma_{AC} + \Gamma_{BC}) \frac{\Delta q_C}{6} \right) \Delta q_C, \end{aligned} \tag{2.46}$$

where Γ_{AB} is the derivative of γ_{AB} with respect to the charge, and γ^h represents the modified term for H-X pairs in the second-order Hamiltonian. Incorporating the third-order term and modifying the second-order term improve the overall performance of DFTB calculations, especially for hydrogen-bonded systems.

Beyond SCC-DFTB and third-order DFTB method, DFTB has been expanded for various systems and applications. The extensions of SCC-DFTB include DFTB+U [97] for correlated materials, non-equilibrium Green's function (NEGF) [98] for transport calculations, real-time time-dependent DFTB (TD-DFTB) for excited state simulations [99] and DFTB with different non-covalent interactions [100, 101, 102, 103] for some chemical and biological systems.

The last line in Eq. (2.27) is the so-called repulsive term. By approximating the repulsive interaction as pairwise interactions, it can be written as:

$$E_{\text{rep}} = \frac{1}{2} \sum_A \sum_{B \neq A} E^{AB}_{\text{rep}}(|\mathbf{R}_A - \mathbf{R}_B|), \tag{2.47}$$

where $E^{AB}_{\text{rep}}(|\mathbf{R}_A - \mathbf{R}_B|)$ is the repulsive interaction between atom A and B, and depends only on the distances between atom A and B. The repulsive term is usually fitted from

the difference between reference energies and DFTB electronic energies E_{elect}. The reference energies are usually obtained from DFT calculations and the electronic energies are usually the summation of band structure energies and SCC energies. Further details of the repulsive potential parametrization will be discussed later.

Periodic boundary conditions

For crystal systems, due to translational symmetry, we can use one unit cell to perform the simulations under periodic boundary conditions. The Bloch condition must be fulfilled for crystal systems:

$$\Phi_i^{\mathbf{k}}(\mathbf{r}+\mathbf{R}) = \Phi_i^{\mathbf{k}}(\mathbf{r})e^{i\mathbf{k}\mathbf{R}}, \tag{2.48}$$

where $\mathbf{R}$ is the translation vector in terms of unit vectors and $\mathbf{k}$ is the crystal momentum vector. In order to satisfy the Bloch condition, we use the following basis expansion in periodic systems

$$\Phi_i^{\mathbf{k}}(\mathbf{r}) = \sum_{\mu} C_{i\mu}^{\mathbf{k}} \beta_{\mu}^{\mathbf{k}}(\mathbf{r}), \tag{2.49}$$

where $C_{i\mu}^{\mathbf{k}}$ are the eigenvector coefficients of the Bloch functions $\beta_{\mu}^{\mathbf{k}}$ and eigenstate i at crystal momentum $\mathbf{k}$. The Bloch functions $\beta_{\mu}^{\mathbf{k}}(\mathbf{r})$ are

$$\beta_{\mu}^{\mathbf{k}}(\mathbf{r}) = \frac{1}{\sqrt{N}} \sum_{\mathbf{R}} \varphi_{\mu}(\mathbf{r}-\mathbf{R})e^{i\mathbf{k}\mathbf{R}}, \tag{2.50}$$

where the N means the number of unit cells in the system. Then we obtain

$$\Phi_i^{\mathbf{k}}(\mathbf{r}) = \frac{1}{\sqrt{N}} \sum_{\mu} C_{i\mu}^{\mathbf{k}} \sum_{\mathbf{R}} \varphi_{\mu}(\mathbf{r}-\mathbf{R})e^{i\mathbf{k}\mathbf{R}}. \tag{2.51}$$

The Hamiltonian and overlap matrices elements will be

$$\begin{aligned} H_{\mu\nu}(\mathbf{R}) &= \langle \varphi_{\mu}(\mathbf{r}) | \hat{H} | \varphi_{\nu}(\mathbf{r}-\mathbf{R}) \rangle e^{i\mathbf{k}\mathbf{R}} \\ H_{\mu\nu}^{\mathbf{k}} &= \sum_{\mathbf{R}} H_{\mu\nu}(\mathbf{R})e^{i\mathbf{k}\mathbf{R}} \\ S_{\mu\nu}(\mathbf{R}) &= \langle \varphi_{\mu}(\mathbf{r}) | \varphi_{\nu}(\mathbf{r}-\mathbf{R}) \rangle e^{i\mathbf{k}\mathbf{R}} \\ S_{\mu\nu}^{\mathbf{k}} &= \sum_{\mathbf{R}} S_{\mu\nu}(\mathbf{R})e^{i\mathbf{k}\mathbf{R}}. \end{aligned} \tag{2.52}$$

The eigenvalues and eigenvectors are determined at each $\mathbf{k}$-point separately using the following generalized eigenvalue problem:

$$\sum_{\nu} H_{\mu\nu}^{\mathbf{k}} C_{i\nu}^{\mathbf{k}} = \epsilon_i^{\mathbf{k}} \sum_{\nu} S_{\mu\nu}^{\mathbf{k}} C_{i\nu}^{\mathbf{k}}. \tag{2.53}$$

With a set of $\mathbf{k}$-points sampling the Brillouin zone and with the weighting parameter of each $\mathbf{k}$-point, we can perform DFTB calculations in periodic systems. High-symmetry $\mathbf{k}$-points and the weights are included for the band structure calculations.

2.3.2 DFTB Electronic parametrization

The Hamiltonian from atom A and B in DFTB can be seen in Eq. (2.32). There are two ways to construct V_{eff} between A and B as discussed before. The first is potential superposition $V_{\text{eff}}[n_0^A]+V_{\text{eff}}[n_0^B]$, and the second is density superposition $V_{\text{eff}}[n_0^A+n_0^B]$. Previous work [104] shows that using density superposition can improve energies, vibrational frequencies and reaction barriers while using potential superposition can achieve better results in band structure calculations. In DFTB electronic parametrization, a confining potential is added to tune both the electronic wavefunctions and the electron densities. The confining potential is often written in the form of

$$V_{\text{conf}} = (\frac{r}{r_0})^n, \tag{2.54}$$

where the value of the power parameter n is usually set to 2. The so-called compression radius r_0 is used to tune the confinement of wavefunctions and electron densities. The key of DFTB electronic parametrization is to get optimized confinement parameters, and many previous works have been developed to search the optimized parameters [105, 47, 44, 40, 36]. The empirical values based on covalent radii are not accurate enough for some applications, and compression radii are needed to be tuned according to the applications of interest [50, 106]. In addition, the on-site energies derived from the eigenvalues of free atoms are also crucial in the electronic parametrization. Both compression radii and on-site energies will be discussed later in this thesis.

2.3.3 DFTB Repulsive Parametrization

The repulsive term is essential for some DFTB calculations, such as geometry optimization calculations or molecular dynamics (MD) simulations. The pairwise repulsive interaction is described in Eq. (2.47). In this thesis, the curvature constrained splines (CCS) [45, 46] method has been applied for repulsive parametrization. The constraints used in CCS ensure the pairwise repulsive potentials without spurious oscillations. The repulsive potentials in CCS are constructed in cubic spline format. The repulsive in CCS is written as:

$$E_{\text{rep}} = \mathbf{v}^T\mathbf{c} + \mathbf{w}^T\boldsymbol{\epsilon}, \tag{2.55}$$

where $\mathbf{w}^T\boldsymbol{\epsilon}$ represents the one-body term to correct the atomic energy difference between reference energies and DFTB energies. $\boldsymbol{\epsilon}$ is the one-body energy term and $\mathbf{w}$ represents the number of atoms. $\mathbf{v}$ is the vector of energies [45] and $\mathbf{c}$ are the coefficients of the cubic spline functions to be determined. When fitting repulsive potentials, the objective function (J) of total K configurations has been defined, which is the difference between reference and DFTB energies. The J function can be written as

$$\begin{aligned} J &= \frac{1}{2}\sum_{k=1}^{K}(E_{\text{rep}}^{k} + E_{\text{elec}}^{k} - E_{\text{ref}}^{k})^2 \\ &= \frac{1}{2}||\mathbf{e}^{\text{rep}} - (\mathbf{e}^{\text{ref}} - \mathbf{e}^{\text{elect}})||_2^2 = \frac{1}{2}||\mathbf{Vc} + \mathbf{W}\boldsymbol{\epsilon} - \mathbf{e}||_2^2 \end{aligned}, \tag{2.56}$$

where

$$\mathbf{V} = \underbrace{\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{K1} & v_{K2} & \cdots & v_{KN} \end{bmatrix}}_{v\in\mathbb{R}^{K\times N}}, \mathbf{e} = \underbrace{\begin{bmatrix} E_{\text{ref}}^{1} - E_{\text{elec}}^{1} \\ E_{\text{ref}}^{2} - E_{\text{elec}}^{2} \\ \vdots \\ E_{\text{ref}}^{K} - E_{\text{elec}}^{K} \end{bmatrix}}_{E_{\text{ref}}-E_{\text{elec}}\in\mathbb{R}^{K}}, \mathbf{W}^T = \underbrace{\begin{bmatrix} \mathbf{w}^1 \\ \mathbf{w}^2 \\ \vdots \\ \mathbf{w}^K \end{bmatrix}}_{\mathbf{w}\in\mathbb{R}^{m\times 1}}, \tag{2.57}$$

where N is the number of nodes in cubic spline functions, $\mathbf{e}$ is the error between the reference and DFTB electronic energy, and $\mathbf{W}$ is a collection of $\mathbf{w}$ in Eq. (2.55) for all configurations. Eq. (2.56) can be written as

$$J = \frac{1}{2}||\boldsymbol{M}\boldsymbol{x} - \boldsymbol{e}||_2^2, \tag{2.58}$$

where $\boldsymbol{M} = [\boldsymbol{V} \quad \boldsymbol{W}]$ and $\boldsymbol{x}^T = [\boldsymbol{c} \quad \boldsymbol{\epsilon}]$. The minimization of the objective function $\frac{1}{2}||\mathbf{Mx\text{-}e}||_2^2$ can be transferred to a quadratic programming (QP) problem

$$\begin{aligned} &\min(\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x}) \\ &\text{Subject to } \mathbf{G}\mathbf{x} < \mathbf{h} \end{aligned}, \tag{2.59}$$

where $\mathbf{P} = \mathbf{M}^T\mathbf{M}$ and $\mathbf{q} = -\mathbf{M}^T\mathbf{e}$. The $\mathbf{G}$ and $\mathbf{h}$ are constraint matrices. The CCS method [46] can effectively avoid the oscillations in the second derivatives of the repulsive potentials and the issue of sparse data by incorporating monotonous and sparsity constraints. However, the repulsive potentials obtained from CCS have limited global transferability, and this is also a common problem for pairwise potentials. Other approaches have been proposed to address this limitation, such as considering many-body effects using non-linear neural networks [61] or using a Chebyshev polynomial-based approach [47].

3 Machine Learning Review

Tom M. Mitchell [107] proposed a widely accepted definition of machine learning, which is often cited: *A computational algorithm learns from experience E concerning task T and performance P of task T improves with experience E.* The machine learning tasks in the current landscape can be broadly classified into two main categories: classification and regression. Classification is about predicting the labels or categories of a given input, usually discrete, while regression is about predicting the continuous values with a given input. Regression tasks are commonly encountered within the domains of physics and chemistry, involving the prediction of physical properties. Machine learning approaches can be further categorized into three main types: supervised, unsupervised, and reinforcement. For our purposes, we will primarily focus on supervised learning. Supervised learning typically requires input and output data (targets), while unsupervised learning focuses on learning concise representations of unlabelled input data. Reinforcement learning, on the other hand, is based on reward and teaches a learner how to behave in order to maximize the cumulative reward. For supervised learning, the input is usually derived from geometries for atomic simulation purposes, while the output data of the simulations represents the machine learning targets. These input and output data can be split for training and testing data sets. The supervised learning algorithms usually undergo iterative optimization using the training data set, allowing them to learn from the input-output pairs. Subsequently, the trained models are evaluated using the testing data to assess their performance.

In this chapter, our attention shifts toward the theoretical aspects of supervised machine learning, specifically focusing on the machine learning algorithms and methods employed in this thesis. We examine the commonly used workflow for supervised machine learning in scientific research, encompassing data collection, techniques for generating machine learning input through data representation, selecting appropriate machine learning models, and the algorithms utilized for the learning process. Subsequently, we discuss each component individually, emphasizing the machine learning algorithms employed in this thesis: the random forest algorithm and neural networks.

3.1 Machine Learning Workflow

The development of science paradigms accompanies the research workflow evolutions. As shown in Figure 3.1, Butler et al. [18] introduced the evolution of the research workflow from the computational science paradigm to the data-driven science paradigm. The first generation follows a traditional approach where the input is geometric data and is processed using computational methods. The second generation employs global optimization algorithms, such as the evolutionary algorithm, to generate chemical structures based on their composition. The third generation focuses on the machine learning workflow, encompassing the following procedures:

1. Data set collection, usually includes geometries and machine learning targets

2. Representation of chemical systems and machine learning input generation
3. Selection of type of learning according to the learning task
4. Selection of an appropriate machine learning algorithm, fine-tuning hyperparameters (parameters controlling the learning process, such as the learning rate), training, and validating the machine learning model

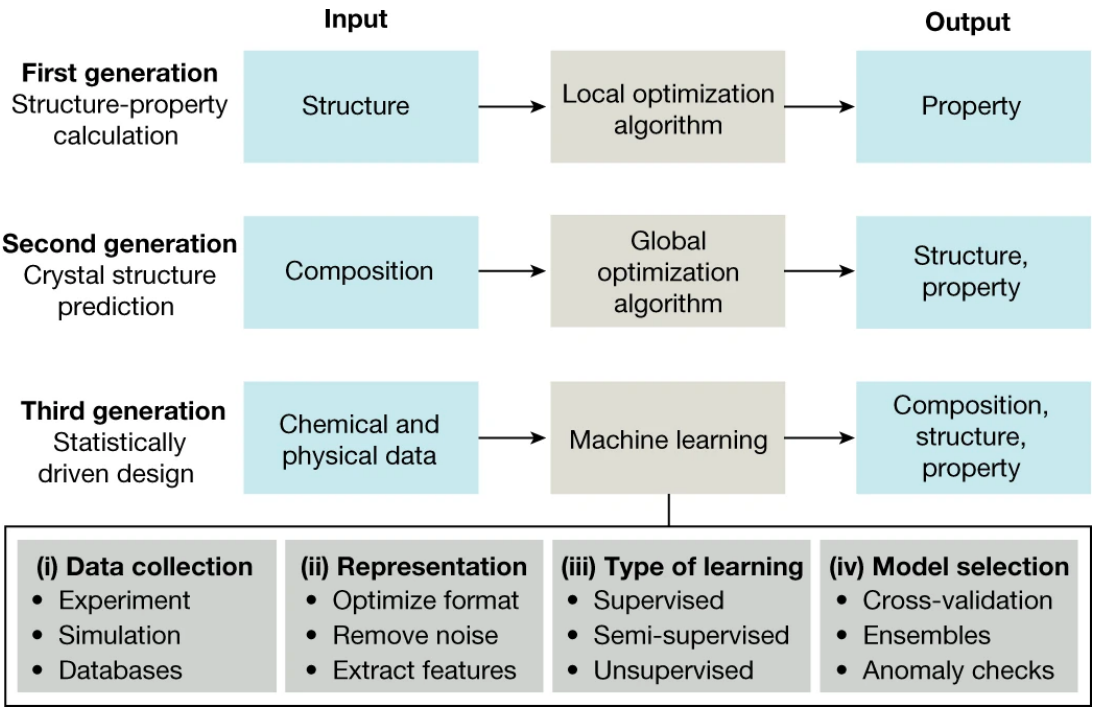


Figure 3.1: Evolution of the research workflow in computational chemistry. Adapted from reference [18].

The outlined steps provide a comprehensive overview of the machine learning workflow employed in this thesis. Machine learning serves as a means of parametrizing the data set. The data set typically consists of geometries and corresponding machine learning targets, which can be physical properties like energies [108] or physical model parameters [57] for learning used in calculations. The chosen representation method effectively extracts pertinent information from the geometries as machine learning input, enabling a representation of the chemical environment based on geometries. Once machine learning input and output have been acquired, the subsequent steps involve selecting suitable machine learning algorithms and determining their corresponding hyperparameters for model training. Commonly used machine learning algorithms include neural networks (NNs) [22, 34], support vector machine (SVM) [109], random forest (RF) [110, 111], and Gaussian process regression (GPR) [4]. For many scientific problems, machine learning algorithms have been extensively tested, and the results have shown that certain algorithms exhibit more significant promise than others for specific predictions [111, 112]. For example, the random forest algorithm has demonstrated superior prediction capabilities in the context of synthetic reactions within multidimensional chemical space [111]. When dealing with a

specific task that has yet to be previously investigated, it is crucial to assess the predictive capabilities of machine learning algorithms on testing data sets. Furthermore, the tuning of hyperparameters, particularly in the case of neural networks, holds significant importance [53]. Properly optimizing hyperparameters can greatly enhance the performance of machine learning algorithms. Subsequent sections will provide comprehensive explanations of each step, emphasizing the methods employed in this thesis.

3.2 Data Collection and Visualization

The generation of data sets is a critical component of the data-driven science paradigm, as it directly impacts the reliability and applicability of machine learning models in practical applications [113]. The data set used for machine learning should represent the task at hand well. If the distributions of the data set used for training and testing are too narrow, the resulting machine learning model can not be applied to real-world problems. Typical data sets used for machine learning to enhance atomic simulations include data from *ab initio* molecular dynamics (AIMD) trajectories [114, 115] and data sets of small molecules and molecular conformers. Well-known materials data resources, such as the materials project [116], automatic-flow for materials discovery (AFLOW) [117], computational materials repository (CMR) [118], novel-materials-discovery (NOMAD) [119], and organic materials database (OMD) [120] provide valuable data sets for research purposes. Moreover, specific data sets of small molecules like ANI-1 [108] and ANIx [121], which use the ANAKIN-ME model (Accurate NeurAl networK engINe for Molecular Energies), or ANI for short, have been used in this thesis and previous research [66]. ANI-1 is a data set that spans conformational and configurational space and contains small organic molecules of up to 8 heavy atoms, demonstrating its applicability to much larger systems of 10–24 heavy atoms [108]. Other widely used data sets include quantum mechanical (QM) based data sets QM7 [122] and QM9 [123], and molecular dynamics 17 (MD17) [124].

Visualizing and analyzing the data sets is desirable once the machine learning data set is generated. Though these visualization methods have not been applied in this thesis, they are important to comprehend the underlying patterns and identify data points with unusual attributes, especially for new data sets. A typical data set for organic molecules comprises millions of geometries [121, 122, 124], along with ranges of properties such as energies and band gaps. The coordinates of geometries are high-dimensional, $3n$ for n atoms. Dimension reduction methods have been used to represent and visualize the geometric patterns of a data set. Commonly used methods include kernel principal component analysis (KPCA) [125], t-distributed stochastic neighbour embedding (t-SNE) [126], and sketch-map [127]. The basic idea of these dimension reduction methods is that if geometries are similar in high-dimensional space, they will remain close to each other in low-dimensional mapping. The dimension reduction methods can be used to analyze the data set distributions. For instance, Cheng et al. [20] used KPCA to map amorphous carbon in two-dimensional projections. The results show that carbon atoms with different chemical environments have been automatically separated.

3.3 Data Representation

The representation method, also called descriptor, fingerprint, or feature, is crucial in applied machine learning. It involves transforming Cartesian coordinates to machine learning input and extracting patterns and regularities from geometries. Significant efforts in this field [128, 129] have greatly advanced the application of machine learning in physics, chemistry, and materials science. To make the learning efficient, the representation method should be invariant to translational, rotational, and permutational symmetries of given geometries [130, 131]. Besides symmetry invariances, practical representation requirements include completeness, smoothness, and additivity [19]. Completeness means that inequivalent geometries should be different, smoothness requires that smooth deformations correspond to smooth representation method output, and additivity suggests that a representation of geometry should be allowed to be decomposed into a sum of local environments (such as atom centred environments). Several previous reviews [19, 132] summarise current representation development. Michele et al. [19] have summarised the current most commonly used features of atomic geometries in seven phylogenetic trees. The commonly used groups include potential fields, density correlation features, and atomic symmetry functions. The limitations of different representations are still under debate [22, 133, 134, 135, 136, 137]. The chosen method should uniquely represent the chemical environments and be computationally accessible. The choice of representation method usually depends on the machine learning task, data set size, and machine learning algorithms. Imbalzano and co-workers [138] suggest that a simple representation method with fewer dimensions can perform better using the testing data set because complex representation tends to overfit the training data set. Guyon and coworkers [139] introduce a general workflow of representation method selection.

In this section, we introduce two representation methods that have been widely used in machine learning applications in science: smooth overlap of atomic positions (SOAP) [137] and atom-centred symmetry functions (ACSFs) [128]. These methods will be further utilized in the subsequent parts of this thesis.

Smooth overlap of atomic positions

SOAP is a widely used method, especially in kernel-based machine learning algorithms [140, 4]. SOAP belongs to the density correlation family of the representation method. Besides SOAP, density correlation family includes Faber-Christensen-Huang-Lilienfeld (FCHL) [134, 141], N-body iterative contraction of equivariants (NICE) [142], spectral neighbour analysis potential (SNAP) [143], the moment tensor potential (MTP) [144], and atomic cluster expansion (ACE) [145]. SOAP represents the atomic environment by expanding a Gaussian atomic density on each atom.

$$\rho^Z(\mathbf{r}) = \sum_i^Z e^{-\frac{1}{2\sigma^2}|\mathbf{r}-\mathbf{R}_i|^2}, \tag{3.1}$$

where the summation for i runs over all the atoms in the system with the atomic number

Z and position $\mathbf{R}_i$. The width of Gaussian atomic density is controlled by the smearing σ. When expanding the Gaussian density with orthonormal radial basis functions and spherical harmonics, we get

$$\rho^Z(\mathbf{r}) = \sum_{nlm} c^Z_{nlm} g_n(\mathbf{r}) Y_{lm}(\theta, \phi), \tag{3.2}$$

where $g_n(\mathbf{r})$ are the orthonormal radial basis functions, $Y_{lm}(\theta, \phi)$ are the spherical harmonics and n, l, m are the quantum numbers. The parameter c^Z_{nlm} can be generated from an inner product:

$$c^Z_{nlm} = \iiint_{\Re^3} dV g_n(\mathbf{r}) Y_{lm}(\theta, \phi) \rho^Z(\mathbf{r}). \tag{3.3}$$

With parameters c^Z_{nlm}, the partial power spectra defined in previous work [137] as SOAP output is

$$p^{Z_1,Z_2}_{nn'l} = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c^{Z_1}_{nlm})^* c^{Z_2}_{nlm}, \tag{3.4}$$

where $n_{\max}$ are the number of radial basis functions and $l_{\max}$ are the maximum degree of spherical harmonics. The partial power spectra vector $p^{Z_1,Z_2}_{nn'l}$ encompasses the interactions between unique atomic pairs Z_1 and Z_2, unique radical basis functions n and n' up to $n_{\max}$, and angular degree values up to $l_{\max}$. By predefining the element species, $n_{\max}$, and $l_{\max}$, the final output of the SOAP descriptor will be a collection of $p^{Z_1,Z_2}_{nn'l}$ with different Z, n and l to represent atomic environments.

Atom-centred symmetry functions

The idea of ACSFs is to transform positions into symmetry functions and satisfy translational and rotational invariant principles. Besides, ACSFs show reasonable computational efficiency and achieve high accuracy in a previous work [146]. With the introduction of cutoff functions f_c, radial functions, and angular functions, ACSFs can be defined as:

$$
\begin{aligned}
G_i^1 &= \sum_{j\neq i}^{\text{all}} f_c(R_{ij}) \\
G_i^2 &= \sum_{j\neq i}^{\text{all}} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}) \\
G_i^3 &= \sum_{j\neq i}^{\text{all}} \cos(\kappa R_{ij}) f_c(R_{ij}) \\
G_i^4 &= 2^{1-\zeta} \sum_{j,k\neq i}^{\text{all}} (1+\lambda\cos\theta_{ijk})^{\zeta} e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \\
G_i^5 &= 2^{1-\zeta} \sum_{j,k\neq i}^{\text{all}} (1+\lambda\cos\theta_{ijk})^{\zeta} e^{-\eta(R_{ij}^2+R_{ik}^2)} f_c(R_{ij}) f_c(R_{ik}),
\end{aligned}
\tag{3.5}
$$

where G^1 is the sum of the predefined cutoff function f_c, G^2 is a term that multiplies Gaussians and cutoff function values, and G^3 is a Fourier-series-like expansion with the radial environment, which should be carefully used because there may appear both positive and negative terms in the sum. G_i^4 and G_i^5 are typical angular environment parameters. R_{ij} is the distance between atoms i and j, R_s is a radial distance parameter to control the distance shift, and parameter η controls the width of the Gaussian functions. Parameter κ in G_3 can tune the period length in the cosine functions. In G_i^4 and G_i^5, the parameter ζ is responsible for the angular resolution while λ is -1 or +1.

The ACSFs are defined for combinations of unique elements and unique element pairs, resulting in a large number of unique pairs of angular symmetry functions when the system contains multiple element species, given by $\frac{1}{2}N_{elem}(N_{elem}+1)$, where N_{elem} is the number of unique element species. The number of unique pairs increases rapidly with the number of unique element species. To overcome this problem, weighted ACSFs (wACSFs) have been proposed [147]. The so-called element-dependent weighting functions have been introduced in wACSFs. The weighting functions can represent the chemical environment of different element pairs and avoid the need for using element pair combinations. The wACSFs were tested using the QM9 data set and achieved comparable performance to the results obtained with ACSFs.

3.4 Machine Learning Model Selection and Estimation

With given data representation as input and machine learning targets as output, the machine learning model selection aims to find the best machine learning algorithms on the learning data set D. In general, we have to split the data set into the training set and the testing set. Hyperparameters θ are used to control the machine learning algorithms. Therefore, the task is to evaluate the learning algorithms and tune hyperparameters θ, so that the selected algorithm can perform well with the testing data set. The errors on the

training data set and testing data set are called training errors and generalization errors. The ideal model generally has well-tuned hyperparameters that generate the minimum generalization error. The learning models must be trained with chosen hyperparameters to yield generalization errors. The training errors can not be used to evaluate the performance of the learning models since we can not exclude the possibility of overfitting and underfitting. Overfitting refers to a phenomenon where a machine learning model achieves a high level of performance on the training data but fails to predict the testing set. Overfitting is typically accompanied by the inclusion of irrelevant details and noise, which suggests that the model is simply memorizing the training data instead of learning from it. In contrast, underfitting is the opposite, where the model fails to capture the underlying patterns in the training data. Unlike overfitting, the solution for underfitting can be increasing the complexity of the training model by adding more parameters.

Many methods have been developed to select and estimate machine learning models. The frequently used methods include hold out, $k-$fold cross validation, and bootstrapping. When using the hold out method, the data set D will be split into training data set S and testing data set T, where $D = S \cup T$ and $S \cap T = \varnothing$. The testing data set is used to estimate the performance of the generalization error. One important issue is to determine the ratios of training and testing data sets. If the training ratio is high, the training data set S is close to data set D, and the estimation based on testing data set T will become unstable. In contrast, if the training ratio is low, the training data set S will fail to capture the underlying patterns of data set D, resulting in a fidelity issue. Usually the training ratio is between $\frac{2}{3}$ and $\frac{4}{5}$ [125]. The $k-$fold cross validation splits the data set D into k folds $D_1 \cdots D_k$, where $D = D_1 \cup D_2 \cdots \cup D_k, D_i \cap D_j = \varnothing (i \neq j)$. The model is then trained and evaluated sets for k times, with each fold serving as a training and testing set. In each iteration, one fold is used as the testing set, and the remaining $k-1$ folds are used for training the model. The final performance of the model is typically assessed by aggregating the performance measures obtained from k iterations. The $k-$fold cross validation is unsuitable when the data set is huge because of the high computational expense. The basic idea of another sampling technique, bootstrapping, is to create multiple data sets by sampling with replacement from the original data set D with n samples. By repeatedly sampling from the data, new data set D' is generated, each time with the same size. For n samplings, the limit of the samples that had not been selected is $\lim_{n\to\infty}(1-\frac{1}{n})^n \approx 0.368$. Bootstrapping is especially popular in ensemble learning [148].

For the machine learning algorithm, the hyperparameters will also influence the performance of model selection. The practical solution to tune hyperparameters is to choose values from the hyperparameter ranges. For example, if we have three hyperparameters and we choose five values for each hyperparameter, we have 125 models in total. Hyperparameter tuning is the essential yet computationally expensive step in some cases, especially in neural networks with large training sets. To distinguish it from the testing data set used for model selection, the data set used for hyperparameter tuning is usually called the validation data set. The workflow with validation data set for hyperparameter tuning is the so-called training-validation-test protocol, and this protocol is frequently used to tune hyperparameters in machine learning, especially in neural networks.

3.5 Machine Learning Algorithms

3.5.1 Random Forest

The random forest is a widely used ensemble algorithm for both classification and regression tasks [148, 149, 150, 151]. Ensemble learning, a powerful machine learning technique, aims to enhance the predictive performance of a model by combining multiple individual learners (known as weak learners) to create a more robust and accurate strong learner. In the context of classification tasks, a weak learner [152] refers to a relatively simple and less accurate learning algorithm. However, it is expected to perform better than random classification, achieving an accuracy of more than 50%. On the other hand, a strong learner represents a powerful learning algorithm (such as the random forest) with high accuracy and impressive predictive capabilities in the given task. The key objective of ensemble learning lies in determining whether the performance obtained from multiple weak learners surpasses that of any individual constituent weak learning algorithm alone. According to Hoeffding's inequality [153], assuming weak learners are independent of each other, the errors from the ensemble learning (strong learner) decrease significantly and eventually tend towards zero as the number of weak learners increases. This property allows ensemble methods like the random forest to achieve remarkable predictive accuracy.

The decision tree algorithm [154] is a commonly employed weak learner within the random forest ensemble algorithm. The random forest is an extension of the bagging method pioneered by Breiman [148]. Both the random forest and bagging algorithms rely on the decision tree algorithm as its fundamental building block. To provide a clear understanding of the concept of the random forest algorithm, we will briefly introduce the decision tree and bagging algorithms before delving into the random forest.

Decision Tree

A decision tree is a hierarchical, tree-like model representing decisions and their possible consequences or outcomes and can be used for both classification and regression tasks. The hierarchy of the tree suggests that it is an ordered structure and a directed acyclic graph composed of a set of nodes. Each node represents a decision, with the simplest case being binary, as depicted in Figure 3.2. The crucial matter is the division of the data set within the decision node, which will be elaborated upon in subsequent discussions.

The node at the top of the tree is the root node, while the leaf nodes (leaves) are at the bottom. Figure 3.2 shows an example of a decision tree, where the root node has two child nodes, indicating a binary decision is made to split the root node. The node to be split is the parent node, and the resulting nodes after splitting are called child nodes. In the case of binary decision trees, the child nodes are typically referred to as the left child node and the right child node. The subsets (data sets of child nodes) should be non-empty and disjoint. Training a decision tree should be based on a given data set, and we consider a data set shown in Table 1 to clarify how to train (grow) a decision tree and how to split the data set of a decision node into two subsets.

