

Machine learning for spectroscopic properties of organic molecules

Annika Stuke



Machine learning for spectroscopic properties of organic molecules

Annika Stuke

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, remote connection Zoom link <https://aalto.zoom.us/j/68194827869>, on 14th August 2020 at 16:00.

Aalto University
School of Science
Department of Applied Physics
Computational Electronic Structure Theory (CEST)

Supervising professor

Professor Patrick Rinke, Aalto University, Finland

Thesis advisor

Doctor Milica Todorović, Aalto University, Finland

Preliminary examiners

Professor Flyura Djurabekova, University of Helsinki, Finland

Professor Bjørk Hammer, Aarhus University, Denmark

Opponent

Professor Rampi Ramprasad, Georgia Institute of Technology, USA

Aalto University publication series

DOCTORAL DISSERTATIONS 108/2020

© 2020 Annika Stuke

ISBN 978-952-60-3966-4 (printed)

ISBN 978-952-60-3967-1 (pdf)

ISSN 1799-4934 (printed)

ISSN 1799-4942 (pdf)

<http://urn.fi/URN:ISBN:978-952-60-3967-1>

Unigrafia Oy

Helsinki 2020

Finland



Author

Annika Stuke

Name of the doctoral dissertation

Machine learning for spectroscopic properties of organic molecules

Publisher School of Science

Unit Department of Applied Physics

Series Aalto University publication series DOCTORAL DISSERTATIONS 108/2020

Field of research Engineering Physics

Manuscript submitted 2 April 2020

Date of the defence 14 August 2020

Permission for public defence granted (date) 5 June 2020

Language English

☐ **Monograph**

☒ **Article dissertation**

☐ **Essay dissertation**

Abstract

The efficient design of new and advanced materials is hindered by a shortfall of suitable methods to rapidly and accurately identify potential materials that meet a desired application. Conventional approaches involve either expensive and time-consuming experiments or computations that often require significant human input. The materials design process could be greatly expedited by utilizing artificial intelligence (AI) tools that are capable of learning effectively from known historic or intentionally generated data that is already available for millions of chemical compounds.

In this dissertation, we develop and apply machine learning (ML) – a subcategory of AI - to infer spectral properties from molecular datasets. Once trained, our ML models predict molecular spectra and spectral properties instantly and at negligible computational cost. We find that the ML algorithms need to be trained on large and diverse datasets to ensure robustness and predictive accuracy. However, publicly available molecular datasets with realistic structures and spectral properties are rare. Therefore, we generated our own structurally diverse benchmark spectroscopy dataset of 62k large organic molecules. We computed electronic geometries at different levels of density functional theory (DFT) for all 62k molecules as well as quasiparticle orbital eigenvalues at high numerical accuracy with the GoWo approach for a subset of 5k molecules. A particular difficulty that is often overlooked in current ML applications are model parameters that cannot be learned directly during training, so called hyperparameters. We solve this challenge by applying Bayesian optimization to automatically tune the hyperparameters of our kernel ridge regression (KRR) model with two different descriptors for the molecular structure, one of which introduces its own set of hyperparameters to the method. Furthermore, we study how the performance of our KRR model varies for molecular datasets of different chemical diversity. We find that the learning success of molecular orbital energies inherently depends on the structural complexity of individual molecules as well as on the diversity within a dataset. Our findings benchmark the accuracy of orbital energy predictions with KRR for publicly available molecular datasets, two of which are lesser-known than the widely used QM9 chemical dataset of small molecules. Finally, we employ deep neural network models to predict molecular excitation spectra with up to 97 % accuracy. The results of this dissertation facilitate instant spectra prediction with machine learning for large molecular databases and pave the way for high throughput screening of materials to find new materials with advanced functionality.

Keywords Machine learning, Spectroscopy, Molecular Datasets, Density Functional Theory, Chemical Space, Materials Design

ISBN (printed) 978-952-60-3966-4

ISBN (pdf) 978-952-60-3967-1

ISSN (printed) 1799-4934

ISSN (pdf) 1799-4942

Location of publisher Helsinki

Location of printing Helsinki **Year** 2020

Pages 153

urn <http://urn.fi/URN:ISBN:978-952-60-3967-1>

Preface

My sincere gratitude goes to all of those who made developing my ideas in this thesis possible and who brought joy to my everyday life. I wholeheartedly thank my supervisor, Patrick Rinke, for leading me through the vast field of scientific research and for supporting me throughout all stages of this journey. I cannot thank you enough for your invaluable advice, encouragement and for creating a social, engaging and supportive atmosphere in the group. My deepest thanks to Milica Todorović, who was always there to help, advise, challenge and inspire me. Thank you for your constant guidance and countless spot-on comments. My genuine gratitude extends to Lei Xie, who welcomed me at Hunter College and introduced me to computational chemistry methods during my visit in 2017. My warmest thanks to Bjørk Hammer from Aarhus University and to Flyura Djurabekova from University of Helsinki, who made the effort to pre-examine my dissertation. Your feedback helped me improve the quality of this thesis. I am especially grateful to Rampi Ramprasad from Georgia Tech for acting as my opponent in the public defense of my dissertation. My profound appreciation goes to my other co-authors – Kunal Ghosh, Dorothea Golze, Lauri Himanen, Aki Vehtari, Christian Kunkel, Johannes Margraf, Karsten Reuter, Harald Oberhofer, Peter Jørgensen, Mikkel Schmidt and Matthias Rupp. The publications included in this thesis are the result of your hard work, profitable inputs and rewarding discussions. You all taught me a lot and it was my great pleasure working with you.

Many wonderful people made my lunch breaks at Aalto and my days in Finland special. I am deeply grateful to all my LIT friends and colleagues who enlightened my PhD life. Veera! We may have suffered through some unlit times, but together we made PhD life lit again! Rina, Ms President, I am thankful to be your first lady. I can always count on your Russian honesty and your schwifty party mode. Marc, thank you for creating value with your jaw-dropping ideas and your ugly dancing skills. Special thanks to your mother! Saani, your karaoke performances and bike repair services are highly recommended! Thank you for many deep conversations (potato potahto) and for your department coffee. Yash, thank you for being

Yash. Ya boiii! The LIT group is devoted to you, thanks to your litness. Sugam, thank you for being proper rap support and competition at karaoke. Matthew, thank you for your research visit to the LIT group. I am sure you learned much. Azeema, my heavy co-drinker, thank you for entertaining our coffee-breaks with stories about the GOD and DEV. To the Italian mafia – Fabio, Riccardo, Filippo – thank you for your great taste in fashion and your (justified) food complaints. I still have to perform diss rap for you guys. No worries. I also thank all other members of the CEST and SIN groups for providing such a fun working environment.

My Servinkuja alumni – Alligator Sanja, thank you for welcoming me to Finland and for introducing me to the Computer Science community. I would have been lost without you during my first weeks here. Heeryung, my No. 1 In Touch reader and shopping advisor, thank you for moving into our living room. Mareike, thank you for many great memories throughout all this time and for sharing delicious wine and drama-rama stories from Austria. Zainab, I will miss our past-midnight life and work analytics. Kata, I'm glad you stayed at our flat during your visit in Finland to show us who is the true Servinkuja party queen! Marius, thank you for regular beer deliveries from Lithuania and for your bad movie taste. Gopika & Rajat, thank you for all the BBQ evenings in Otaniemi and for upgrading my kitchen with your plants. Thank you, Maria, for your contagious laugh and our lit Hanko experience. Klaudia, thank you for sweaty dance battles and Polish drinking nights. Thanks also to my new neighbour Sid, and to Mohit(o) for co-suffering through the Easter mass. The Fat Lizard people – Antuané, Omar, Chang, Anastasia, Matthias – I am grateful for many good memories we share. I would like to acknowledge the Aalto Inn fire alarm for reliably sounding for no reason. That's how people meet in Finland. Sonja, thank you for keeping me company during my first days in Finland.

My beloved companions from the Spasstenverein – Norah and Aurélie – thank you for being the most awesome WG in Berlin, for challenging my IQ with many intellectual conversations (14 variables) and for staying true Spassten even when living apart. You always stood by me and supported me during tough times. Hegdl! I am devoted to my old-but-gold friends from the Öi-Club and all the memories we share. Katja, Julia, Alexandra, Ann-Kathrin, Alexa – I am glad to have you as friends since day 1.

In the end, heartfelt thanks go to my family (also the new one) for always believing in me, no matter what goals I pursue. Thank you for your endless support and encouragement. Yunes, I am glad to have met you.

Espoo, July 15, 2020,

Annika Stuke

Contents

Preface	i
Contents	iii
List of Publications	v
Author's Contribution	vii
Abbreviations	ix
1. Introduction	1
1.1 Chemical space exploration and data-driven materials science	1
1.2 Research objective	2
1.3 Research approach and questions	4
1.4 Thesis structure	6
2. Photoemission spectroscopy	7
2.1 Experimental photoemission spectroscopy	7
2.2 Computational photoemission spectroscopy	9
2.2.1 Density functional theory	9
2.2.2 The <i>GW</i> method	13
2.2.3 Numerical representation of spectra	14
2.3 Summary	15
3. Molecular datasets	17
3.1 Data-driven materials science and existing datasets . . .	17
3.2 Generating molecular datasets	24
3.3 Numerical representation of molecular structures	27
3.4 Summary	30
4. Machine learning approach	33
4.1 Machine learning principle	33

4.2	Kernel ridge regression	36
4.2.1	KRR for HOMO energy prediction	40
4.3	Artificial neural networks	41
4.3.1	ANNs for photoemission spectra prediction	44
4.4	Hyperparameter optimization	47
4.4.1	Grid search	48
4.4.2	Bayesian optimization	49
4.4.3	Comparison of Bayesian optimization and grid search for KRR hyperparameter tuning	51
4.5	Summary	57
5.	Machine learning application	59
5.1	Prediction of molecular orbital energies with kernel ridge regression	59
5.2	Prediction of photoemission spectra with deep neural net- works	64
5.3	KRR and ANN predictions for materials discovery	67
5.4	Summary	70
6.	Summary and Outlook	71
6.1	Summary	71
6.2	Outlook	74
	References	75
	Appendices	87
	Publications	91

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

- I** Annika Stuke, Christian Kunkel, Dorothea Golze, Milica Todorović, Johannes T. Margraf, Karsten Reuter, Patrick Rinke and Harald Oberhofer. Atomic structures and orbital energies of 61,489 crystal-forming organic molecules. *Scientific Data*, 7, 58, February 2020.
- II** Annika Stuke, Patrick Rinke and Milica Todorović. Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization. Submitted to *The Journal of Chemical Physics*, March 2020.
- III** Annika Stuke, Milica Todorović, Matthias Rupp, Christian Kunkel, Kunal Ghosh, Lauri Himanen and Patrick Rinke. Chemical diversity in molecular orbital energy predictions with kernel ridge regression. *The Journal of Chemical Physics*, 150, 204121, April 2019.
- IV** Kunal Ghosh, Annika Stuke, Milica Todorović, Peter B. Jørgensen, Mikkel N. Schmidt, Aki Vehtari and Patrick Rinke. Deep Learning Spectroscopy: Neural Networks for Molecular Excitation Spectra. *Advanced Science*, 6, 1801367, January 2019.

Author's Contribution

Publication I: "Atomic structures and orbital energies of 61,489 crystal-forming organic molecules"

The author curated the data and carried out calculations at the DFT- and G_0W_0 -level of theory. The author postprocessed and validated the calculations and co-wrote the manuscript.

Publication II: "Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization"

The author developed the ML models and the algorithms for hyperparameter optimization. The author performed all simulations and wrote the manuscript.

Publication III: "Chemical diversity in molecular orbital energy predictions with kernel ridge regression"

The author carried out calculations at the DFT-level of theory for 134k small organic molecules. The author postprocessed and analyzed the data, developed the ML models, performed the simulations and wrote the manuscript.

Publication IV: "Deep Learning Spectroscopy: Neural Networks for Molecular Excitation Spectra"

The author carried out calculations at the DFT-level of theory for 134k small organic molecules that serve as training data for the deep neural

Author's Contribution

network models. The author postprocessed the DFT calculations to provide broadened spectral lines for the continuous learning. The author contributed to the analysis of the neural network models and the results.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
BO	Bayesian Optimization
CM	Coulomb Matrix
CNN	Convolutional Neural Network
DFT	Density Functional Theory
DTNN	Deep Tensor Neural Network
GP	Gaussian Processes
GPR	Gaussian Process Regression
HOMO	Highest Occupied Molecular Orbital
KRR	Kernel Ridge Regression
KS	Kohn-Sham
LUMO	Lowest Unoccupied Molecular Orbital
MBTR	Many-Body Tensor Representation
MAE	Mean Absolute Error
MO	Molecular Orbital
PES	Photoelectron/Photoemission Spectroscopy
QM	Quantum Mechanics
RMSE	Root Mean Squared Error
RSE	Relative Spectral Error

1. Introduction

1.1 Chemical space exploration and data-driven materials science

The concept of chemical space is widely used in materials design and drug discovery, for example to design chemical libraries, to classify or select chemical compounds or to explore structure-property relationships. However, a precise and unique definition of chemical space is not simple, and an even more challenging task is the navigation through this largely unknown space. In the most widely accepted definition, chemical space is the set of all possible stable compounds that are distributed according to their structures and properties [1, 2]. This includes an inconceivable number of possibilities. Even when only considering small organic molecules with fewer than 30 atoms, there are not enough atoms in the universe (estimated to be 10^{78} to 10^{82}) to synthesize a single molecule for each of the 10^{60} possible atomic permutations [3, 4]. Exploring such a vast space is a formidable task, especially considering that the largest current public database of molecules that have already been synthesized, PubChem, contains only around 70 million molecules [5].

Materials discovery aims to design new materials of industrial relevance to address technological challenges, such as creating improved solar cell materials to collect energy from the sun, new battery materials for energy storage or quantum materials for novel forms of computing. One of the major limitations in materials discovery is the need to identify relevant molecules or materials from the vast chemical space of possible targets. How would we decide which molecules to synthesize for a given task or application? How would we find the ones that feature the best properties? Nobody knows what exceptional materials might hide in chemical space, only waiting to be discovered. It is very well possible that some hidden treasures will never be found – or that we are already stumbling through the most fruitful area of chemical space.

Conventional forms of materials discovery and chemical space explo-

ration are experimentation and computation. Advances in experimental techniques have continuously increased the resolution, volume and application domain of measurements. However, experiments are often restricted to examine only one compound at a time and therefore not suitable to explore chemical space on large scale. Theoretical methods based on modern quantum-mechanical (QM) first principles can be used to complement experiments. Computer resources have progressed to a level where materials properties can be calculated with reasonable accuracy for many properties. However, the computational cost increases rapidly with system size, and accuracy requirements limit the number of tractable compounds once again. Hence, chemical space exploration with current experimental or theoretical approaches remains selective and restricted.