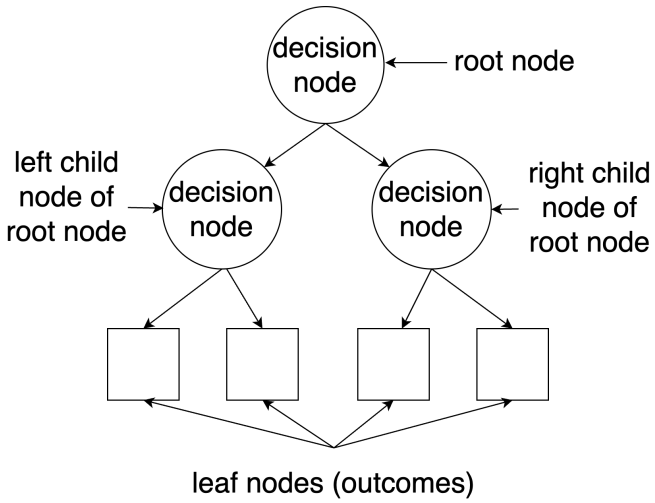


Figure 3.2: Illustration of a decision tree.

Table 1: Data set D for the binary classification task.

label	x_1	x_2	x_3	y
1	0	0	0	0
2	0	1	1	0
3	1	1	1	0
4	0	0	0	1
5	1	1	0	1
6	1	1	1	1

In Table 1, the variables x_1, x_2, and x_3 represent the input data (or features), while y denotes the reference data to be learned. The label {1, 2, 3, 4, 5, 6} indicates the presence of six samples in the data set. The decision tree algorithm recursively splits the data into subsets to construct the tree structure. This process of splitting will continue until a predetermined threshold is reached. When using the data set from Table 1, at each split, the decision tree algorithm selects the best feature among x_1, x_2, and x_3. The criterion for selecting the best feature is based on maximizing the so-called impurity decrease. A predefined objective function is employed to quantify the impurity decrease, and the highest value of this function corresponds to the maximum impurity reduction. The objective function of node i and feature j can be defined as:

$$\Delta(D^i, x_j) = I(D^i, x_j) - p_L I(D^i_L, x_j) - p_R I(D^i_L, x_j), \tag{3.6}$$

where x_j are the chosen features, and D^i_R and D^i_L are the right child data set and left data set after splitting. p_L is the ratio of samples of the subset D^i_L to the data set D^i, the same is for p_R. The I is the impurity function, and there are two main impurity functions: the entropy and the Gini impurity functions. The entropy impurity function is $I_E = -\sum_k p_k \log_2 p_k$, where k is the samples with k-th class and p_k is the ratio of such samples. In the data set presented in Table 1, the variable k can take on values of 0 or 1, as there are only two classes for y. The Gini impurity function is $I_G = 1 - \sum_k p_k^2$. With

defined impurity functions, the objective function and the given data set, we can search over all the features and obtain the best feature corresponding to the maximum objective function value. Using the best feature, the data set of the parent node is split into D^L and D^R. When using the data set from Table 1 and the entropy impurity function, and choosing feature x_1, we can get the $I_E(D, x_1) = -(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}) = 1$. The subset D^1 where the feature x_1 equals 0 includes the data set labels {1, 2, 4}. The y values of these three labels are $\frac{2}{3}$ for $y = 0$ and $\frac{1}{3}$ for $y = 1$. Similarly, the subset D^2 where the feature x_1 equals to 1 includes the data set labels {3, 5, 6}. The y values of these three labels are $\frac{1}{3}$ for $y = 0$ and $\frac{2}{3}$ for $y = 1$. The returned entropy function values are:

$$\begin{aligned} I_E(D^1, x_1) &= -(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}) = 0.92, \\ I_E(D^2, x_1) &= -(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}) = 0.92, \end{aligned} \tag{3.7}$$

then we get the returned objective function value of x_1 is

$$\Delta(D, x_1) = I_E(D, x_1) - (\frac{3}{6} * 0.92 + \frac{3}{6} * 0.92) = 0.08. \tag{3.8}$$

When using the entropy impurity function, the values of the objective function are also called entropy gain, which measures the impurity decrease of the data set splitting with a chosen feature. Similarly, we get that the $\Delta(D, x_2)$ and $\Delta(D, x_3)$ are 0 and 0.08. When choosing x_2 to split the data set D, the possibilities of y equals 0 and 1 are 50 percent in both two subsets, which equals the possibilities in data set D. This means the entropy gain of feature x_2 is 0, suggesting no impurity decrease. The value of the impurity gain of x_1 and x_3 are the same. In this case, we randomly choose one of these two features as the best feature to split the data set D.

We have stated how to split the node and determined the best feature. For practical implementations, we will set a minimum entropy gain (impurity decrease) value, minimum data set size in the node, and the maximum depth. The depth of a decision tree is the length of the longest path from a root node to leaf nodes. For instance, if we set the minimum size of the data set to be 4, the decision tree using the data set from Table 1 will only split once since the left and right subsets contain only 3 samples after splitting. When reaching the threshold of the setting values, the tree will stop splitting and obtain leaf nodes in Figure 3.2. In addition to the training process of a decision tree, the prediction process and overfitting are also crucial aspects of the decision tree algorithm. Previous reviews [155] have discussed overfitting in the training process of decision trees, and we will not delve further into this topic.

From Bagging to Random Forest

The bagging algorithm uses the bootstrapping sampling method to select data set D_{bs} from the original data set. The workflow of bagging is shown in Algorithm 1. $\mathbb{I}$(input) is a function that returns one if the input is True else zero. The decision tree is chosen as the training algorithm $\mathfrak{L}$ in Algorithm 1. Finally, bagging usually uses the majority

vote for the output based on weak learners for calcification tasks and the average value for regression tasks.

Algorithm 1 Bagging Algorithm

Input:

Training data set $D = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)$ with n samples
Learning algorithm $\mathfrak{L}$
Training step T

1: **for** each $i \in [1, T]$ **do**
2: $h_t = \mathfrak{L}(D, D_{\text{bs}})$
3: **end for**

Output:

$H(\mathbf{x}) = \text{argmax} \sum_i \mathbb{I}(h_i(\mathbf{x}) = y)$

Based on the bagging method, the random forest introduces so-called feature bagging, which means that we will randomly choose the subset of the feature for each decision node in the learning processes. Assuming the dimension of the feature is d and the parameter k controls the random choosing features. When k equals d, it is the bagging algorithm. In general, the recommended value of k in the random forest method is $\log_2 d$ [148]. The recommended value ensures that each tree in the random forest method is unique and captures different aspects of the data. The random forest training will be more efficient than bagging and can achieve lower generalization errors—the higher efficiency and better performance result from the random feature selection. The diversity of the single weak learner decision tree usually contributes to the lower generalization error. In bagging, one or a few features can be strong predictors and dominate many single learners, decreasing the diversity of the single trees.

The random forest can be used to analyze the variable (feature) importance measurement (VIM). To measure the k-th feature importance, we can use the out-of-bag (OOB) [148] or Gini index to compute the importance. The Gini index is defined using the previously defined Gini impurity function:

$$GI_{it}^k = \text{Gini\_index}_{it}(D, x_k) = \sum_j p_j I_G(D_j), \tag{3.9}$$

where D is the data set of node i in the t-th decision tree, D_j are the subsets of node i, p_j are the ratios of D_j and x_k is the k-th variable. When specifying the Gini index change of node i in the t-th binary decision tree, we get

$$\text{VIM}_{it}^k = GI_{it}^k - GI_{l,it}^k - GI_{r,it}^k, \tag{3.10}$$

where GI_{it}^k is the Gini index before splitting while $GI_{l,it}^k$ and $GI_{r,it}^k$ are the left and right nodes after splitting. Then for the k-th feature importance, we sum over all nodes in a single decision tree and all decision trees used in ensemble learning

$$\text{VIM}^k = \sum_i \sum_t \text{VIM}_{it}^k. \tag{3.11}$$

Usually, when we get all the variable importance, we will do a normalization for the VIM. The obtained VIM values enable the analysis of the random forest model and help to understand the relationships between the input-output pairs. The detailed information for VIM using OOB can be found in previous reference [148].

3.5.2 Neural Networks

The concept of artificial neural networks draws inspiration from biological neural networks. In biological systems, neural networks consist of interconnected neurons communicating through chemical and electrical synapses. Artificial neural networks emulate this idea, where neurons become activated when their values surpass a certain threshold. In 1943, McCulloch and Pitts [156] proposed a simple M-P model known as the perceptron. However, it was later acknowledged by Marvin and Seymour [157] in 1969 that a single-layer perceptron was inadequate for solving non-linear problems, and implementing a multilayer perceptron (MLP) was deemed unrealistic due to computational constraints and hardware limitations. The development of the backpropagation algorithm in the 1970s and 1980s [12] significantly advanced research on neural networks. This algorithm played a crucial role in enabling efficient training of multilayer perceptrons. Furthermore, in the 2010s, the emergence of big data propelled neural networks into the spotlight with the advent of deep learning. We will introduce the neural networks starting from the perceptron model, followed by MLP and backpropagation algorithm.

Perceptron model

An M-P perceptron neuron has been illustrated in Figure 3.3.

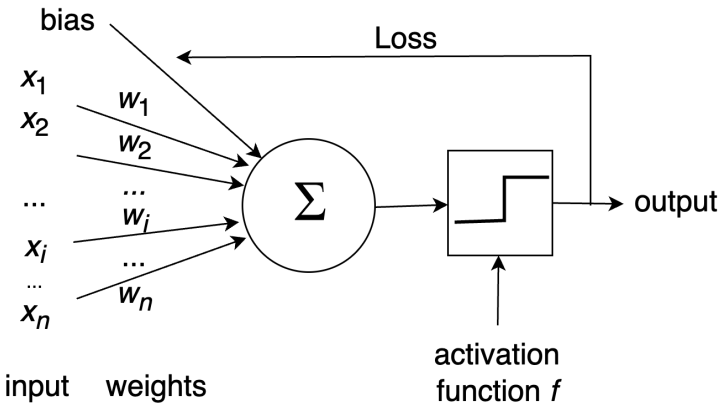


Figure 3.3: Illustration of a M-P perceptron neuron.

As shown in Figure 3.3, a neuron connects with the input and weight parameters, and we get the prediction of $\hat{y} = f(\sum_i w_i x_i + b - \theta)$, where b is the bias, f is the used activation function and θ is the threshold of the activation function. The value of

the neuron received is $\sum_i w_i x_i + b$. Once the neuron receives the value, this value will be compared with the threshold and generate output by utilizing the activation function. The activation function controls the connections between neurons. The simplest activation function is a binary function, which returns 1 if the output reaches the threshold else 0. The extension of the binary function is a linear function $f(a) = a$, where a is $\sum_i w_i x_i + b$. Besides linear activation functions, commonly used non-linear activation functions include rectified linear unit (ReLU) and sigmoid. We have introduced the forward procedure in the perceptron model. Another key issue is updating the weight parameters to minimize errors between predictions and references. With a defined loss function J:

$$J = \frac{1}{2}\sum_{j=1}^{N}(y^j - \hat{y}^j)^2, \tag{3.12}$$

where y^j is the reference value of sample j and $\hat{y}^j$ is the prediction value of of sample j. A frequently used optimization algorithm is gradient descent, where the weight parameters w_i are updated by moving in the direction opposite to the first-order gradient, resulting in the steepest ascent:

$$w_i := w_i + \Delta w_i. \tag{3.13}$$

The gradient descent, or the batch gradient descent, involves calculating and summing up every sample. The batch in this context refers to using the entire training data set to compute the gradient during each iteration. The resulting equation is as follows:

$$\begin{aligned} \Delta w_i &= -\eta \frac{\delta J}{\delta w_i} \\ &= -\eta \frac{\delta}{\delta w_i}\frac{1}{2}\sum_j (y^j - \hat{y}^j)^2 \\ &= \eta \sum_j (y^j - \hat{y}^j) x_i^j, \end{aligned} \tag{3.14}$$

where x_i^j is the i-th dimension value of the sample j, and η is the learning rate. The batch gradient descent is simple, but it will be slow for large data sets. Stochastic gradient descent (SGD), or iterative gradient descent, is an option when optimizing a big data set. The SGD will update the weight using only one sample in each iteration:

$$\omega_i := \omega_i + \eta(y^j - \hat{y}^j)x_i^j \tag{3.15}$$

On the one hand, SGD is generally easier to converge than gradient descent since it updates the gradient more frequently. On the other hand, the batch gradient descent can be smoother than SGD. Between the batch gradient descent and SGD is the mini-batch gradient descent (select a fixed number of training samples as a mini-batch), often applied in machine learning. Another popular algorithm, adaptive moment estimation (Adam) [158] is a combination of two gradient descent methods [159], Momentum, and root mean squared propagation (RMSP), which is given by

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \end{aligned} \tag{3.16}$$

where m_t and v_t are parameters that consider the past gradients and past squared gradients at step t, g_t is the gradient of SGD at step t, while β_1 and β_2 are decay rates which are often close to 1.0. m_t and v_t are initialized to zero and β_1 and β_2 are close to 1. Therefore m_t and v_t in the initial time steps are biased towards zero. To avoid using the biased parameters m_t and v_t, the so-called bias-corrected $\hat{m}_t$ and $\hat{v}_t$ can be constructed as:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2}. \end{aligned} \tag{3.17}$$

Finally, the parameters w can be updated as:

$$w_{t+1} := w_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{3.18}$$

where the default value of ϵ is 10^{-8} and η is the learning rate. The Adam method tunes the past gradient and past squared gradients, which implies the previous-step gradients have been considered. Besides, Broyden–Fletcher–Goldfarb–Shanno (BFGS), Limited-memory BFGS (LBFGS) [160] and root mean square propagation (RMSP) [161] are also some methods frequently used for machine learning.

Multilayer perceptron model

The MLP model has multilayer linear functions, and each node will be a neuron with an applied activation function. In MLP, the layers between the input and output layers are hidden layers. First, we define a multilayer neural network with n layers, as shown in Figure 3.4.

The forward calculations in MLP can be written as follows:

$$\begin{aligned} a_j^1 &= f^1(\sum_i w_{ij}^1 x_i + b_j^1) = f^1(z_j^1) \\ &\cdots \\ a_j^k &= f^k(\sum_i w_{ij}^k a_i^{k-1} + b_j^k) = f^k(z_j^k) \\ &\cdots \\ a^n &= \sum_i w_i^n a_i^{n-1} + b^n = z^n, \end{aligned} \tag{3.19}$$

where x_i represent the i-th dimension values of the input. f^k represent the activation functions in the k-th layer, while a_j^k represent the output values of neuron j in the k-th layer. w_{ij}^k are the weight parameters in neural network in k-th layer, where i and j mean the neuron i in $(k-1)$-th layer and neuron j in k-th layer, and b_j^k are the bias parameters of neurons j in k-th layer. For the last layer, the MLP will be a linear multiplication

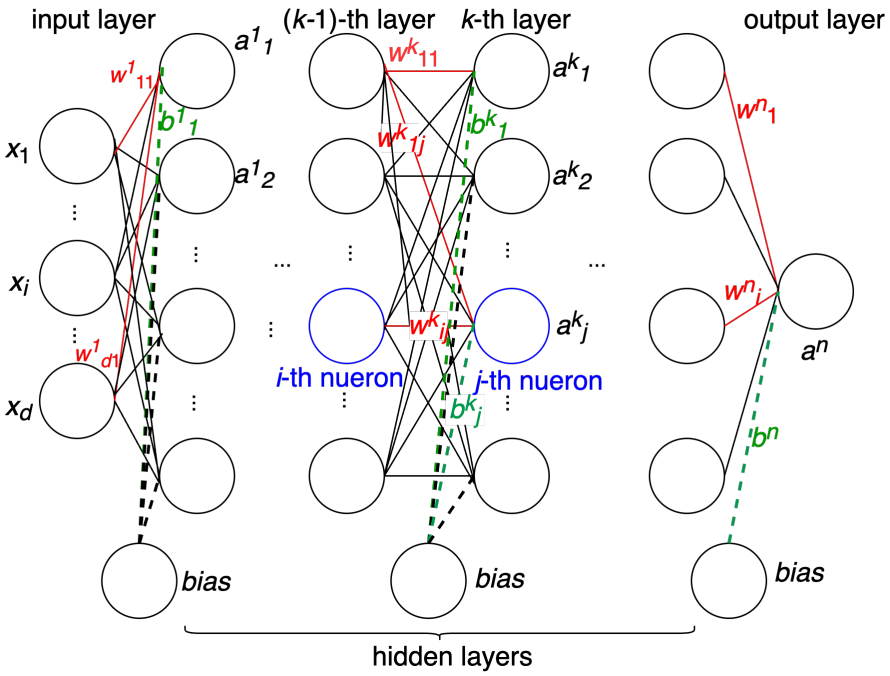


Figure 3.4: Illustration of the MLP algorithm.

to generate output a^n and return a loss function $J = \frac{1}{2}(a^n - y)^2$, where y represent the reference values.

So far, we have obtained the forward output of an MLP model. To minimize the error, backpropagation is typically applied to update the weights and biases. The algorithm for backpropagation is outlined in Algorithm 2. Here, we introduce a vectorized format for Eq. (3.19): $\mathbf{a}^k = f^k(\mathbf{z}^k) = f^k(\mathbf{w}^k\mathbf{a}^{k-1} + \mathbf{b}^k)$.

Algorithm 2 Backpropagation algorithm in MLP

1: compute forward calculations in Eq. (3.19) to get loss function J
2: compute the last layer gradient: $\boldsymbol{\delta}^n = \frac{\partial J}{\partial \mathbf{z}^n} = (\mathbf{z}^n - \mathbf{y})$
3: **for** $i = n - 1$ to 1 **do**
4: compute

$$\boldsymbol{\delta}^i = \frac{\partial J}{\partial \mathbf{z}^i} = ((\mathbf{w}^{i+1})^{\mathrm{T}}\boldsymbol{\delta}^{i+1}) \odot (f^i(\mathbf{z}^i))'$$

5: compute

$$\frac{\partial J}{\partial \mathbf{w}^{i+1}} = \boldsymbol{\delta}^{i+1}(\mathbf{a}^i)^{\mathrm{T}} \qquad \frac{\partial J}{\partial \mathbf{b}^{i+1}} = \boldsymbol{\delta}^{i+1}$$

6: **end for**

The $\odot$ is Hadamard product. With the backpropagation, the weight and bias for each layer can be updated. At this point, both the forward pass and the backward gradient update have been defined for the MLP model.

4 DFTB Parametrization for Lithium-Ion Batteries

Lithium-ion batteries (LIBs) have become ubiquitous energy storage devices in electronic cars and cell phones. However, to meet the demands of the industry, there is a pressing need for LIBs that are safer, have a longer lifetime, are more affordable, and have higher energy density. Solid-state batteries, Li-S, and Li-O_2/air batteries have been considered as potential candidates for the next generation of LIBs, with solid-state batteries being particularly promising due to their high energy densities and superior safety compared to conventional LIBs that rely on flammable liquid electrolytes. Recent studies have highlighted the challenges of developing solid-state batteries, including optimizing the interface between the solid electrolyte and electrodes, which is critical for achieving high performance in LIBs [162, 163].

This chapter details the development of a DFTB parameter set for the modelling of $Li_6(PS_4)SCl$ and $Li_5(PS_4)Cl_2$, both of which hold great potential as solid electrolytes in solid battery systems. These materials have attracted significant attention in both experimental [164, 165, 166] and theoretical fields [167, 168]. Previous studies have employed force field methods [167] to simulate the diffusion properties of such systems. DFTB was selected for its ability to provide a favourable compromise between computational efficiency and accuracy. This choice allows for calculating large-scale systems while simultaneously exploring their electronic properties. Our DFTB parametrization focused on solid-state batteries containing lithium, phosphorus, sulfur, and chlorine. Using the DFTB parameters, we can simulate the battery's geometric and electronic properties.

4.1 Data Sets and Methods

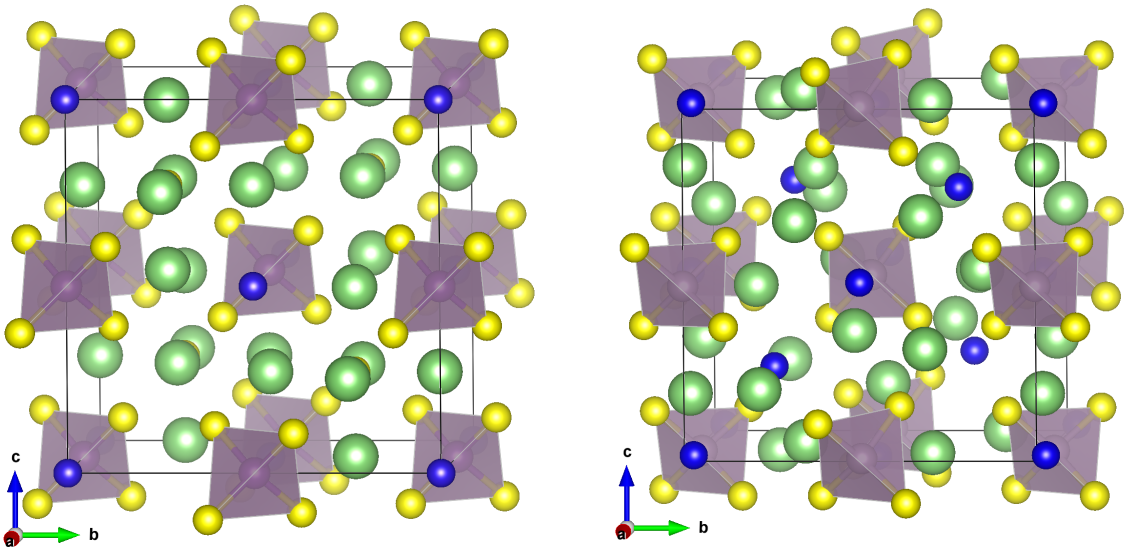
Data sets

The crystal structures of $Li_6(PS_4)SCl$ and $Li_5(PS_4)Cl_2$, as shown in Figure 4.1, were obtained from previous theoretical calculations reported in references [167, 169, 170]. $Li_6(PS_4)SCl$ has a cubic space group $F\bar{4}3m$ with a lattice parameter of 9.898 Å, based on X-ray synchrotron diffraction data [171]. $Li_5(PS_4)Cl_2$ was modelled by modifying $Li_6(PS_4)SCl$ through the removal of one lithium ion and the replacement of one sulfur atom with a chloride atom, as previously described in references [172, 173]. Unless otherwise specified, these geometries were used for DFT and DFTB calculations.

The crystal structures of cubic lithium, cubic sulfur, Li_3P, Li_2S, and LiCl were obtained from the materials project [116]. High-symmetry points for band structure calculations were automatically generated using the method introduced by Wahyu and Stefano [174].

DFT calculations

The reference data for the DFTB parametrization was obtained from full-electron



(a) Geometry of $Li_6(PS_4)SCl$ (b) Geometry of $Li_5(PS_4)Cl_2$

Figure 4.1: The green, purple, yellow, and blue balls represent lithium, phosphorus, sulfur, and chlorine atoms, respectively

DFT calculations using the Fritz Haber Institute *ab initio* molecular simulations (FHI-aims) [175]. For cubic $Li_6(PS_4)SCl$ and $Li_5(PS_4)Cl_2$, the **k**-mesh used for band structure and SCF-DFT calculations was set to $5 \times 5 \times 5$, whereas for geometry optimization, it was set to $3 \times 3 \times 3$. The lattice parameters obtained from DFT calculations using $3 \times 3 \times 3$ and $5 \times 5 \times 5$ grids are identical for geometry optimization. Additionally, the differences in atomic positions are all less than 1×10^{-3} Å. In the case of $Li_6(PS_4)SCl$, the maximum difference of band structure values between calculations performed using $3 \times 3 \times 3$ and $5 \times 5 \times 5$ is greater than 1 eV. However, the maximum difference between calculations performed using $5 \times 5 \times 5$ and $7 \times 7 \times 7$ grids is only at the level of 1×10^{-2} eV. As a result, a $5 \times 5 \times 5$ was selected for the band structure calculations. For geometry optimizations in this section, the Broyden–Fletcher–Shanno–Goldfarb (BFGS) algorithm [176] was utilized with a tolerance of 1×10^{-4} eV/Å. The SCF calculations utilized a total energy convergence of 1×10^{-4} eV and an eigenvalue convergence of 1×10^{-3} eV, with the Perdew-Burke-Ernzerhof (PBE) [89] functional. For FHI-aims, the different *tiers* [175] represent different accuracy levels of basis functions. If the *tier* is not mentioned, *tier* 2 (tight) was applied.

DFTB calculations

All SCC-DFTB calculations were performed using the DFTB+ package [177]. The maximum angular momentum for lithium, phosphorus, sulfur, and chlorine were set to *p*, *d*, *d*, and *d*, respectively. The **k**-mesh used in the DFTB calculations was set to match the one used in the DFT calculations. The electronic temperature was set to 300 Kelvin for all DFTB calculations, and the SCC tolerance was set to 1×10^{-6} electrons. Limited-memory

BFGS (LBFGS) [178] was used for geometry optimizations.

4.2 Electronic Parametrization

The DFTB parametrization was divided into the electronic parametrization and the creation of the repulsive potential. The electronic parametrization began with the `3ob-3-1` Slater-Koster files developed in a previous study [50]. For the initial step, we used the `3ob-3-1` parameters for phosphorus, sulfur, and chlorine while the lithium parameters were optimized. To generate basis parameters for lithium, we used a cubic lithium system with a primitive cell to determine the compression radii for lithium's s and p orbitals. The compression radii grid points for the lithium's s orbital were set to 2.25, 2.5, 3.0, 3.5, 4.0, and 4.5 Bohr, while those for the lithium's p orbital were set to 3.0, 3.5, 4.0, 4.5, 5.0, and 6.0 Bohr. We calculated the MAEs of the band structures for all grid points. The loss function was calculated as follows:

$$\text{Loss} = \frac{1}{N_i}\sum_i \frac{1}{N_v}\frac{1}{N_{\mathbf{k}}}\sum_v\sum_{\mathbf{k}} |\epsilon^{\text{DFT}}_{i,\mathbf{k},v} - \epsilon^{\text{DFTB}}_{i,\mathbf{k},v}| + |\frac{\partial\epsilon^{\text{DFT}}_{i,\mathbf{k},v}}{\partial\mathbf{k}} - \frac{\epsilon^{\text{DFTB}}_{i,\mathbf{k},v}}{\partial\mathbf{k}}|, \tag{4.1}$$

where N_i represents the number of geometries, $N_{\mathbf{k}}$ denotes the number of selected $\mathbf{k}$ points and N_v is the number of selected energy states. To prevent the selection of compression radii that lead to flattened conduction bands with small MAEs, the first derivative of band structure values in Eq. (4.1) is employed. Instances of flat conduction bands and small MAEs might arise during parametrization for particular compression radii. Hence, solely considering the MAEs of band structure values is insufficient, and by incorporating the first derivative into the loss function, the occurrence of flat conduction bands and small MAEs can be effectively avoided.

To minimize errors between DFT band structures and those obtained from DFTB calculations on the cubic lithium system, compression radii of 3.0 and 4.5 Bohr were used for the lithium's s and p orbitals, respectively. Slater-Koster tables were generated based on the compression radii of phosphorus, sulfur, and chlorine from the `3ob-3-1` set and the optimized compression radii of lithium. They were then used for the band structure calculations of $Li_6(PS_4)SCl$.

Figure 4.2 presents the band structure of $Li_6(PS_4)SCl$ obtained from FHI-aims and DFTB+. The results demonstrate that optimizing only the lithium basis parameters is insufficient for reproducing band structures from the DFT calculations. The band structures of the DFTB calculations exhibit three issues: a much larger gap between the valence bands near the valence band maximum (VBM) than that from DFT calculations (0.5 eV for DFT and 2.0 eV for DFTB), swapped valence bands (shown in Figure 4.2 by using arrows), and a smaller band gap than the gap from the DFT-PBE calculations, which is already known to underestimate the band gap.

We analyzed the density of states (DOS) and the projected DOS (PDOS) obtained from DFT calculations to identify the atomic orbitals responsible for the observed discrepancies. The analysis indicated that the p orbitals of sulfur and lithium dominate in the energy

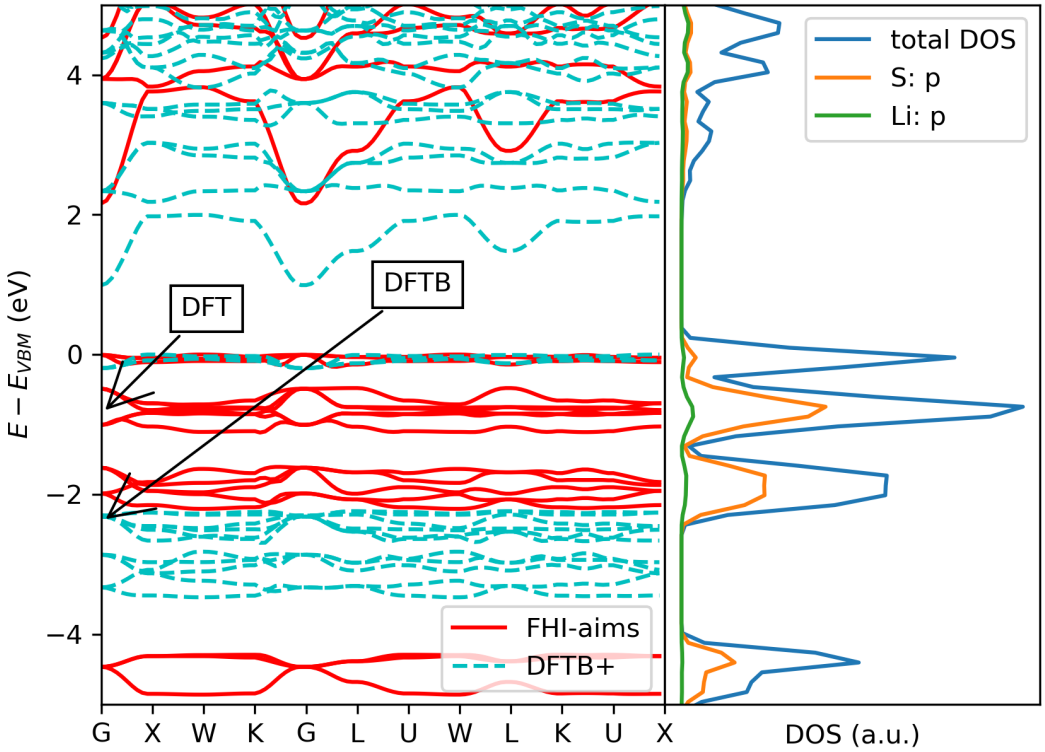


Figure 4.2: Band structure calculations of $Li_6(PS_4)SCl$ were obtained from DFT and DFTB calculations and DOS calculations of $Li_6(PS_4)SCl$ from DFT calculations. DFTB calculations were carried out using DFTB+ with `3ob-3-1` parameters for phosphorus, sulfur, and chlorine and optimized compression radii of lithium. In the SCC-DFTB calculations, the wavefunction compression radii of lithium's s and p orbitals were set to 3.0 and 4.5 Bohr, respectively. Band structures, DOS and PDOS calculations were performed using FHI-aims with the tight level basis set. The arrows show the swapped valence bands.

range of -1 eV to -3 eV. Based on our findings, we focused on refining the electronic parametrization of the lithium and sulfur orbitals. We used compression radii grid points of 3.0, 3.5, 4.0, 4.5, 5.0, 6.0, and 7.0 Bohr for the s and p orbitals of sulfur, and 3.5, 4.0, 4.5, 5.0, 6.0, 7.0, and 9.0 Bohr for the d orbital of sulfur. The compression radii grid points for lithium's s orbital were 2.25, 2.5, 3.0, 3.5, 4.0, and 4.5 Bohr, and for lithium's p orbital, they were 3.0, 3.5, 4.0, 4.5, 5.0, and 6.0 Bohr. Additionally, we optimized the on-site energies of sulfur, which systematically improves the band structures.