Past advances in experimental and computational methods created an abundance of molecular and materials data. The availability of more and more data has given rise to data-driven materials science as a new scientific paradigm [6, 7]. Outcomes from many experiments and simulations are now stored in large materials repositories, such as NOMAD [8], the Materials Project [9, 10], the Cambridge Structural Database [11], the Cambridge Crystallographic Data Centre (CCDC) [12] or the Materials Cloud [13]. The view on data has changed significantly over the last decade in the natural sciences. Data are now regarded as a resource instead of only a by- or endproduct of simulations or experiments. Many of the readily available data from experiments or simulations are correlated. For instance, QM calculations that are run with the same method for various similar compounds will yield outputs with repetitive information. Hence, instead of running another QM calculation, it makes sense to apply statistical tools that can discover relationships in the available outputs, thus profiting from the redundancy in the data. Especially machine learning – a subfield of artificial intelligence (AI) – has seen a steep rise of popularity in materials science. Machine learning utilizes statistical tools and algorithms that are able to learn from data [14]. Applied to numerous chemical systems, they can predict the outcomes of computationally demanding electronic structure calculations. Plenty of excellent review articles are available in the literatures that demonstrate the importance of machine learning in materials science [15–24]. The trend of utilizing and exploiting publicly available materials data from experiments and simulations allows an accelerated exploration of chemical space and constitutes the base for the study described in this dissertation.

1.2 Research objective

The objective of this dissertation is to implement machine learning models that accelerate scientific discovery and help us with chemical space ex-

ploration. Specifically, I am interested in models that can predict spectra and spectroscopic properties of organic molecules based on their chemical structure.

Spectroscopy – the interaction between matter and radiation – is of fundamental importance to the natural sciences and one of the dominant experimental techniques to characterize materials [25]. The response of matter to radiation (e.g. electro-magnetic, sound, particle beams) reveals different kinds of characteristic spectroscopic properties and spectra that are relevant for various technological applications. For example, vibrational spectra can be used to find new thermo-electrics for waste heat recovery, X-ray spectra to discover new medical diagnostic materials or conductivity spectra to manufacture new batteries with high storage capacity. Photoemission spectra expose the distribution of electronic energy levels in a material and play an essential role in the development of new optoelectronic devices. While semiconductors used in solid-state electronics are prevalently based on inorganic materials – such as gallium, germanium or silicon – organic semiconductors have recently emerged as a new class of electronic and optoelectronic materials [26–32]. Organic semiconductors are solids whose building blocks are small molecules or conjugated polymers made up mainly of carbon and hydrogen atoms [33]. They can be employed in various applications, such as organic photovoltaic devices (OPVs), organic light-emitting diodes (OLEDs) and organic field-effect transistors (OFETs). The advantage of organic semiconductors over inorganic semiconductors lies in their low cost, light weight, mechanical flexibility, processability and the ability to tune their properties. Organic materials therefore provide the possibility to realize novel applications such as flexible displays, biological sensors, wearable electronics and solar cells. While significant technological developments have been achieved in the field of organic optoelectronics, there is a high demand for new organic solids with advanced properties before the large-scale production of organic electronic devices becomes possible. To improve, for example, the device performance of OLEDs, which are progressively adopted as a standard display technology for TV and phone screens, new materials need to be found or created that exhibit a specific band gap. The band gap of a material defines the wavelength of the light that the material is able to emit and is given by the energy difference between the HOMO (highest occupied molecular orbital) and LUMO (lowest unoccupied molecular orbital). For this aim, it is desirable to know the HOMO and LUMO energies for a large number of possible materials, so that one can pick the best suited material for a specific OLED application.

In regard of this demand for new organic compounds in optoelectronic applications, I will develop machine learning models that can predict photoemission spectra and frontier molecular orbital energies of organic molecules. The ability to produce instant energy and spectra predictions

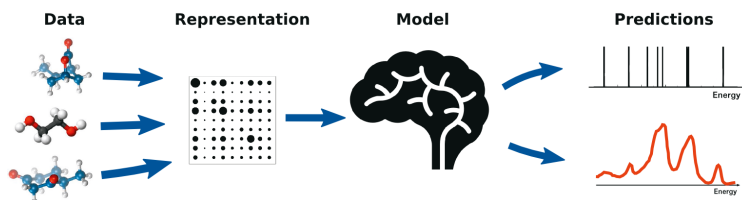


Figure 1.1. Schematic diagram of the workflow described in this thesis. The machine learns from mapping reference molecular structures to their spectral properties. Once the training is finished, the model is able to make out-of-sample predictions for unseen molecules.

with machine learning will be useful to bypass conventional laborious approaches in the materials design process. One could predict photoemission spectra with machine learning for a wide range of molecules at low computational cost in order to narrow down the number of candidates tried in laboratory tests. Experimental photoemission spectra would have to be measured only for a small number of compounds.

The machine learning models developed in this dissertation have the potential to systematically change and facilitate the chemical space exploration and materials discovery process. Instead of experimentally mixing chemicals to see how they behave, scientists and manufacturers will be able to instantly screen through a database of materials with their predicted spectra to find those that exactly show a desired spectral behavior, before these materials are even fabricated. Molecular structures could be identified in a matter of seconds based on their predicted spectra, revealing which one makes, e.g., a good converter of solar energy to electricity. Materials modeling will be brought closer to industrial and societal exploitation.

1.3 Research approach and questions

To achieve the objective of this thesis, I will follow the schematic procedure sketched in Figure 1.1 throughout this dissertation. It shows the four basic components of my machine learning workflow: Training data, a data representation, the machine learning algorithm and a prediction target. For every component in this workflow, I formulate a research question that will be addressed in one or more of the publications throughout this thesis.

Training data The foundation of any machine learning model lies in trustworthy data of high quality. Therefore, the first step in the development of my machine learning model is to assemble my own reference datasets and to find appropriate and useful molecular structures that are relevant for optoelectronic applications. There are several molecular datasets available that are well known in the machine learning community. But which one is suitable for our needs, if any? Moreover, the data should be of high quality,

consistent and reliable, that is, computed with the same computational method. Therefore, once I have chosen what kind of molecules to use for model training, I will assemble my own datasets of reference spectra and orbital energies, computed with first-principle QM methods. The output of these QM reference computations will serve as input to my machine learning model.

This leads to the first research question:

RQ1 What kind of datasets should I use to train my machine learning model and how can I obtain reliable reference data of high quality?

Data representation Before I can input the computed reference data into the model, I have to think about how to make them accessible to the machine. That is, the 3D structures of the molecules and their corresponding spectra need to be represented appropriately in a mathematical form such that the model can process them. This is not a trivial problem, and a lot of research has been put into finding an adequate molecular representation. Therefore, I will try different representations in this dissertation. I ask the following research question:

RQ2 How can molecules and their spectra be numerically represented in such a way that the machine can establish a relationship between each molecule and its spectra?

Machine learning algorithm Another fundamental question is which machine learning method to use for the problem at hand. The choice of method to generate the best spectra predictions is not always obvious due to the large variety of available machine learning techniques. Moreover, most machine learning models involve hyperparameters, which are parameters that can usually not be learned by the model itself, but instead must be specified separately before training. This is often a burden for machine learning practitioners, requiring intuition, expert knowledge or large computing resources for brute-force search over a wide range of possible values. Once the right method has been chosen and the hyperparameters have been set, I can train the model on my generated reference data. The following research question arises:

RQ3 Which machine learning methods should I choose and how can I determine the hyperparameters of my models?

Predictions When the training is finished, the machine learning model needs to be validated on a set of test molecules that were not used for training to see how well the model can predict spectral properties. I will assess model performance based on predefined quality metrics and will compare

various algorithms. Then, I need to determine whether the predictions of the model are good enough with respect to the reference data. Finally, when the model delivers predictions of satisfying quality, it can be used to produce fast and accurate approximations of spectra and orbital energies for unseen molecules that have not been used in the training or validation phase of the model. This yields the following questions:

RQ4 How can machine learning methods be applied in practice to predict molecular orbital energies and photoemission spectra and to explore chemical space?

1.4 Thesis structure

The dissertation is structured as follows. Chapter 2 dives into the subject of photoemission spectra and describes how they are traditionally obtained experimentally and computationally. Chapter 3 reviews currently available molecular datasets that can be used for developing and benchmarking machine learning models in material science. I will discuss Publication I, which provides a new benchmark spectroscopy dataset of high numerical accuracy. Chapter 4 explains the general concept of machine learning and introduces three methods: kernel ridge regression, artificial neural networks and Bayesian optimization based on Gaussian processes. I then discuss Publication II, which deals with the tuning of hyperparameters in machine learning models, employing Bayesian optimization. Chapter 5 discusses the results obtained in Publications III and IV. In Publication III, kernel ridge regression is applied to three different molecular datasets to predict HOMO energies. In Publication IV, neural networks are used to predict photoemission spectra for small organic molecules. The results are put into context with other state-of-the-art machine learning models in materials science. A general discussion and concluding remarks are presented in Chapter 6. The publications are presented after Chapter 6.

2. Photoemission spectroscopy

Having learnt about the importance of photoemission spectra and orbital energies for the design of new functional materials, I will now review experimental and computational techniques that can determine these quantities. One central spectroscopic technique is photoelectron spectroscopy (PES), also known as photoemission spectroscopy, which determines the energies and shapes of electronic states in atoms and molecules. This chapter explains what photoemission spectra are, how they are obtained experimentally and computationally and how I aim to predict them with machine learning. At the end of this chapter, I will be able to partly answer the second research question,

RQ2 How can molecules and their spectra be numerically represented in such a way that the machine can establish a relationship between each molecule and its spectra?

2.1 Experimental photoemission spectroscopy

Experimental PES measures the energy of electrons emitted from a substance by the photoelectric effect in order to disclose the binding energies of electrons in the material. Thereby, the bonding in molecules or the elemental composition of materials can be studied. In PES, a sample is exposed to visible or ultraviolet light (UPS) or to X-rays (XPS) in synchrotron facilities, which causes electrons to be ejected from their bound states within the sample, as sketched in Figure 2.1a). An energy analyzer records the kinetic energies of the ejected electrons, and then a detector counts the number of photoelectrons at various kinetic energies.

The energy that is required to emit an electron from a substance is the electron's ionization potential (IP), also known as the energy required to remove an electron from a bound state ϵ_s that lies below the Fermi level ϵ_F . By monitoring the kinetic energy E_{kin} of the ejected photoelectron, the

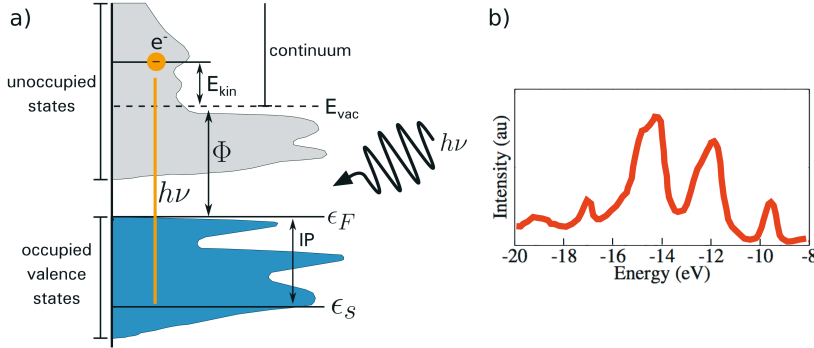


Figure 2.1. a) Schematic principle of the photoemission process. An incoming photon of energy $h\nu$ excites an electron from an occupied valence state ϵ_s into the continuum. Adapted from [25]. b) Example illustration of a photoemission spectrum.

IP can be reconstructed as

$$IP = -\epsilon_s = h\nu - E_{kin} - \Phi, \quad (2.1)$$

where $h\nu$ is the energy of the incoming photon and Φ is the work function, which is specific to the sample. The ionization potential of an electron in an atom or molecule depends on its location relative to the nucleus. Valence electrons have the lowest ionization potential since they are on average farther away from the nucleus and more shielded by other electrons. Core electrons are closer to the nucleus and less shielded, therefore exhibiting higher ionization potentials.

Figure 2.1b) shows an example valence spectrum resulting from a photoemission process. The photoelectron count is plotted as a function of energy. The spectrum shows several peaks at different energies, which correspond to electronic states in the sample. The peaks represent the energy required to remove an electron from its state, and the intensity represents the number of electrons in that state. Thus, photoemission spectra reflect the electronic structure of energy levels within a sample. PES is one of the most accurate and sensitive methods for determining the energies and shapes of electronic states, atomic orbitals and molecular orbitals. Every element and molecule is characterized by its own photoemission spectrum. By analyzing the distribution and intensities of the peaks in a photoemission spectrum of unknown origin, one can identify the elemental composition of the sample. However, the experimental implementation is expensive and time consuming and therefore, it is not feasible to produce experimental photoemission spectra for a high number of new substances.

2.2 Computational photoemission spectroscopy

Computer simulations are a fundamental part of scientific research, allowing researchers to complement experimental measurements. The growth in available computing power and the developments of new methods are responsible for the increasing impact of computational methods in materials science and their practical applications to real systems. Simulations can serve as a means of confirming already existing experimental results, or as a means of guiding and predicting future experimental outcomes.

The most commonly used computational methods to calculate the electronic and atomic structure of matter are first-principles quantum mechanical approaches, such as wave function based methods (Hartree-Fock, Quantum Monte Carlo), Green’s function based methods (*GW* [25, 34]) or density based methods, such as density functional theory (DFT) and time-dependent density functional theory (TDDFT). In this thesis, *GW* and DFT are employed to generate large sets of reference molecular orbital energies that can later be used for model training. *GW* is now the standard approach to calculate ionization energies of molecules and solids as measured in photoemission experiments. However, it is computationally intensive and it is not feasible to compute molecular orbital energies for a large amount (e.g. tens of thousands) of molecules. Since DFT is computationally cheaper, it is the dominant method used in this thesis to generate reference spectra and orbital energies for machine learning. DFT is a strict ground-state theory and thus describes photoemission spectra only approximately. However, approximate spectra are sufficient to build my methodology, and enable me to use large volumes of training data that can be produced at manageable cost. The methodology can later be refined with more accurate *GW* spectra, if necessary.

In the following, I first describe the DFT framework, followed by the *GW* approach. Then, I explain how discrete orbital energy values resulting from DFT calculations are turned into approximate photoemission spectra, and how these spectra will numerically be represented to the machine for learning.

2.2.1 Density functional theory

Density functional theory (DFT) has become a standard tool in computational chemistry, physics and materials science due to its favorable computational scaling. DFT is derived from the N -particle Schrödinger equation and is completely formulated in terms of the ground state density $n_0(r)$. It is applicable to nuclei, atoms, molecules and solids, reducing the computation of ground-state properties of systems of interacting particles to the solution of single-particle equations. The fact that DFT is applicable to relatively large systems of several hundreds of atoms explains the

success of this method.

DFT is usually formulated in the Born-Oppenheimer approximation [35], which postulates that nuclear positions can be considered as fixed, since the nuclear motion is of many orders slower than the motion of electrons. The motion of electrons is then governed by a fixed external potential V_{ext} , which is the Coulomb potential imposed by the fixed nuclei. This approximation greatly simplifies the solution of the Schrödinger equation $\hat{H}\Psi = E\Psi$, where Ψ is the molecular wavefunction, \hat{H} is the Hamiltonian operator and E is a proportionality constant corresponding to the total energy of the system.