Effect of on-site energies

Typically, in DFTB, the on-site energies are determined from the eigenvalues of free

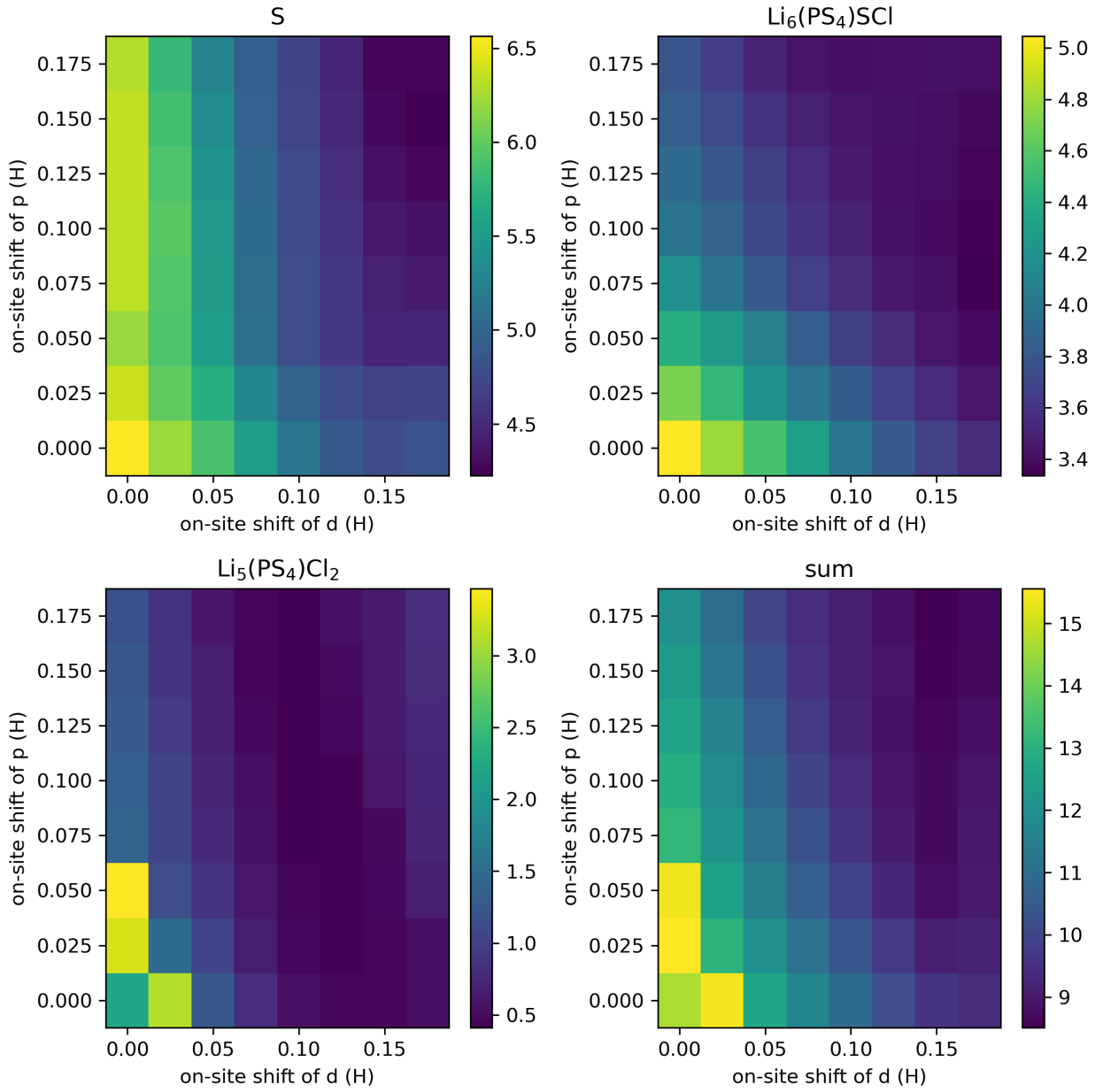


Figure 4.3: Effect of on-site energies of sulfur on the accuracy of band structure calculations. The heat maps represent the sum of MAEs of band structures and their first derivatives. The sum of errors for all three materials is shown in the last sub-figure with the title sum.

atoms, and the on-site energies of occupied orbitals are not adjusted. However, our investigation revealed that tuning the on-site energies of sulfur can significantly improve the band structures for all systems containing sulfur. As shown in Figure 4.3, by systematically shifting the on-site energies of sulfur p and d orbitals, we were able to enhance the accuracy of the DFTB band structure calculations for cubic sulfur, $Li_5(PS_4)Cl_2$, and $Li_6(PS_4)SCl$. We used a loss function defined in Eq. (4.1) to determine the on-site ener-

gies for the band structure optimization. The geometries used in the loss function include cubic sulfur, $Li_5(PS_4)Cl_2$ and $Li_6(PS_4)SCl$. The loss function was defined as the difference between the DFT and DFTB band structures, including both the MAEs of band structures and the first derivative of the band structures, as shown in Eq. (4.1). To determine the compression radii and on-site energies for lithium and sulfur, it is necessary to calculate the loss function for all possible combinations of these radii and on-site energies and select the combination that minimizes the loss function value.

To improve the accuracy of the DFTB band structure calculations, we systematically shifted the on-site energies of the sulfur p and d orbitals by 0.0, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15, and 0.175 Hartree. Figure 4.3 shows that for cubic sulfur, $Li_5(PS_4)Cl_2$, and $Li_6(PS_4)SCl$ systems, tuning the on-site energies of sulfur p and d orbitals can systematically improve the DFTB band structures. For the band structure of $Li_6(PS_4)SCl$, significant errors between DFT and DFTB occur where the p orbital of sulfur dominates the contributions to the DOS. Tuning the on-site energies of the p orbital significantly decreases the MAEs of $Li_6(PS_4)SCl$. We also found that shifting the on-site energy of the p orbital is crucial to solve the swapped energy states in the $Li_6(PS_4)SCl$ band structure.

When selecting the minimum MAEs based on Eq. (4.1), we obtained compression radii for the s and p orbitals of sulfur of 7.0 Bohr, while that of the d orbital is 4.0. The shifts in the on-site energies of the p and d orbitals of sulfur are 0.125 and 0.175 Hartree, respectively. However, choosing these basis parameters may pose difficulties in fitting repulsive potentials, as shown in Figure 4.4. This parameter set using the optimized compression radii and on-site energies tends to decrease electronic energies as the scaling ratio of cubic sulfur increases, making it challenging to fit repulsive potentials. The electronic energies refer to the energies from SCC-DFTB calculations without repulsive energies. The ideal electronic energies should increase as the scaling ratios increase. Therefore, we added a constraint condition that ensures that the electronic energy with a scaling ratio of 0.8 is lower than that with a scaling ratio of 1.2. This constraint condition ensures a general increasing tendency of electronic energies as the scaling ratio increases when searching for the compression radii and on-site energies in electronic parametrization.

With the constraint condition, new optimized parameters have been shown in Table 2. The on-site energies of the p and d orbitals of sulfur were optimized by shifting them by 0.125 and 0.15 Hartree to -0.13 and 0.17 Hartree, respectively. The compression radii for the s and p orbitals of sulfur are 3.5, and for d orbital is 4.0 Bohr, respectively, while for lithium, the compression radii for the s and p orbitals are 3.0 and 5.0 Bohr, respectively. By using the new optimized compression radii and the on-site energies of sulfur listed in Table 2, we can fix the issue of swapping energy states, as demonstrated in Figure 4.5 at the Gamma point.

Table 2: Optimized on-site energies (Hartree) and compression radii (Bohr). The subscript on-site indicates the value as the on-site energies, while r denotes the value for compression radii. The superscripts represent atomic orbitals.

$\text{sulfur}^{d}_{\text{on-site}}$	$\text{sulfur}^{p}_{\text{on-site}}$	$\text{sulfur}^{s,p}_{r}$	sulfur^{d}_{r}	lithium^{s}_{r}	lithium^{p}_{r}
0.17	-0.13	3.5	4.0	3.0	5.0

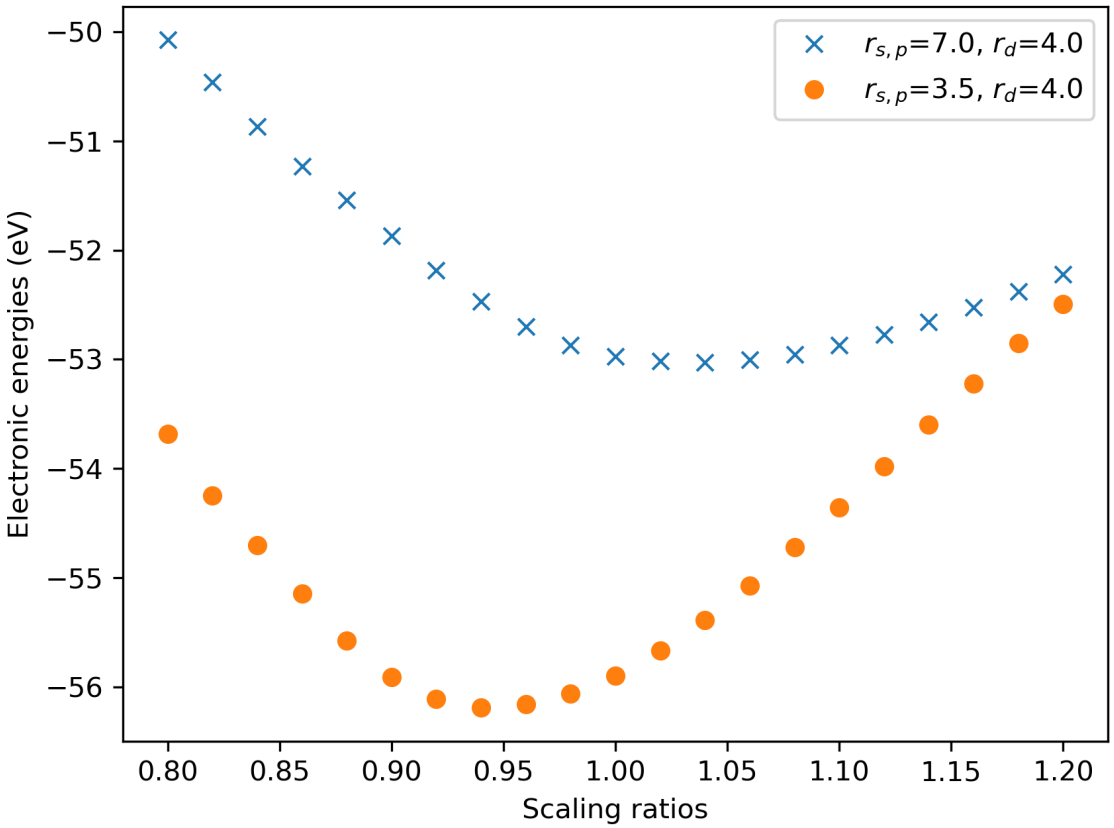


Figure 4.4: The effect of compression radii of sulfur on electronic energies was investigated for various scaled volumes of the cubic sulfur system. The volume scaling ratio refers to the scaling parameters used to multiply the lattice parameters and atomic positions of the optimized geometry from DFT.

4.3 Repulsive Parametrization and Geometry Optimization

The DFTB parametrization consists of two parts: electronic structure parametrization and repulsive potential parametrization. The latter is crucial for geometry optimization or molecular dynamics since these calculations need total energies and forces. In this section, our focus is on generating repulsive parameters for atomic pairs containing lithium and sulfur. To achieve this, we use scaling parameters that range from 0.8 to 1.2 to scale the lattice parameters and atomic positions on seven different systems, including cubic lithium, cubic sulfur, Li_3P, Li_2S, LiCl, $Li_6(PS_4)SCl$, and $Li_5(PS_4)Cl_2$. Each material contains a total of 21 geometries. The repulsive fitting is performed by scanning the unit cells of these systems with those scaling ratios. We utilized CCS [45, 46] to fit the repulsive potentials. Subsequently, we applied the obtained Slater-Koster tables based on the electronic parametrization and the repulsive parametrization by carrying out geometry optimizations.

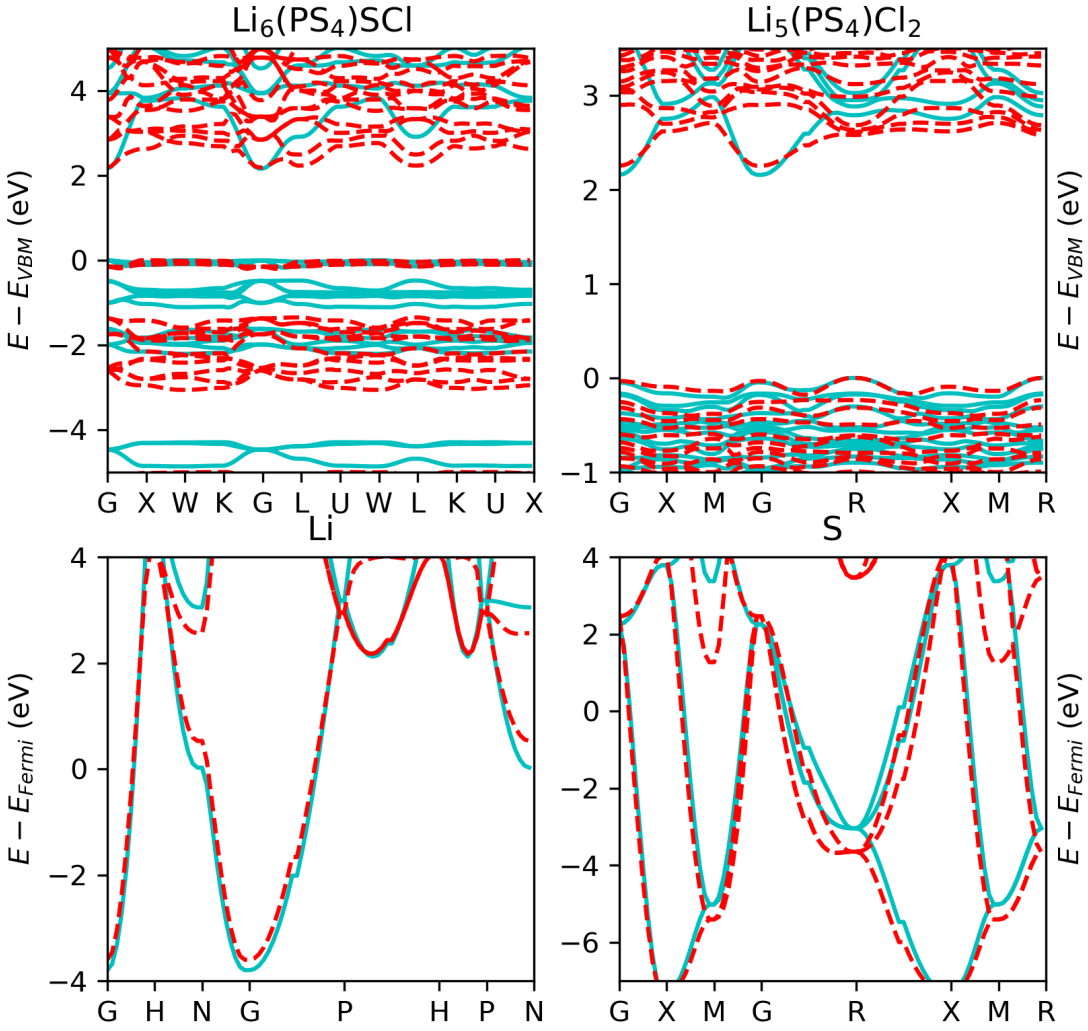


Figure 4.5: Band structures from DFT (solid line) and DFTB (dash line) calculations of $Li_6(PS_4)SCl$, $Li_5(PS_4)Cl_2$, lithium and cubic sulfur systems. The parameters used for the DFTB calculations have been listed in Table 2.

The lattice parameters presented in Table 3 were obtained using DFT and DFTB calculations. The original geometries were taken from previous studies and the materials project database. We optimized the geometries using the settings described in the Methods section. To perform SCC-DFTB geometry optimization calculations, we first obtained optimized geometries using DFT calculations. Then we applied DFTB calculations on these optimized geometries from DFT to get the final SCC-DFTB optimized geometries. Our results show that the electronic and repulsive parametrization we used for the SCC-DFTB calculations effectively and reasonably reproduces the optimized lattice parameters from DFT calculations. Due to the symmetries inherent in these materials, the atomic

Table 3: Optimized lattice parameters (Å) from DFTB and DFT

	DFT			DFTB		
	a	b	c	a	b	c
Li	4.33	4.33	4.33	4.14	4.14	4.14
S	2.58	2.58	2.58	2.55	2.55	2.55
Li_3P	4.22	4.22	7.55	4.30	4.30	7.54
Li_2S	5.67	5.67	5.67	5.46	5.46	5.46
LiCl	5.08	5.08	5.08	5.01	5.01	5.01
$Li_5(PS_4)Cl_2$	9.90	9.90	9.90	9.07	9.20	9.14
$Li_6(PS_4)SCl$	10.28	10.28	10.28	9.90	9.90	9.90

positions obtained from the DFTB calculations exhibit a good agreement with the DFT results.

4.4 Conclusions

In this chapter, we presented our approach for parametrizing both the electronic and the repulsive parameters of LIBs. Initially, we used confinement parameters from `3ob-3-1` for phosphorus, sulfur, and chlorine and only optimized the confinement parameters for lithium. However, the swapped valence bands and small band gap in the $Li_6(PS_4)SCl$ band structure indicated that optimizing only the lithium confinement parameters was insufficient for the electronic parametrization. Therefore, through PDOS and DOS analysis, we included confinement parameters for lithium and sulfur in the electronic parametrization. We also adjusted the on-site energies of sulfur to improve the band structure calculations systematically. However, the optimized compression radii for sulfur and lithium, which generated the minimum MAEs of band structures, made it challenging to fit repulsive potentials. We therefore determined compromised values for the on-site energies and compression radii that successfully resolved the swapped energy states in $Li_6(PS_4)SCl$ and yielded the band structures of cubic sulfur, cubic lithium, $Li_5(PS_4)Cl_2$, and $Li_6(PS_4)SCl$ with reasonable accuracy, and also allowed to fit repulsive potentials.

5 Tight Binding Machine Learning Toolkit Implementation

As previously introduced in the first chapter, data-driven research represents the fourth research paradigm. Machine learning has emerged as a popular tool for data-driven research, enabling the development of accurate models with minimal computational cost. Machine learning frameworks [59, 179] have significantly boosted the development of machine learning applications in various fields. These frameworks are packaged libraries that incorporate fundamental machine learning algorithms, including various neural network architectures, and facilitate the training of complex models by offering pre-constructed modules, simplifying the process for users. Additionally, these frameworks provide functions for data preprocessing, analysis, visualization, and other tasks.

Machine learning-based data-driven research can be implemented as a pure machine learning model or incorporate physical models developed from the third paradigm. Incorporating physically motivated models enhances the transferability of data-driven models [53]. To incorporate machine learning with tight-binding-based methods, we have developed an open source framework called TBMaLT (tight binding machine learning toolkit). TBMaLT facilitates machine learning techniques to improve the accuracy of tight-binding calculations, making it a valuable tool for materials science research and development. I have to emphasize that TBMaLT is a collaborative project, and the author's contributions are detailed in our previous work [180]. In this chapter, I will highlight the aspects I have contributed. Additionally, I will provide an overview of the entire project, discussing its motivation, general design, key features, and fundamental workflow.

This chapter introduces the implementation and performance of the TBMaLT. We begin by discussing the general structure and design principles of the TBMaLT, which elucidate the rationale behind reimplementing the DFTB method, the selection of PyTorch as the machine learning framework, the pivotal inclusion of batch operability designed for training, and the foundation of the base `Calculator` as an essential element of the training workflow. Subsequently, we offer a concise overview of the distinctive features inherent to TBMaLT. In sequence, we delve into the diverse methods employed to construct diatomic integrals and assess the interpolation techniques. Moving forward, we detail the implementation of the DFTB method and compare single and batch DFTB calculations. We then elucidate the electronic properties that form the focal point of our training endeavors. Finally, we provide a succinct introduction to the implementations intertwined with the DFTB-ML implementation.

5.1 Structure and Design

The overarching goal of TBMaLT is to enable machine learning-based tight-binding calculations and parametrization. This necessitates calculating gradients that link output properties to input variables. Such computations can be effectively achieved by har-

nessing the automatic gradient engine within cutting-edge machine learning frameworks. Consequently, reimplementing tight binding methods is essential within machine learning frameworks. PyTorch [59] and TensorFlow [179] are two well-known machine learning frameworks that were released in 2016 and 2015, respectively. These frameworks are primarily developed using Python and C++. While there are similarities between these packages, such as model building and training, visualizations, widespread applications in academia and industry, and suitability for training on graphics processing units (GPUs), they also differ. In TensorFlow, the computational graph, which represents the flow of computations and mathematical expressions, is defined before the program execution. This is known as a static computational graph. On the other hand, PyTorch uses a dynamic computational graph and follows a define-by-run execution approach. The dynamic computational graph makes PyTorch more Pythonic than TensorFlow and easier to use. For example, PyTorch allows for more natural control flow statements, such as loops, to be used in the model, while in TensorFlow, extra functions must be used to realize loops. The scikit-learn framework is another popular machine learning tool that focuses on traditional machine learning algorithms, such as support vector machine and random forest, rather than solely concentrating on neural network-based algorithms. Scikit-learn also integrates a powerful toolkit for data analysis and processing, making it useful in applied machine learning. Recent developments in PyTorch have made it possible to perform many linear algebra operations, making it a viable option for scientific programming. In addition to PyTorch's dynamic computational graph, its stability in transitioning between different releases makes the framework more robust and reliable. Python is a popular choice for machine learning due to the availability of numerous libraries and frameworks. Therefore, TBMaLT is implemented using the PyTorch and Python programming languages.

The concepts of batch and epoch are important in machine learning. The batch size determines the number of samples processed in each iteration for gradient updates during training, while an epoch signifies a complete traversal through the training data set. To illustrate, if we divide a data set containing 2000 samples into 4 batches of size 500, then it necessitates 4 iterations to complete a single epoch. For the application of machine learning to DFTB, the functionality of TBMaLT must encompass batch operability, enabling the execution of DFTB calculations for multiple systems simultaneously, departing from the traditional single-system approach. Batch operability within TBMaLT entails a technique known as padding, which involves expanding a set of n rank-k arrays to a uniform size and concatenating them into a single rank $(k + 1)$ array. Using batch operability leverages vectorized operations [181], effectively sidestepping the performance bottlenecks associated with Python loops. In the next section, we will conduct a performance comparison between batch DFTB calculations and conventional single DFTB calculations.

In TBMaLT, a base `Calculator` is constructed by assembling the necessary `Feed` objects, as depicted in Figure 5.1. These `Feed` objects serve as input suppliers for desired calculations, as exemplified in the SCC-DFTB calculation shown in Figure 5.1. When creating the `Calculator` for an SCC-DFTB calculation, specific feeds are required to extract parameters from Slater-Koster files. These `Feed` objects initialize by parsing Slater-Koster files and storing pertinent data based on element species. The data in-

cludes diatomic Hamiltonian integrals, overlap matrices, etc. Subsequently, using input targets like the CH_4 molecule in Figure 5.1, the initialized `Feed` objects facilitate the provision of necessary inputs to an SCC-DFTB calculation. The `Calculator` computes various properties upon request for a conventional SCC-DFTB calculation without machine learning integration. In this case, the reference and the loss function are not part of the conventional SCC-DFTB calculation workflow. In an SCC-DFTB calculation merged with machine learning to optimize requested properties, the loss function is constructed using `Calculator`-derived properties and the reference data. The PyTorch autograd engine comes into play, leveraging resulting gradients to update properties within some or all `Feed` objects.

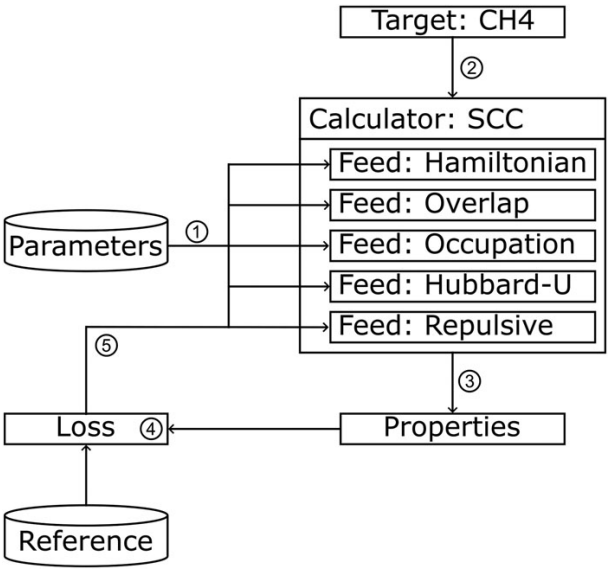


Figure 5.1: Illustration outlining the prediction and update process within an SCC-DFTB style `Calculator` instance. This process involves leveraging `Feed` objects and targeting the CH_4 molecule to compute desired properties.

5.2 Implementation and Performance

In this section, we will provide a comprehensive overview of the existing features of TBMaLT. Subsequently, we will dissect these features and, guided by the workflow depicted in Figure 5.1, systematically introduce each component individually. As illustrated in Figure 5.1, the DFTB-ML framework within TBMaLT can be compartmentalized into two principal segments: the forward part, accountable for executing DFTB calculations and generating the requisite target electronic properties, and the backward part, which leverages the PyTorch autograd engine to iteratively update gradients and optimize these electronic properties. Initiating the forward DFTB calculations, the foremost stride involves constructing Hamiltonian matrices pivotal for subsequent DFTB computations. Hence, our initial focus will be on presenting the methodologies employed for DFTB Hamiltonian matrix construction, notably emphasizing the indispensable role of interpolation techniques since the accuracy of the interpolation implementation determine the

final accuracy of DFTB calculations. Subsequently, we will delve into the comprehensive implementation of DFTB and introduce the targeted electronic properties. Lastly, we will provide a succinct synthesis of the existing state of the DFTB-ML framework's implementation.

5.2.1 Summary of the TBMaLT features

TBMaLT is a software package that includes various interpolation methods, mixers, machine learning toolkits, and modules for DFTB and DFTB with machine learning (DFTB-ML). The package provides a range of functionality, such as generating DFTB parametrization for specific chemical systems, performing calculations with non-SCC DFTB and SCC-DFTB, and training and using DFTB-ML models for predicting electronic properties of materials. Here, we provide a summary to highlight the current functionality of TBMaLT.

- non-SCC and SCC DFTB calculations for molecules and solids
- support both GPUs and central processing units (CPUs)
- support high throughput DFTB calculations
- support Hamiltonian and overlap integrals optimization for DFTB with various machine learning approaches
- support real and complex numbers in forward and backward calculations
- well tested and documented

5.2.2 DFTB Hamiltonian

We have developed different ways to generate the two-centre integrals in Eq. (2.32), from which the Hamiltonian and the overlap matrices can be constructed in a DFTB calculation. If optimizing two-centre integrals directly, we build the two-centre integrals using the cubic spline interpolation. When optimizing the targeted physical properties, we directly optimize the spline parameters and update the two-centre integrals. Besides optimizing Hamiltonian and overlap integrals directly, another strategy is to optimize the parameters of the atomic basis function parameters (the compression radii) and calculate the diatomic integrals with well-defined basis functions. In order to efficiently calculate diatomic two-centre integrals for arbitrary basis functions, we have first pre-generated integral tables for various compression radii pairs on a grid. Then bi-cubic interpolation will be applied to generate two-centre integrals with chosen compression radii. When training the targeted physical properties, the compression radii can be updated and optimized. We can stipulate that the compression radii are the same for each element specie, constituting global training. Alternatively, we can optimize the compression radii individually for each atom, constituting local training that considers the chemical environment. Hence,

we will introduce bi-cubic and cubic spline interpolation methods and demonstrate the performance of the accuracy of the bi-cubic interpolation.

Bi-cubic interpolation

Bi-cubic interpolation is an extension of the cubic interpolation, allowing for interpolating two variables simultaneously. We obtain bi-cubic interpolated values with two variables x and y

$$p(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{i,j}x^i y^j \quad , \tag{5.1}$$

where $a_{i,j}$ are the coefficients to be determined. In TBMaLT, x and y are usually compression radii of element pairs used in basis function combinations in diatomic two-centre integral calculations. If we write bi-cubic interpolation in matrix multiplication form, we obtain

$$p(x,y) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \underbrace{\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}}_{a_{i,j}\in\mathbb{R}^{4\times4}} \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix} . \tag{5.2}$$

The problem is to determine the $a_{i,j}$ matrix elements. Eq. (5.1) yields the derivatives

$$\begin{aligned} \frac{\partial p(x,y)}{\partial x} &= p_x(x,y) = \sum_{i=1}^{3}\sum_{j=0}^{3} i a_{i,j}x^{i-1}y^j \\ \frac{\partial p(x,y)}{\partial y} &= p_y(x,y) = \sum_{i=0}^{3}\sum_{j=1}^{3} j a_{i,j}x^{i}y^{j-1} \quad . \\ \frac{\partial p(x,y)}{\partial x \partial y} &= p_{xy}(x,y) = \sum_{i=1}^{3}\sum_{j=1}^{3} ij a_{i,j}x^{i-1}y^{j-1} \end{aligned} \tag{5.3}$$

When x and y in Eq. (5.1) and Eq. (5.3) are equal to 0 or 1, the values of the p function are already known. We can use this information to construct 16 equations containing $a_{i,j}$ to determine the 16 coefficients in Eq. (5.1). Then the concise matrix P which contains $a_{i,j}$ can be written as:

$$
\begin{aligned}
P &= \begin{bmatrix} p(0,0) & p(0,1) & p_y(0,0) & p_y(0,1) \\ p(1,0) & p(1,1) & p_y(1,0) & p_y(1,1) \\ p_x(0,0) & p_x(0,1) & p_{xy}(0,0) & p_{xy}(0,1) \\ p_x(1,0) & p_x(1,1) & p_{xy}(1,0) & p_{xy}(1,1) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 \end{bmatrix}
\end{aligned} \quad . \tag{5.4}
$$

Solving the above 16 equations, we obtain all 16 coefficients $a_{i,j}$ in Eq. (5.4). We can calculate the interpolated value with given x and y inputs by applying Eq. (5.2).