In 1964, Hohenberg and Kohn [35, 36] derived density-functional theory, in which the electron density $n(\mathbf{r})$ replaces the wavefunction Ψ as the central quantity in the Schrödinger equation. The electron density $n(\mathbf{r})$ can be interpreted as the probability of detecting an electron at the position \mathbf{r} . For a system of interacting electrons that move in an external potential V_{ext} , Hohenberg and Kohn showed that the external potential V_{ext} and the ground-state wavefunction Ψ are uniquely determined by the ground-state electron density $n(\mathbf{r})$. With that, all other observables of the system, such as kinetic energy or electronic properties, are uniquely determined by the ground-state electron density as well.

Kohn-Sham DFT In 1965, Kohn and Sham replaced the system of interacting electrons by a fictitious system of non-interacting electrons that move within a local effective Kohn-Sham (KS) single-particle potential V_{eff} [35–37]. The KS method is still exact, as it produces the same ground-state density as the interacting system, but largely facilitates the solution of the Schrödinger equation. The energy of the whole system is given as a unique functional of the electronic density $n(\mathbf{r})$ [37]:

$$E[n(\mathbf{r})] = T[n(\mathbf{r})] + E_{el}[n(\mathbf{r})] + E_{Ha} + E_{xc}[n(\mathbf{r})], \quad (2.2)$$

where \mathbf{r} denotes the position of electrons T is the kinetic energy of the non-interacting electrons, E_{el} is the external potential energy acting on the interacting system – usually resulting from interactions with nuclei – E_{Ha} is the Hartree energy (the electrostatic interaction energy between electrons) and E_{xc} is the non-classical exchange-correlation energy. $E_{xc}[n(\mathbf{r})]$ incorporates all exchange and correlation effects that are not captured by the Hartree term and, moreover, the difference between the kinetic energy of the non-interacting system and the fully interacting system. The effective KS potential is defined as

$$V_{eff}(\mathbf{r}) = V_{ext}(\mathbf{r}) + e^2 \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + v_{xc}(\mathbf{r}), \quad (2.3)$$

where the last term is the exchange-correlation potential given by

$$v_{xc}(\mathbf{r}) = \frac{\delta E_{xc}[n(\mathbf{r})]}{\delta n(\mathbf{r})}. \quad (2.4)$$

This leads to the central equations in Kohn-Sham DFT, that is, the set of one-electron Schrödinger equations:

$$\left(-\frac{\hbar^2}{2m} \nabla^2 + V_{\text{eff}}(\mathbf{r}) \right) \phi_i(\mathbf{r}) = \epsilon_i \phi_i(\mathbf{r}). \quad (2.5)$$

Here, $\phi_i(\mathbf{r})$ are the KS one-electron orbitals with corresponding KS eigenvalue ϵ_i . The electron density for the N electron system is defined as

$$n(\mathbf{r}) = \sum_{i=1}^N |\phi_i(\mathbf{r})|^2. \quad (2.6)$$

Since the effective potential $V_{\text{eff}}(\mathbf{r})$ relies on the electron density $n(\mathbf{r})$, the KS equations (2.5) are solved self-consistently. In practice, $V_{\text{eff}}(\mathbf{r})$ is constructed from an initial guess of the ground-state density $n(\mathbf{r})$, and Eq. (2.5) is then solved for the KS orbitals $\phi_i(\mathbf{r})$. With these orbitals, the ground-state density is updated and the cycle repeats until convergence is reached. The final ground-state energy is obtained from Eq. (2.2) using the final electron density. If each term in the KS energy functional in Equation (2.2) was known, one could obtain the exact ground-state density and total energy. However, the exact form of the exchange-correlation functional $E_{xc}[n(\mathbf{r})]$ is not exactly known and therefore has to be approximated.

Exchange-correlation functionals There is an almost infinite number of approximate exchange-correlation functionals with different levels of complexity. The exchange part in the exchange-correlation functional emerges from the Pauli principle, which states that two electrons cannot take the same quantum-mechanical state. The correlation term includes screening effects of electrons, which move collectively in order to reduce the net interaction among any pair of electrons [37].

The most widely used approximation is the *local-density approximation* (LDA), which treats the electron density locally as a homogeneous electron gas. LDA assumes that the exchange-correlation energy at each location in the system is the same as that of a uniform electron gas of equal density. Thus, the functional only depends on the density at the point where the functional is computed. Most local parameterizations suit the description of the uniform electron gas, for which the exchange-correlation energy is known accurately. Such LDA functionals are a good approximation when the densities are close to that of a uniform electron gas, as is the case for simple metals.

Generalized gradient approximations (GGA) goes beyond the LDA by incorporating the gradient of the electron density. The most common

functional of this type is PBE (Perdew-Burke-Ernzerhof) [38], where well-defined limits are used to calculate the values of certain parameters in the functional. PBE is generally more accurate for ground-state problems than LDA, but strongly underestimates orbital energies and energy gaps. This is due to the delocalization error of local and semi-local DFT functionals and the absence of the derivative discontinuity. Delocalization errors occur for systems with fractional charges, for which GGA is not able to properly interpolate between the electronic energy and fractional charges, which leads to an artificial tendency towards the delocalization of electrons. These limitations can be remedied by including a fraction of the exact exchange. Such functionals are called *hybrid functionals*. The exact exchange is given by the Hartree-Fock expression, which is computationally much more expensive than evaluating purely density or gradient dependent exchange-correlation expressions. An additional challenge is to choose the right fraction of exact exchange. Hybrid functionals such as PBEh have been shown to better describe materials properties than LDAs and GGAs [37].

Limitations of DFT DFT is strictly a ground-state theory, although in principle, any excited state property that could be expressed as a functional of the ground-state density would also be accessible by DFT. However, in practice, no known (approximate) functionals exists that map the ground state density to excitation energies or other excited-state properties. Thus, excited states and their properties are practically unreachable by DFT.

One exception is the highest occupied molecular orbital (HOMO). According to the IP-theorem [39–41], the KS eigenvalue corresponding to the highest occupied molecular orbital (HOMO) equals the exact first vertical ionization potential. For all other (occupied and unoccupied) orbitals, no such theorem exists, i.e. the KS eigenvalues are not equal to ionization potentials. However, while ground-state properties and the HOMO can in principle be computed exactly with DFT, the approximate exchange-correlation functionals introduce errors, such as incorrect description of the long-range behavior of the electron density [42] and self-interaction [43]. In the latter case, DFT predicts a non-physical self-interaction energy for a system of just one electron. Semi-local functionals such as PBE typically underestimate the ionization potential of the HOMO. Hybrid functionals typically do better for orbital energies, but their success hinges on finding a suitable fraction of exact exchange.

Despite the above mentioned limitations, KS eigenvalues from DFT have been used quite extensively in the literature to complement photoemission spectroscopy and to gain useful insight from the comparison of experimental measurements with DFT calculations [44].

2.2.2 The GW method

The GW approximation for the computation of quasiparticle energies was first proposed by Hedin in 1965 [45]. However, it was not before the mid-eighties that this method gained track in large-scale electronic structure computations. Nowadays GW has become the standard method to compute electronic structure properties related to photoemission spectroscopy, such as molecular excitations and band structures.

Excited states and quasi-particle energies are calculated based on many-body perturbation theory (MBPT). The quasi-particle energy of an excitation corresponds to the energy required to remove or add an electron to a many-body system. In the DFT framework, the response of a system of interacting electrons to an external potential V_{ext} is replaced by the response of a system of non-interacting electrons to an "effective" potential. A similar idea assumes that the long-range and fairly strong Coulomb forces screen single electrons with a charge cloud of the other surrounding electrons. These electrons with their surrounding screening cloud are known as quasi-particles. The response of interacting electrons can be expressed in terms of weakly interacting quasi-particles.

The mathematical formulation of quasi-particles is founded on the single-particle Green's function $G(\mathbf{r}, t, \mathbf{r}', t')$, also known as propagator. It represents the probability amplitude for the propagation of an electron from position \mathbf{r}' at time t' to the position \mathbf{r} at time t . The exact computation of the Green's function requires the complete knowledge of the quasi-particle self-energy Σ . The self-energy holds all quantum-mechanical exchange and correlation interactions of the hole created in an excitation process and its surrounding electrons. It can be approximated using a perturbative expansion with respect to the quasi-particle interaction. A working scheme for the quantitative calculation of excitation energies is the "dynamically screened interaction", or GW approximation.

In practice, GW is carried out within first-order perturbation theory (G_0W_0) and often starts from DFT single-particle orbitals ϕ_n and corresponding eigenvalues ϵ_n [25,34]. For a molecular orbital ϕ_n , the corrections to the DFT-KS orbital energies ϵ_n are given by

$$\epsilon_n^{G_0W_0} = \epsilon_n + \text{Re} \left\langle \phi_n \left| \Sigma(\epsilon_n^{G_0W_0}) - v^{\text{XC}} \right| \phi_n \right\rangle, \quad (2.7)$$

where $\epsilon_n^{G_0W_0}$ are the G_0W_0 quasiparticle energies, and v^{XC} is the exchange potential from DFT. The self-energy Σ is the product of the noninteracting KS Green's function G_0 and the screened Coulomb interaction W_0

$$\Sigma(\mathbf{r}, \mathbf{r}', \omega) = \frac{1}{2\pi} \int d\omega' e^{i\omega'\eta} G_0(\mathbf{r}, \mathbf{r}', \omega + \omega') W_0(\mathbf{r}, \mathbf{r}', \omega') \quad (2.8)$$

with $\eta > 0$. The self-energy is usually split into a correlation part Σ^C and an exchange part Σ^X . The correlation part Σ^C is computed from

$W_0^C(\mathbf{r}, \mathbf{r}', \omega) = W_0(\mathbf{r}, \mathbf{r}', \omega) - v(\mathbf{r}, \mathbf{r}')$, where $v(\mathbf{r}, \mathbf{r}')$ is the Coulomb interaction. The exchange part Σ^X is the Hartree-Fock exact exchange self-energy. The mean-field Green's function is given by

$$G_0(\mathbf{r}, \mathbf{r}', \omega) = \sum_m \frac{\phi_m(\mathbf{r})\phi_m(\mathbf{r}')}{\omega - \epsilon_m - i\eta \operatorname{sgn}(\epsilon_F - \epsilon_m)}, \quad (2.9)$$

where ϵ_F is the Fermi energy. The sum involves all occupied and unoccupied KS orbitals ϕ_m with the corresponding KS orbitals ϵ_m . The screened Coulomb interaction in the random phase approximation (RPA) is defined as

$$W_0(\mathbf{r}, \mathbf{r}', \omega) = \int d\mathbf{r}'' \epsilon^{-1}(\mathbf{r}, \mathbf{r}'', \omega) v(\mathbf{r}'', \mathbf{r}'), \quad (2.10)$$

where ϵ is the dielectric function.

2.2.3 Numerical representation of spectra

Despite the limitations in DFT mentioned in Section 2.2.1, I will use DFT-KS energies within this dissertation to approximate molecular ionization energies, and with that, photoemission spectra. Compared to *GW*, DFT is cheaper and thus it will be more convenient to compute reference datasets containing large numbers of molecules. My focus in this thesis lies on the development of machine learning models for spectra and energy predictions. The methodology can always be refined with more accurate data, for example with the dataset produced Publication I, which consists of 5k molecules and their orbital energies computed with *GW*.

In order to predict spectra, the discrete KS energies need to be transformed into a continuous curve, so that the model is able to map each molecular structure to its corresponding photoemission spectrum. Within this thesis, two types of spectral representations are employed:

- i. Discrete energy spectrum made up of a defined number of KS energies from the molecular valence energy region (starting from the HOMO). The task of the machine learning model is to predict all KS energies simultaneously.
- ii. Approximate photoemission spectrum resulting from broadening the discrete KS energies within a pre-defined energy range. The continuous curve is discretized into 300 points, and the task of the machine learning model is to predict all 300 points simultaneously.

Figure 2.2 shows how the discrete energy spectrum of KS eigenvalues is transformed into an approximate photoemission spectra. Each eigenvalue that lies within a pre-defined valence energy region (in this example from -30 to 0 eV) is broadened with a Gaussian distribution. These distributions are then added together to form a continuous curve.

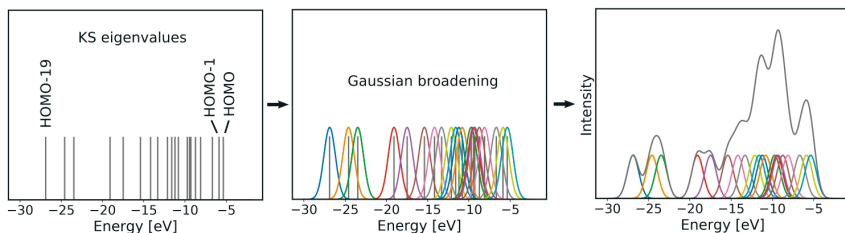


Figure 2.2. Numerical representation of photoemission spectra: For each molecule, the DFT-computed KS eigenvalues are broadened into Gaussian distributions and then transformed into a continuous spectrum by adding the distributions.

2.3 Summary

Photoemission spectroscopy (PES) determines the energies of electrons in atoms and molecules. In experimental PES, a sample is ionized by exposing it to high-energy radiation. The kinetic energies of the ejected electrons are measured, converted to binding energies and plotted as a spectrum. The energy locations of the peaks in the spectrum correspond to the electronic energy states in the atom or molecule. Computational photoemission spectra can be obtained using first-principles quantum mechanical methods derived from the Schrödinger equation, ideally excited-state theories such as TDDFT or *GW*. In this thesis, I will use approximate photoemission spectra computed with ground-state DFT serving as reference spectra to train my machine learning models. Although DFT-KS energies have limited accuracy, it is convenient to assemble large and consistent molecular datasets with DFT to develop my methodology. While not yet used within my machine learning model framework, I also generated a spectroscopy benchmark dataset containing molecular orbital energies of 5k compounds computed with the more accurate G_0W_0 approach. This and other public molecular benchmark datasets will be discussed in the following chapter.

This chapter gives a partial answer to the second research question,

RQ2 How can molecules and their spectra be numerically represented in such a way that the machine can establish a relationship between each molecule and its spectra?

I will represent spectra by Gaussian-broadening the DFT-KS energies into a continuous curve that resembles a photoemission spectrum. The curve is discretized into 300 points, and the task for the machine will be to predict these 300 points simultaneously. The second part of **RQ2**, how to represent molecular structures, will be addressed in the next chapter.

3. Molecular datasets

In the previous chapter, experimental and quantum mechanical computational methods were presented that can determine the electronic structure and properties of molecules. In this chapter, I will now present molecular databases that contain structures and properties resulting from such methods. Given the relatively high cost of computing spectra, only a limited number of available molecular datasets provide spectral properties useful for my work. In this chapter, I will assess a dataset's value for my work based on 7 criteria. Then, I will elaborate how I generated my own molecular datasets using DFT and *GW*, in order to provide consistent reference data to my machine learning model. In particular, I processed and standardized two already existing datasets and created a new spectroscopy dataset of 62k diverse molecules computed with accurate state-of-the-art first principles computations. This new dataset was published in Publication I, which is summarized within this chapter. Previously, we learned how the target (molecular spectra and energies) will be numerically represented to the machine. At the end of this chapter, I will explain how molecular structures can be turned into machine-readable and exploitable input. The first two research questions will be addressed,

RQ1 What kind of datasets should I use to train my machine learning model and how can I obtain reliable reference data of high quality?

RQ2 How can molecules and their spectra be numerically represented in such a way that the machine can establish a relationship between each molecule and its spectra?

3.1 Data-driven materials science and existing datasets

With ever growing amounts of data that is openly shared from experiments and computations in materials science, researchers have begun to change their attitude towards data. Data is now frequently regarded as a resource

itself, and scientists aim to extract knowledge from materials datasets. This paradigm shift of effectively using existing data to obtain desired results is often referred to as data-driven materials science [7]. However, the technological developments in data-driven materials science are still far from reaching the proportions in e-commerce or social media, where infrastructures, algorithms and analysis tools are available for large-scale data. Especially in spectroscopy, data generation is slow and expensive. Data are often high-dimensional and difficult to store or share, and there tools to utilize large amounts of spectroscopic data do not yet exist.

However, for machine learning applications it is crucial to train and validate on large quantities of high quality data. The quality of the resulting models and their predictions will only be as good as that of the data used for training. Hence, datasets should be chosen carefully. In this thesis, I select molecular datasets for my machine learning framework according to the following criteria:

- i. **Relevance** The molecules in the dataset should be relevant for technological applications, such as photovoltaic devices or OLEDs. In particular, I am looking for datasets of organic molecules or polymers. Preferably, the molecules should exhibit great structural diversity.
- ii. **Properties** All relevant information about the molecules should be available in the dataset. This includes molecular 3D structures and photoemission spectra or molecular orbital energies.
- iii. **Availability** The dataset should be publicly available or at least freely available.
- iv. **Size** The dataset should be large in size, i.e. contain several thousands of molecules. The prediction error of a machine learning model typically decreases with more available training data. Therefore, it is important to have large volumes of training data from which the learning algorithm can draw relationships. A large dataset size furthermore ensures that a variety of different structures is included, i.e. the dataset is likely to be diverse.
- v. **Diversity** Formally, diversity within a dataset is characterized by the distribution of data points. If the deviation of data points from their collective average is low, then all data points most likely have similar values. High deviations infer that the points in the set are far from the mean and from each other, and the dataset is called diverse. Similarity in datasets can be the result of a biased data collection procedure by humans and by systematic factors, resulting in a distribution mismatch between dataset and reality.

To broaden and enrich the correlations made by my machine learning model during training, I am looking for datasets that are as diverse

as possible. The datasets should contain many different molecular structures, different bonding patterns and a variety of different elements. This ensures that the training data represent the full range of cases that the model is likely to confront in real-world applications. When working with diverse datasets, it is especially important to have large data sizes so that the model sees enough examples of different structures throughout training.

- vi. **Accuracy and consistency** Accuracy and consistency of a dataset can be described by its bias and variance. Bias describes systematic errors in experimental or computational data resulting from, e.g. incorrectly calibrated experimental equipment, inaccuracies in the measuring devices, errors in the computation software or the use of different numerical methods or settings. This leads to a systematic offset from the true result. Variance describes random errors in the data. Random errors are unavoidable and emerge from the uncertainty inherent in the measuring or computation process, or from the variation in the quantity that is being measured. This causes a probabilistic deviation of the data from the true results which is likely clustered around the true value. The variance in computational data is usually insignificant since there is little unpredictability or uncertainty in the simulation process. However, the bias of computational data can be significant, that is, the offset from experimental results or from more accurate computational methods. For experimental data, a classification into bias and variance is not always possible. Systematic and random errors are both the consequence of insufficient adjustment of the measured system, lack of resolution and unpredictable interactions between the sample and the environment.

When utilizing different datasets for machine learning, it is important that the data are compatible within and across datasets, that is, low in bias. Data generated with different numerical settings, different computational methods (e.g. GW or DFT or different exchange-correlation functionals), different experimental methods or different instruments can lead to systematic deviations in the computed or measured spectra, which can negatively affect the predictive accuracy of my machine learning models.

- vii. **Format** When working with multiple datasets from different sources, the data need to be presented in a consistent format to the machine learning algorithm. Machine learning algorithms can only provide valid results and predictions if the data is correctly formatted. Published data resulting from experiments or computations are available in a variety of different forms and formats, often in a form that only suits a particular study. Moreover, computational or experimental data is in many cases not published in its entirety, however, metadata

or results that seem unimportant for a particular study often prove valuable for another one.

Equipped with a list of criteria that a suitable machine learning dataset should fulfill, we can now take a look at several specific materials datasets that might be of use for my work. In the following, I present computational datasets that are publicly available and discuss their usefulness with respect to my machine learning framework.

GDB A convenient source for obtaining large amounts of simple molecular structures is the Generated DataBase (GDB) universe, a set of databases that exhaustively enumerates parts of organic chemical space. GDB lists billions of combinatorial possible molecules that have not been chemically synthesized so far, exceeding by far the number of known molecules of similar size. The subset GDB-17 [46] includes 166.4 billion small organic molecules with up to 17 heavy atoms made of C, N, O, S and halogens (the term 'heavy' atoms refers to non-hydrogen atoms). To date, GDB-17 is the largest publicly available database of small organic molecules. Other databases of this project are GDB-11 [47, 48] of 26.4 million structures with up to 11 heavy atoms of C, N, O and F, and GDB13 [49] of 977 million structures with up to 13 heavy atoms of C, N, O, S and Cl. One drawback of the GDB databases is that they contain only molecular structures, but no properties. Several studies therefore computed their own reference properties with QM methods for subsets of the GDB-13 and GDB-17 datasets, listed in the following.

QM9 The QM9 dataset [50] is a subset of GDB-17 and contains 133,885 molecules with up to 9 heavy atoms of C, N, O and F. It includes optimized geometries and 18 different molecular properties calculated in DFT using the B3LYP functional. This dataset is known as the golden standard for machine learning studies in materials science and is used to benchmark, for instance, the prediction of atomization energies [51, 52] or a wide variety of other molecular properties [53].

I choose this dataset for the early-stage development of my machine learning framework. QM9 is one of the standard datasets for machine learning studies in materials science and hence it will be easy to compare my work to other studies. Moreover, its size provides decent training ranges; more than many other datasets offer. QM9 contains small organic molecules, with simple and similar structures that will be relatively fast and easy to learn. The QM9 dataset contains 3D structure information and DFT-computed frontier orbital energies, but no other orbital energies of deeper states. Therefore, I computed the HOMO energies and other molecular orbital energies employing my own DFT calculations, as described in the next section. Another problem with QM9 is the lack of diversity. QM9 is made of small molecules of only H, C, N, O and F, and thus, there is not

a big variety of different molecular structures. Considering the large size of the dataset, the structures only differ slightly from each other. The high degree of redundancy within QM9 will produce machine learning models that are not necessarily generalizable to molecules with more complex structures. However, QM9 will constitute a good starting point to build a simple machine learning "prototype" model. At a later stage, when the model works as expected, other datasets containing larger molecules with technological relevance can be considered.

QM7/QM7b The QM7 dataset [54] is a subset of GBD-13, consisting of 7,165 molecules with up to 7 heavy atoms of elements C, N, O or S. It includes relaxed structures and atomization energies calculated with the Perdew-Burke-Ernzerhoff hybrid functional (PBE0). Many machine learning models for predicting the atomization energy with the QM7 dataset have been studied using KRR [51, 54–57]. The QM7b dataset [58] extends the QM7 dataset to molecules containing Cl and includes, in addition to the atomization energy, 13 properties including spectroscopically relevant HOMO and LUMO energies, excitation energies and polarizabilities, calculated at numerous levels of theory (ZINDO, SCS, PBE0, GW). This vast range of properties has facilitated past development of multi-output regression and neural network models that are able to predict various properties simultaneously [55, 57, 58].

The QM7(b) dataset is also used within this thesis for the same reasons elaborated above for QM9 (except for its size). QM7(b) provides easy and free access to structures of small molecules that are widely used for benchmarking in other studies. Containing 7k molecules, QM7(b) is considerably smaller in size than QM9, and also includes molecules that are on average smaller (molecules with up to 7 heavy atoms in QM7(b) vs. molecules with up to 9 heavy atoms in QM9). The QM7(b) dataset includes molecules with sulfur, which is not present in QM9, and thus adds to the overall element diversity.

QM8 The QM8 dataset is a subset of GDB-17 and the result of recent work [46, 59]. It contains the lowest two vertical electronic excited states computed with TDDFT and second-order approximate coupled-cluster (CC2) for 20k small organic molecules with up to 9 heavy of C, O, N or F.

I will not consider this dataset further within my thesis. While this dataset comprises high-quality results for excited states, I am not interested in predicting electronic transition energies and oscillator strengths with machine learning, but photoemission spectra.

AA The AA dataset [60] contains 45,892 conformers of amino acids and dipeptides with DFT-relaxed structures (PBE+vdW, light), total energies and KS eigenvalues (PBE+vdW, tier2 bases, tight). Dipeptides are flexible chains of amino acids linked by amide bonds and may fold and assemble into organized 3D-structures. Many peptides are biologically active

(hormones, antibiotics, toxins). The AA dataset has been used in several studies to develop machine learning models [61–63].

This is the second dataset chosen for my machine learning framework. The dataset is large in size and more diverse than QM9 and QM7(b). The compounds are on average larger and show more variety in chemical structures. The AA dataset contains 3D structures and orbital energies computed with the same DFT functional I chose for the other datasets in my study, thus providing data without introducing new bias. No further computation is needed for this dataset in order to be usable for my work. I can use this dataset after my first prototype machine learning model is validated on the small organic molecules datasets from GDB.

10k diastereomers of C₇H₁₀O₂ constitutional isomers This dataset [64] comprises structures of 9,868 diastereomers isomers of parent C₇H₁₀O₂ isomers, which were relaxed at the B3LYP level of DFT. The original 6,095 isomers are part of the QM9 dataset.

This dataset is used in Publications III and IV to showcase the applicability of my machine learning model. After training the model on structures from other datasets, it was applied to the 10k previously unseen diastereomers to predict spectra and orbital energies.

Clean Energy Project Database The Clean Energy Project Database (CEPD) [65] contains molecular structures and DFT calculations of 2.3 million molecules and polymers resulting from the Harvard Clean Energy Project [66]. It represents the most extensive first-principles quantum chemical investigation ever conducted. The library was built using a combinatorial molecule generator based on around 10k molecular motifs that are of potential interest for small molecule organic photovoltaic applications.

The CEPD fulfills many of the criteria for a suitable dataset, especially with regard to size and diversity. However, during my Doctoral studies I was generated my own dataset (OE62) with diverse structures from the Cambridge Structural Database (CSD). While there was not the need for me to utilize the CEPD to develop my machine learning models, it surely would have been a valuable alternative.

Harvard organic photovoltaic dataset The Harvard organic photovoltaic dataset [67] contains 350 small molecules and polymers that were used as *p*-type materials in organic photovoltaic devices in the literature. For each compound atomic coordinates, experimental properties and corresponding quantum-mechanical calculations including HOMO and LUMO energies are provided.

The molecules in this dataset and their properties are suitable for my machine learning objective, however, with only 350 molecules this dataset is not large enough for my purposes.

Dataset of chemical shifts Paruzzo et. al developed a machine learning framework to predict nuclear magnetic resonance (NMR) chemical shifts in

solids [68]. For model training a reference dataset of DFT-calculated chemical shifts was produced for organic structures taken from the Cambridge Structural Database (CSD), which is available on Materials Cloud [69]. The dataset contains relaxed structures and chemical shifts calculated for 2,500 molecular solids made of H, C, N and O.

With 2.5k structures, this dataset is comparably small and includes only four different elements. Moreover, I am not interested in predicting NMR spectra, but photoemission spectra. Therefore, I will use the OE62 dataset within this thesis instead, which was generated in the scope of Publication I. It is also derived from the CSD, but considerably larger in size (62k structures), exhibits a higher element diversity, and includes molecular orbital energies that can be used to predict photoemission spectra.

Multi-fidelity bandgap database This dataset contains bandgaps of 599 double perovskite halides (or elpasolites), computed with DFT at two fidelity levels of DFT [70]. For all 599 compounds, the crystal structures were relaxed and bandgaps were computed using the PBE functional (low-fidelity dataset). For a subset of 250 compounds, bandgaps were calculated employing the hybrid Heyd-Scuseria-Ernzerhof (HSE06) exchange-correlation functional (high-fidelity dataset), starting from DFT-PBE relaxed structures.

Similar to the previous dataset, this dataset is too small in size to be of use for my machine learning objective. Moreover, I am interested in molecules, not solids.

There are plenty of other datasets for different materials and electronic properties. For example, the Automatic Flow for Materials Discovery (AFLOW) [71] database and the Materials Project [9, 10] provide large datasets for solids, computed with DFT. AFLOW includes more than 3 million materials that were structurally relaxed with the same DFT functional and numerical settings for all calculations to keep consistency. Utilizing these databases for my purposes would require the extraction of suitable molecules from the crystal structures of these databases, along with their corresponding spectral properties. This is a laborious task, and it is easier to use pre-existing datasets of molecular structures and to compute, if necessary, missing electronic structures with DFT or other methods.

Materials data from experiments is available only to a limited extent. Obtaining precise and accurate results from measurements requires specialized instruments, human supervision and time. As a result, there is only a finite number of experimental datasets available, and these are typically much smaller than the computational datasets described above. One example is a compilation of 370 high entropy alloys (HEAs) and complex concentrated alloys (CCAs) that were published in the literature between 2004 and 2016 [72]. One of the few larger experimental databases is the High Throughput Experimental Materials (HTEM) database [73] that consists of 140k inorganic thin film materials.

The reason for the limited availability of experimental data is that these are often generated by informal processes adapted based on a specific research question. The resulting data depends on unique experimental designs and specialized instruments, which makes it difficult to assess and compare the quality of data resulting from different sources. Another problem is that there are generally no accepted standards on how to store data from experiments. Data is often saved in various formats that are not consistent with forms from other sources. Every institution has the freedom to choose to what extent the research data is saved and made publicly available, and meta-data is often provided incompletely. Experimental data are especially prone to systematic and random errors that emerged during measurement due to lack of control over the measured system, inaccurately calibrated equipment, problems with the measuring device or the variation of quantities over time.

Computational materials data, on the contrary, are currently more freely available in larger numbers and sizes. The resulting data are less prone to variance and easy to standardize. Within this thesis, I will therefore only use computational materials data to develop my machine learning models for spectra and energy prediction.