Figure 5.2 illustrates the differences in Mulliken charges of all the atoms in the ANI-1 data set [108] obtained from standard DFTB calculations based on two-centre integrals using bi-cubic interpolation and standard DFTB calculations. Standard DFTB calculations involve diatomic Hamiltonian integrals directly from traditional Slater-Koster tables. For DFTB calculations with a bi-cubic interpolation, we have first pre-generated integral tables for various compression radii pairs on a grid. The compression radii grid points used are 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 6.0, 8.0, and 10.0 Bohr. The integrals were then calculated using bi-cubic interpolation between the pre-calculated integral tables according to the compression radii of the atomic basis functions. The compression radii used for the bi-cubic interpolation and standard Slater-Koster tables of H, C, N, and O are 3.0, 2.7, 2.2, and 2.3 Bohr, respectively. The minor errors in the results indicate the accuracy of the implemented bi-cubic interpolation.

Cubic spline interpolation

In TBMaLT, we have also implemented cubic spline interpolation, which can be used to generate diatomic integrals directly. In the DFTB-ML workflow, the cubic spline parameters, as we will discuss later, can be updated directly to optimize the targeted electronic properties. A cubic spline function can usually be defined as

$$
f_n(x) = a_n + b_n(x - x_n) + c_n(x - x_n)^2 + d_n(x - x_n)^3 \tag{5.5}
$$

where a_n, b_n, c_n, and d_n are the parameters of the cubic spline interpolation method to be determined. When optimizing targeted physical properties in DFTB-ML workflow, these parameters can be updated and optimized. In TBMaLT, the x_n values represent the distance grid points in Slater-Koster files. The spline function with N gird points satisfies the constraints:

$$
\begin{aligned}
f_n(x_n) &= f_{n+1}(x_n) \quad n \in \{1, 2, 3, \ldots, N-1\}, \\
f'_n(x_n) &= f'_{n+1}(x_n) \quad n \in \{1, 2, 3, \ldots, N-1\}, \\
f''_n(x_n) &= f''_{n+1}(x_n) \quad n \in \{1, 2, 3, \ldots, N-1\}, \\
f''(x_0) &= f''(x_N) = 0,
\end{aligned} \tag{5.6}
$$

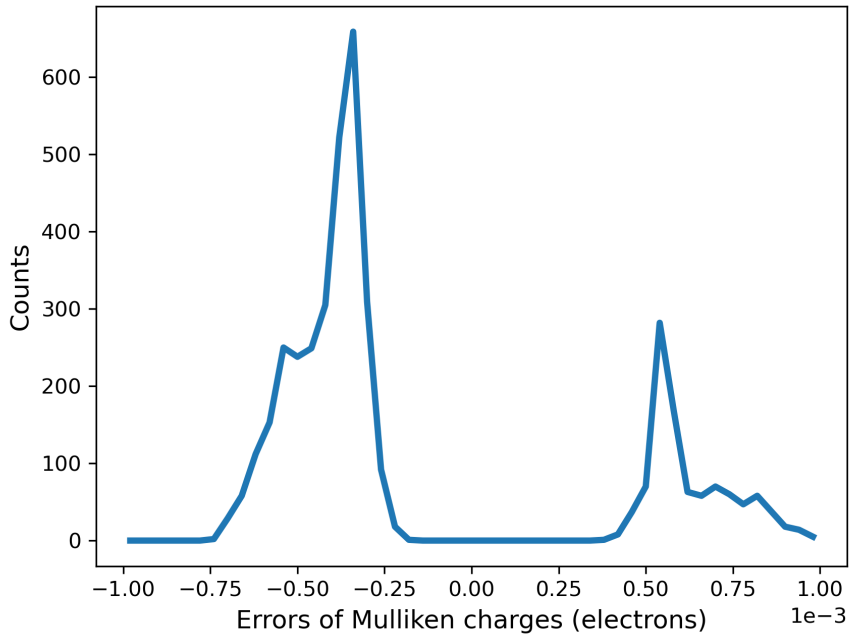


Figure 5.2: Errors in Mulliken charges between standard DFTB calculations and DFTB calculations with a bi-cubic interpolation. The results are obtained from SCC-DFTB calculations based on 1000 molecules from the ANI-1 data set [108] with one heavy atom.

where $f''(x_0) = f''(x_N) = 0$ represents so-called natural boundary condition. When defining $b_i = x_{i+1} - x_i$, we obtain

$$\begin{aligned} a_{i+1} &= a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 \\ b_{i+1} &= b_i + 2c_i h_i + 3d_i h_i^2 \\ c_{i+1} &= c_i + 3d_i h_i, \end{aligned} \tag{5.7}$$

and then we can construct b_i and d_i with c_i

$$\begin{aligned} b_i &= \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(2c_i + c_{i+1}) \\ d_i &= \frac{c_{i+1} - c_i}{3h_i}. \end{aligned} \tag{5.8}$$

When substituting b_i in equation $b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2$, we obtain

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}) \tag{5.9}$$

In Eq. (5.9), a_i and h_i are already known. We can get c_i, b_i and d_i by solving this equation. Then, we can calculate the interpolated values by applying Eq. (5.5) with input x.

5.2.3 DFTB calculations

Once the Hamiltonian and overlap matrices had been generated, we can solve the generalized eigenvalue problem to obtain the desired eigenvalues and eigenvectors, from which physical properties can be constructed. In TBMaLT, the generalized eigenvalue problem is converted to the standard eigenvalue problem since PyTorch can only solve the standard eigenvalue problem. In TBMaLT, two methods have been implemented to solve this generalized eigenvalue problem.

Generalized eigenvalue problem

The Cholesky decomposition is one solution to turn the generalized eigenvalue problem into the standard eigenvalue problem. For a real Hermitian positive matrix $\mathbf{A}$, the Cholesky decomposition is written as

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{T}}, \tag{5.10}$$

where $\mathbf{L}$ is a lower triangular matrix with real and positive diagonal entries. The generalized eigenvalue equation of DFTB is $\mathbf{HC} = \lambda\mathbf{SC}$, where $\mathbf{H}$ is the Hamiltonian matrix and $\mathbf{S}$ is the overlap matrix. The λ is the eigenvalue, and $\mathbf{C}$ is the corresponding eigenvector. When substituting the overlap with a Cholesky decomposed form, we obtain

$$\mathbf{HC} = \lambda\mathbf{L}\mathbf{L}^{\mathrm{T}}\mathbf{C}, \tag{5.11}$$

which equals to

$$(\mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-\mathrm{T}})(\mathbf{L}^{\mathrm{T}}\mathbf{C}) = \lambda(\mathbf{L}^{\mathrm{T}}\mathbf{C}), \tag{5.12}$$

where λ is the eigenvalue of $\mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-\mathrm{T}}$. With Cholesky decomposition, the generalized eigenvalue problem has been transferred to a the eigenvalue problem PyTorch can solve. Löwdin orthogonalization is another method used for solving the generalized eigenvalue problem using PyTorch, and Löwdin orthogonalization has also been implemented in TBMaLT. For Löwdin orthogonalization, we first construct the matrix $\mathbf{S}^{-1/2}$, which then allows us to transform the generalized eigenvalue problem into the following equations

$$(\mathbf{S}^{-\frac{1}{2}}\mathbf{H}\mathbf{S}^{-\frac{1}{2}})(\mathbf{S}^{\frac{1}{2}}\mathbf{C}) = \lambda\mathbf{S}^{\frac{1}{2}}\mathbf{C}. \tag{5.13}$$

Another challenge in solving the generalized eigenvalue problem is symmetric eigendecomposition, which arises when implementing Cholesky decomposition for degenerate eigenstates. This happens for degenerate eigenstates when updating the gradients using the autograd engine in PyTorch. To address this issue, Lorentzian broadening techniques have been employed in previous studies [182, 183].

Batch operability

One of the crucial features of TBMaLT is its batch operability, which enables DFTB calculations for multiple single systems simultaneously using packed padding. The vectorized feature in batch DFTB calculations avoids the slow loops in Python as much as possible, resulting in significant efficiency improvements compared to loops of the single system.

The starting point for DFTB calculations in TBMaLT is to read the Slater-Koster tables for DFTB calculations. TBMaLT supports traditional DFTB Slater-Koster tables as input and supports Slater-Koster tables with various compression radii grid points for different element pairs. In machine learning or high-throughput DFTB calculations, we perform DFTB calculations of multiple geometries simultaneously instead of the single DFTB calculation in traditional packages. Incorporating the batch DFTB calculation effectively circumvents redundant input/output (IO) operations, which can otherwise be prevalent in conventional software packages. The batch operability also effectively eliminates the need for inefficient Python loops. However, to establish a fair comparison between batch DFTB calculations and iterations of single DFTB calculations, it is crucial to concentrate exclusively on the DFTB computations. This entails excluding the time associated with IO operations and the initialization of various `Feed` objects. As depicted in Figure 5.3, the CPU time exclusively encapsulates the interval starting from Slater-Koster transformations [26] to the convergence of SCC-DFTB calculations. Figure 5.3 shows that the batch calculation can speed up SCC-DFTB calculations by at least one order of magnitude for a large data set.

Figure 5.3 shows that the Slater-Koster transformation, the generalized eigenvalue problem and other operations should be implemented using vectorized and batch processing code to improve computational efficiency. This approach allows calculating all corresponding SKT operations simultaneously, resulting in significant time savings.

5.2.4 Electronic properties

In this section, we will introduce the electronic properties which can be calculated using TBMaLT. These electronic properties are the machine learning targets in this thesis.

Charge population analysis

Charge population analysis (CPA) [184] can be used to derive effective atomic C_6 coefficients in DFTB, which is based on the method developed by Tkatchenko and coauthors [185, 186]. The method developed by Tkatchenko et al. [185] uses effective atomic C_6 coefficients in van der Waals interactions depending on the bonding environment. The C_6 coefficients can be obtained by exploiting the linear relationship that exists between atomic polarizabilities and the Hirshfeld volume [187]. With the Hirshfeld volume V_A and polarizability α_A in the environment of atom A, Hirshfeld volume V_A^{free} and polarizabil-

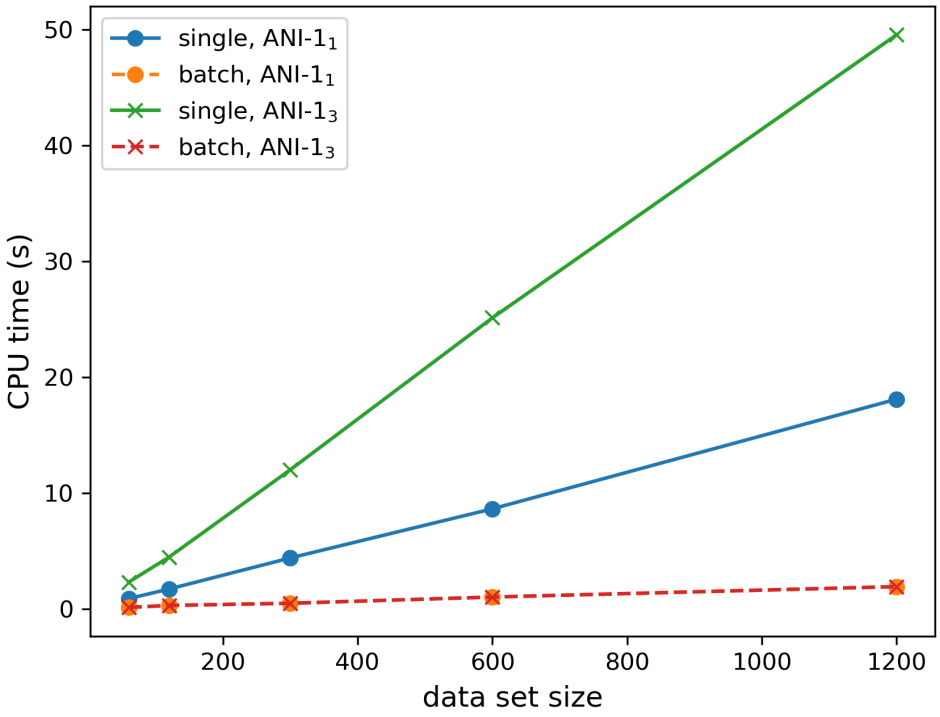


Figure 5.3: Performance of single and batch DFTB calculations using ANI-1 data set [108] with one (ANI-1_1) and three heavy atoms (ANI-1_3). All molecules were calculated sequentially in the single calculations, while in the batch calculations they were calculated together. All the times exclude the IO time and only consider the DFTB calculations starting from Slater-Koster transformations. The data set contains 60, 120, 300, 600, and 1200 molecules. The tests were run on a machine with an Apple M1 Pro processor using one thread.

ity α_A^{free} of the free atom A, and taking advantage of the direct relation [187] between polarizability and Hirshfeld volume, we have

$$\frac{C_6^{AA}}{C_6^{AA,\text{free}}} \approx (\frac{\alpha_A}{\alpha_A^{\text{free}}})^2 \approx (\frac{V_A}{V_A^{\text{free}}})^2. \tag{5.14}$$

where C_6^{AA} is the homonuclear coefficient and the superscript free means the corresponding property of a free atom. An extension of this applicability exists in DFTB, known as CPA [184]. In this electron density partitioning scheme, a similar relationship is utilized between polarizabilities and the ratio of on-site contribution to Mulliken populations and atomic charge Z_A of atom A. For an atomic basis set $|\Phi_i\rangle = \sum_\mu C_{i\mu}|\varphi_\mu\rangle$, the atom-projected trace h_A of the density matrix is defined as

$$h_A = \sum_i f_i \sum_{\mu \in A} |C_{i\mu}|^2, \tag{5.15}$$

where occupation of state i is f_i and $C_{i\mu}$ are the associated coefficients. h_A measures the

hybridization-induced charge transfer due to the interactions with other atoms, similar to the atom-in-molecule Hirshfeld volume. Therefore the approximation of the polarizability of an atom-in-molecule can be expressed as follows:

$$\frac{C_6^{AA}}{C_6^{AA,\text{free}}} \approx (\frac{\alpha_A}{\alpha_A^{\text{free}}})^2 \approx (\frac{h_A}{Z_A})^2. \tag{5.16}$$

The so-called CPA ratios are $\frac{h_A}{Z_A}$ and Hirshfeld volume ratios are $\frac{V_A}{V_A^{\text{free}}}$. Within a machine learning process, the workflow involves learning CPA ratios in DFTB using reference data on Hirshfeld volume ratios from DFT calculations.

Density of states

Density of states (DOS), as well as projected DOS (PDOS) have been implemented in TBMaLT as

$$\text{DOS}(\epsilon) = \sum_i \delta^\sigma(\epsilon - \epsilon_i) \tag{5.17}$$

$$\text{PDOS}(\epsilon, \nu) = \sum_i \sum_\mu c_{\mu i}^* c_{\nu i} S_{\mu\nu} \delta^\sigma(\epsilon - \epsilon_i), \tag{5.18}$$

where ϵ_i are the calculated eigenvalues of state i, ϵ are energy values, δ can be either the Dirac delta-function or the Gaussian function with broadening parameter σ, $c_{\mu i}$ is the coefficient of state i and orbital μ, and $S_{\mu\nu}$ is the overlap between orbitals μ and ν.

Band structures

The theory behind band structure is discussed in the section on periodic boundary conditions in chapter 2. Band structures can be computed by solving generalized eigenvalue problems using defined high-symmetry **k**-points. The SCC cycles should be set to 1 for band structure calculations, and Mulliken charges are obtained from well-converged SCC-DFTB calculations.

Periodic boundary conditions and band structure calculations have been implemented in TBMaLT. In order to test our implementation, we compared the band structures of TiO_2 obtained from TBMaLT and DFTB+ using the same parameter set. Figure 5.4 illustrates the band structure of TiO_2 using our implementation and DFTB+. The SCC-DFTB calculations were performed with a tolerance of 1×10^{-6} electrons, and a **k**-mesh of 5×5×5 was used. The maximum angular momentum used for oxygen was p, while titanium was d. The compression radii for titanium were all set to 4.3 Bohr, while for oxygen they were set to 3.5 Bohr. The results demonstrate that the band structure calculations in DFTB+ and TBMaLT yield identical results.

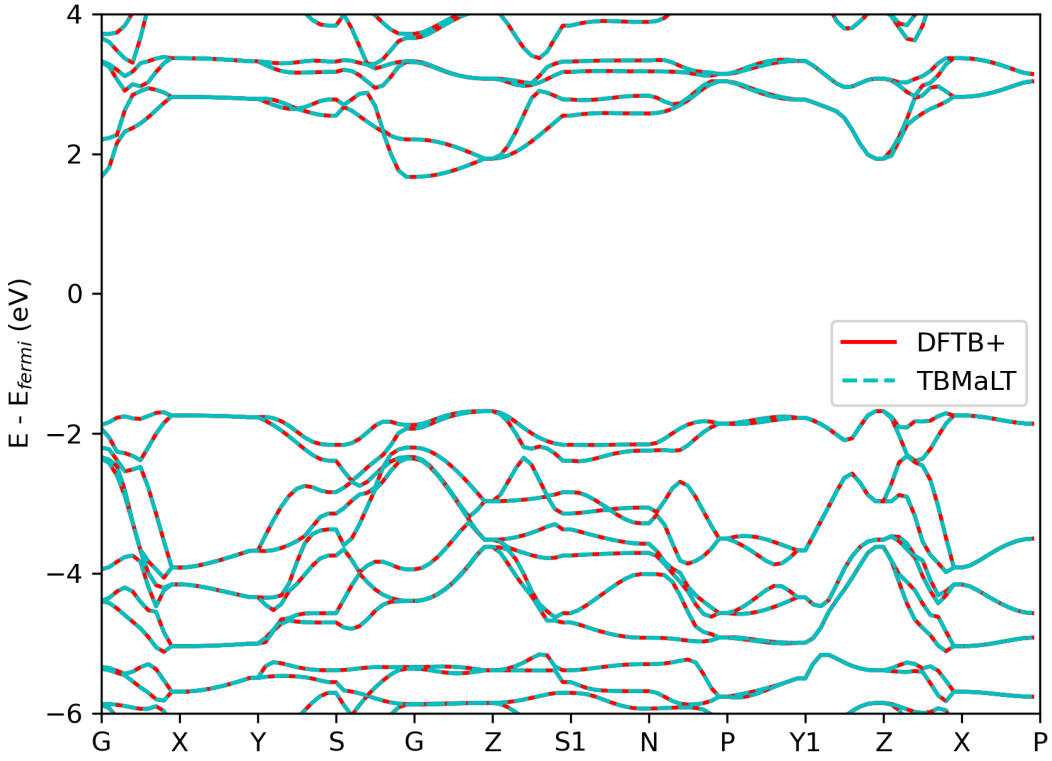


Figure 5.4: Band structure of anatase TiO_2 using TBMaLT and DFTB+.

5.2.5 DFTB-ML framework

We have outlined the workflow of a `Calculator` in Figure 5.1. In this section, we will delve into the implementation details of DFTB-ML within TBMaLT. We will begin by presenting the feature implementation in TBMaLT, and then by introducing the various approaches used for constructing Hamiltonian matrices and the workflow of DFTB-ML.

Feature engineering plays a pivotal role in the field of applied machine learning. We introduced a diverse set of machine learning features in chapter 3. In TBMaLT, we have integrated and extensively validated the usage of ACSFs for molecular and solid-state systems. This enables us to accurately depict the chemical environment of molecules and solids. To optimize the computational efficiency of machine learning calculations, we have employed Cython in the implementation.

The conventional DFTB calculations are based on the Slater-Koster tables that contain distance-dependent diatomic integrals and on-site energies. These basis parameters used for diatomic integral generations are identical for all atoms of a given element. With the DFTB-ML framework, it is possible to optimize the diatomic integrals and on-site energies globally, which is similar to the conventional approach. Additionally, the DFTB-ML approach can also be employed in the local method, which adapts the atomic parameters

and on-site energies of each atom individually depending on the chemical environment, providing greater flexibility in DFTB calculations. In the following two chapters, these approaches and the DFTB-ML frameworks will be introduced in detail.

5.3 Conclusions

In this chapter, we introduced the general design of the TBMaLT, methods used in the implementation of the TBMaLT, including various interpolations, electronic properties, and batch-designed DFTB calculations. Our interpolation methods showed reasonable accuracy, ensuring the accuracy of DFTB calculations. Moreover, the efficiency of training large batch systems was enhanced by utilizing batch-designed DFTB calculations, which outperformed the use of single calculations.

6 Machine Learning of Molecular Electronic Properties

Based on the implementation of TBMaLT, we introduce a DFTB-ML model for optimizing DFTB molecular electronic properties (such as dipole moments). The application is based on a data set of molecules, enabling the optimization of DFTB basis parameters or diatomic Hamiltonian and overlap integrals directly. Our results demonstrate that the DFTB-ML model improves both single DFTB electronic property calculations and calculations of multiple electronic properties, and exhibits good transferability. Additionally, we demonstrate the importance of incorporating basis functions to ensure that the trained and predicted Hamiltonian and overlap integrals remain within physically reasonable ranges.

6.1 Data Sets and Methods

Data collection

This work performed all training and testing using molecular geometries from the ANAKIN-ME data set, also known as ANI-1 [108]. ANI-1 consists of four element species: hydrogen, carbon, nitrogen, and oxygen, with the latter three referred to as heavy atoms. Separate models were trained using molecules of different sizes to investigate the impact of molecule size. In our notation, the subscript after the data set name ANI-1 indicates the number of heavy atoms in the molecules. For example, ANI-1_1 represents a data set comprising methane, ammonia, and water molecules, each containing one heavy atom. On the other hand, ANI-1_3 represents a data set where each molecule contains three heavy atoms.

DFT calculations

The geometries used in this study were taken from the ANI-1 data set, and all-electron DFT calculations were performed using the FHI-aims code [175]. The basis set employed was at *tier* 2, or tight level, and the PBE functional [89] was used. The electronic properties calculated from DFT included dipole moments, Mulliken charges, and Hirshfeld partitioning, allowing for the calculation of effective atomic polarizabilities. [185].

DFTB calculations

All DFTB calculations in this work were performed based on SCC-DFTB calculations using TBMaLT. The difference between standard DFTB calculations and DFTB-ML models lies in the way of generating the two-centre Hamiltonian and overlap integrals. The electronic properties investigated in this section included dipole moments, Mulliken

charges, and charge population analysis (CPA) ratios [184], which is directly related to the atomic polarizabilities.

Representations of atomic geometries

In this chapter, we utilized the ACSFs [128] as machine learning features, specifically employing the cutoff function G_1, the radial symmetry function G_2, and the angular function G_4 introduced in Eq. (3.5). The ACSFs implementation in this study utilized a cutoff parameter R_c of 6.0 Angstrom for G_1, as well as η and R_s values of 1.0 and 1.0 Angstrom for G_2, and η, ζ, and λ values of 0.02, 1.0, and -1.0 for G_4.

Machine learning methods

This work employed machine learning algorithms to predict compression radii and on-site energies using the scikit-learn package. The algorithms used were neural networks (NNs) and the random forest (RF). The neural networks used for training and testing were multilayer perceptrons with five layers and the ReLU activation function. The random forest regression utilized 100 estimators.

During the training process, the learning rate for two-centre integrals was set to 0.02 and 0.001 when training the basis function parameters and the diatomic integrals directly without basis functions, respectively. The Adam optimizer [158] was used to optimize the parameters through backward propagated gradients. The learning rates for on-site energies were set to 5×10^{-4} for basis function training and 2×10^{-6} for the model without basis functions. Mean squared errors (MSEs) were chosen as the default loss function if not otherwise specified.

6.2 DFTB-ML Workflow

The workflow for molecular electronic training consists of two parts: forward calculations and backward gradient calculations. The entire workflow of the DFTB-ML model for electronic properties training in molecule systems is illustrated in Figure 6.1. To perform the forward calculations, the first step is to construct the Hamiltonian and overlap matrices for SCC-DFTB calculations, and various methods were employed in this work to construct these matrices, as will be described below. SCC-DFTB calculations are then carried out to obtain the electronic properties. Loss functions are then constructed using the reference electronic properties from FHI-aims calculations and electronic properties from SCC-DFTB calculations. The parameters are updated through backward gradient calculations using an autograd engine in PyTorch.

Three approaches to construct Hamiltonian and overlap

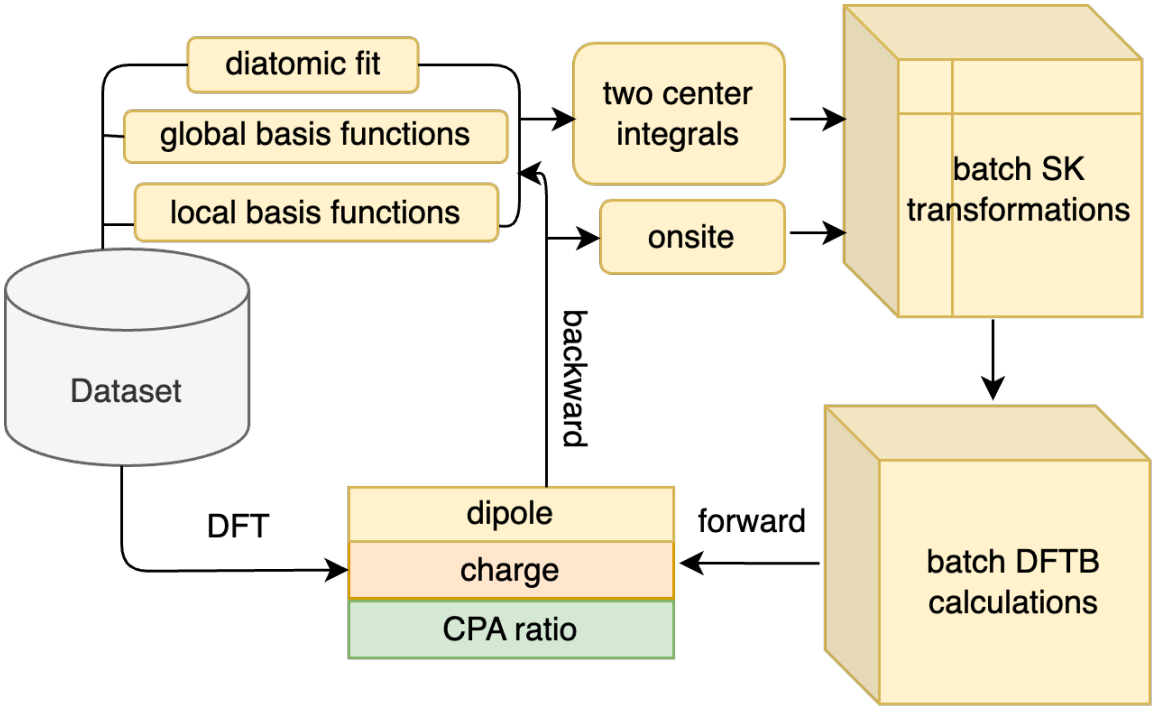


Figure 6.1: Illustration of the DFTB-ML workflow.

In this chapter, we will provide a detailed explanation of the different approaches we used to construct Hamiltonian integrals. These approaches are global and local methods with well-defined basis functions and the direct diatomic fit using cubic spline interpolation. In SCC-DFTB calculations, distance dependent two-centre integrals and on-site energies are needed to construct Hamilton and overlap matrices. The confining potential used in calculating the two-centre Hamiltonian and overlap integrals is shown in Eq. (2.54).

The first approach is to globally tune the compression radii r_0 and on-site energies for each element species. The first global approach involves assigning the same parameters to atoms of the same element species in different chemical environments. This global approach is similar to the traditional DFTB parametrization methods [36, 40, 44, 105], and the only difference is that we use a machine learning based framework to optimize the parameters of the confinement term and the on-site energies.

The second approach involves tuning the compression radius and on-site energies depending on the local chemical environment. This local approach also applies confining potentials as in Eq. (2.54) to generate distance dependent Hamiltonian and overlap integrals, but unlike the first global method, the compression radii in the basis functions are locally determined for each atom separately based on the chemical environment. Additionally, the on-site energies are also local in this approach. By adjusting the compression radii and on-site energies, the Hamiltonian and overlap matrices can be built for the SCC-DFTB calculations. The initial values of compression radii in the global and local approaches were chosen as in the `mio-1-1` parameter set.

The global approach and the local approach involve tuning compression radii to build the two-centre integrals based on basis functions. Conversely, the third approach involves skipping the basis functions and directly generating the two-centre integrals using cubic spline interpolation. The Hamiltonian and overlap two-centre integrals and on-site energies are treated globally for each element specie as in this approach. A similar approach was introduced in a previous work [66]. In this third diatomic approach, the spline parameters and on-site energies will be updated. To have a good initial guess, these spline parameters are initialized using the `mio-1-1` parameter set [27]. The way to generate the spline parameters has been introduced in chapter 5.

To implement the global and local approaches, diatomic integrals have been pre-calculated for all element specie pairs with defined compression radii grid points and then interpolated using the given compression radii. Hence, an efficient and accurate interpolation is crucial. For this purpose, a bi-cubic interpolation was employed to obtain integrals from the pre-calculated diatomic integrals, which was introduced and tested in chapter 5 and was found to be sufficiently accurate for the global and local approaches. ACSFs and machine learning algorithms have been applied to train and predict the chemical environment dependent compression radii and on-site energies in the local approach. The optimized parameters in the global and diatomic approaches can be directly applied to new geometries.

Forward DFTB calculations

The distance-dependent diatomic integrals and on-site energies obtained from the three approaches are used to generate Hamiltonian and overlap matrices through the Slater-Koster transformations (SKT) [26]. Solving the generalized eigenvalue problems in Eq. (2.34) yields eigenvalues and eigenvectors, which can be used to compute the desired electronic properties. All SKT and SCC-DFTB calculations are performed using batch calculations to ensure reasonable efficiency. The batch operability has been described in chapter 5 and has been proven to be reasonably efficient.