3.2 Generating molecular datasets

The computational datasets chosen in the previous section – QM9, QM7(b) and AA – fulfill most of the listed criteria for a suitable machine learning dataset. However, the properties in these three datasets are computed in DFT with different exchange-correlation functionals, and for that reason are not yet consistent with each other. Moreover, for QM9 and QM7(b), not all orbital energies are available. QM9 and QM7(b) contain energies only for the frontier orbitals, computed with the B3LYP exchange-correlation functional in references [54, 58], while AA provides all orbital energies computed with PBE+vdW. In order to establish consistency among the three datasets, I performed DFT computations for QM9 and QM7(b) with the PBE+vdW functional (tight settings, tier2 basis sets) in FHI-aims [74]. In particular, I optimized the structures towards their energy minimum (structure relaxation) and computed the KS eigenvalues serving as orbital energies. More computational details about the FHI-aims code are described in the Appendix. For QM9, 71 out of 133,885 structure relaxations did not converge, and these molecules were excluded from my reference dataset. The number of molecules in the QM9 dataset is so large that 71 fewer reference molecules is not a big loss. The remaining 133,814 molecules constitute my reference QM9 dataset. For QM7, 239 calculations out of 7102 calculations failed to converge during structural optimization and were discarded. The remaining 6,926 molecules constitute my QM7

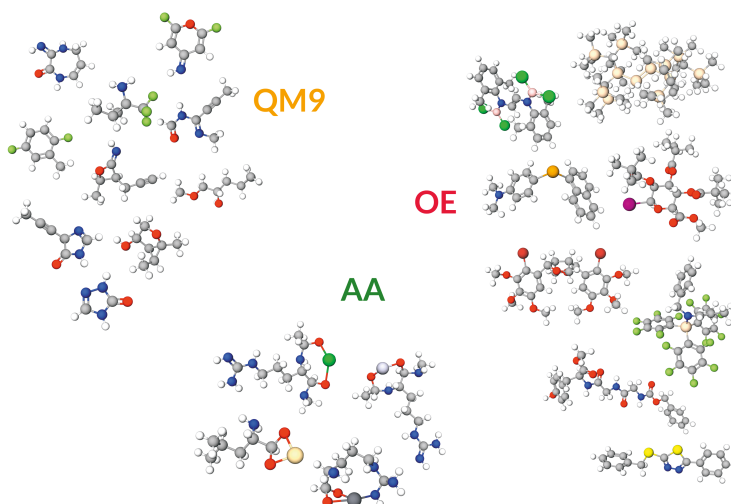


Figure 3.1. The three datasets employed in this dissertation. QM9 contains small organic molecules with simple bonding patterns, while more complex structures are found in AA and OE62. The amino acids and dipeptides in AA share a common backbone, but vary in the sidechains and dihedral angles. OE62 contains a diverse collection of large organic molecules that exhibit conjugated and complex aromatic backbones.

reference dataset. QM7 and QM9, together with the AA dataset, will be used to train and validate my machine learning models.

The large size of the dataset and the small size of the molecules that it contains makes QM9 an ideal dataset for the early stage development of machine learning models. However, the small molecular structures do not represent molecules that are usually used in optoelectronic applications. As shown in Figure 3.1, the bonding patterns of the molecules in QM9 are relatively simple. For realistic machine learning applications, one would wish to train models on a more diverse set of molecules that contains larger and more complex structures. However, as mentioned in the previous section, publicly available datasets that provide reliable spectroscopic properties of technologically relevant molecules are rare.

OE62 In the scope of Publication I, I participated in the creation of a new, structurally diverse benchmark spectroscopy dataset of 62k large molecules, referred to as the OE62 dataset. This dataset is based on a diverse collection of organic crystals taken from the Cambridge Structural Database (CSD) [75]. For each crystal, the molecular structure that makes up the organic crystal is extracted. The obtained molecular geometries were then relaxed in vacuum at the PBE+vdW level of DFT for all 62k molecules. Moreover, total energies and orbital eigenvalues were computed at the PBE and PBE0 levels of theory. In addition, total energies and orbital energies were computed in a water solvent at the PBE0 level of theory, for a subset of 31k molecules. Finally, for a subset of 5k molecules in vacuum,

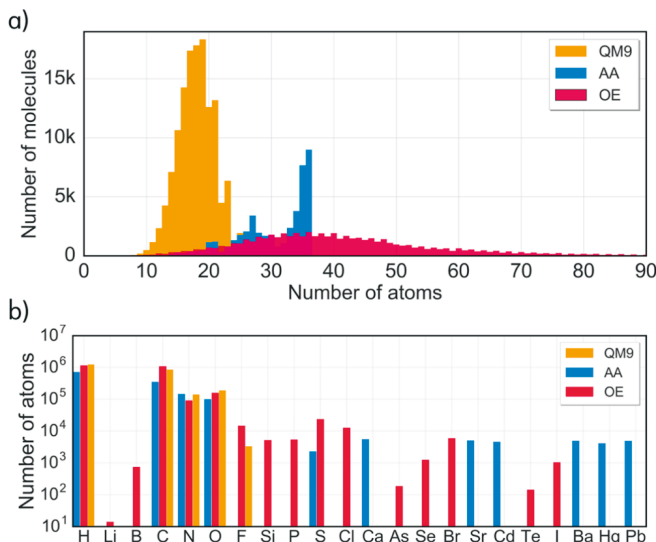


Figure 3.2. Comparison of the three molecular datasets used in this dissertation. a) Distribution of molecular size including H-atoms. b) Distribution of element types. Reproduced from Publication III.

quasiparticle energies at the G_0W_0 level were calculated. This latter subset constitutes a rare and valuable benchmark dataset of molecular orbital energies computed at high numerical accuracy.

The new OE62 dataset provides molecular structures and quantum chemical properties of technologically relevant molecules. Figure 3.2 shows distributions of molecular size and element types for OE62 in comparison with QM9 and AA. While QM9 and AA both contain similar sized molecules, OE62 exhibits a much broader distribution in molecular size, with molecules of up to 174 atoms. Moreover, OE62 includes a large range of different organic elements, while QM9 and AA are restricted to a smaller set of elements. Figure 3.1 further illustrates that OE62 spans a diverse combinatorial space of scaffold-functional group pairings. For instance, OE62 contains molecules with conjugated and aromatic backbones and offers many structures with different functional groups. This degree of chemical diversity cannot be found QM9 or AA. Detailed illustrations of the scaffold diversity in OE62 can be found in [76, 77].

Figure 3.3 shows the spread of 3k randomly picked molecules from each of the QM9, AA and OE62 datasets, produced with the dimensionality reduction technique t-SNE [78]. In a), molecular structures are represented by the Coulomb matrix (CM) and in b), molecules are represented by the many-body tensor representation (MBTR). These two molecular descriptors are introduced in the following section. In both cases a) and b), molecules of OE62 are widely spread out, while QM9 and AA molecules form clusters that are restricted in chemical space. Figure 3.3 again illus-

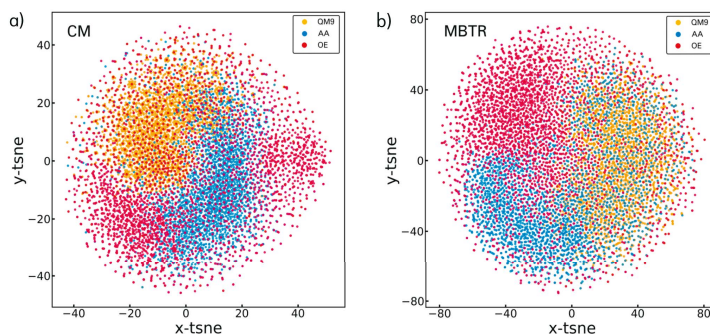


Figure 3.3. Chemical diversity within the datasets QM9, AA and OE62, as seen through the a) CM and b) MBTR molecular descriptor. Reproduced from Publication III.

trates the chemical diversity in OE62 compared to the other two datasets. OE62 contains both small and large organic molecules, while QM9 and AA include small molecules of similar size and similar element composition. The AA dataset includes amino acid conformers with different protonation states of the backbone and sidechains. Moreover, different divalent cations (Ca^{2+} , Ba^{2+} , Sr^{2+} , Cd^{2+} , Pb^{2+} , and Hg^{2+}) are attached to some of the amino acids and dipeptides. The chemical diversity of the three datasets and their impact for learning is further discussed in Section 5.1, where Publication III is summarized. In Publication III machine learning performance was investigated with respect to the three datasets QM9, AA and OE62.

3.3 Numerical representation of molecular structures

My aim in this dissertation is to learn spectra from the atomic structure of molecules alone, without having to supply any additional information to the machine, such as molecular properties. However, simply inserting Cartesian coordinates of atomic positions as input to the machine would not prove successful, since fixed coordinates are not invariant to translation of the molecule. That is, when the molecule moves with respect to a reference coordinate system, a new set of Cartesian coordinates is needed to describe the molecule. The same molecule can thus be represented by many different sets of atomic positions. Moreover, Cartesian coordinates alone do not reflect any prior knowledge about the molecule. In order to model molecular spectra and energies, it would be desirable to include information about the underlying physics into the molecular representation. All relevant information about the molecule should be appropriately encoded into the representation, so that the machine learning algorithm is able to draw the proper relation between structures and spectra.

An ideal molecular descriptor fulfills the following requirements [51]: i) invariance to translation, rotation and permutation of the same element in the structure, ii) uniqueness, iii) continuity, iv) generality, and v) efficiency (both in terms of the representation being fast to evaluate as well as requiring the least amount of data points in training). Given finite datasets, the last point can make a considerable difference in the predictive performance and may necessitate alternative representations. The choice of molecular representation is a prevalent area of research in machine learning, and various descriptors that fulfill most of the above stated requirements have been developed. Examples are the Coulomb matrix (CM) [54], bag of bonds (BoB) [57], bispectrum [79], smooth overlap of atomic positions (SOAP) [80], symmetry functions [81], bonding angular machine learning [82] or the many-body tensor representation (MBTR) [83]. Other representations satisfy the requirements only partly, such as cheminformatics fingerprint descriptors [84], the Fourier series of atomic radial distribution functions [52], partial radial distribution functions [85], and rotationally invariant internal vectors [86]. In this dissertation, two of the aforementioned molecular descriptors are used: The CM, because it is a simple representation that is cheap to compute and widely used in other studies, and the MBTR, which is computationally more expensive, but describes molecular structures more accurately than the CM, as demonstrated in Chapter 5.

CM The CM represents each molecule by a matrix C , whose entries are given by

$$C_{ij} = \begin{cases} 0.5Z_i^{2.4} & \text{if } i = j \\ \frac{Z_i Z_j}{\|\mathbf{R}_i - \mathbf{R}_j\|^2} & \text{if } i \neq j, \end{cases}$$

where Z_i is the atomic number (nuclear charge) of atom i and \mathbf{R}_i is the atomic position. A typical CM is shown in Figure 3.4. The main diagonal of the CM embodies a polynomial fit of atomic nuclear charges to the total energies of the free atoms, encoding element types. Off-diagonal elements express the Coulomb repulsion for each pair of nuclei in the molecule, encoding geometry. The CM is symmetric and the number of rows and columns corresponds to the number of atoms in the molecule. The CM is a unique representation of data in the sense that no two molecules will have the same CM unless they are identical or enantiomers. However, one problem of the CM is that it has no well-defined ordering of atoms, that is, there are many different CMs representing the same molecule. One way to uniquely order the atoms in the CM is to permute the matrix so that rows and columns of the CM are ordered by their norm. Another drawback is that the CM is size-dependent, that is, the size of the CM is determined by the number of atoms in a molecule. However, machine learning models need inputs of constant size. This issue can be solved by "zero-padding" matrices of smaller molecules up to a size that corresponds to the largest

molecule found in a given dataset. Filling matrices with zeros, however, adds no new information to the problem. The machine learning model might interpret meaning into the zeros it is given as input, while there is none. Since the zeros in the matrices do not represent actual vacancies in the molecular structure, they are physically meaningless, but the machine learning model is not aware of that.

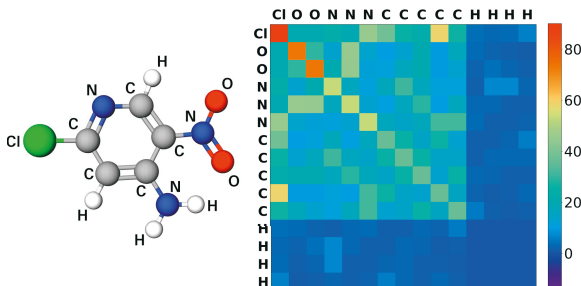


Figure 3.4. In the Coulomb matrix representation, a 3D molecular structure is transformed into a matrix, based on atomic coordinates \mathbf{R}_i and nuclear charges Z_i . The matrix represents one atom per row (and column) and is symmetric. The main diagonal of the matrix contains atomic self-interactions (encoding atom types) while off-diagonal elements contain interactions between different atoms (encoding geometry information). The shown matrix is governed by self-interactions and pairwise interactions resulting from heavier atoms in the molecule (Cl). Reproduced from Publication III.

MBTR The MBTR is a more sophisticated molecular descriptor than the CM, encoding not only geometry and atom types, but also bonding and angle information. Molecular structures are encoded by decomposing them into a set of many-body terms (species, interatomic distances, bond angles, dihedral angles, etc.). A set of constant sized vectors represents each k -body term, where k numbers the level of the many-body term. One-body terms ($k=1$) encode the element types that exist in the molecule. Two-body terms ($k=2$) encode pairwise inverse distances between any two atoms (bonded and non-bonded). Three-body terms ($k=3$) encode angular distributions for any triple of atoms. A geometry function g_k transforms each configuration of k atoms into a single scalar value. These scalar values are broadened into continuous representations \mathcal{D}_k by a Gaussian distribution:

$$\mathcal{D}_1^l(x) = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(x - g_1(Z_l))^2}{2\sigma_1^2}} \quad (3.1)$$

$$\mathcal{D}_2^{l,m}(x) = \frac{1}{\sigma_2 \sqrt{2\pi}} e^{-\frac{(x - g_2(\mathbf{R}_l, \mathbf{R}_m))^2}{2\sigma_2^2}} \quad (3.2)$$

$$\mathcal{D}_3^{l,m,n}(x) = \frac{1}{\sigma_3 \sqrt{2\pi}} e^{-\frac{(x - g_3(\mathbf{R}_l, \mathbf{R}_m, \mathbf{R}_n))^2}{2\sigma_3^2}}, \quad (3.3)$$

where σ_k are the broadening widths for the different k -terms. The variable x runs over a pre-defined range of possible values for the geometry

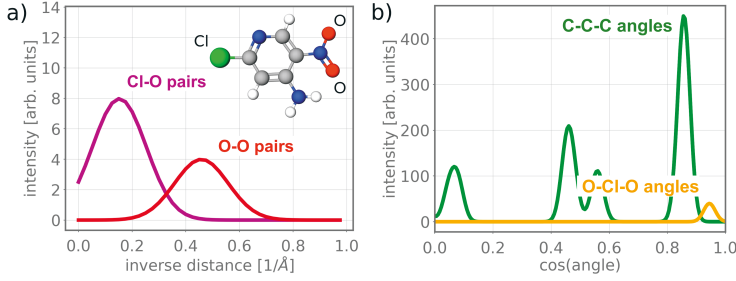


Figure 3.5. Visualization of the many-body terms that constitute the MBTR molecular descriptor. a) Inverse distance distributions ($k=2$ terms). Cl and O atoms are relatively far apart in the molecule. Therefore, a peak occurs at small inverse distances. In contrast, O atoms are located closer to other O atoms. b) Angle distribution ($k=3$ terms). There is a high number of C atoms in the molecule, which gives rise to several C-C-C spikes in the angle distribution. Contrary, there is only one possible configuration for O-Cl-O angles, yielding a single peak in the distribution.

functions g_k . For $k = 1, 2, 3$, the geometry functions are given by $g_1(Z_l) = Z_l$ (atomic number), $g_2(\mathbf{R}_l, \mathbf{R}_m) = |\mathbf{R}_l - \mathbf{R}_m|$ (distance) or $g_2(\mathbf{R}_l, \mathbf{R}_m) = \frac{1}{|\mathbf{R}_l - \mathbf{R}_m|}$ (inverse distance), and $g_3(\mathbf{R}_l, \mathbf{R}_m, \mathbf{R}_n) = \cos(\angle(\mathbf{R}_l - \mathbf{R}_m, \mathbf{R}_n - \mathbf{R}_m))$ (cosine of angle). For each possible combination of k elements occurring in the dataset, a weighted sum of distributions \mathcal{D}_k is produced. For $k = 1, 2, 3$, these final distributions for each many-body term are given by

$$\text{MBTR}_1^{Z_1}(x) = \sum_l^{|Z_1|} w_1^l \mathcal{D}_1^l(x) \quad (3.4)$$

$$\text{MBTR}_2^{Z_1, Z_2}(x) = \sum_l^{|Z_1|} \sum_m^{|Z_2|} w_2^{l,m} \mathcal{D}_2^{l,m}(x) \quad (3.5)$$

$$\text{MBTR}_3^{Z_1, Z_2, Z_3}(x) = \sum_l^{|Z_1|} \sum_m^{|Z_2|} \sum_n^{|Z_3|} w_3^{l,m,n} \mathcal{D}_3^{l,m,n}(x). \quad (3.6)$$

The sums over l , m , and n run over all atoms with atomic numbers Z_1 , Z_2 and Z_3 . The weighting functions w_k balance the relative importance of different k -terms and/or limit the range of inter-atomic interactions.