Backward gradients updates and machine learning methods

Electronic properties can be obtained from SCC-DFTB calculations using one of three approaches to construct Hamiltonian and overlap matrices. The reference electronic properties have been pre-calculated using FHI-aims. The loss functions for optimizing spline parameters, compression radii, and on-site energies have been defined as:

$$\text{Loss} = \frac{1}{N}\sum_{j=1}^{m}\sum_{i=1}^{N}\omega_j(P_{ij}^{\text{DFT}} - P_{ij}^{\text{DFTB}})^2, \tag{6.1}$$

where N describes the number of systems in the training data set, and m is the number of physical properties taken into account. P^{DFT} and P^{DFTB} are the target physical property values from the reference DFT and the DFTB calculations, respectively, and ω_j is the

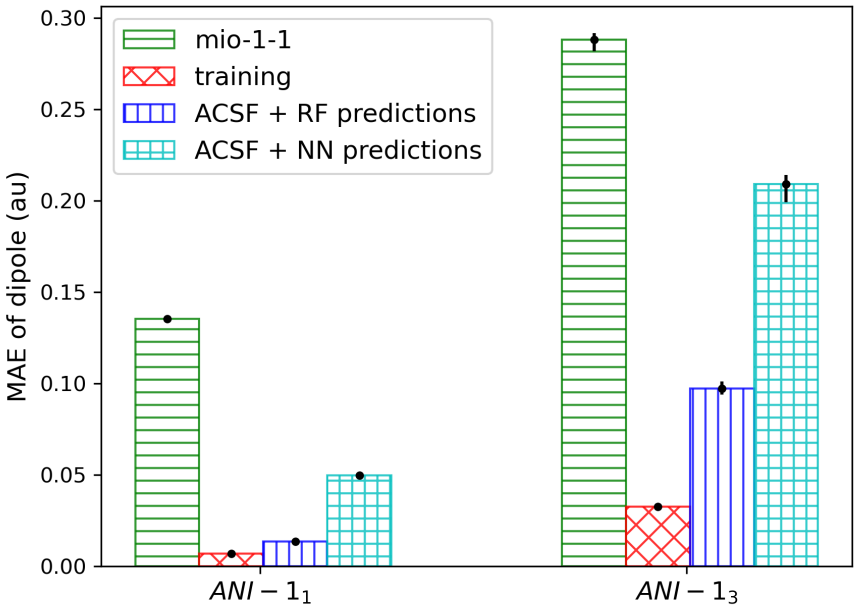


Figure 6.2: MAEs of dipole moments from traditional SCC-DFTB calculations based on the `mio-1-1` parameter set using testing set, ML training results using training set, and ML testing results with different ML algorithms using testing set. MAEs are the average MAEs of three dependent training or testing runs. The unit of dipole moments is the atomic unit. ANI-1_1, ANI-1_3 are the data set[108] with one and three heavy atoms. The data set size for training is 1000 and the testing data set size is 400.

weight associated with a given physical property. The default value for ω is 1, unless otherwise specified.

For the local approach, we tested different machine learning algorithms and compared their performance for dipole moment calculations based on predicted compression radii and on-site energies. Figure 6.2 illustrates the performance of two different machine learning algorithms and traditional SCC-DFTB calculations with the `mio-1-1` parameter set. The geometries were represented using was ACSFs, which generated the machine learning input. The machine learning output consists of the optimized compression radii and the on-site energies. The neural networks achieve MAEs of 0.05 and 0.21 per molecule with one (ANI-1_1) and three (ANI-1_3) heavy atoms, respectively. On the other hand, the ensemble method random forest with 100 estimators achieves MAEs of 0.01 and 0.10 for molecules with one and three heavy atoms, respectively. These results indicate that the random forest method outperforms neural networks for all molecules with different sizes in our applications. Therefore, we have chosen the random forest as the machine learning algorithm.

6.3 Results and Discussions

Training and testing set size

The size of the data set used in machine learning significantly impacts the chemical environment patterns that the applied algorithm can learn. Generally, a larger data set provides more patterns for the algorithm to learn. Therefore, we tested different training set sizes to investigate their effect using data sets with one and three heavy atoms with all three approaches. Figure 6.3 shows the MAEs when training sets with different molecule sizes and optimization methods. When examining the effect of the molecule sizes on convergence, the ANI-1_3 data set exhibits higher fluctuations, likely resulting from the more complex chemical environment. The advantage of the local approach is that it has the smallest error bar, suggesting the robustness of this approach. From Figure 6.3, we can conclude that all three approaches with different molecule sizes reach reasonable convergence when the data set size is 1000. Therefore, for subsequent training, 1000 molecules have been used.

6.3.1 Single Electronic Property Training

The training was tested on single and multiple electronic properties, starting with single properties, using data sets with one and three heavy atoms. Figure 6.4 illustrates the training loss and predictions of dipole moments, Mulliken charges, and CPA ratios separately using molecules with one heavy atom, while Figure 6.5 illustrates the training loss and predictions of dipole moments, Mulliken charges, and CPA ratios separately using molecules with three heavy atoms. The loss functions of all three properties decrease remarkably. Considering the starting point of the Slater-Koster files being `mio-1-1`, the decreases of loss functions suggest that the DFTB-ML framework can optimize electronic properties and decrease the errors between DFT and DFTB.

Figures 6.4 and 6.5 only depict the performance based on the local approach, whereas Figure 6.6 compares the mean absolute errors (MAEs) of the predictions for dipole moments, Mulliken charges, and CPA ratios to evaluate the performance of all three approaches. The performance of Mulliken charges and dipole moments is better for the data set with one heavy atom than that of CPA ratios. However, for the data set with three heavy atoms, the errors of dipole moments increase considerably. This implies that the complexity of the chemical environment weakens the performance of dipole moment predictions. The MAEs for dipole moments, charges, and CPA ratios for the ANI-1_1 data set based on the local approach are 0.01, 0.02, and 0.07, respectively, compared to values of 0.18, 0.15, and 0.29 obtained using the `mio-1-1` parameter set. For the ANI-1_3 data set based on the local approach, the MAEs are 0.10, 0.08, and 0.10 for dipole moments, charges, and CPA ratios, respectively, compared to values of 0.29, 0.41, and 0.34 obtained using the `mio-1-1` set. The local basis function optimization method significantly improves all electronic properties, indicating its potential for predicting electronic properties in complex chemical environments. Although the diatomic model and the global basis function optimization methods also show improvements for the ANI-1_1 data set,

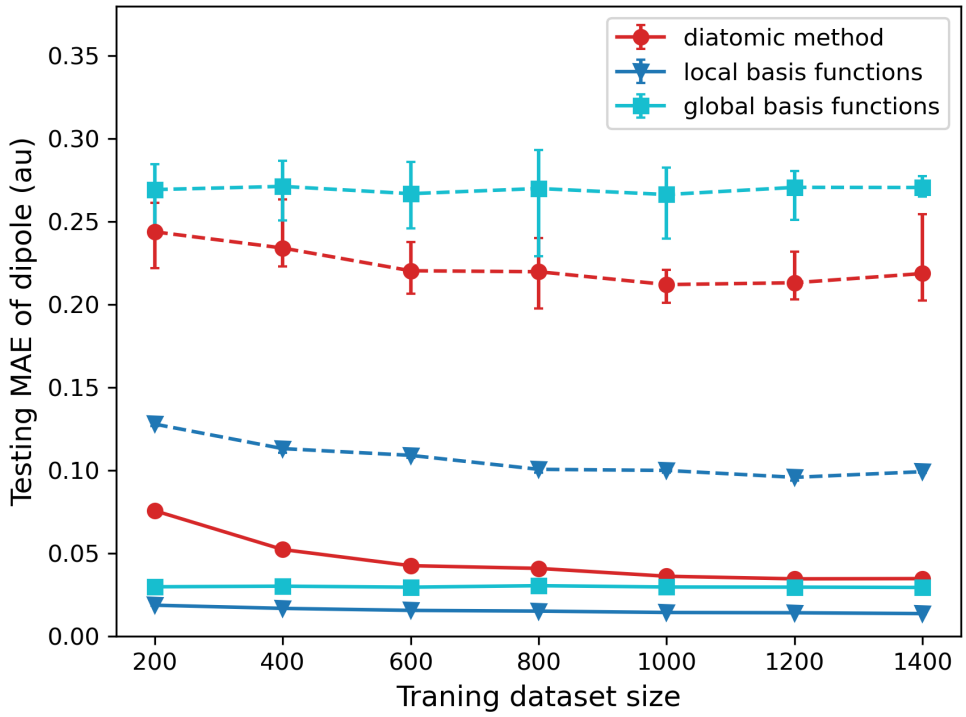


Figure 6.3: Effect of the training set size on dipole moment prediction. The dashed lines represent the ANI-1_3 data set with three heavy atoms, while the solid lines represent the ANI-1_1 set with one heavy atom. The line colors of the diatomic method, the global method and the local method are red, cyan and blue, respectively. All MAEs are given in atomic units per molecule. The MAEs are the average of seven independent runs of training and prediction for the diatomic method and the global method with ANI-1_3 data set because of the fluctuations. The other MAEs are the average over three independent runs. All testing data sets consist of 400 molecules.

these two approaches only slightly improve for the ANI-1_3 data set. The slight improvement suggests that the `mio-1-1` set was already globally optimized to a reasonable range.

6.3.2 Multiple Physical Properties Training

Machine learning on multiple targets is a challenging task. We evaluated the predictivity of our approach by training on two properties simultaneously. Figure 6.7 displays the predictions for combinations of two electronic properties, and only the local approach was selected since it gives the best performance in all cases. All predictions show improvements compared to the results from the `mio-1-1` set. In single property predictions as discussed before, Mulliken charges and dipole moments perform better than CPA ratios, especially for data sets with one heavy atom. In multiple property predictions, Mulliken charges and dipole moments also outperform CPA ratios in charge-CPA and dipole-CPA predictions.

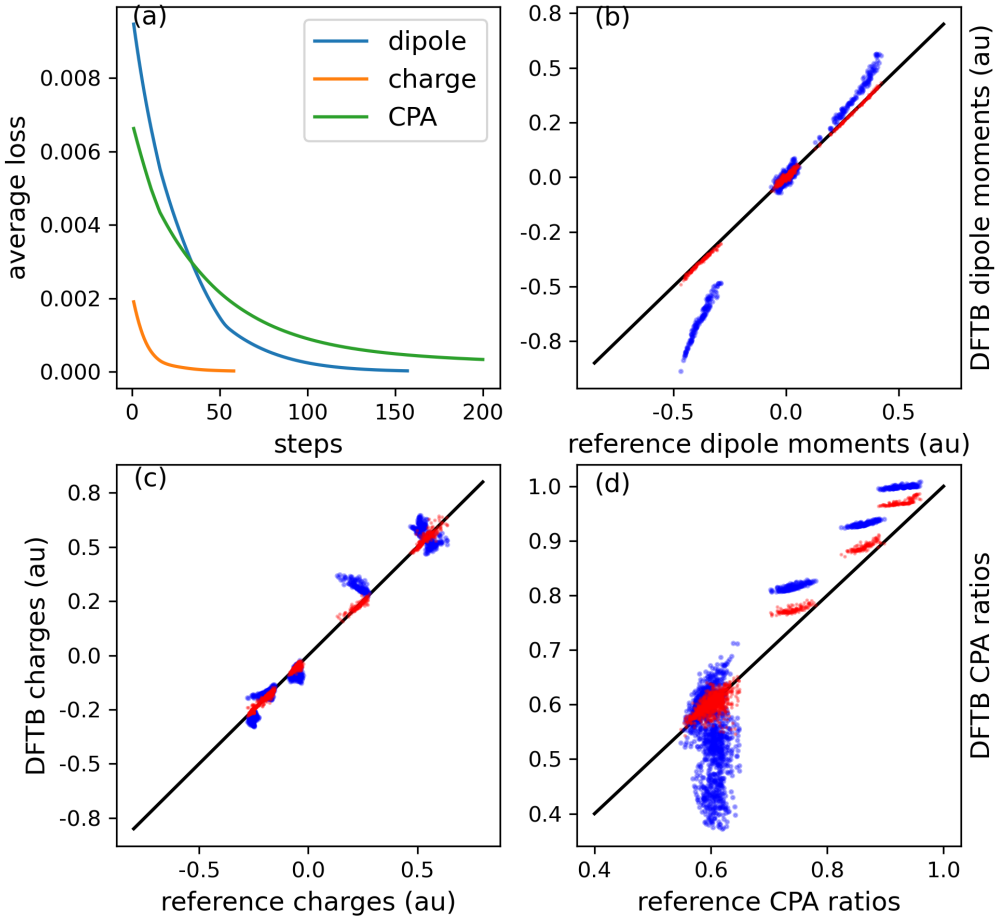


Figure 6.4: (a) Training loss functions, (b) predictions of dipole moments, (c) predictions of Mulliken charges, and (d) predictions of CPA ratios. The training and prediction are based on ANI-1_1 data set. Each of the electronic properties has been trained and predicted separately. The blue points are based on the `mio-1-1` parameter set, while the red points are based on the local approach. The training and the testing data set sizes were 1000 and 400, respectively.

This indicates that properties that perform well in single property predictions will also perform well in multiple property predictions.

In the dipole-charge and CPA-charge training, the MAEs of Mulliken charges over the ANI_1 data set are 0.11 and 0.05, respectively. The MAEs of Mulliken charges are significantly higher than that obtained during the single property training. This difference can be attributed to different optimized distributions of on-site energies and compression radii for the Mulliken charge training and dipole-charge training, as depicted in Figure 6.8, which takes Mulliken charges of oxygen atoms as an example. It is important

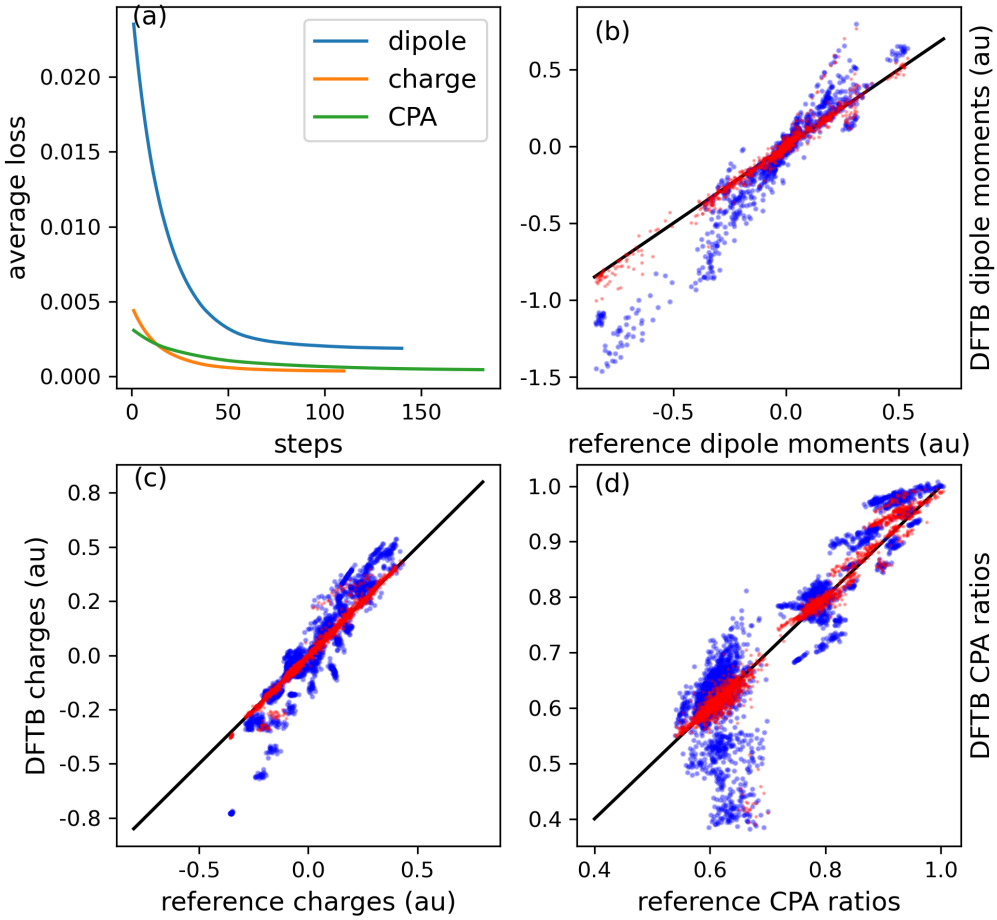


Figure 6.5: (a) Training loss functions, (b) predictions of dipole moments, (c) predictions of Mulliken charges, and (d) predictions of CPA ratios. The training and prediction are based on ANI-1_3 data set. Each of the electronic properties has been trained separately. The blue points are based on the `mio-1-1` parameter set, while the red points are based on the local approach. The training and the testing data set sizes were 1000 and 400, respectively.

to note that the data set used in Figure 6.8 remains fixed, while the data sets are selected randomly for other purposes. The fixed data set is helpful in checking the effects of on-site energy distributions or compression radii distributions on the single property and multiple properties training. Figure 6.8 illustrates that the on-site energy distributions of oxygen atoms play an important role in optimizing Mulliken charges of oxygen atoms. Notably, when incorporating on-site energies derived from the Mulliken charge training and using the compression radii from dipole-charge training, the performance of Mulliken charges demonstrates a significant enhancement compared with the Mulliken charges in dipole-charge training. The outcomes depicted in Figure 6.8 elucidate the rise in Mul-

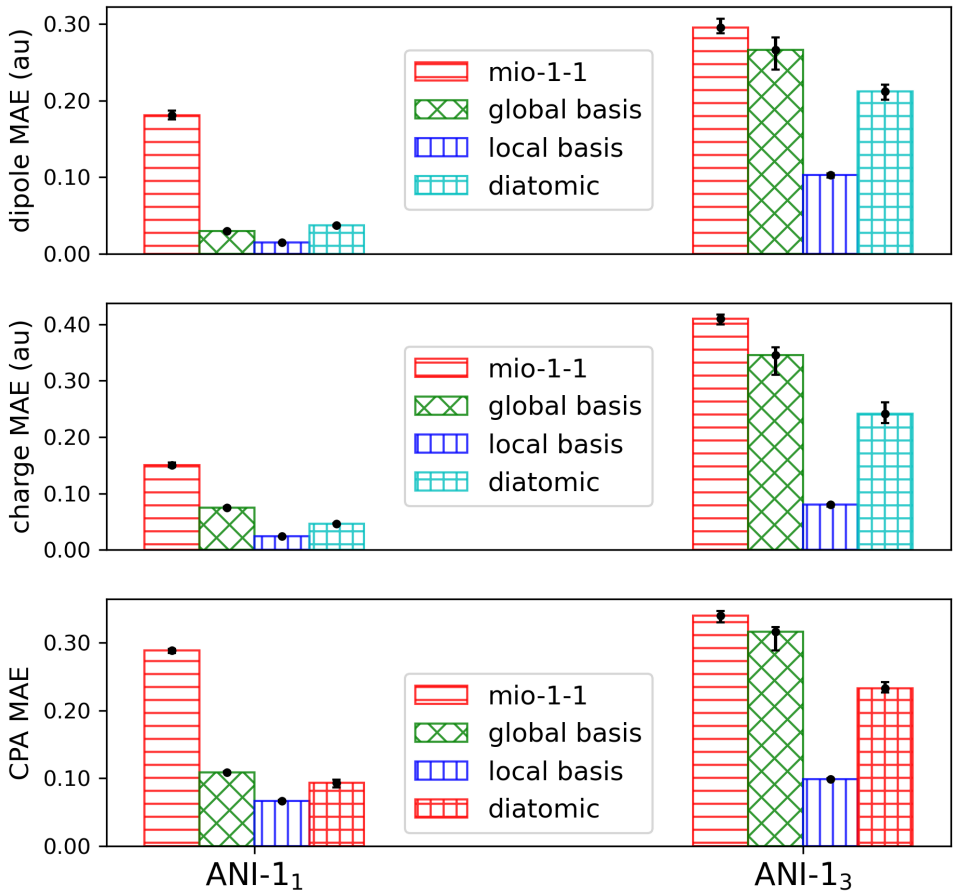


Figure 6.6: MAEs of dipole moments, Mulliken charges and CPA ratios using data sets with one and three heavy atoms. The average MAEs were obtained from three independent training and testing runs. The training and the testing data sets contained 1000 and 400 molecules, respectively. All values of MAEs are average of three training runs except global basis functions and diatomic model for the ANI-1_3 data set, which were generated from seven training runs.

liken charges during dipole-charge training, which can be attributed to the variations in the distributions of on-site energies. The different distributions of on-site energies and compression radii can also explain the results of the dipole-CPA training. These findings indicate that multiple property training is a compromise optimization. We can introduce weight parameters in multiple property training to achieve a balance in performance. We tested weight parameters ω between 0.5 and 3.0 using Eq. (6.1), and the results are presented in Table 4.

The MAEs presented in Table 4 were calculated in atomic units and based on the

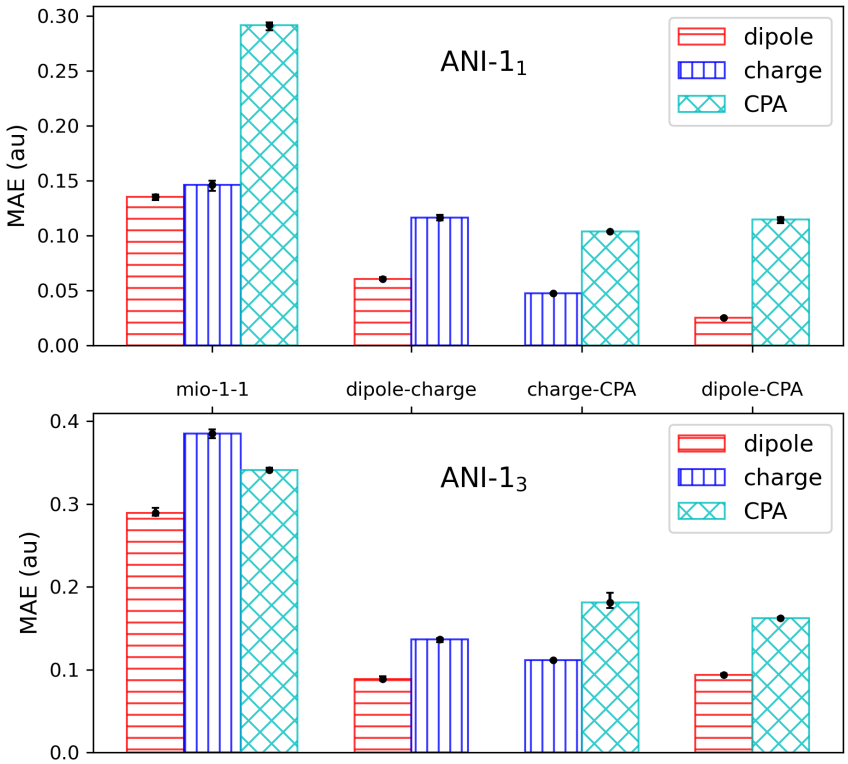


Figure 6.7: Average MAEs of electronic properties from standard SCC-DFTB calculations using the `mio-1-1` parametrization and predictions based on the local approach on two electronic properties (dipole-charge, dipole-CPA, and charge-CPA). The MAEs were obtained from three training and testing runs. The training and predictions data set sizes were 1000 and 400, respectively. All weights of the electronic properties in the loss functions were chosen to be 1.

Table 4: Effect of weights in loss functions on the predictions of multiple properties

weights in CPA loss functions	0.5	1.0	2.0	3.0
weights in dipole loss functions	1.0	1.0	1.0	1.0
MAEs of dipole moments	0.02	0.03	0.03	0.04
MAEs of CPA ratios	0.11	0.10	0.09	0.09

ANI-1$_1$ data set. The results demonstrate that tuning the weight parameters can be beneficial in achieving global optimization in multiple properties training. By adjusting the weights, it is possible to balance the relative importance of each physical property in the loss function and improve the predictivity of the model for all properties. This finding is particularly relevant in the context of machine learning, where multiple task learning

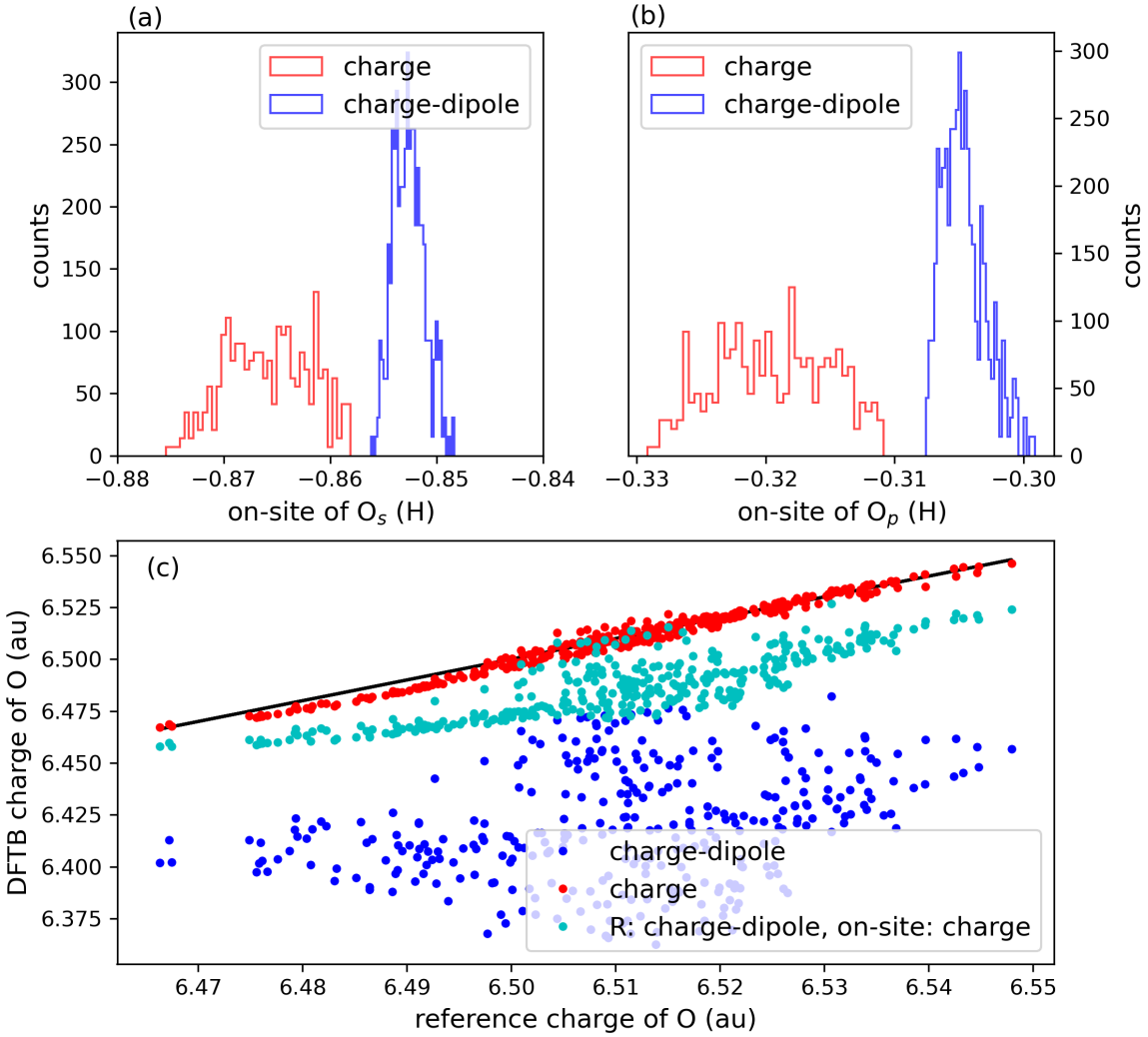


Figure 6.8: On-site energies distributions of (a) oxygen s orbital, (b) oxygen p orbital from single Mulliken charge training and charge-dipole training. (c) Mulliken charges of oxygen atoms were obtained from three scenarios: directly from training Mulliken charges, from training dipole-charge, and employing on-site energies from training Mulliken charges combined with compression radii from training dipole-charge. The training data set contains 1000 samples, and the data set is the ANI-1_1 data set.

is a challenging task due to the complexity of the data and the need to optimize multiple properties simultaneously. To conclude, these results highlight the importance of careful parameter tuning in achieving accurate and reliable predictions in multiple task machine learning.

6.4 Transferability

The transferability of a machine learning model is a crucial aspect that determines its usability in real-world systems. This chapter focuses on two types of transferability: scaling transferability and transferability between physical properties. Scaling transferability measures whether a learning model can be applied to more complex chemical environments, while the latter evaluates whether the training model is limited to predicting specific physical properties used during training.

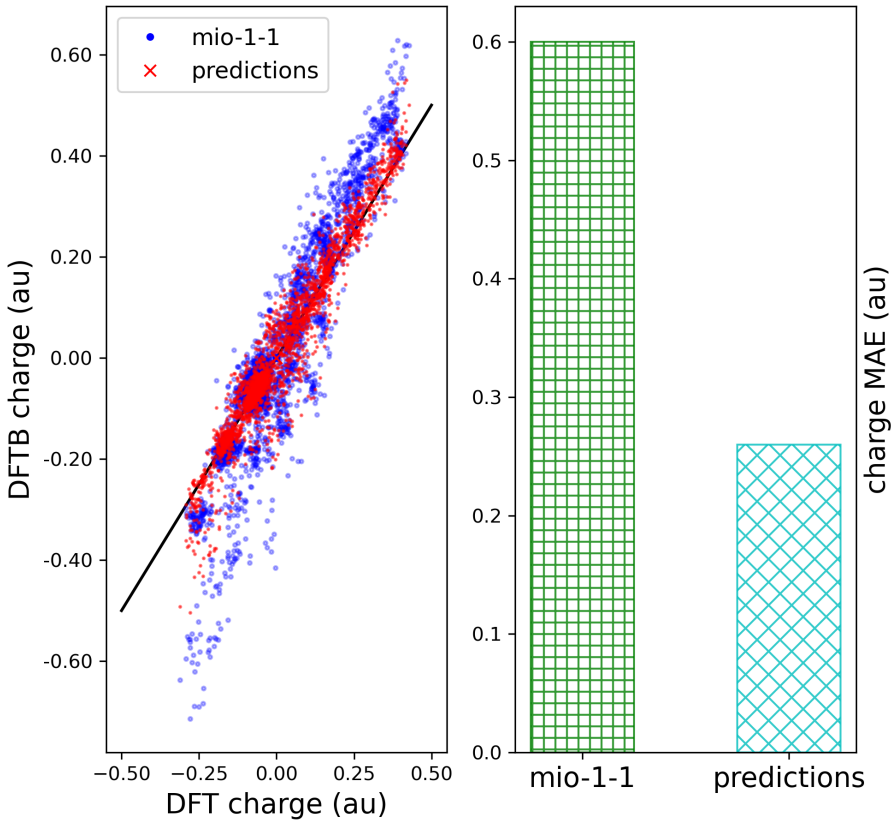


Figure 6.9: Predictions of the Mulliken charges using a data set containing molecules with five heavy atoms after training on a data set containing molecules with three heavy atoms only. The local basis function training scheme was applied; the sizes of the training and prediction sets were 1000 and 400, respectively.