The MBTR contains many levels of information about the molecular structure, which clearly exceeds the information content encoded by the CM. Another advantage of the MBTR is that each molecular structure is represented by a vector or tensor of same dimensions, hence, it is a constant-size descriptor.

3.4 Summary

In this chapter, I first defined seven characteristics that a materials dataset should have to be of use in my machine learning framework, thereby

answering the first part of the research question:

RQ1 What kind of datasets should I use to train my machine learning model and how can I obtain reliable reference data of high quality?

Materials datasets used within this thesis should contain compounds relevant for organic electronic applications, including 3D structures and spectral properties. The datasets should be publicly and freely available, large in size and diverse. Moreover, the data should be accurate (low bias and variance), consistent within and across datasets and presented in a compatible format. While consistent experimental data are rare in materials science, there are plenty of computational materials datasets that fulfill the majority of mentioned criteria. Among those are the QM9 and the QM7 datasets of 134k and 7k small organic molecules, as well as the AA dataset of 44k amino acids and dipeptides. These three datasets are used within this thesis as reference datasets. To guarantee reliability and consistency across these three datasets I employed my own DFT computations for QM9 and QM7, while the AA dataset did not need any further processing. As a result, I am equipped with three large molecular datasets that contain a diverse collection of organic molecular structures and their orbital energies, all of which are calculated at the same level of DFT. I also generated a new molecular dataset, OE62, which contains 62k diverse molecular structures extracted from the CSD. In addition to DFT-computed structures and orbital energies, OE62 contains high-quality results of G_0W_0 computations for a subset of 5k molecules. This dataset constitutes a valuable spectroscopy benchmark dataset for computational materials science and is published in Publication I.

Moreover, after having learned about the representation of spectra in the previous chapter, this chapter introduced two types of representations for the molecular structure, thereby providing an answer to

RQ2 How can molecules and their spectra be numerically represented in such a way that the machine can establish a relationship between each molecule and its spectra?

For the numerical representation of molecular structures, the Coulomb matrix (CM) and the many-body tensor representation (MBTR) will be employed. The CM is a simple and computationally affordable descriptor encoding information about atomic positions and types. The MBTR is a more sophisticated but computationally more expensive descriptor encapsulating information about atomic types, bonds and angles.

4. Machine learning approach

Having generated enough reference data of molecular structures and their orbital energies, I now review suitable machine learning methods for my research objective of mapping molecular structures (in form of a numerical descriptor) to corresponding energies (scalar values) or spectra (continuous curve of multiple values).

Three machine learning methods are used within this thesis: Kernel ridge regression (KRR), artificial neural networks (ANNs) and Bayesian optimization (BO). Each of these machine learning methods serves a slightly different purpose. KRR seeks to infer a relationship between one or more input values and one output. In other words, it maps multi-dimensional inputs to scalar outputs. I will therefore use KRR to predict a single spectral property for each molecule in my datasets, that is, the energy of the HOMO. The results of HOMO energy predictions with KRR are summarized in Chapter 5. BO is a technique to direct an efficient and effective global minimum search. Hence, I will employ BO to optimize the parameters of my KRR machine learning model. ANNs are highly adaptable machine learning tools consisting of multiple interconnected processing units that receive inputs and deliver outputs. They are able to map multi-dimensional input to multi-dimensional output. Hence, I will use ANNs to predict molecular photoemission spectra that consist of multiple data points. In the last section, I will discuss Publication II, which tackles the optimization of model hyperparameters in machine learning. At the end of this chapter, I address the third research question,

RQ3 Which machine learning methods should I choose and how can I determine the hyperparameters of my models?

4.1 Machine learning principle

Many definitions of machine learning are available in the literature, reflecting the immense popularity and versatility of this fast developing field.

For instance, machine learning is described as a "field of study that gives computers the ability to learn without being explicitly programmed" [87], the "computational study of algorithms that improve performance based on experience" [88] or as a "process through which we use data to train models" [89]. At its root, machine learning is a subfield of artificial intelligence (AI) that can be further categorized into three main branches: Supervised learning, unsupervised learning and reinforcement learning [14, 90].

In supervised learning, both the input data x and the target data t are available, and the machine learning algorithm learns a mapping function $f(x, \theta)$ with internal parameters θ from input data to target data. The goal is to fit the parameters θ of the mapping function so well that targets can be predicted for new input data that was not used for fitting. The mapping function $f(x, \theta)$ is called the machine learning model and the parameters θ are called the model parameters. Supervised learning is the most extensively applied form of machine learning. Further sub-branches of supervised learning are regression and classification. In regression, the targets can take any possible value, while in classification, the targets can take only discrete values.

In unsupervised learning, there are only inputs x , but no targets t . The task of the learning algorithm is to find patterns in the input data. This technique is mostly used as a clustering method to group samples [91].

Reinforcement learning algorithms learn how to behave in an environment by taking actions and quantifying the results [92]. The training is based on exploration and exploitation [14], which is an important concept in Bayesian optimization, as explained in Section 4.4. During exploration, the algorithm searches over the whole training sample space, gathering information that might lead to better decisions in the future. During exploitation, this information is used to exploit only promising areas of the sample space that are close to already found minima. Applications of reinforcement learning include game theory or control theory [93].

In this thesis, I focus on supervised learning, since the task of mapping molecular structures to spectra conveniently fits this type of learning principle. From a mathematical standpoint, the task in supervised machine learning is to find a function $f \in \mathcal{F}$ that maps an input vector x onto a corresponding target value t , where \mathcal{F} is the space of possible functions that depends on the learning method. In the beginning, the data are divided into a *training set* and a *test set*, where the test set typically comprises 10-30% of the whole data. Both training and test data contain input/target pairs. Given a set of n training data $\{(x_i, t_i)\}_{i=1}^n$ that consist of input vectors $x \in \mathbb{R}^d$ and corresponding target values $t \in \mathbb{R}$, the goal during model training is to find a function $f \in \mathcal{F}$ that predicts the target t for a new input x , while minimizing an error function E . This task is

formulated as the following minimization problem:

$$\arg \min_f E(t, f(x)) \quad \text{with } f \in \mathcal{F}, \quad (4.1)$$

where $f(x)$ is a vector containing the model outputs $f(x)$ for all training inputs x , and t is a vector containing the corresponding training targets t . The error function E quantifies the quality of the model output $f(x)$ by comparing it to the target vector t . A popular choice for the error function is the mean absolute error (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |t_i - f(x_i)|, \quad (4.2)$$

the mean squared error (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (t_i - f(x_i))^2, \quad (4.3)$$

or the root mean squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - f(x_i))^2}, \quad (4.4)$$

where i runs over all n input/target pairs.

When all training pairs (x, t) were seen by the model during training, the model is evaluated on the *test set* data. Test data x are given as input to the final model f and the model output $f(x)$ is recorded. The prediction error on the test set is then computed by using, for example, the MAE metric in Equation 4.2, which is based on the deviation between model outputs and test targets.

With increasing training set size, it is expected that the prediction error on the test set decreases. The evolution of the test prediction error with training set size is known as *learning curve*. In many machine learning applications, the prediction error E is found to be inversely proportional to the number of training data n to some power $b > 0$ [58, 94, 95]:

$$E \approx \frac{a}{n^b}, \quad (4.5)$$

As a double logarithmic plot, the learning curves are then expected to decrease linearly with slope b and offset a :

$$\log(E) \approx \log(a) - b \log(N). \quad (4.6)$$

Learning curves not only illustrate how well a machine learning model performs with increasing amount of training data, but also allow to compare different model settings or algorithms. The law in Eq. (4.6) reveals that a good machine learning model is linearly decaying, has a low offset

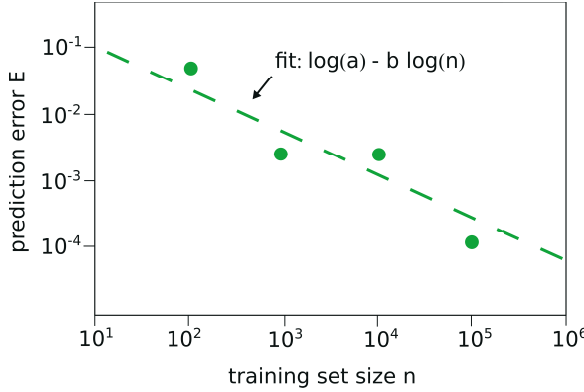


Figure 4.1. Prediction error on the test set as a function of training set size n , plotted on a double logarithmic scale. The error decay with increasing training samples can be fitted with a straight line.

a (achieved for example by using a more suitable representation of the model input) and has a steep learning slope, i.e. large b . Figure 4.1 shows a typical example for such a learning curve. Learning curves also expose the ideal amount of training data needed to obtain an acceptable prediction error. Knowing the minimal amount of data that will yield sufficient model performance is especially useful when working with small datasets where no large volumes of training data are available.

Each machine learning algorithm is a combination of model function f , error function E and optimization strategy. There is no recipe of a universally best learning algorithm – each of these components need to be selected with regard to a specific problem at hand, guided by human intuition and experience. In this dissertation, I employ two machine learning models based on kernel ridge regression (KRR) and artificial neural networks (ANNs), which are widely used in the context of applying machine learning in materials science. Moreover, I employ Bayesian optimization (BO) to optimize the hyperparameters of my KRR model. In the following, I will give an overview over these three methods.

4.2 Kernel ridge regression

Kernel-based methods have been popular in machine learning since they emerged in the 1990s [96–98]. They are based on linear regression, combined with regularization and the kernel trick [97–99].

Linear regression We first consider linear regression, the simplest regression technique. Given the input x , linear regression algorithms compute the model output $f(x)$ as a linear combination of the features of the input

vector, each weighted by a regression weight w_i

$$f(\mathbf{x}) = \sum_{i=1}^d w_i x_i = \mathbf{w}^T \mathbf{x}, \quad (4.7)$$

where d is the dimension of the input space and \mathbf{w} is a vector containing all regression weights w_i . The goal of the learning algorithm during training is to find regression weights w_i that minimize the error on new training inputs. The optimization problem from Equation 4.1 can be rewritten as

$$\arg \min_{\mathbf{w} \in \mathbb{R}^n} \|\mathbf{X}\mathbf{w} - \mathbf{t}\|_2^2, \quad (4.8)$$

where \mathbf{X} is a matrix containing all training input vectors \mathbf{x} and \mathbf{t} is a vector containing all training targets t . Here, the L_2 or Euclidean norm $\|\mathbf{x}\|_2 = \sum_{i=1}^n \sqrt{x_i^2}$ is used as a specific choice for the error function. The problem with this approach is that the model is fitted exactly to the training inputs and targets. This means that any noise in the data, such as deviations in the data due to, e.g. different implementations or numerical settings, are fitted exactly as well. Fitting these meaningless differences in the data can lead to large regression weights w_i that almost balance out the training inputs x_i . The corresponding model has low errors on the training data, but cannot generalize well to new data and therefore exhibits high errors on the test set. This problem is known as overfitting.

Ridge regression Ridge regression is linear regression in combination with regularization to prevent overfitting. Regularization minimizes the regression weights by adding a penalty term based on the squared norm of the weights to the minimization problem of 4.8, yielding

$$\arg \min_{\mathbf{w} \in \mathbb{R}^n} \|\mathbf{X}\mathbf{w} - \mathbf{t}\|_2^2 + \alpha \|\mathbf{w}\|_2^2, \quad (4.9)$$

where α is a hyperparameter that determines the strength of the regularization. The combination of the two terms in Equation 4.9 ensures that among all regression weights \mathbf{w} that map inputs \mathbf{x} to corresponding targets \mathbf{t} well, the smallest ones are chosen. This results in the simplest and least complex possible model, which should then also generalize well to unseen data. It is possible to apply complexity measures other than the norm, however, using the squared norm is a convenient choice since it allows for a simple analytic solution of Equation 4.9. It should be noted that solving the minimization problem only determines the regression weights w_i , but not the regularization strength α . This hyperparameter has to be set separately, typically by cross-validation, which will be further discussed in the next section dealing with the optimization of such hyperparameters.

Kernel ridge regression Linear regression and ridge regression algorithms produce outputs based on a linear combination of training inputs. The

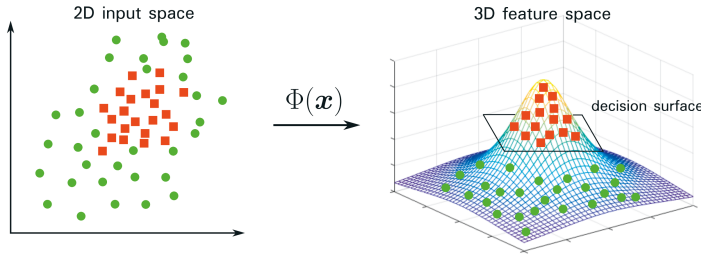


Figure 4.2. In 2D input space, the two classes (red squares, green circles) are not linearly separable through a line. When mapping them into 3D feature space with a suitable mapping function Φ , they become linearly separable through a decision surface.

drawback with these methods is that they can only be applied to input data that are linearly separable with respect to the relationship between the inputs and the targets. For a 2-dimensional input space this means that two sets of input data can be separated by a line such that all data of one set are on one side of the line and all data from the other set are on the other side of the line.

For many machine learning problems, the input data are not linearly separable. In the real world, data are often randomly distributed, which makes it difficult to make predictions based on linear relationships between the input data. This is when kernel-based methods become relevant, such as KRR, support vector machines (SVM), [100], principal component analysis [101], or Gaussian process regression [102]. KRR constructs non-linear learning algorithms from linear learning algorithms by projecting the input data into a higher-dimensional space and employing the linear learning algorithm there. In Figure 4.2, there are two classes of data in input space: green circles and red squares. There is no way to divide them by a linear line. But if there was a way to map the input data from the 2-dimensional space into a 3-dimensional space, then a decision surface could be drawn that separates the two classes. However, it is not easy to find an appropriate mapping function $\Phi(x)$.

This is when the *kernel trick* is applied [103]. The kernel trick allows to operate in the original input space without knowing the coordinates of the data in the higher-dimensional space or doing any computations in that space at all. As an example, consider a 3-dimensional input space with two vectors $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$. Let us assume these vectors need to be projected into a 9-dimensional space in order to linearly separate the input data. The mapping may be accomplished by applying the following mapping functions:

$$\Phi(\mathbf{x}) = (x_1^2, x_1x_2, x_1x_3, x_2x_1, x_2^2, x_2x_3, x_3x_1, x_3x_2, x_3^2)^T \quad (4.10)$$

$$\Phi(\mathbf{y}) = (y_1^2, y_1y_2, y_1y_3, y_2y_1, y_2^2, y_2y_3, y_3y_1, y_3y_2, y_3^2)^T \quad (4.11)$$

In this 9-dimensional feature space, where the input data are linearly separable, a linear learning algorithm (e.g., linear regression) may then be applied for learning. However, finding the appropriate mapping functions and applying them might be difficult. The kernel trick takes advantage of the fact that many machine learning algorithms only require the evaluation of scalar products between input vectors. In feature space, the scalar product between the input vectors is

$$\Phi(\mathbf{x})^T \Phi(\mathbf{y}) = \sum_{i,j=1}^3 x_i x_j y_i y_j. \quad (4.12)$$

However, if we had applied the function $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y})^2$ in the original 3D input space, we would have reached the same result:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y})^2 \quad (4.13)$$

$$= (x_1y_2 + x_2y_3 + x_3y_3)^2 \quad (4.14)$$

$$= \sum_{i,j=1}^3 x_i x_j y_i y_j. \quad (4.15)$$

Thus, we could have avoided the complicated computation in the 9-dimensional feature space. The function $k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})^T \Phi(\mathbf{y})$ is known as a kernel function. A kernel function is a function that operates on data in input space, but whose output corresponds to the scalar product between mapped input data to a higher-dimensional space. In other words, kernel functions can replace the mapping of input data into higher dimensional space and instead evaluate dot products between input data in the original space. A kernel function can be interpreted as a similarity measure between inputs. Commonly used kernel functions are the Gaussian kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp \left(-\frac{1}{2\gamma^2} \|\mathbf{x} - \mathbf{y}\|^2 \right) \quad (4.16)$$

or the Laplacian kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp \left(-\frac{1}{\gamma^2} \|\mathbf{x} - \mathbf{y}\| \right). \quad (4.17)$$

The kernel width γ is another hyperparameter that needs to be chosen separately before model training. Given the input \mathbf{x} , KRR models compute the model output $f(\mathbf{x})$ as a linear combination of kernel outputs, each weighted by a regression weight w_i

$$f(\mathbf{x}) = \sum_{i=1}^n w_i k(\mathbf{x}_i, \mathbf{x}), \quad (4.18)$$

where the sum runs over all n training inputs. The corresponding minimization problem is

$$\arg \min_{\mathbf{w} \in \mathbb{R}^n} \sum_{i=1}^n (f(\mathbf{x}_i) - t_i)^2 + \alpha \|\mathbf{f}\|_{\Phi}^2 \quad (4.19)$$

$$= \arg \min_{\mathbf{w} \in \mathbb{R}^n} \|\mathbf{K}\mathbf{w} - \mathbf{t}\|_2^2 + \alpha \mathbf{w}^T \mathbf{K}\mathbf{w}, \quad (4.20)$$

where $\|\mathbf{f}\|_{\Phi}^2$ is the norm of f in the higher-dimensional feature space and $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the *kernel matrix* between training inputs, with its elements defined as $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. An analytic solution for the regression weights is obtained by setting the gradient to zero:

$$\nabla_{\mathbf{w}} (\|\mathbf{K}\mathbf{w} - \mathbf{t}\|_2^2 + \alpha \mathbf{w}^T \mathbf{K}\mathbf{w}) = 0 \quad (4.21)$$

$$\iff \mathbf{w} = (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{t} \quad (4.22)$$

Predictions can then be made for new inputs using Equation 4.18

$$f(\mathbf{x}) = \sum_{i=1}^n w_i k(\mathbf{x}_i, \mathbf{x}) \quad (4.23)$$

$$= \mathbf{w}^T \mathbf{k}(\mathbf{x}) \quad (4.24)$$

$$= (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{t}^T \mathbf{k}(\mathbf{x}), \quad (4.25)$$

where $\mathbf{k}(\mathbf{x})$ is a kernel vector with elements $k_i = k(\mathbf{x}_i, \mathbf{x})$. Without the kernel trick, one would have to carry out an expensive matrix inversion of mapped inputs in high-dimensional feature space. By reformulating the regression problem with a kernel, one can instead invert the kernel matrix in a much smaller input space of size $n \times n$.

However, the inversion of the kernel matrix is still computationally expensive and therefore, it is usually not practicable to apply KRR to very large datasets of hundreds of thousands of data entries. On the other hand, KRR is fairly transparent due to its exact analytical solution, unlike many other machine learning methods. Predictions are based on mathematical principles and can be interpreted more easily. Moreover, KRR involves only two hyperparameters and thus, little hyperparameter engineering is needed (as long as the descriptors do not introduce many hyperparameters).

4.2.1 KRR for HOMO energy prediction

KRR methods are widely applied in materials science to predict scalar properties of molecules [104–106] and crystals [107–109]. The resulting

models provide results within chemical accuracy. For example, predictions for atomization energies of small organic molecules are reported with an MAE of 1 kcal/mol [108], using only 2000 training molecules. Formation energies of bulk systems are predicted with an MAEs of 0.1 eV/atom [107].

In this thesis, KRR is employed to predict molecular HOMO energies. The input for KRR are molecular structures represented either by the CM or the MBTR, and the prediction target is the scalar HOMO energy. KRR maps each molecular structure to its corresponding HOMO energy. Predicting multiple outputs of full photoemission spectra consisting of several hundreds of data points is not feasible with KRR. While a few regression approaches exist that deal with the task of predicting multi-dimensional outputs [110, 111], these approaches are challenging and costly. Generally, multi-output regression methods can be categorized as problem-transforming methods and algorithm-adaption methods. Problem transforming methods transform the multi-output problem into independent single-output problems, each using an expensive single-output regression routine. Algorithm adaption methods adapt a particular scalar-output method, e.g. KRR, to produce multiple outputs. These are typically even more challenging than problem-transforming methods since they not only predict multiple targets but also draw relationships between these targets. Hence, multi-output regression methods are costly in time and in computational resources, and are not always guaranteed to deliver desired accuracy. Recently, a study on multi-output KRR for the prediction of multiple excited-states of CH_2NH_2^+ cations was published [112], employing the algorithm-adaption method. Ab-initio calculations of electronic states were encoded explicitly in the inputs (in addition to the molecular representation) in order to predict multiple excited-state properties simultaneously, such as forces, nonadiabatic couplings between different states and transition dipole moments. However, when learning all properties simultaneously, significantly worse results were reported than when learning only one target at a time. Therefore, I will use KRR to predict a only scalar targets, in this case the HOMO energy. The results of HOMO energy prediction with KRR are summarized and discussed in Chapter 5. For multiple-output predictions, ANNs are naturally a better fit. They are able to map multi-dimensional inputs to multi-dimensional outputs, as explained in the following.

4.3 Artificial neural networks

Artificial neural networks are versatile machine learning algorithms that yield state-of-the-art results in a variety of different fields such as text classification [113], information retrieval [114], image recognition [115–117], speech processing [115, 118, 119] and machine translation [120]. At its

base, a neural network model is a function f that maps inputs to outputs and that consists of a set of interconnected units, called neurons [121]. An example neuron is shown in Figure 4.3.

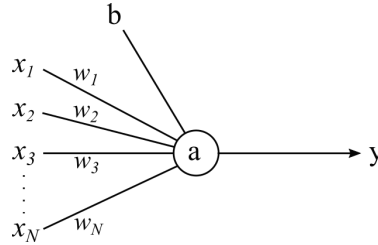


Figure 4.3. Single neuron with its inputs x_i , weights w_i , bias b and output y .

The input to a neuron is the sum of weighted outputs from other neurons, plus a bias term. Then, a so-called activation function is applied, resulting in a single scalar output y for each neuron, which is given by

$$y = a \left(\sum_{i=1}^K w_i x_i + b \right), \quad (4.26)$$

where a is the activation function, x_i are signals coming to the neuron from other neurons, w_i are the weights and b is the bias term. The activation function transforms the linear combination of incoming signals in Equation 4.26 into a nonlinear scalar output of the neuron. This is the reason why ANNs are able to learn nonlinear relationships between input and output. Commonly, smooth functions such as the sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (4.27)$$

or the hyperbolic tangent

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (4.28)$$

are used as activation functions, since they are easy to interpret and differentiable, which facilitates training [122]. The neurons are placed in a layered architecture with an input layer, output layer and a hidden layer, as shown in Figure 4.4. Every element in the input layer is linked to every element in the hidden layer via w_{kl} , which is the weight of the link between the k^{th} input element and the l^{th} hidden neuron. The same connection structure is present between the hidden layer and the output layer with w'_{lm} , which is the weight associated with the connection between the l^{th} hidden neuron and the m^{th} output neuron. Output neurons usually employ another type of activation function than input neurons and hidden neurons, since the output is typically represented by a larger range of values than the information within hidden layers.

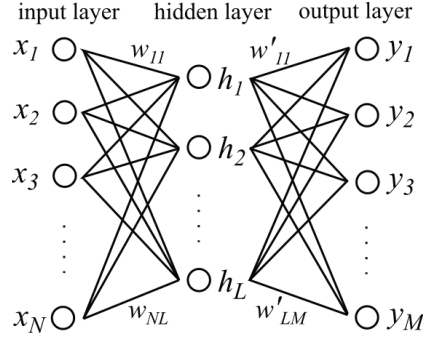


Figure 4.4. Example of a simple feedforward ANN with one hidden layer. The layers of the network consist of N input neurons, L hidden neurons and M output neurons. The input $\mathbf{x} = \{x_1, \dots, x_N\}$ flows to the hidden layer $\mathbf{h} = \{h_1, \dots, h_L\}$ and from there to the output $\mathbf{y} = \{y_1, \dots, y_M\}$. The outputs from preceding neurons are weighted and biased before being passed to neurons in the next layer, where the inputs are processed by an activation function.

In a simple feedforward neural network, each hidden neuron is linked to each hidden neuron in the following layer. The information travels from the input layer to the hidden layer and then to the output layer. In this process, the output from the previous layer is used as input for the next layer. During training, the weights and biases are optimized with respect to an error function in such a way that the computations performed by the neurons lead to a desired output, as further explained in the next paragraph. One might think that simply placing small computational units in a series of layers and connecting them through activations would not approximate the underlying phenomena in the data properly. However, according to the universal approximation theorem, a feedforward ANN with only a single hidden layer can fit practically any continuous function [123, 124]. Thus, a network that has a sufficient number of hidden layers, each containing many hidden neurons, can fit almost any complex dependency between inputs and outputs. Thus, neural networks are highly flexible and adaptable to specific problems.

However, when the neural network becomes too large, tuning the model parameters during training can be challenging. Model training is typically done by employing the backpropagation algorithm [125, 126], which subsequently repeats two cycles: propagation and weight update. Input data $\mathbf{x}_i \in \{x_1, \dots, x_N\}$ travels layer by layer forward through the network until reaching the output layer. The prediction of each neuron y_i in the output layer is then compared to the desired target. The resulting error values are then propagated backwards from the output to the input layer. After this backward pass, each neuron in the network is associated with an error value, which approximately corresponds to its contribution to the overall error. In order to minimize the overall error function, the gradient of this error function is computed with respect to weights and biases. This

gradient is then passed to an optimization method, commonly a gradient descent algorithm, which updates the weights and biases according to their overall error contribution.

When applying ANNs to real-life problems, the most demanding task is the selection of the network architecture. While there is usually not much freedom in the design of input and output layers of a neural network, the structure of hidden layers is in most cases not straightforward. It is often not that obvious which activation function to apply for which layers, how many hidden layers to employ and how many neurons within these layers to connect with each other. Fully connected architectures, such as feedforward ANNs tend to be computationally expensive, since all neurons are connected with each other. Hence, these architectures are not feasible for data that require a large number of input neurons or for large datasets.

4.3.1 ANNs for photoemission spectra prediction

Like many other fields, also materials science exploits the broad applicability of ANNs. ANNs have become powerful tools for large-scale molecular dynamics (MD) simulations to predict forces and potential energy surfaces [81, 127–130] significantly faster than most of the efficient electronic structure methods, thereby enabling MD simulations of large systems. Moreover, relationships between molecular structures and correlation energies [131] as well as bond dissociation enthalpies [132] were successfully modeled by ANNs.

Within this thesis, when applying neural networks for the prediction of molecular photoemission spectra, large molecular datasets of several thousands of molecules are used. The input layer, represented by the molecular descriptor, involves hundreds to thousands of entries, which requires a large amount of input neurons. Therefore, fully connected neural networks are computationally too expensive for the prediction of molecular photoemission spectra. Instead, I will consider two types of deep neural networks: convolutional neural networks (CNNs) and deep tensor neural networks (DTNNs). Deep learning networks typically involve a large number of hidden layers, but with fewer neural connections than fully connected networks [126, 133, 134]. Due to the large number of hidden layers, deep neural networks are able to adapt to very complex problems while little feature engineering is needed, that is, raw data can often be used as input. Many deep neural networks are able to learn internal parameters or to form an efficient representation of input features. The increasing amount of available data in almost all fields of research and industry is the driving source for the rapid development of deep learning algorithms, which vitally require large amounts of training data. The first computational model for neural networks was created in 1943, but it was only over the last two decades when large amounts of data was available

to adequately train them.

Convolutional neural networks Convolutional neural networks are network architectures whose layers are not fully connected, offering a computationally affordable training for large quantities of data and for data with large input dimensions [122, 135]. The connectivity scheme between neurons in CNNs is motivated by the arrangement of the animal visual cortex, which contains a complex structure of cells that are responsive to small sub-regions of the visual field. The cells act as local filters to the visual field and are able to exploit strong local correlations in images. Instead of fully-connected layers, CNNs employ a local connectivity scheme between neurons of neighbouring layers, thereby processing spatial input structures. CNNs are widely used in image and video recognition, recommendation systems and language processing [115]. Also in materials science, CNNs have been applied to various recognition problems, such as classifying structures by the symmetry of their diffraction patterns [136], characterizing complex molecular assemblies on surfaces from scanning tunneling microscope (STM) images [137] or identifying molecules from atomic force microscopy (AFM) images [138].