6.4.1 Scaling Transferability

The scaling transferability of a machine learning model is evaluated based on whether the model can be trained on small systems but applied to predict the properties of larger systems. In this work, we investigate the scaling transferability of our model by training

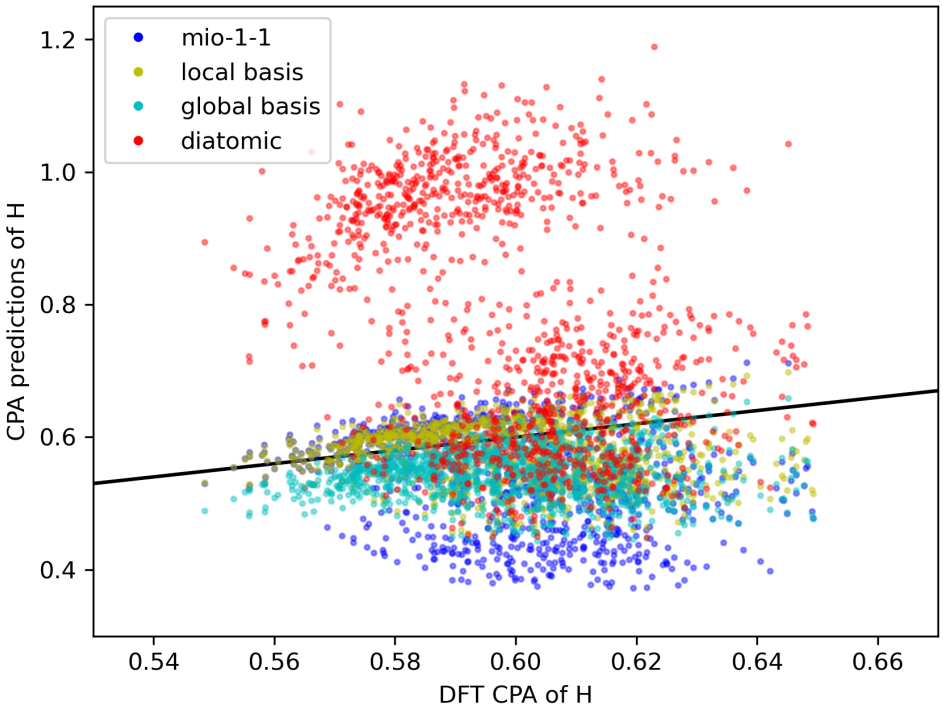


Figure 6.10: CPA predictions for hydrogen atoms in molecules of the ANI-1_1 data set. The models were trained on dipole moments. The black line represents the DFT reference. The data set sizes for the training and predictions were 1000 and 400, respectively.

it on the ANI-1_3 data set and testing it on the ANI-1_5 data set. The two-centre integrals and on-site energies used for scaling transferability evaluation are based on the local approach, which involves local compression radii and on-site energies training, and the electronic property considered is Mulliken charges. Figure 6.9 demonstrates that the predictions obtained using the local DFTB-ML model are about 55% lower than those obtained from the `mio-1-1` Slater-Koster files, indicating that the DFTB-ML model is scalable and can be used to predict more complex chemical environments of the testing set than the training set.

6.4.2 Transferability of Physical Properties

Incorporating physical models into machine learning models can enhance their transferability across different physical properties. In this section, we compared three approaches for constructing the DFTB Hamiltonian's two-centre integrals: direct prediction with diatomic fit, global basis functions, and local basis functions. We trained these models on dipole moments using the ANI-1_1 data set and evaluated them by predicting CPA ratios. The results shown in Figure 6.10 indicate that the diatomic model, which directly predicts the two-centre integrals, produced some physically implausible predictions. In contrast,

the global and local approaches, which employ well-defined basis functions, consistently produced physically plausible predictions. This suggests that training on parameters in basis functions better preserves physicality and yields values of untrained properties within a reasonable range.

6.5 Conclusions and Outlook

In this section, we combined the DFTB method with machine learning to predict the electronic properties of molecules. Our workflow achieved a cheap, accurate, and transferable scheme for predicting electronic structures by optimizing the two-centre integrals and on-site energies in the DFTB method using machine learning.

We compared three different approaches for building the DFTB Hamiltonian's two-centre integrals: direct prediction and explicit calculation using globally or locally optimized atomic basis functions. All of these approaches showed improvement compared to traditional SCC-DFTB calculations using `mio-1-1` parameter set, with local compression radii optimization giving the best overall predictions. The local compression radii approach allows for tuning the confinement term parameters in a chemically dependent manner, providing greater flexibility to the model. We emphasize the importance of well-defined basis functions. The diatomic approach, which directly predicts the diatomic integrals with low errors, showed poor transferability between physical properties, leading to physically unreasonable predictions of electronic properties not considered during training.

The next step for the DFTB-ML framework is to extend it to include solid systems, which will be discussed in the following chapter. Another ongoing work is to incorporate the repulsive potential into the model to predict total energies and forces.

7 Machine Learning of Band Structures

The band structure is a fundamental physical property of solid-state materials, and accurately predicting it is crucial in materials science. However, DFT with GGA has been found to underestimate band gaps, which limits its accuracy [188]. To overcome this limitation, more computationally expensive methods such as the HSE hybrid functional [94, 95] and GW method [68] have been developed. Although HSE can provide accurate results that match experimental data, its computational expense makes it suitable only for systems of limited size.

Recently, machine learning has emerged as a promising approach to predict band structures [68, 189, 190, 191, 192]. Machine learning models achieve a good balance between accuracy and computational expense, making them suitable for larger systems. These models use various techniques to predict the materials' electronic structures and learn the relationship between the input features and the band structure outputs. By training on an extensive data set of known band structures, these models can accurately predict band structures of new materials, making them a valuable tool for materials design and discovery.

This chapter presents a novel approach combining machine learning and the DFTB method to reproduce hybrid functional results. Our DFTB-ML method can accurately predict band structures in bulk systems and various more complex chemical environments, such as slab models and defect systems. Our framework enables high-accuracy predictions of band structures at a low computational cost. The DFTB-ML approach is built upon a machine learning model that has been trained using DFT results with a hybrid functional. By learning the two-centre Hamiltonian integrals and on-site energies to minimize the errors between DFTB band structure calculations and hybrid functional band structures, our machine learning model achieves high accuracy in predicting band structures compared to hybrid functional calculations. Moreover, this approach can be applied to systems that are more accessible to hybrid functional calculations. The DFTB-ML framework has the potential to significantly reduce the computational cost of accurate band structure predictions and open up new possibilities for materials design and discovery.

7.1 Methods and Data Collection

Data set generation

In order to increase the diversity of our machine learning data set, we constructed models for different systems. Specifically, we included bulk silicon, carbon, and silicon carbide systems with several different lattice structures. We used a three-layer system with a 15.0 Angstrom vacuum for slab models. The supercell system used for the bulk systems had 64 atoms based on the diamond lattice structure. Additionally, we included defect systems by removing one atom from the supercell system, resulting in a system with 63 atoms. Specifically, one silicon atom has been removed in silicon carbide systems with one point defect.

To capture a variety of chemical environments, we performed MD simulations using the DFTB+ package [177] and the `pbc-0-3` Slater-Koster files [193]. Geometry optimizations were performed before MD simulations. We used the NVT ensemble at 1273 K with a Nosé-Hoover thermostat [194, 195, 196]. A time step of 1 fs with 1000 steps was used, and every 20 steps, the geometries were recorded as the structures for learning. In the case of slab models, only the surface layer was allowed to move during MD simulations, while for defect systems, the first and second neighbouring atoms of the vacancy were allowed to move. All geometries for band structure calculations in the following sections have been based on the MD simulations.

Table 5: Data collection for machine learning band structure calculations.

	lattice	size	number	slab	defect	K-mesh
Si	hexagonal	2	50	No	No	$9 \times 9 \times 5$
	tetragonal	2	50	No	No	$9 \times 9 \times 9$
	tetragonal	4	50	No	No	$5 \times 5 \times 9$
	diamond	2	50	No	No	$9 \times 9 \times 9$
	diamond	8	50	No	No	$7 \times 7 \times 7$
	diamond	64	50	No	No	$5 \times 5 \times 5$
	diamond	24	50	100	No	$7 \times 7 \times 5$
	diamond	24	50	110	No	$7 \times 7 \times 5$
	diamond	24	50	111	No	$7 \times 7 \times 5$
	diamond	63	50	No	Yes	$5 \times 5 \times 5$
C	hexagonal	2	50	No	No	$9 \times 9 \times 5$
	hexagonal[1]	4	50	No	No	$9 \times 9 \times 5$
	hexagonal[2]	4	50	No	No	$9 \times 9 \times 5$
	diamond	2	50	No	No	$9 \times 9 \times 9$
	diamond	8	50	No	No	$7 \times 7 \times 7$
	diamond	24	50	100	No	$7 \times 7 \times 5$
	diamond	24	50	110	No	$7 \times 7 \times 5$
	diamond	24	50	111	No	$7 \times 7 \times 5$
	diamond	64	50	No	No	$5 \times 5 \times 5$
	diamond	63	49	No	Yes	$5 \times 5 \times 5$
SiC	cubic	2	50	No	No	$9 \times 9 \times 9$
	cubic[1]	8	50	No	No	$7 \times 7 \times 7$
	diamond[2]	2	50	No	No	$9 \times 9 \times 9$
	diamond	8	50	No	No	$7 \times 7 \times 7$
	diamond	24	50	100	No	$7 \times 7 \times 5$
	diamond	24	50	110	No	$7 \times 7 \times 5$
	diamond	24	50	111	No	$7 \times 7 \times 5$
	diamond	64	50	No	No	$5 \times 5 \times 5$
	diamond	63	50	No	Yes	$5 \times 5 \times 5$

Table 5 shows the geometries used for machine learning training. The data set includes a range of bulk systems with different lattice structures, providing a representative sample

of the diversity of materials in nature. Hexagonal[1] and hexagonal[2] structures of carbon correspond to AA- and AB-stacked bilayer graphene, respectively, while hexagonal structures with two atoms represent a lattice cell containing a single layer of graphene. Both cubic[1] and diamond[2] in Table 5 are cubic crystals. In cubic[1], one silicon atom is bonded to six equivalent carbon atoms to form a mixture of corner and edge-sharing SiC_6 octahedra, while in diamond[2], one silicon atom is bonded to four equivalent carbon atoms to form corner-sharing SiC_4 tetrahedra. Additionally, we included slab and defect models, allowing for the prediction of band structures in these systems. The **k**-mesh in Table 5 was used for DFT band structure calculations. We obtained 50 geometries for the carbon vacancy system; however, only 49 were used in a carbon diamond system with one point defect. One of the geometries failed to converge in the DFT calculations with the hybrid functional and thus was not included in our study. The **k**-mesh shown in Table 5 has been applied to each geometry for DFT and DFTB band structure calculations and DFTB MD simulations. MD simulations allowed us to capture the effects of thermal fluctuations and environmental interactions on the electronic structure of the materials, making our data set more realistic and applicable to real-world scenarios.

DFT calculations

The geometries used in DFT calculations were obtained from molecular dynamics calculations conducted previously. With a light-level basis set, the band structure calculations were performed using the FHI-aims package [175]. To generate more precise reference band structures for our machine learning models, we employed the HSE functional (introduced in chapter 2) and adopted the parameters from the HSE06 method [95]. To assess the accuracy of the light level basis set, we compared hybrid band structures based on both tight and light levels, as shown in Figure 7.1. Our results demonstrate that the light level basis set is sufficient for accurately predicting band structures, with good agreement between the tight and light level calculations. The bandgap obtained using the light and tight basis sets were 5.42 eV and 5.51 eV, respectively, comparable to the experimental value of 5.47 eV [197]. Additionally, the high-symmetry points were generated automatically using the method introduced by Wahyu and Stefano [174].

Representations of atomic geometries

In this chapter, we employed ACSFs [128] as machine learning features. Specifically, we utilized the cutoff function G_1, the radial symmetry function G_2, and the angular function G_4, introduced in chapter 3. The implementation of ACSFs utilized a cutoff parameter R_c of 10 Angstrom, along with η and R_s values of 1.0 and 1.0 Angstrom for G_2, and η, ζ, and λ values of 0.02, 1.0, and -1.0 for G_4. The ACSFs utilized for on-site energies are conventional atomic symmetry functions, while those for scaling parameters are diatomic features based on conventional ACSFs. The diatomic scaling parameters will be discussed in the DFTB-ML workflow section. The G_4 function in Eq (3.5) can both be used to represent atomic and diatomic chemical environments. G_4 can represent the

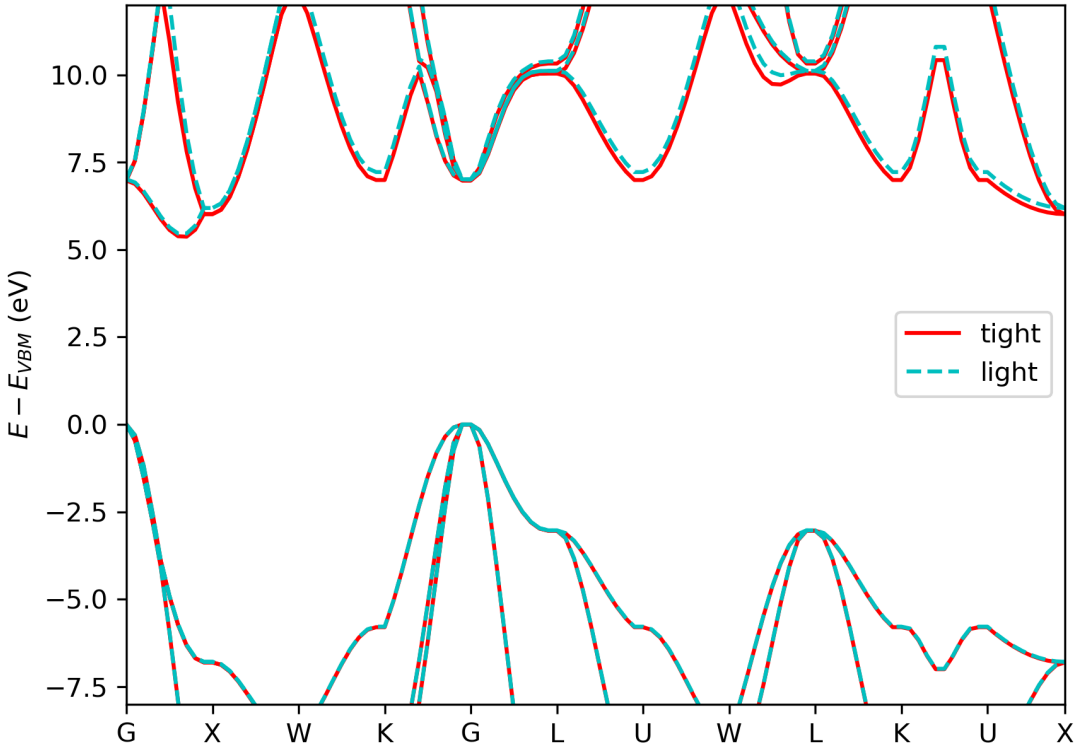


Figure 7.1: Band structures of carbon diamond using the HSE hybrid functional method in the FHI-aims software. We compare the results obtained using two different basis sets: light and tight.

interactions of atom i with two other atoms j and k while also representing the atomic pair i and j interacting with atom k. On the other hand, the G_1 and G_2 functions can only represent atomic chemical environments. To represent the diatomic chemical environment between atom i and atom j, we add the G_2 function of atom i and G_2 function of atom j with a decay function f_c. The revised G_2 function is as follows:

$$G_{ij}^2 = f_c(R_{ij}) \left[\sum_{k \neq i}^{\text{all}} e^{-\eta(R_{ik}-R_s)^2} f_c(R_{ik}) + \sum_{k \neq j}^{\text{all}} e^{-\eta(R_{jk}-R_s)^2} f_c(R_{jk}) \right]. \tag{7.1}$$

Here, the notation employed is consistent with that utilized in Equation (3.5). The G_1 function has also been revised in a similar manner. Additionally, the feature representations in this chapter incorporate periodic boundary conditions, and thus the periodic ACSFs were utilized for diatomic scaling parameters and on-site energies.

DFTB-ML parameters

All DFTB calculations in the DFTB-ML process were performed using TBMaLT [180]. The initial DFTB Hamiltonian and overlap integrals were obtained from a traditional DFTB parametrization. The learned parameters consisted of scaling factors for the two-body Hamiltonian integrals and on-site energies. During the training process, the learning rate for scaling factors of the Hamiltonian integrals in carbon systems was set to 3×10^{-3}, and on-site energies were set to 1×10^{-3}. The learning rate for scaling factors of the Hamiltonian integrals in silicon carbide systems was set to 3×10^{-3}, and for on-site energies was set to 2×10^{-3}. The learning rate for scaling factors of the Hamiltonian integrals in silicon systems was set to 5×10^{-4}, and for on-site energies was set to 4×10^{-4}. The Adam optimizer [158] was employed to optimize the parameters via backward propagated gradients. The default loss functions were MAEs unless otherwise specified. The convergence tolerance for all training was set at 1×10^{-4} eV. We have used the random forest algorithm to predict the on-site energies and scaling parameters based on the geometric features, and 100 estimators were applied in random forest.

7.2 DFTB-ML Workflow

In order to obtain a good initial starting point for machine learning-based DFTB band structure optimization, we applied two-step optimization procedures. Firstly, we globally optimized the basis parameters and on-site energies to minimize the band structure errors between DFT and DFTB. Based on the two-centre Hamiltonian integrals and on-site energies obtained in the first step, we further optimized the Hamiltonian integrals and on-site energies using machine learning.

Global optimization

For silicon systems, the `siband-1-1` parameter set [41, 42] have been fitted based on experimental results and perform state-of-the-art DFTB band structure predictions. The `siband-1-1` parameter set was used as the basis set parameters for generating the initial Slater-Koster parameter set. For carbon systems, the existing parameters do not satisfactorily predict the band structures. Therefore, we initially screened the basis set parameters to optimize the Slater-Koster tables for carbon and silicon carbide systems globally. We then utilized these optimized parameters as the starting point for subsequent training.

The compression radii of the carbon s and p orbitals are the same, with grid points at 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.5, 4.0, and 4.5 Bohr. Meanwhile, the grid points of compression radii for the d orbital are 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.5, 4.0, and 4.5 Bohr. We determined the final compression radii by calculating the MAEs between hybrid functional DFT and DFTB calculations to obtain the optimized compression radii. Additionally, the on-site energy of the d orbital was adjusted, and the grid points of on-site energy are 0.0, 0.05, 0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, and 0.5 Hartree. The errors were computed as follows:

$$\text{Loss} = \frac{1}{N_i}\sum_i \frac{1}{N_v}\frac{1}{N_{\mathbf{k}}}\sum_v\sum_{\mathbf{k}} |\epsilon^{\text{DFT}}_{i,\mathbf{k},v} - \epsilon^{\text{DFTB}}_{i,\mathbf{k},v}|, \tag{7.2}$$

where N_i represents the number of geometries, including carbon and silicon carbide with various lattice types, and $N_{\mathbf{k}}$ denotes the number of selected $\mathbf{k}$ points, and N_v is the number of selected energy states. Unlike Eq. (4.1), the first derivative is not applied since there is no flat band issue in this system shown in Table 5. As a result, we chose a compression radius of 4.0 Bohr for the s and p orbitals and 2.75 Bohr for the d orbital. We also shifted the on-site energy of the carbon d orbital from 0.02 to 0.45 Hartree. The Hamiltonian and overlap integrals based on these parameters were used as initial Slater-Koster tables for machine learning.

DFTB-ML optimization

The diatomic integrals and on-site energies obtained from the global optimization were used as the initial parameters for the DFTB-ML optimization. To incorporate a scaling parameter $\alpha_{AB}^{l_A l_B}$ for the two-center Hamiltonian integral $H_{AB}^{l_A l_B}$ of atomic pair A and B and azimuthal quantum number pair l_A and l_B, we set the initial value of $\alpha_{AB}^{l_A l_B}$ to 1.0 and then used $\alpha_{AB}^{l_A l_B} H_{AB}^{l_A l_B}$ instead of $H_{AB}^{l_A l_B}$ for the subsequent DFTB calculations. Here, two different integrals of the same atomic pair and the same azimuthal quantum number pair share the same scaling parameter. For instance, the scaling parameters of $H_{AB}^{pp^\sigma}$ and $H_{AB}^{pp^\pi}$ are the same. The scaling parameters of all atomic pairs were then optimized to minimize the band structure errors between DFTB and DFT. The on-site energies of each atom were optimized directly. The overlap integrals remain the same as the overlap in global optimization. For the DFTB-ML model, the machine learning targets were the scaling parameters and on-site energies.

When building DFTB-ML models, one of the most important tasks is to define the loss functions. The chosen loss functions are MAEs for band structure learning, which is

$$\text{Loss} = \frac{1}{N_i}\frac{1}{N_v}\frac{1}{N_{\mathbf{k}}}\sum_i\sum_v\sum_{\mathbf{k}} |\epsilon^{\text{DFT}}_{i,\mathbf{k},v} - \epsilon^{\text{DFTB}}_{i,\mathbf{k},v}|. \tag{7.3}$$

The notations used in this section are the same as those mentioned in the previous section. The slight difference between Eq. (7.3) and Eq. (7.2) is that the loss function in Eq. (7.3) is directly the average of all selected eigenvalues. We use different loss functions in Eq. (7.3) and Eq. 7.2. We use Eq. 7.2 due to the differences in eigenvalue summations among various geometries. This variance is attributed to differences in geometric size and element species. Consequently, calculating the average eigenvalue error for each geometry is reasonable. However, for DFTB-ML optimization, we train systems with different element species and sizes separately. As elucidated in Chapter 5, we employ padding for batch operability. Padding diverse geometric sizes generates sparse tensors with numerous zeros. To ensure efficient training, we train geometries of varying sizes and element species separately. In this scenario, the total eigenvalue errors for each geometry are similar, enabling direct summation of all eigenvalues in the loss function using Eq. (7.3). We have chosen all the

eigenvalues of high symmetric **k** points for use in the loss function. The **k** path between two high symmetric points consists of ten grid points, and the grid point located at the midpoint of the path is also included in the loss function. During our machine learning training, all valence bands were taken into consideration. Additionally, the number of conduction bands included in the loss function was determined based on the number of atoms. Specifically, in the case of a diamond with two atoms, two conduction bands were used in the loss function.

7.3 Training on Bulk Systems

Effect of training size

We began with optimized global parameters in our machine learning training of DFTB parameters for band structures. Then the on-site energies and the scaling parameters of the Hamiltonian integrals were optimized based on Slater-Koster tables from global optimization. Each sub-training set, as shown in Table 5, comprised 50 geometries, except for the carbon diamond defect system. To determine the optimal size of the training sets, we used data set ratios of 0.1, 0.2, 0.3, 0.4, and 0.5, as illustrated in Figure 7.2. The testing data set ratio was 0.2, and we tested the performance of our training model using the random forest algorithm for predicting on-site energies and scaling parameters and ACSFs for atomic geometry representations. Our results indicate that a ratio of 0.4 provides the best trade-off between the training set's size and the predicted results' accuracy. Therefore this ratio was used for the subsequent training.

Training on bulk systems

Using the data set outlined in Table 5, we initiated the machine learning training process on bulk materials with varying lattice types and geometric sizes. Geometries sharing the same lattice-type but differing sizes were trained independently and evaluated collectively. This segregated training approach helps circumvent the issue of padding zeros, as discussed in chapter 5. Figure 7.3 illustrates the average MAEs of carbon diamond systems, encompassing those with 2, 8, and 64 atoms. The parametrization of the previous `pbc-0-3` parameter set [193] was primarily focused on the periodic system, specifically targeting carbon and silicon elements. It is important to note that the `pbc-0-3` parameter set represents a minimal basis set. The comparison between `pbc-0-3` parameter set and the globally optimized parameter set highlights that optimizing the basis parameters and on-site energies can improve the performance of the band structures. Additionally, including d orbitals in the basis function can enhance the band structure's performance. We compared band structure calculations based on the DFTB-ML model with band structures based on the previous `pbc-0-3` parameter set and our globally optimized parameter set. The results in Figure 7.3 indicate that, compared with `pbc-0-3`, all band structure calculations based on the globally optimized parameter set have remarkably improved. By applying the DFTB-ML model, which uses the globally optimized Slater-Koster tables as

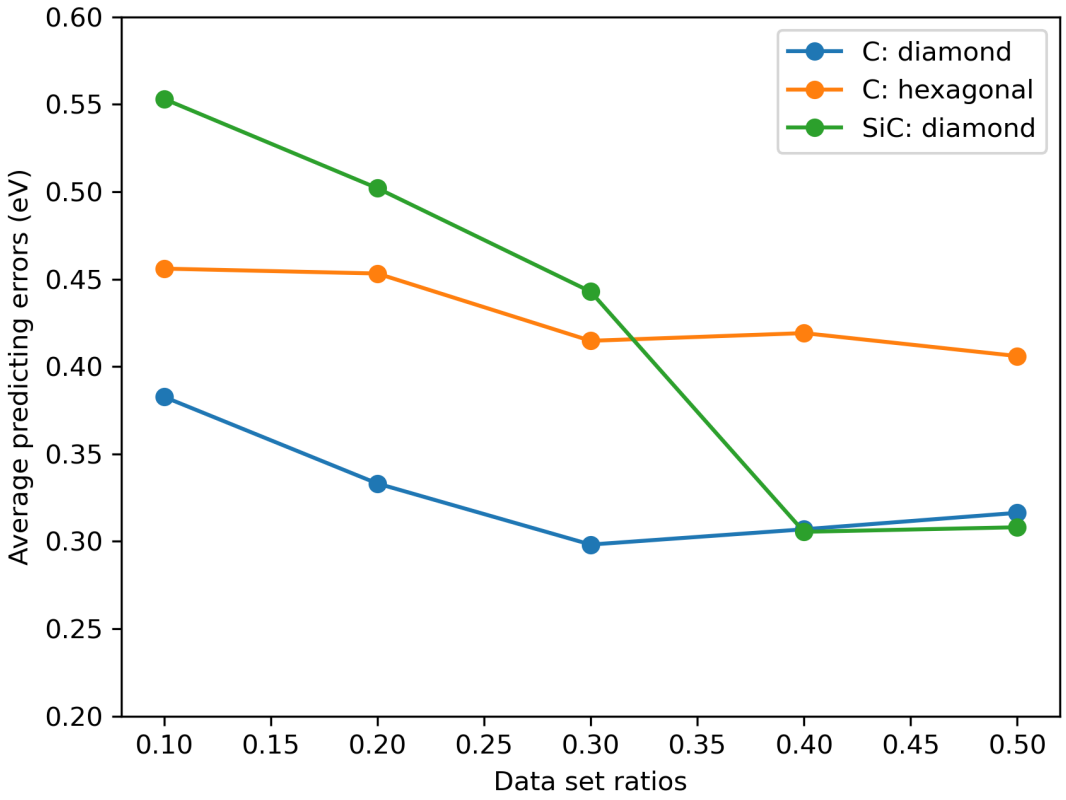


Figure 7.2: Effect of training size on the testing errors.

the initial parameters for training, band structure calculations based on Mulliken charges from well-converged SCC-DFTB calculations can be further improved. The results suggest that the chemical environment adaptive machine learning model can enhance band structure calculations with various lattice types and geometric sizes.

To demonstrate the accuracy of machine learning-based band structure predictions, we present Figure 7.4 to visualize the band structure performance. The geometry used for this calculation was taken from MD step 120, and the MAE value is 0.26 eV, close to the average value of carbon diamond shown in Figure 7.3. The DFTB-ML model accurately reproduces the valence band maximum (VBM) and conduction band minimum (CBM) of the DFT-HSE band structures and the band gap. However, for energy states approximately 10 eV below the VBM, the prediction error remains around 1 eV, contributing to a significant portion of the overall error.

Similar results were obtained for band structure predictions of other materials with different geometries. The DFTB-ML model can perfectly reproduce the VBM and CBM of these materials and the main contribution to the prediction errors is from energy states that are far away from the VBM.

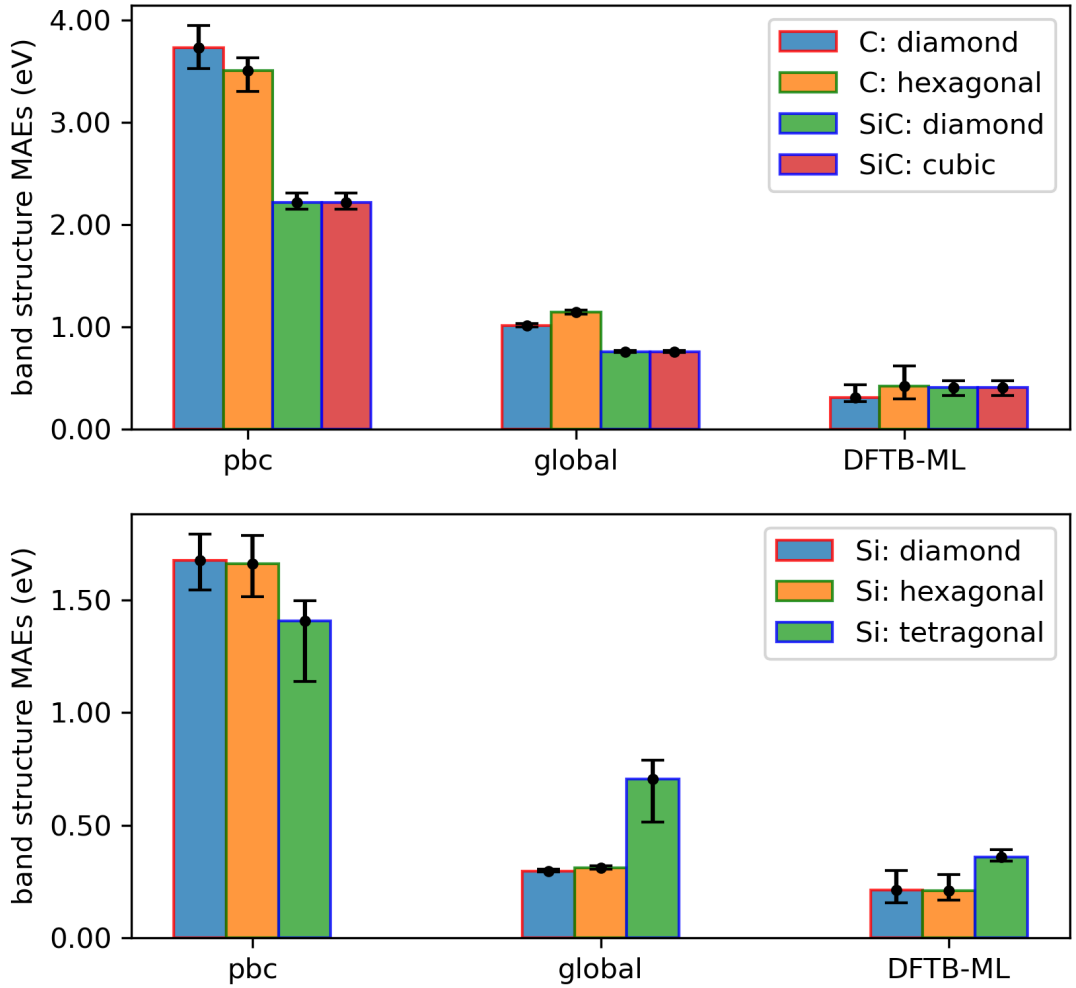


Figure 7.3: Band structure MAEs of bulk carbon, silicon, and silicon carbide geometries using `pbc-0-3` set, the global optimized Slater-Koster tables, and the DFTB-ML model. The testing ratio was 0.2, and the MAEs are reported in eV. The reference values are from DFT-HSE calculations.

7.4 Training on Defect and Slab Systems

The results show that the predictions of band structures based on the training model on various bulk systems can improve the band structure performance. This section further investigates more complex environments, including slab models and defect systems. Figure 7.5 shows the predictions of band structures by using previous training models, including bulk, slab, and defect systems. The slab models include (100), (110), and (111) surfaces of silicon, carbon, and silicon carbide diamond structured geometries, which are

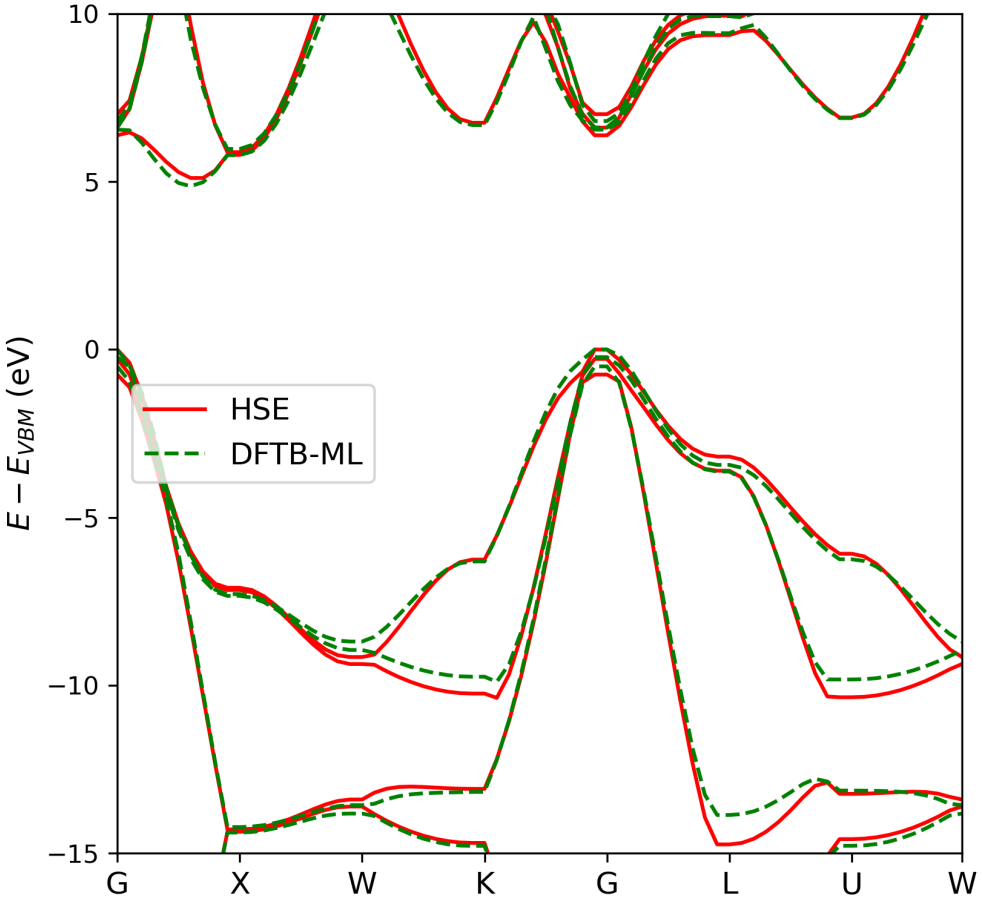


Figure 7.4: The band structure calculations of the carbon diamond system were performed using the DFTB-ML model, with the geometry taken from a snapshot of MD (step 120). The resulting band structure predictions were compared to DFT-HSE calculations.

generated from MD calculations. The defect systems considered in this study include silicon, carbon, and silicon carbide diamond structures, each with a point vacancy resulting from removing one atom. In the case of silicon carbide, one silicon atom has been removed. As shown in Figure 7.5, the global optimization of Slater-Koster tables has improved the band structures of all systems. Further improvements have been achieved by using the DFTB-ML model. This result indicates that DFTB-ML is capable of performing better in complex chemical environments.

To visualize the performance of the DFTB-ML model, we chose the carbon diamond system with a point defect in Figure 7.6. The value of MAEs of this testing geometry is 0.24 eV, slightly higher than the average MAE of 0.18 eV in the carbon diamond systems with point defects. Figure 7.6 shows that the DFTB-ML model can accurately reproduce

the band structures near VBM and CBM.

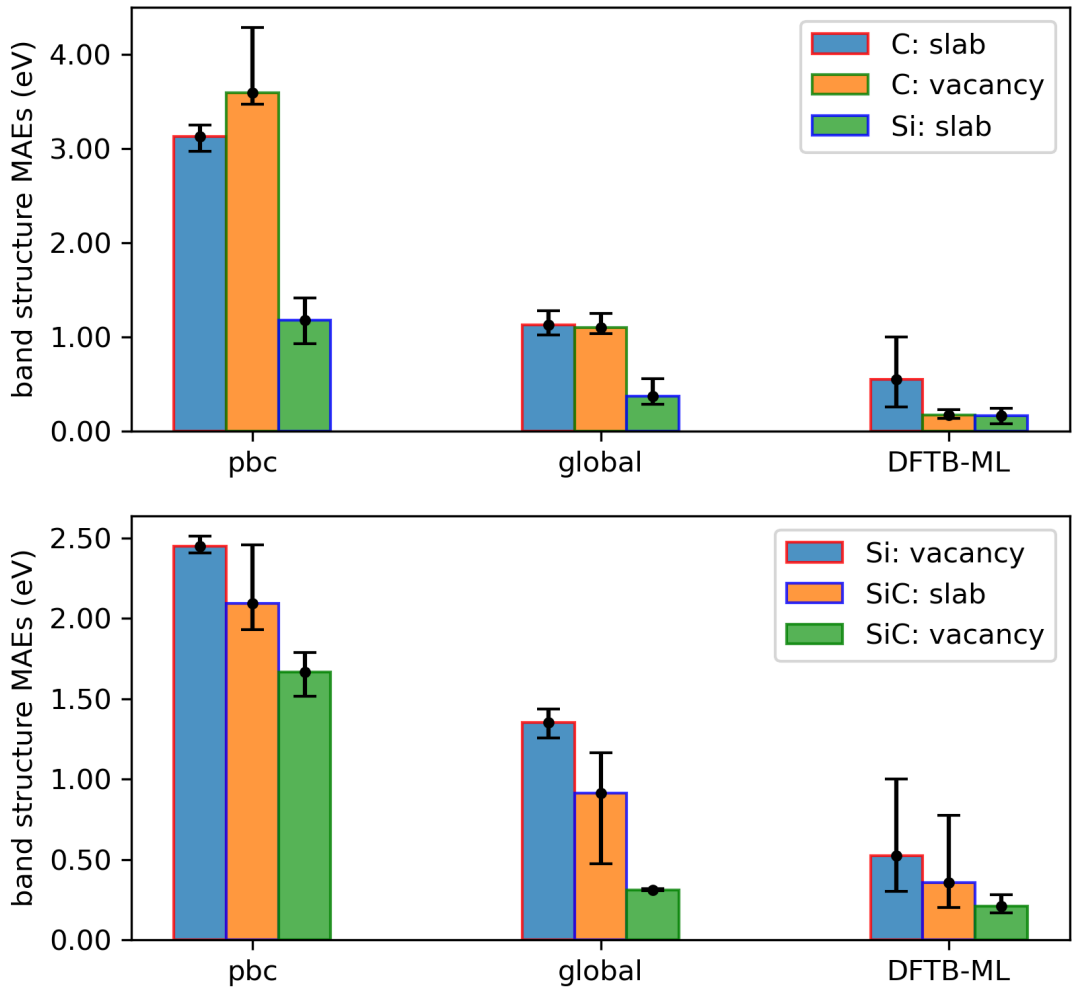


Figure 7.5: Band structure MAEs of slab and defect carbon, silicon, and silicon carbide geometries from `pbc-0-3` set with minimal basis functions, global optimized Slater-Koster tables with d orbitals, and DFTB-ML predictions based on global optimized parameter set. The testing ratio was 0.2, and the MAEs are reported in eV. The reference values are from DFT-HSE calculations.

In this section, we have trained and tested slab models and defect models. In all geometries, DFTB-ML gives the most accurate band structure predictions. The DFTB-ML model performs incredibly well near VBM and CBM, similar to the bulk systems.

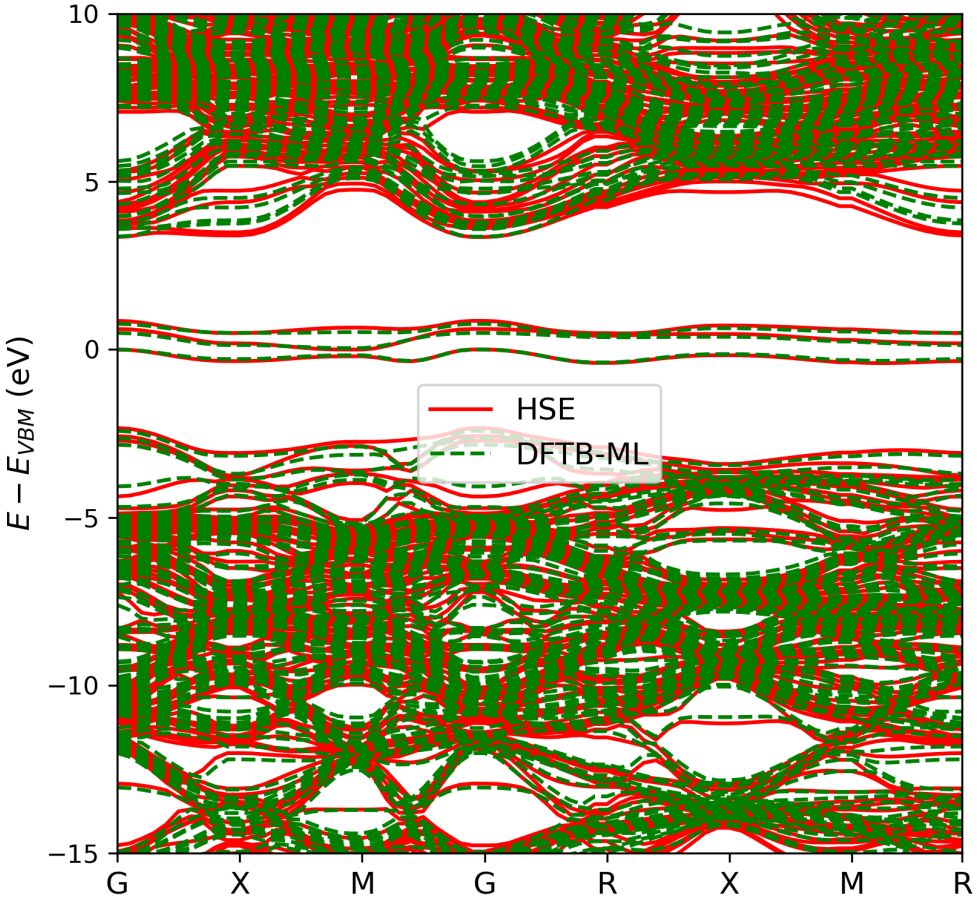


Figure 7.6: The band structure calculations of the carbon diamond defect system with 63 atoms were performed using the DFTB-ML model, with the geometry taken from a snapshot of MD (step 360). The resulting band structure predictions were compared to DFT-HSE calculations.

7.5 Transferability

In this chapter, we have investigated the transferability of our method by training our models using small carbon diamond geometries with 2 and 8 atoms and applying the training models to predict the band structures of carbon diamond systems with 64 atoms and 63 atoms with one point defect. The results, as shown in Figure 7.7, indicate that both the global optimized method and DFTB-ML model significantly improve the accuracy of band structures from DFTB calculations for larger systems. The MAEs for all testing geometries between DFTB calculations and DFT-HSE results decrease from 3.7 eV using `pbc-0-3` parameter set to 1.0 eV using global optimized parameter set and 0.3 eV from DFTB-ML models, respectively. Similarly, using small geometries for training for carbon

defect systems decreases the error from 3.6 eV using `pbc-0-3` parameter set to 1.1 eV using global optimized parameter set and 0.6 eV from DFTB-ML models, respectively. Both the global optimized Slater-Koster parameter set and DFTB-ML model accurately reproduce the band gap and VBM, with the latter providing better predictions for the CBM. These results demonstrate the potential of DFTB-ML for accurately predicting the electronic structure of larger and more complex systems.

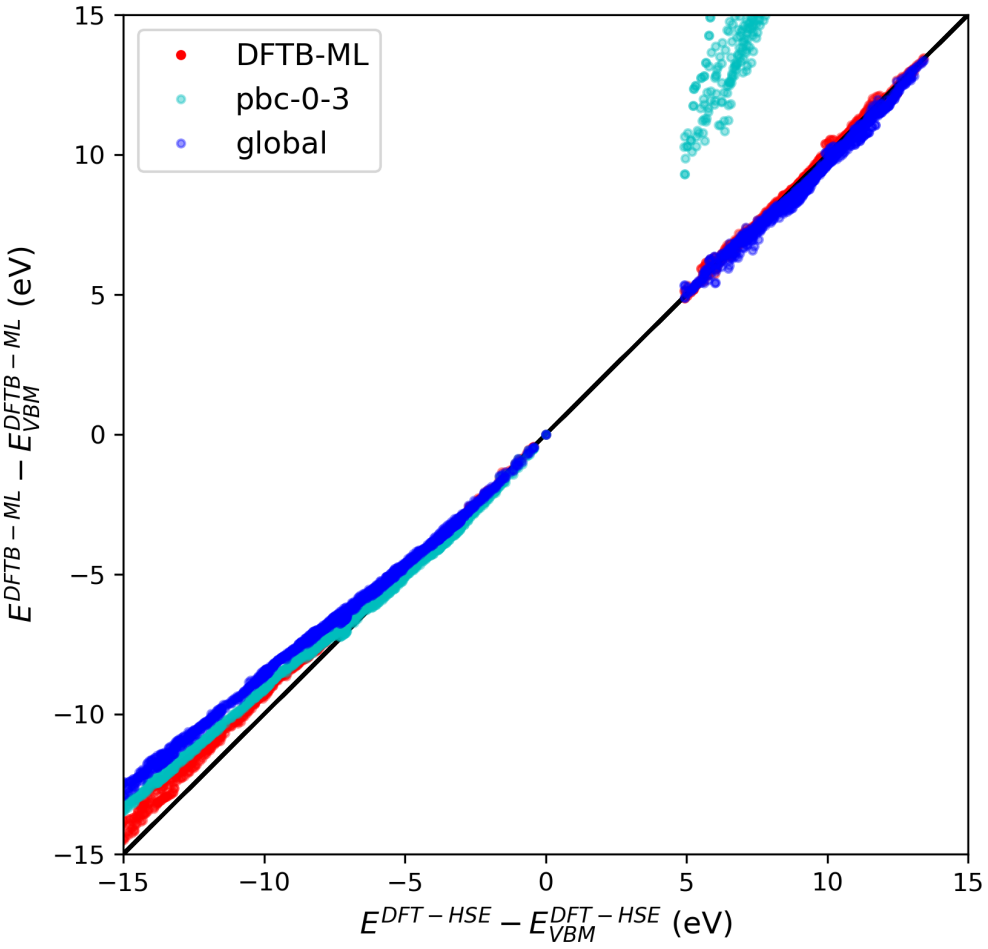


Figure 7.7: Testing transferability of DFTB-ML models in predicting band structures of carbon diamond with 64 atoms based on training small carbon diamond systems (2 and 8 atoms). The testing ratio was 0.2. The geometries were carbon diamond crystals.

In summary, our results demonstrate that the DFTB-ML model based on small geometries can significantly improve the accuracy of band structure calculations for larger systems, even for systems with defects not included in training sets. This highlights the excellent transferability of our DFTB-ML models.

7.6 Conclusions and Outlook

In this section, we propose a two-step machine learning workflow to improve the accuracy of band structure predictions for carbon, silicon, and silicon carbide systems. The first step is global optimization of the basis parameters of the carbon element. As the `siband-1-1` Slater-Koster tables have shown exemplary performance in band structure calculations for silicon and silica systems, we adopt the same basis parameters for our study. We optimize the basis parameters of carbon by tuning the compression radii and on-site energies to minimize the band structure errors between DFT-HSE and DFTB calculations for bulk carbon and silicon carbide systems. Using the globally optimized parameter set can significantly decrease the MAEs compared to the previously used `pbc-0-3` parameter set. This improvement demonstrates that incorporating optimized basis parameters, on-site energies, and d orbitals in carbon and silicon elements can enhance band structure calculations. In the second step, we utilize the DFTB-ML model to further optimize the DFTB band structure calculations. This optimization is performed using the globally optimized Slater-Koster tables as the initial parameter set, resulting in a further reduction in the MAEs in all cases.

The geometries considered in our study encompass bulk structures with various lattice types, slab models, and defect systems, representing a diverse range of chemical environments. The application of DFTB-ML improves the band structure calculations for all these geometries, showcasing the advantages of machine learning in handling complex systems. Additionally, our model demonstrates good scaling transferability for carbon diamond structured geometries, indicating that the DFTB-ML approach can efficiently and accurately calculate the band structures of large systems using the cost-effective DFTB method.

In the previous chapter, we implemented our DFTB-ML model in TBMaLT for molecule data sets. In this section, we extend our approach to periodic boundary conditions, enabling the prediction of band structures. As a next step, we plan to apply the DFTB-ML model to real systems, such as lithium batteries. Furthermore, our objective encompasses expanding the DFTB-ML model to encompass repulsive potentials, facilitating the computation of total energies and forces. This expansion enables us to conduct molecular dynamics simulations and geometry optimizations.

8 Conclusions

This thesis starts by presenting the traditional parametrization of DFTB for LIBs, followed by a focus on implementing the DFTB-ML framework TBMaLT and its applications in both molecular and solid-state systems. The incorporation of machine learning shows promising potential for enhancing the DFTB parametrization with machine learning based techniques.

DFTB parametrization for Lithium-ion batteries

In this study, we investigated the DFTB parametrization of $Li_6(PS_4)SCl$ and $Li_5(PS_4)Cl_2$, which are promising solid-state electrolytes for next-generation LIBs due to their high ionic conductivity and stability.

For the initial step, we used the `3ob-3-1` Slater-Koster tables for phosphorus, sulfur, and chlorine and tuned the compression radii of lithium only. However, this approach alone led to the underestimation of the band gap and swapped certain valence bands of $Li_6(PS_4)SCl$. Through PDOS analysis, we found that the valence bands near the VBM were influenced by the p orbitals of lithium and sulfur. The compression radii parameters of the lithium and sulfur elements were adjusted, while the parameters of phosphorus and chlorine were kept the same as in the `3ob-3-1` Slater-Koster files since they had minimal impact on the band structures. We applied cubic lithium, cubic sulfur, $Li_6(PS_4)SCl$, and $Li_5(PS_4)Cl_2$ systems for electronic parameterization by minimizing the band structure errors between DFT and DFTB calculations of these systems. We found that increasing the compression radii of the sulfur p orbital improved the band structure performance. However, the values of compression radii that minimized the band structure errors made it challenging to fit the repulsive parameters of the sulfur system. Therefore, we determined compromised compression radii for lithium and sulfur, enabling reasonable DFTB band structure calculations and subsequent repulsive fitting. We also found that tuning the on-site energies for sulfur p and d orbitals can systematically improve the DFTB band structure performance.

Finally, we determined the repulsive energies of cubic lithium, cubic sulfur, Li_3P, Li_2S, LiCl, $Li_6(PS_4)SCl$, and $Li_5(PS_4)Cl_2$ systems using the CCS method, which reproduced the geometry optimization results obtained from DFT calculations with reasonable accuracy.

Tight binding machine learning toolkit implementation

This work introduced the implementation of TBMaLT that enables standard DFTB calculations and machine learning-based automatic parametrization. TBMaLT is compatible with both molecular and solid systems using standard Slater-Koster files and offers a range of electronic property calculations. It supports high-throughput DFTB calculations, and the batch implementation feature allows for reasonably efficient high-throughput calculations. TBMaLT also offers flexibility by allowing the incorporation of different machine

learning-based Hamiltonian and overlap matrices. Several machine learning methods have been developed for constructing Hamiltonian and overlap matrices.

Machine Learning of Molecular Electronic Properties

Based on the current TBMaLT implementation, we have integrated machine learning into the DFTB method to predict the electronic properties of molecules. Our workflow has resulted in a cheap, accurate, and transferable scheme for predicting electronic structures. This workflow optimized the two-centre integrals and on-site energies using machine learning, which enhanced the accuracy of predictions for both single and multiple electronic properties.

We compared three approaches for constructing two-centre integrals and on-site energies for SCC-DFTB calculations. The first approach adjusted the compression radii and on-site energies in the confinement term for each element species, while the second approach modified them in a chemically-dependent manner for each atom. The third approach predicted and calculated the diatomic integrals globally and directly using a cubic spline method with global on-site energies. All approaches exhibit improvements over traditional SCC-DFTB calculations with the `mio-1-1` parameter set, with the second approach (local optimization of compression radii and on-site energies) yielding the best predictions. The second approach allows for chemically-dependent tuning of the confinement term parameters, enhancing model flexibility.

We found the importance of well-defined basis functions for accurate predictions of electronic properties. The third approach, which directly predicts the diatomic integrals with low errors, showed poor transferability between physical properties and resulted in physically unreasonable predictions of electronic properties. This highlights the need for caution when using machine learning-based approaches for predicting electronic properties.

Machine Learning of Band Structures

In this work, we presented a two-step machine learning workflow to enhance the accuracy of band structure predictions for carbon, silicon, and silicon carbide systems, including bulk, slab, and defect geometries.

In the first step, we optimized the basis parameters of the carbon element by tuning the compression radii and on-site energy of the carbon d orbital. This was achieved by minimizing the band structure errors between DFT-HSE and DFTB calculations for selected bulk carbon and silicon carbide systems. We used the `siband-1-1` basis parameters for silicon, demonstrating exemplary performance in band structure calculations for silicon and silica systems. Using these globally optimized parameter set resulted in a notable improvement compared to the previously employed minimal basis `pbc-0-3` parameters, emphasizing the effectiveness of incorporating optimized basis parameters, on-site energies, and the significance of d orbitals in carbon and silicon elements for improving band

structure calculations.

In the second step, we applied the DFTB-ML model to optimize the DFTB band structure calculations using the globally optimized Slater-Koster tables as the initial parameters. The DFTB-ML model trained and predicted the scaling parameters for the diatomic Hamiltonian integrals for each atomic pair and local on-site energies for each atom. The DFTB-ML-based parameters significantly improved the band structure calculations in all bulk, slab, and defect systems, resulting in chemically environment-adaptive machine learning-based predictions. This DFTB-ML model also demonstrated excellent scaling transferability, allowing for training on small systems and prediction on larger ones. This showcases the potential applications of this approach in modelling large systems that are not accessible to hybrid functional calculations.

List of Publications

- Adam McSloy, **Guozheng Fan**, Wenbo Sun, Christian Hölzer, Marvin Friede, Sebastian Ehlert, Nils-Erik Schütte, Stefan Grimme, Thomas Frauenheim, and Bálint Aradi. Tbmalt, a flexible toolkit for combining tight-binding and machine learning. *The Journal of Chemical Physics*, 158(3):034801, 2023 (**Chapter 5**)

- **Guozheng Fan**, Adam McSloy, Bálint Aradi, Chi-Yung Yam, and Thomas Frauenheim. Obtaining electronic properties of molecules through combining density functional tight binding with machine learning. *The Journal of Physical Chemistry Letters*, 13(43):10132–10139, 2022 (**Chapter 6**)

- Machine learning-based parameterization of density functional tight-binding for band structures in bulk, slab, and defect systems. 2023. In preparation (**Chapter 7**)

- Wenbo Sun, **Guozheng Fan**, Tammo van der Heide, Adam McSloy, Thomas Frauenheim, and Bálint Aradi. Machine learning enhanced dftb method for periodic systems: learning from electronic density of states. *Journal of Chemical Theory and Computation*, 19(13):3877–3888, 2023

- Hongwei Fu, **Guozheng Fan**, Jiang Zhou, Xinzhi Yu, Xuesong Xie, Jue Wang, Bingan Lu, and Shuquan Liang. Facilitating phase evolution for a high-energy-efficiency, low-cost O3-type $Na_xCu_{0.18}Fe_{0.3}Mn_{0.52}O_2$ sodium ion battery cathode. *Inorganic Chemistry*, 59(18):13792–13800, 2020

- Yong Chen, Yuanming Zhang, **Guozheng Fan**, Lizhu Song, Gan Jia, Huiting Huang, Shuxin Ouyang, Jinhua Ye, Zhaosheng Li, and Zhigang Zou. Cooperative catalysis coupling photo-/photothermal effect to drive sabatier reaction with unprecedented conversion and selectivity. *Joule*, 5(12):3235–3251, 2021

Acknowledgements

After four years of studies in Germany, my mind is flooded with memories, and I am filled with gratitude towards the many people who have helped and encouraged me along the way. Foremost, I would like to express my appreciation to Prof. Dr. Thomas Frauenheim, who provided me with the opportunity to study in Bremen. I am also grateful to Prof. Dr. Jianping Xiao and Prof. Dr. Xie Zhang for their invaluable assistance during my application process. Furthermore, I would like to thank Prof. Dr. Chi-Yung Yam, who provided excellent guidance during my stay in CSRC.

Throughout my four years of study, Dr. Bálint Aradi provided outstanding supervision for my projects. I would also like to express my gratitude to Dr. Adam McSloy, who offered me much-needed support in code development. I hope you find your ideal position soon. I would like to thank WenBo Sun and Tammo van der Heide for the numerous insightful discussions we had. I would also like to express my appreciation to Prof. Dr. Guanhua Chen for providing me with the opportunity to visit HK for three months. I would like to extend my thanks to all members of BCCMS. It was a pleasant time to stay in BCCMS with all of you, and I am grateful for the numerous memories we have shared.

I would like to express my gratitude to the committee members, including Dr. Bálint Aradi and Prof. Dr. Thomas Niehaus, for their valuable contributions as reviewers. I am also grateful to Priv. Doz. Dr. Christopher Gies, Prof. Dr. Gordon Callsen, Dr. Carlos Raul Medrano, and Mr. Jonas Müller for their roles as committee members.

Finally, I want to extend my heartfelt thanks to my parents, beloved girlfriend, all my families, and all my friends, who have provided me with unwavering support, especially during the challenging times of the COVID pandemic. Your love and encouragement have been a constant source of motivation and strength for me throughout my studies.

Guozheng Fan

March 2023, Bremen, Germany

References

[1] Gabriel R Schleder, Antonio CM Padilha, Carlos Mera Acosta, Marcio Costa, and Adalberto Fazzio. From dft to machine learning: recent approaches to materials science–a review. *Journal of Physics: Materials*, 2(3):032001, 2019.

[2] Paul Adrien Maurice Dirac. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 123(792):714–733, 1929.

[3] Gordon Bell, Tony Hey, and Alex Szalay. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009.

[4] Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

[5] Lauri Himanen, Amber Geurts, Adam Stuart Foster, and Patrick Rinke. Data-driven materials science: status, challenges, and perspectives. *Advanced Science*, 6(21):1900808, 2019.

[6] Alan M Turing. Computing machinery and intelligence. In *Parsing the turing test*, pages 23–65. Springer, 2009.

[7] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[8] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[9] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[11] Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, (3):299–307, 1967.

[12] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

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

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

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[19] Felix Musil, Andrea Grisafi, Albert P Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Physics-inspired structural representations for molecules and materials. *Chemical Reviews*, 121(16):9759–9815, 2021.

[20] Bingqing Cheng, Ryan-Rhys Griffiths, Simon Wengert, Christian Kunkel, Tamas Stenczel, Bonan Zhu, Volker L Deringer, Noam Bernstein, Johannes T Margraf, Karsten Reuter, et al. Mapping materials and molecules. *Accounts of Chemical Research*, 53(9):1981–1991, 2020.

[21] Volker L Deringer, Miguel A Caro, and Gábor Csányi. A general-purpose machine-learning force field for bulk and nanostructured phosphorus. *Nature communications*, 11(1):5461, 2020.

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

[23] Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1):13890, 2017.

[24] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

[25] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

[26] John C Slater and George F Koster. Simplified lcao method for the periodic potential problem. *Physical review*, 94(6):1498, 1954.

[27] Marcus Elstner, Dirk Porezag, G Jungnickel, J Elsner, M Haugk, Th Frauenheim, Sandor Suhai, and Gotthard Seifert. Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties. *Physical Review B*, 58(11):7260, 1998.

[28] Th Frauenheim, G Seifert, M Elsterner, Z Hajnal, G Jungnickel, D Porezag, S Suhai, and R Scholz. A self-consistent charge density-functional based tight-binding method for predictive materials simulations in physics, chemistry and biology. *physica status solidi (b)*, 217(1):41–62, 2000.

[29] Qiang Cui, Marcus Elstner, Efthimios Kaxiras, Thomas Frauenheim, and Martin Karplus. A qm/mm implementation of the self-consistent charge density functional tight binding (scc-dftb) method. *The Journal of Physical Chemistry B*, 105(2):569–585, 2001.

[30] Marcus Elstner. The scc-dftb method and its application to biological systems. *Theoretical Chemistry Accounts*, 116:316–325, 2006.

[31] Mathias Rapacioli, Aude Simon, Leo Dontot, and Fernand Spiegelman. Extensions of dftb to investigate molecular complexes and clusters. *physica status solidi (b)*, 249(2):245–258, 2012.

[32] Felix RS Purtscher, Leo Christanell, Moritz Schulte, Stefan Seiwald, Markus Rödl, Isabell Ober, Leah K Maruschka, Hassan Khoder, Heidi A Schwartz, El-Eulmi Bendeif, et al. Structural properties of metal–organic frameworks at elevated thermal conditions via a combined density functional tight binding molecular dynamics (dftb md) approach. *The Journal of Physical Chemistry C*, 2023.

[33] Kiyoshi Yagi, Shingo Ito, and Yuji Sugita. Exploring the minimum-energy pathways and free-energy profiles of enzymatic reactions with qm/mm calculations. *The Journal of Physical Chemistry B*, 125(18):4701–4713, 2021.

[34] Claudia L Gómez-Flores, Denis Maag, Mayukh Kansari, Van-Quan Vuong, Stephan Irle, Frauke Gr´ater, Tomas Kubar, and Marcus Elstner. Accurate free energies for complex condensed-phase reactions using an artificial neural network corrected dftb/mm methodology. *Journal of Chemical Theory and Computation*, 18(2):1213–1226, 2022.

[35] Dirk Porezag, Th Frauenheim, Th Köhler, Gotthard Seifert, and R Kaschner. Construction of tight-binding-like potentials on the basis of density-functional theory: Application to carbon. *Physical Review B*, 51(19):12947, 1995.

[36] Chien-Pin Chou, Yoshifumi Nishimura, Chin-Chai Fan, Grzegorz Mazur, Stephan Irle, and Henryk A Witek. Automatized parameterization of dftb using particle swarm optimization. *Journal of chemical theory and computation*, 12(1):53–64, 2016.

[37] "a software tool intended to automate the optimisation of parameters for the Density Functional Tight Binding (DFTB) theory", https://github.com/dftbplus/skpar.

[38] Mohammad Wahiduzzaman, Augusto F Oliveira, Pier Philipsen, Lyuben Zhechkov, Erik Van Lenthe, Henryk A Witek, and Thomas Heine. Dftb parameters for the periodic table: Part 1, electronic structure. *Journal of chemical theory and computation*, 9(9):4006–4017, 2013.

[39] Stanislav Markov. Skpar. `https://github.com/dftbplus/skpar`, 2018.

[40] Glen R Jenness, Caitlin G Bresnahan, and Manoj K Shukla. Adventures in dftb: Toward an automatic parameterization scheme. *Journal of Chemical Theory and Computation*, 16(11):6894–6903, 2020.

[41] Stanislav Markov, Balint Aradi, Chi-Yung Yam, Hang Xie, Thomas Frauenheim, and Guanhua Chen. Atomic level modeling of extremely thin silicon-on-insulator mosfets including the silicon dioxide: Electronic structure. *IEEE Transactions on Electron Devices*, 62(3):696–704, 2015.

[42] Stanislav Markov, Gabriele Penazzi, YanHo Kwok, Alessandro Pecchia, Bálint Aradi, Thomas Frauenheim, and GuanHua Chen. Permittivity of oxidized ultra-thin silicon films from atomistic simulations. *IEEE Electron Device Letters*, 36(10):1076–1078, 2015.

[43] Zoltán Bodrog, Bálint Aradi, and Thomas Frauenheim. Automated repulsive parametrization for the dftb method. *Journal of chemical theory and computation*, 7(8):2654–2664, 2011.

[44] Maxime Van den Bossche, Henrik Gronbeck, and Bjørk Hammer. Tight-binding approximation-enhanced global optimization. *Journal of chemical theory and computation*, 14(5):2797–2807, 2018.

[45] Akshay Krishna AK, Eddie Wadbro, Christof Köhler, Pavlin Mitev, Peter Broqvist, and Jolla Kullgren. Ccs: A software framework to generate two-body potentials using curvature constrained splines. *Computer Physics Communications*, 258:107602, 2021.

[46] Akshay Krishna Ammothum Kandy, Eddie Wadbro, Bálint Aradi, Peter Broqvist, and Jolla Kullgren. Curvature constrained splines for dftb repulsive potential parametrization. *Journal of Chemical Theory and Computation*, 17(3):1771–1781, 2021.

[47] Nir Goldman, Kyoung E Kweon, Babak Sadigh, Tae Wook Heo, Rebecca K Lindsey, C Huy Pham, Laurence E Fried, Bálint Aradi, Kiel Holliday, Jason R Jeffries, et al. Semi-automated creation of density functional tight binding models through leveraging chebyshev polynomial-based force fields. *Journal of Chemical Theory and Computation*, 17(7):4435–4448, 2021.

[48] Jan M Knaup, Ben Hourahine, and Th Frauenheim. Initial steps toward automating the fitting of dftb e rep (r). *The Journal of Physical Chemistry A*, 111(26):5637–5641, 2007.

[49] Cong Huy Pham, Rebecca K Lindsey, Laurence E Fried, and Nir Goldman. High-accuracy semiempirical quantum models based on a minimal training set. *The Journal of Physical Chemistry Letters*, 13(13):2934–2942, 2022.

[50] Michael Gaus, Albrecht Goez, and Marcus Elstner. Parametrization and benchmark of dftb3 for organic molecules. *Journal of Chemical Theory and Computation*, 9(1):338–354, 2013.

[51] Pavlo O Dral, Xin Wu, Lasse Spórkel, Axel Koslowski, and Walter Thiel. Semiempirical quantum-chemical orthogonalization-corrected methods: Benchmarks for ground-state properties. *Journal of Chemical Theory and Computation*, 12(3):1097–1120, 2016.

[52] Anders S Christensen, Tomas Kubar, Qiang Cui, and Marcus Elstner. Semiempirical quantum mechanical methods for noncovalent interactions for chemical and biochemical applications. *Chemical reviews*, 116(9):5301–5337, 2016.

[53] Pavlo O Dral and Tetiana Zubatiuk. Improving semiempirical quantum mechanical methods with machine learning. In *Quantum Chemistry in the Age of Machine Learning*, pages 559–575. Elsevier, 2023.

[54] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Big data meets quantum chemistry approximations: the δ-machine learning approach. *Journal of chemical theory and computation*, 11(5):2087–2096, 2015.

[55] Albert P Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Science advances*, 3(12):e1701816, 2017.

[56] Xu Chen, Pinyuan Li, Eugen Hruska, and Fang Liu. δ-machine learning for quantum chemistry prediction of solution-phase molecular properties at the ground and excited states. *Physical Chemistry Chemical Physics*, 25(19):13417–13428, 2023.

[57] Guoqing Zhou, Nicholas Lubbers, Kipton Barros, Sergei Tretiak, and Benjamin Nebgen. Deep learning of dynamically responsive chemical hamiltonians with semiempirical quantum mechanics. *Proceedings of the National Academy of Sciences*, 119(27):e2120333119, 2022.

[58] Pavlo O Dral. Quantum chemistry in the age of machine learning. *The journal of physical chemistry letters*, 11(6):2336–2347, 2020.

[59] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[60] Julian J Kranz, Maximilian Kubillus, Raghunathan Ramakrishnan, O Anatole von Lilienfeld, and Marcus Elstner. Generalized density-functional tight-binding repulsive potentials from unsupervised machine learning. *Journal of chemical theory and computation*, 14(5):2341–2352, 2018.

[61] Martin Stöhr, Leonardo Medrano Sandonas, and Alexandre Tkatchenko. Accurate many-body repulsive potentials for density-functional tight binding from deep tensor neural networks. *The Journal of Physical Chemistry Letters*, 11(16):6835–6843, 2020.

[62] Dylan Bissuel, Tristan Albaret, and Thomas A Niehaus. Critical assessment of machine-learned repulsive potentials for the density functional based tight-binding method: A case study for pure silicon. *The Journal of Chemical Physics*, 156(6):064101, 2022.

[63] Tammo van der Heide, Jolla Kullgren, Peter Broqvist, Vladimir Bačić, Thomas Frauenheim, and Bálint Aradi. Fortnet, a software package for training behler-parrinello neural networks. *Computer Physics Communications*, 284:108580, 2023.

[64] Chiara Panosetti, Artur Engelmann, Lydia Nemec, Karsten Reuter, and Johannes T Margraf. Learning to use the force: Fitting repulsive potentials in density-functional tight-binding with gaussian process regression. *Journal of chemical theory and computation*, 16(4):2181–2191, 2020.

[65] Junmian Zhu, Bobby G Sumpter, Stephan Irle, et al. Artificial neural network correction for density-functional tight-binding molecular dynamics simulations. *MRS Communications*, 9(3):867–873, 2019.

[66] Haichen Li, Christopher Collins, Matteus Tanha, Geoffrey J Gordon, and David J Yaron. A density functional tight binding layer for deep learning of chemical hamiltonians. *Journal of chemical theory and computation*, 14(11):5764–5776, 2018.

[67] Vittorio Peano, Florian Sapper, and Florian Marquardt. Rapid exploration of topological band structures using deep learning. *Physical Review X*, 11(2):021052, 2021.

[68] Nikolaj Rørbæk Knøsgaard and Kristian Sommer Thygesen. Representing individual electronic states for machine learning gw band structures of 2d materials. *Nature Communications*, 13(1):468, 2022.

[69] Christoph Schattauer, Milica Todorović, Kunal Ghosh, Patrick Rinke, and Florian Libisch. Machine learning sparse tight-binding parameters for defects. *npj Computational Materials*, 8(1):116, 2022.

[70] Max Born and Robert Oppenheimer. Quantentheorie der molekeln. *Annalen der Physik*, 389(20):457–484, 1927.

[71] Douglas R Hartree. The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24, pages 89–110. Cambridge university press, 1928.

[72] Douglas Rayne Hartree. The wave mechanics of an atom with a non-coulomb central field. part ii. some results and discussion. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24, pages 111–132. Cambridge University Press, 1928.

[73] John C Slater. Note on hartree's method. *Physical Review*, 35(2):210, 1930.

[74] Vladimir Fock. Näherungsmethode zur lösung des quantenmechanischen mehrkörperproblems. *Zeitschrift für Physik*, 61(1):126–148, 1930.

[75] Llewellyn H Thomas. The calculation of atomic fields. In *Mathematical proceedings of the Cambridge philosophical society*, volume 23, pages 542–548. Cambridge University Press, 1927.

[76] Enrico Fermi. Statistical method to determine some properties of atoms. *Rend. Accad. Naz. Lincei*, 6(602-607):5, 1927.

[77] Edward Teller. On the stability of molecules in the thomas-fermi theory. *Reviews of Modern Physics*, 34(4):627, 1962.

[78] Attila Szabo and Neil S Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 2012.

[79] Michael JS Dewar and Walter Thiel. Ground states of molecules. 38. the mndo method. approximations and parameters. *Journal of the American Chemical Society*, 99(15):4899–4907, 1977.

[80] Michael JS Dewar and Donn M Storch. Development and use of quantum molecular models. 75. comparative tests of theoretical procedures for studying chemical reactions. *Journal of the American Chemical Society*, 107(13):3898–3902, 1985.

[81] James JP Stewart. Optimization of parameters for semiempirical methods ii. applications. *Journal of computational chemistry*, 10(2):221–264, 1989.

[82] James JP Stewart. Optimization of parameters for semiempirical methods vi: more modifications to the nddo approximations and re-optimization of parameters. *Journal of molecular modeling*, 19:1–32, 2013.

[83] Walter Kohn. Nobel lecture: Electronic structure of matter—wave functions and density functionals. *Reviews of Modern Physics*, 71(5):1253, 1999.

[84] John P Perdew and Karla Schmidt. Jacob's ladder of density functional approximations for the exchange-correlation energy. In *AIP Conference Proceedings*, volume 577, pages 1–20. American Institute of Physics, 2001.

[85] John P Perdew and Yue Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Physical review B*, 45(23):13244, 1992.

[86] John P Perdew. Accurate density functional for the energy: Real-space cutoff of the gradient expansion for the exchange hole. *Physical Review Letters*, 55(16):1665, 1985.

[87] Chengteh Lee, Weitao Yang, and Robert G Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Physical review B*, 37(2):785, 1988.

[88] John P Perdew, John A Chevary, Sy H Vosko, Koblar A Jackson, Mark R Pederson, Dig J Singh, and Carlos Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Physical review B*, 46(11):6671, 1992.

[89] John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

[90] John P Perdew, Matthias Ernzerhof, and Kieron Burke. Rationale for mixing exact exchange with density functional approximations. *The Journal of chemical physics*, 105(22):9982–9985, 1996.

[91] Axel D Becke. A new mixing of hartree–fock and local density-functional theories. *The Journal of chemical physics*, 98(2):1372–1377, 1993.

[92] Axel D Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Physical review A*, 38(6):3098, 1988.

[93] Seymour H Vosko, Leslie Wilk, and Marwan Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Canadian Journal of physics*, 58(8):1200–1211, 1980.

[94] Jochen Heyd, Gustavo E Scuseria, and Matthias Ernzerhof. Hybrid functionals based on a screened coulomb potential. *The Journal of chemical physics*, 118(18):8207–8215, 2003.

[95] Aliaksandr V Krukau, Oleg A Vydrov, Artur F Izmaylov, and Gustavo E Scuseria. Influence of the exchange screening parameter on the performance of screened hybrid functionals. *The Journal of chemical physics*, 125(22):224106, 2006.

[96] M Elstner. Scc-dftb: what is the proper degree of self-consistency? *The Journal of Physical Chemistry A*, 111(26):5614–5621, 2007.

[97] B Hourahine, S Sanna, B Aradi, C Köhler, Th Niehaus, and Th Frauenheim. Self-interaction and strong correlation in dftb. *The Journal of Physical Chemistry A*, 111(26):5671–5677, 2007.

[98] A Pecchia, G Penazzi, L Salvucci, and A Di Carlo. Non-equilibrium green's functions in density functional tight binding: method and applications. *New Journal of Physics*, 10(6):065022, 2008.

[99] Franco P Bonafé, Bálint Aradi, Ben Hourahine, Carlos R Medrano, Federico J Hernández, Thomas Frauenheim, and Cristián G Sánchez. A real-time time-dependent density functional tight-binding implementation for semiclassical excited state electron–nuclear dynamics and pump–probe spectroscopy simulations. *Journal of Chemical Theory and Computation*, 16(7):4454–4469, 2020.

[100] Jan Rezac. Empirical self-consistent correction for the description of hydrogen bonds in dftb3. *Journal of Chemical Theory and Computation*, 13(10):4804–4817, 2017.

[101] Stefan Grimme, Jens Antony, Stephan Ehrlich, and Helge Krieg. A consistent and accurate ab initio parametrization of density functional dispersion correction (dft-d) for the 94 elements h-pu. *The Journal of chemical physics*, 132(15):154104, 2010.

[102] Eike Caldeweyher, Sebastian Ehlert, Andreas Hansen, Hagen Neugebauer, Sebastian Spicher, Christoph Bannwarth, and Stefan Grimme. A generally applicable atomic-charge dependent london dispersion correction. *The Journal of chemical physics*, 150(15):154122, 2019.

[103] Alberto Ambrosetti, Anthony M Reilly, Robert A DiStasio Jr, and Alexandre Tkatchenko. Long-range correlation energy calculated from coupled atomic response functions. *The Journal of chemical physics*, 140(18):18A508, 2014.

[104] Michael Sternberg. *The atomic structure of diamond surfaces and interfaces*. PhD thesis, Paderborn, Univ., Diss., 2001, 2001.

[105] A Urban, M Reese, M Mrovec, C Elsässer, and B Meyer. Parameterization of tight-binding models from density functional theory calculations. *Physical Review B*, 84(15):155119, 2011.

[106] Guishan Zheng, Henryk A Witek, Petia Bobadova-Parvanova, Stephan Irle, Djamaladdin G Musaev, Rajeev Prabhakar, Keiji Morokuma, Marcus Lundberg, Marcus Elstner, Christof Köhler, et al. Parameter calibration of transition-metal elements for the spin-polarized self-consistent-charge density-functional tight-binding (dftb) method: Sc, ti, fe, co, and ni. *Journal of chemical theory and computation*, 3(4):1349–1367, 2007.

[107] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[108] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

[109] Hanbing Rao, Zerong Li, Xiangyuan Li, Xiaohua Ma, Choongyong Ung, Hu Li, Xianghui Liu, and Yuzong Chen. Identification of small molecule aggregators from large compound libraries by support vector machines. *Journal of computational chemistry*, 31(4):752–763, 2010.

[110] Andreas Keller, Richard C Gerkin, Yuanfang Guan, Amit Dhurandhar, Gabor Turu, Bence Szalai, Joel D Mainland, Yusuke Ihara, Chung Wen Yu, Russ Wolfinger, et al. Predicting human olfactory perception from chemical features of odor molecules. *Science*, 355(6327):820–826, 2017.

[111] Derek T Ahneman, Jesús G Estrada, Shishi Lin, Spencer D Dreher, and Abigail G Doyle. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*, 360(6385):186–190, 2018.

[112] Andrew F Zahrt, Jeremy J Henle, Brennan T Rose, Yang Wang, William T Darrow, and Scott E Denmark. Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning. *Science*, 363(6424):eaau5631, 2019.

[113] Jacques A Esterhuizen, Bryan R Goldsmith, and Suljo Linic. Interpretable machine learning for knowledge generation in heterogeneous catalysis. *Nature Catalysis*, 5(3):175–184, 2022.

[114] Mohammad Javad Eslamibidgoli, Mehrdad Mokhtari, and Michael H Eikerling. Recurrent neural network-based model for accelerated trajectory analysis in aimd simulations. *arXiv preprint arXiv:1909.10124*, 2019.

[115] Zhenwei Li, James R Kermode, and Alessandro De Vita. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Physical review letters*, 114(9):096405, 2015.

[116] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1):011002, 2013.

[117] Stefano Curtarolo, Wahyu Setyawan, Shidong Wang, Junkai Xue, Kesong Yang, Richard H Taylor, Lance J Nelson, Gus LW Hart, Stefano Sanvito, Marco Buongiorno-Nardelli, et al. Aflowlib. org: A distributed materials properties repository from high-throughput ab initio calculations. *Computational Materials Science*, 58:227–235, 2012.

[118] David D Landis, Jens S Hummelshøj, Svetlozar Nestorov, Jeff Greeley, Marcin Dułak, Thomas Bligaard, Jens K Nørskov, and Karsten W Jacobsen. The computational materials repository. *Computing in Science & Engineering*, 14(6):51–57, 2012.

[119] "OMAD Repository and Archive ", https://nomad-lab.eu/services/repo-arch.

[120] "Organic Materials Database ", https://omdb.mathub.io. Accessed: March 2022.

[121] Justin S Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific data*, 7(1):134, 2020.

[122] Johannes Hoja, Leonardo Medrano Sandonas, Brian G Ernst, Alvaro Vazquez-Mayagoitia, Robert A DiStasio Jr, and Alexandre Tkatchenko. Qm7-x, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific data*, 8(1):43, 2021.

[123] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[124] Joel M Bowman, Chen Qu, Riccardo Conte, Apurba Nandi, Paul L Houston, and Qi Yu. The md17 datasets from the perspective of datasets for gas-phase "small" molecule potentials. *The Journal of Chemical Physics*, 156(24):240901, 2022.

[125] Zhihua Zhou. *Machine Learning*.

[126] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[127] Michele Ceriotti, Gareth A Tribello, and Michele Parrinello. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proceedings of the National Academy of Sciences*, 108(32):13023–13028, 2011.

[128] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics*, 134(7):074106, 2011.

[129] Lauri Himanen, Marc OJ Jäger, Eiaki V Morooka, Filippo Federici Canova, Yashasvi S Ranawat, David Z Gao, Patrick Rinke, and Adam S Foster. Dscribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, 2020.

[130] John A Keith, Valentin Vassilev-Galindo, Bingqing Cheng, Stefan Chmiela, Michael Gastegger, Klaus-Robert Müller, and Alexandre Tkatchenko. Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chemical reviews*, 121(16):9816–9872, 2021.

[131] Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.

[132] O Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Exploring chemical compound space with quantum-based machine learning. *Nature Reviews Chemistry*, 4(7):347–358, 2020.

[133] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O Anatole Von Lilienfeld, Klaus-Robert Muller, and Alexandre Tkatchenko. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The journal of physical chemistry letters*, 6(12):2326–2331, 2015.

[134] Felix A Faber, Anders S Christensen, Bing Huang, and O Anatole Von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *The Journal of chemical physics*, 148(24):241717, 2018.

[135] Haoyan Huo and Matthias Rupp. Unified representation of molecules and crystals for machine learning. *arXiv preprint arXiv:1704.06439*, 2017.

[136] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[137] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

[138] Giulio Imbalzano, Andrea Anelli, Daniele Giofré, Sinja Klees, Jörg Behler, and Michele Ceriotti. Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials. *The Journal of chemical physics*, 148(24):241730, 2018.

[139] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[140] Félix Musil, Michael J Willatt, Mikhail A Langovoy, and Michele Ceriotti. Fast and accurate uncertainty estimation in chemical machine learning. *Journal of chemical theory and computation*, 15(2):906–915, 2019.

[141] Anders S Christensen, Lars A Bratholm, Felix A Faber, and O Anatole von Lilienfeld. Fchl revisited: Faster and more accurate quantum machine learning. *The Journal of chemical physics*, 152(4):044107, 2020.

[142] Jigyasa Nigam, Sergey Pozdnyakov, and Michele Ceriotti. Recursive evaluation and iterative contraction of n-body equivariant features. *The Journal of Chemical Physics*, 153(12):121101, 2020.

[143] Aidan P Thompson, Laura P Swiler, Christian R Trott, Stephen M Foiles, and Garritt J Tucker. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics*, 285:316–330, 2015.

[144] Alexander V Shapeev. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation*, 14(3):1153–1173, 2016.

[145] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 99(1):014104, 2019.

[146] Thuong T Nguyen, Eszter Székely, Giulio Imbalzano, Jörg Behler, Gábor Csányi, Michele Ceriotti, Andreas W Götz, and Francesco Paesani. Comparison of permutationally invariant polynomials, neural networks, and gaussian approximation potentials in representing water interactions through many-body expansions. *The Journal of chemical physics*, 148(24):241725, 2018.

[147] Michael Gastegger, Ludwig Schwiedrzik, Marius Bittermann, Florian Berzsenyi, and Philipp Marquetand. wacsf—weighted atom-centered symmetry functions as descriptors in machine learning potentials. *The Journal of chemical physics*, 148(24):241709, 2018.

[148] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[149] Adele Cutler, D Richard Cutler, and John R Stevens. Random forests. *Ensemble machine learning: Methods and applications*, pages 157–175, 2012.

[150] Leo Breiman. Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849, 1998.

[151] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[152] Zhi-Hua Zhou. Ensemble methods. *Combining pattern classifiers. Wiley, Hoboken*, pages 186–229, 2014.

[153] Shishi Dong and Zhexue Huang. A brief theoretical overview of random forests. *Journal of Integration Technology*, 2(1):1–7, 2013.

[154] Oded Z Maimon and Lior Rokach. *Data mining with decision trees: theory and applications*, volume 81. World scientific, 2014.

[155] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature biotechnology*, 26(9):1011–1013, 2008.

[156] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[157] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 479:480, 1969.

[158] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[159] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[160] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[161] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

[162] Kyungho Yoon, Sunyoung Lee, Kyungbae Oh, and Kisuk Kang. Challenges and strategies towards practically feasible solid-state lithium metal batteries. *Advanced Materials*, 34(4):2104666, 2022.

[163] Florian Strauss, Timo Bartsch, Lea de Biasi, A-Young Kim, Jurgen Janek, Pascal Hartmann, and Torsten Brezesinski. Impact of cathode material particle size on the capacity of bulk-type all-solid-state batteries. *ACS Energy Letters*, 3(4):992–996, 2018.

[164] Luhan Ye and Xin Li. A dynamic stability design strategy for lithium metal solid state batteries. *Nature*, 593(7858):218–222, 2021.

[165] Laidong Zhou, Kern-Ho Park, Xiaoqi Sun, Fabien Lalère, Torben Adermann, Pascal Hartmann, and Linda F Nazar. Solvent-engineered design of argyrodite li6ps5x (x= cl, br, i) solid electrolytes with high ionic conductivity. *ACS Energy Letters*, 4(1):265–270, 2018.

[166] Gaozhan Liu, Wei Weng, Zhihua Zhang, Liping Wu, Jing Yang, and Xiayin Yao. Densified li6ps5cl nanorods with high ionic conductivity and improved critical current density for all-solid-state lithium batteries. *Nano Letters*, 20(9):6660–6665, 2020.

[167] Tridip Das, Boris V Merinov, Moon Young Yang, and William A Goddard III. Structural, dynamic, and diffusion properties of a li 6 (ps 4) scl superionic conductor from molecular dynamics simulations; prediction of a dramatically improved conductor. *Journal of Materials Chemistry A*, 10(30):16319–16327, 2022.

[168] Chuang Yu, Swapna Ganapathy, Jart Hageman, Lambert Van Eijck, Ernst RH Van Eck, Long Zhang, Tammo Schwietert, Shibabrata Basak, Erik M Kelder, and Marnix Wagemaker. Facile synthesis toward the optimal structure-conductivity characteristics of the argyrodite li6ps5cl solid-state electrolyte. *ACS applied materials & interfaces*, 10(39):33296–33306, 2018.

[169] Niek JJ De Klerk, Irek Rosłon, and Marnix Wagemaker. Diffusion mechanism of li argyrodite solid electrolytes for li-ion batteries and prediction of optimized halogen doping: the effect of li vacancies, halogens, and halogen disorder. *Chemistry of Materials*, 28(21):7955–7963, 2016.

[170] Tao Cheng, Boris V Merinov, Sergey Morozov, and William A Goddard III. Quantum mechanics reactive dynamics study of solid li-electrode/li6ps5cl-electrolyte interface. *ACS Energy Letters*, 2(6):1454–1459, 2017.

[171] Marvin A Kraft, Sean P Culver, Mario Calderon, Felix B$'$ocher, Thorben Krauskopf, Anatoliy Senyshyn, Christian Dietrich, Alexandra Zevalkink, J$'$urgen Janek, and Wolfgang G Zeier. Influence of lattice polarizability on the ionic conductivity in the lithium superionic argyrodites li6ps5x (x= cl, br, i). *Journal of the American Chemical Society*, 139(31):10909–10918, 2017.

[172] Rayavarapu Prasada Rao, Haomin Chen, and Stefan Adams. Stable lithium ion conducting thiophosphate solid electrolytes li x (ps4) y x z (x= cl, br, i). *Chemistry of Materials*, 31(21):8649–8662, 2019.

[173] Ardeshir Baktash, James C Reid, Tanglaw Roman, and Debra J Searles. Diffusion of lithium ions in lithium-argyrodite solid-state electrolytes. *npj Computational Materials*, 6(1):162, 2020.

[174] Wahyu Setyawan and Stefano Curtarolo. High-throughput electronic band structure calculations: Challenges and tools. *Computational materials science*, 49(2):299–312, 2010.

[175] Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals. *Computer Physics Communications*, 180(11):2175–2196, 2009.

[176] William H Press and Saul A Teukolsky. Vwt, and fbp, numerical recipes: The art of scientific computing, 2007.

[177] Ben Hourahine, Bálint Aradi, Volker Blum, F Bonafé, A Buccheri, Cristopher Camacho, Caterina Cevallos, MY Deshaye, T Dumitrică, A Dominguez, et al. Dftb+, a software package for efficient approximate density functional theory based atomistic simulations. *The Journal of chemical physics*, 152(12):124101, 2020.

[178] Jorge Nocedal and Stephen J Wright. Numerical optimization 2nd edition, 2006.

[179] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[180] Adam McSloy, Guozheng Fan, Wenbo Sun, Christian Hölzer, Marvin Friede, Sebastian Ehlert, Nils-Erik Schütte, Stefan Grimme, Thomas Frauenheim, and Bálint Aradi. Tbmalt, a flexible toolkit for combining tight-binding and machine learning. *The Journal of Chemical Physics*, 158(3):034801, 2023.

[181] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[182] Matthias Seeger, Asmus Hetzel, Zhenwen Dai, Eric Meissner, and Neil D Lawrence. Auto-differentiating linear algebra. *arXiv preprint arXiv:1710.08717*, 2017.

[183] Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang. Differentiable programming tensor networks. *Physical Review X*, 9(3):031041, 2019.

[184] Martin Stöhr, Georg S Michelitsch, John C Tully, Karsten Reuter, and Reinhard J Maurer. Communication: Charge-population based dispersion interactions for molecules and materials. *The Journal of Chemical Physics*, 144(15):151101, 2016.

[185] Alexandre Tkatchenko and Matthias Scheffler. Accurate molecular van der waals interactions from ground-state electron density and free-atom reference data. *Physical review letters*, 102(7):073005, 2009.

[186] Alexandre Tkatchenko, Robert A DiStasio Jr, Roberto Car, and Matthias Scheffler. Accurate and efficient method for many-body van der waals interactions. *Physical review letters*, 108(23):236402, 2012.

[187] Tore Brinck, Jane S Murray, and Peter Politzer. Polarizability and volume. *The Journal of chemical physics*, 98(5):4305–4306, 1993.

[188] Aron J Cohen, Paula Mori-Sánchez, and Weitao Yang. Insights into current limitations of density functional theory. *Science*, 321(5890):792–794, 2008.

[189] Anand Chandrasekaran, Deepak Kamal, Rohit Batra, Chiho Kim, Lihua Chen, and Rampi Ramprasad. Solving the electronic structure problem with machine learning. *npj Computational Materials*, 5(1):22, 2019.

[190] Zifeng Wang, Shizhuo Ye, Hao Wang, Jin He, Qijun Huang, and Sheng Chang. Machine learning method for tight-binding hamiltonian parameterization from ab-initio band structure. *npj Computational Materials*, 7(1):11, 2021.

[191] R Patrick Xian, Vincent Stimper, Marios Zacharias, Maciej Dendzik, Shuo Dong, Samuel Beaulieu, Bernhard Schölkopf, Martin Wolf, Laurenz Rettig, Christian Carbogno, et al. A machine learning route between band mapping and band structure. *Nature Computational Science*, pages 1–14, 2022.

[192] Evgenii Tsymbalov, Zhe Shi, Ming Dao, Subra Suresh, Ju Li, and Alexander Shapeev. Machine learning for deep elastic strain engineering of semiconductor electronic band structure and effective mass. *npj Computational Materials*, 7(1):76, 2021.

[193] E Rauls, J Elsner, R Gutierrez, and Th Frauenheim. Stoichiometric and non-stoichiometric (1010) and (1120) surfaces in 2h–sic: A theoretical study. *Solid state communications*, 111(8):459–464, 1999.

[194] Shūichi Nosé. A molecular dynamics method for simulations in the canonical ensemble. *Molecular physics*, 52(2):255–268, 1984.

[195] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics*, 81(1):511–519, 1984.

[196] William G Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Physical review A*, 31(3):1695, 1985.

[197] Hongchao Yang, Yandong Ma, and Ying Dai. Progress of structural and electronic properties of diamond: a mini review. *Functional Diamond*, 1(1):150–159, 2022.

[198] Adam McSloy, **Guozheng Fan**, Wenbo Sun, Christian Hölzer, Marvin Friede, Sebastian Ehlert, Nils-Erik Schütte, Stefan Grimme, Thomas Frauenheim, and Bálint Aradi. Tbmalt, a flexible toolkit for combining tight-binding and machine learning. *The Journal of Chemical Physics*, 158(3):034801, 2023.

[199] **Guozheng Fan**, Adam McSloy, Bálint Aradi, Chi-Yung Yam, and Thomas Frauenheim. Obtaining electronic properties of molecules through combining density functional tight binding with machine learning. *The Journal of Physical Chemistry Letters*, 13(43):10132–10139, 2022.

[200] Machine learning-based parameterization of density functional tight-binding for band structures in bulk, slab, and defect systems. 2023. In preparation.

[201] Wenbo Sun, **Guozheng Fan**, Tammo van der Heide, Adam McSloy, Thomas Frauenheim, and Bálint Aradi. Machine learning enhanced dftb method for periodic systems: learning from electronic density of states. *Journal of Chemical Theory and Computation*, 19(13):3877–3888, 2023.

[202] Hongwei Fu, **Guozheng Fan**, Jiang Zhou, Xinzhi Yu, Xuesong Xie, Jue Wang, Bingan Lu, and Shuquan Liang. Facilitating phase evolution for a high-energy-efficiency, low-cost O3-type $Na_xCu_{0.18}Fe_{0.3}Mn_{0.52}O_2$ sodium ion battery cathode. *Inorganic Chemistry*, 59(18):13792–13800, 2020.

[203] Yong Chen, Yuanming Zhang, **Guozheng Fan**, Lizhu Song, Gan Jia, Huiting Huang, Shuxin Ouyang, Jinhua Ye, Zhaosheng Li, and Zhigang Zou. Cooperative catalysis coupling photo-/photothermal effect to drive sabatier reaction with unprecedented conversion and selectivity. *Joule*, 5(12):3235–3251, 2021.