Figure 4.5 shows the CNN architecture used for the prediction of photoemission spectra developed in Publication IV. The CM is used as molecular descriptor, representing the input layer of the CNN. The number of input neurons equals the number of CM entries, which in turn depends on the number of atoms of the largest molecule in the dataset. For QM9, the largest molecule has 29 atoms, hence, the input layer consists of $29 \times 29 = 841$ input neurons. The input layer is connected to the first

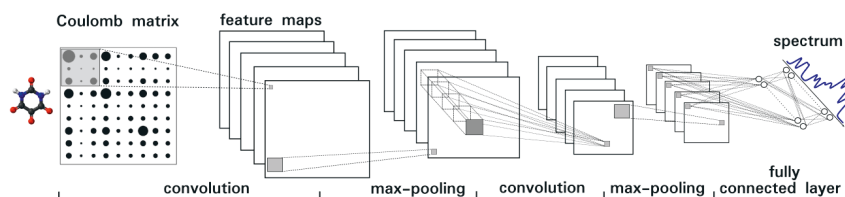


Figure 4.5. CNN architecture for the prediction of photoemission spectra. Each molecule is represented by its CM, which is passed through a multilayer network consisting of convolutional layers and subsequent max-pooling layers. The second max-pooling layer is fully connected to the output, which either contains 16 neurons corresponding to 16 discrete excitation energies, or which contains 300 neurons corresponding to the 300 discretized points of the photoemission spectrum.

convolutional layer, which consists of several feature maps made up of hidden neurons. In each feature map, the hidden neurons exploit a different feature of the input. In contrast to fully-connected feed-forward neural networks, not every input neuron of the CNN is connected to every hidden neuron in the feature maps. Instead, each neuron in the feature maps is connected to only a small region of the CM, called the local receptive field.

The local receptive field is slid across the CM entry by entry to connect to another neuron in each feature map. This is done simultaneously for all feature maps, until each feature map is fully built. During training, the neurons in the feature maps analyze their receptive fields and learn a corresponding weight and bias. In the same layer, each feature map uses the same set of weights for all of its neurons, which implies that each of the hidden neurons in one particular feature map learns the same features for its local receptive field. This way, all neurons of the same feature map detect exactly the same feature of the input, but at different locations in the CM. The weights defining the feature map are thus called shared weights, and the bias defining the feature map is the shared bias. Using shared weights and biases notably lowers the number of parameters determining the network, which results in faster training of the CNN.

After the convolutional layer follows a max-pooling layer, which simplifies the information from the convolutional layer by preparing a condensed feature map. Each unit in the pooling layer outputs the maximum value of its input region. This adds scale-invariance to the learning process of the network and again helps reduce the number of parameters needed in the following layers and adds scale invariance. Max-pooling is applied to each feature map separately, resulting in 5 max-pooling layers. After the max-pooling layer follows another another pair of convolutional and pooling layers. Finally, the output layer represents the photoemission spectrum, which is numerically described either by 16 discrete KS eigenvalues or by 300 discretized data points of the Gaussian-broadened spectral curve.

Deep tensor neural network Another type of neural network used in Publication IV is the deep tensor neural network (DTNN), which is a custom designed deep neural network by Schütt et al. [135]. This type of network can adapt to the complex task of building its own molecular representation based on atomic positions in the molecules. Thus, no CM is needed as input, but only (x, y, z) coordinates of atomic positions within the molecule. The DTNN transforms these positions into its own representation by building a coefficient vector according to their species and nuclear charges. The coefficient vectors are subsequently refined in a sequence of interaction passes by embedding each atom in its neighbourhood with other atoms. In the first interaction pass, interatomic distances are learned. In the second interaction pass, angles between atom triplets are encoded. In subsequent passes, higher order interactions, such as dihedral angles are learned. Thus, by decomposing atomic interactions within the molecule the DTNN is able to learn an efficient representation of embedded atoms, which encodes local atomic environments in a similar way as the MBTR. The DTNN then passes the embedded representation of atoms through two fully connected neural layers. For each atomic coefficient vector, an energy contribution is predicted, which contributes to the final photoemission spectrum in the output layer.

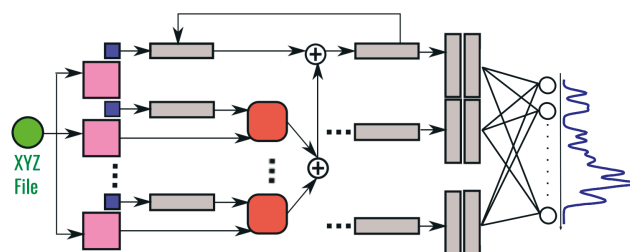


Figure 4.6. Simplified diagram of the DTNN for the prediction of photoemission spectra. Molecular XYZ structures are given as input to the network by a vector of nuclear charges (blue squares) and an interatomic distance matrix (pink squares). According to its nuclear charge, a coefficient vector (grey block) is assigned to each atom. Parameter tensors (red squares) transform the coefficient vectors and the interatomic distance matrix into a refined coefficient vector. This refinement is repeated multiple times, building up more complex interactions between atoms with each pass. Two fully-connected layers (grey blocks) are employed to obtain the final spectral contributions from each atom for each point of the spectrum. The sum over all spectral contributions gives the total prediction for a given point in the spectrum. Adapted from Publication IV.

The results from Publication IV, where the above described CNN and DTNN were used for the prediction of molecular photoemission spectra are summarized and discussed in Chapter 5. Before machine learning models are able to make reliable predictions, problem-specific hyperparameters need to be fine-tuned. This will be the objective of the following section, explained in combination with KRR.

4.4 Hyperparameter optimization

An often difficult challenge when applying machine learning algorithms to real-world data is the choice of so called hyperparameters. Hyperparameters are specific to the machine learning method and to the problem, but cannot be learned by the algorithm itself during training. Instead, they have to be selected separately. For example, KRR involves two hyperparameters, namely the regularization strength α and the kernel width γ , while the regression weights w are internal parameters of the model that are tuned during training via Equation 4.1. In more general terms, one can also view the kernel function k , the error function E or the machine learning algorithm itself as a hyperparameter. For ANNs, there are considerably more hyperparameters, such as the activation function a , the number of hidden layers, the number of neurons in the hidden layers, the number of connections between the neurons as well as the optimization algorithm (e.g. backpropagation) to minimize the error. Just like internal model parameters, hyperparameters crucially influence the performance of machine learning.

However, it is not always obvious which hyperparameter values lead to a good overall model performance. Unfortunately, there is no equation that balances model performance with respect to the hyperparameters. Therefore, hyperparameters are often chosen based on intuition, prior knowledge or trial and error. Another option is to apply suitable search algorithms capable of finding the optimal set of hyperparameters that minimize the machine learning prediction error. Within this thesis, two algorithms based on grid search and Bayesian optimization were invoked to tune the hyperparameters of machine learning with KRR.

4.4.1 Grid search

Within a grid search, a grid of values is setup for each hyperparameter. The dataset is first divided into training and test set and then, a validation set is split off from the training set [139, 140]. Each possible combination of hyperparameters from the grid is used once for model training on the remaining training data, while the prediction error is estimated on the validation data. The hyperparameter combination with the lowest prediction error is chosen. The advantage of splitting off a validation set from the training set is that the optimal hyperparameter settings can be determined on the validation set, while leaving the test set completely untouched. Using the best performing combination of hyperparameters, the final prediction error can then be reported on the test data. In other words, the remaining training set is used for learning the model parameters (regression weights w_i when using KRR), the validation set for learning the hyperparameters (e.g. regularization strength α and kernel width γ) and the test set for evaluating the performance of the trained model.

For very small datasets, however, splitting off validation data from the training data leaves little data to train the model, and the reported prediction errors will be unreliable. In this case, k -fold cross-validation [141] can be used, where the training data is split into k subsets. For each combination of hyperparameters, each subset is used once for training and once for validation, and the validation errors from individual splits are averaged. The combination of hyperparameters with the lowest average error is chosen, and the prediction error is evaluated on the untouched test set. However, the drawback of grid search is that it is computationally and time-wise expensive to try every possible combination of hyperparameters. Each combination requires training and validation, which can quickly eat up computational resources, especially when dealing with large datasets or complex machine learning methods that involve many hyperparameters.

4.4.2 Bayesian optimization

Another, more efficient and effective approach to automate the hyperparameter search is the use of global optimization algorithms, such as Bayesian optimization (BO) [90, 142–144]. BO seeks to find the minimum of an objective function $f(x)$, given data x within the domain \mathcal{X} . In this thesis, BO is applied to find the optimal set of KRR hyperparameters α and γ that minimize the prediction error (in terms of the MAE) of my KRR model. In this problem set up, the objective function is the unknown MAE as a function of model hyperparameters. The output of this function is only known after evaluating it at a selected point x of hyperparameters. Each evaluation of the objective function for a certain setting of hyperparameters involves the training of the KRR model. Due to the matrix inversion that is involved in KRR training, querying many hyperparameters can quickly eat computational resources. Thus, the number of calls to the unknown objective function should be kept as low as possible.

A number of other global optimization approaches different from BO exist [145]. For machine learning, stochastic approximation is a popular concept for the optimization of unknown objective functions [146] that is used in reinforcement learning [93, 147, 148], for Boltzmann machines [149, 150] or deep belief networks [151]. However, stochastic approximation requires a large number of samples, and is therefore not suitable for problems where drawing samples is expensive. The key idea of BO approaches is to use prior knowledge and evidence in order to sample efficiently, drawing as few samples from the unknown objective function as possible. BO utilizes past evaluation results to form a probabilistic model – the surrogate model – of the objective function. To decide where to sample next, BO employs an acquisition function, which assesses the surrogate model for the most promising hyperparameters. These are then evaluated on the objective function in the next iteration. After each objective function evaluation, the surrogate model is updated. BO therefore becomes more accurate with more given data. BO methods determine the next set of hyperparameters based on past trials, while making as few calls to the objective function as possible. This process of informed decision-making is the reason for the success of BO approaches for hyperparameter optimization in machine learning.

In the following, I explain the general working principle of BO. Then, in Section 4.4.3, I summarize and discuss my results of KRR hyperparameter optimization based on BO.

There are two key ingredients for a BO model: A probabilistic model based on Gaussian process regression (GPR) and an acquisition function that determines the next hyperparameters to evaluate on f . Let $x \in \mathbb{R}^n$ be a vector encoding n hyperparameters to be optimized. At iteration i , the

objective function f is sampled at the data point $x_i \in \mathcal{X}$, yielding the scalar value $y_i = f(x_i)$ as a result of evaluating f . After N iterations, let the ensemble of N sampled data points x_i and N corresponding observations y_i constitute the dataset $(x, y) = \{(x_i, y_i)\}_{i=1}^N$.

Gaussian process regression Since the true objective function f is expensive to evaluate, BO seeks to approximate f by a surrogate model. Gaussian process regression (GPR) can be used to build this surrogate model as the mean of a Gaussian process (GP). In general, a GP is a probability distribution over functions [142]. Applied to BO, the GP provides a prior probability distribution over possible objective functions

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')), \quad (4.29)$$

where $\mu(x)$ is the mean function and $k(x, x')$ is the covariance function of the GP, which is a kernel function. GPs provide a prior belief about the space of possible objective functions that are consistent with already observed data x . The prior distribution returns the mean $\mu(x)$ (which is typically assumed to be zero) and the variance $k(x, x')$ of a normal distribution over possible values of f at x . In other words, the true objective function f is modelled by a GP prior with zero mean. A typical choice for the covariance function is the squared exponential function

$$k(x, x') = \exp\left(-\frac{1}{2\theta^2} \|x - x'\|^2\right), \quad (4.30)$$

where θ is a GP-internal hyperparameter controlling the width of the kernel. The squared exponential function describes how closely two points in input space are correlated to each other and determines the smoothness of sample functions drawn from the prior distribution.

The goal in GPR is to compute the GP posterior distribution from the prior distribution. The posterior distribution is the surrogate model of the objective function. This posterior is computed by applying the Bayes' rule and by combining prior belief (about the initial distribution of functions) and evidence (provided by sampled data).

As new data points x_* are acquired, the prior distribution is updated, producing the posterior. The posterior distribution at the new point x_* , conditional on previously sampled data (x, y) and on the GP hyperparameter θ , is again a normal distribution

$$x_*|(x, y), \theta \sim \mathcal{N}(\mu(x_*), \sigma^2(x_*)), \quad (4.31)$$

where

$$\mu(x_*) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{y}, \quad (4.32)$$

$$\sigma^2(x_*) = k(x_*, x_*) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}. \quad (4.33)$$

The posterior distribution in Equation (4.31) is the surrogate model of the objective function f . Hence, the surrogate model is defined by the posterior mean $\mu(\mathbf{x}_*)$ and the posterior variance $\sigma^2(\mathbf{x}_*)$. The prediction of the surrogate model for a new set of hyperparameters \mathbf{x}_* is given by evaluating the posterior mean $\mu(\mathbf{x}_*)$. The variance $\sigma^2(\mathbf{x}_*)$ describes the uncertainty of the model and indicates which regions of the hyperparameter space are less known.

The optimal point $\hat{\mathbf{x}}$ predicted by the surrogate model and its corresponding global minimum are given by

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}_* \in \mathcal{X}} \mu(\mathbf{x}_*), \quad \mu_{\min} = \mu(\hat{\mathbf{x}}) \quad (4.34)$$

The convergence of the surrogate model can be monitored as $\mu(\hat{\mathbf{x}}) - f(\hat{\mathbf{x}})$, which describes the difference between the global minimum prediction of the surrogate model and the true function at the predicted optimal point.

Acquisition function The acquisition function determines which point \mathbf{x}_* within the domain \mathcal{X} to evaluate next on the objective function f , hence guiding the search for the global minimum. The location of the next acquisition \mathbf{x}_* is typically selected by maximizing the acquisition function a :

$$\mathbf{x}_* = \operatorname{argmax}_{\mathbf{x}_* \in \mathcal{X}} a(\mathbf{x}_*) \quad (4.35)$$

The acquisitions a depend implicitly on the previously sampled data (\mathbf{x}, \mathbf{y}) and on the GP-internal hyperparameter θ :

$$a(\mathbf{x}_*) = a(\mathbf{x}_*; \mathbf{x}, \mathbf{y}, \theta). \quad (4.36)$$

The corresponding observation is y_* . In the following iteration, (\mathbf{x}_*, y_*) is added to the dataset. The best observation from all acquisitions is denoted as $(\mathbf{x}_{\text{best}}, y_{\text{best}})$.

4.4.3 Comparison of Bayesian optimization and grid search for KRR hyperparameter tuning

In machine learning, BO is widely used for the optimization of various learning algorithms such as random forest, deep neural network, deep forest or kernel methods [152–156]. However, the machine learning algorithms are mostly applied to standard datasets from the UCI machine learning repository [157]. It is not yet common to apply BO to optimize machine learning setups in materials science, where typically high-dimensional hyperparameter spaces are encountered. The objective of Publication II is to quantify the efficiency and accuracy of BO against grid search for the optimization of up to 4 hyperparameters. From a more practical perspective, this work also intends to provide a guideline to machine learning practitioners who work in materials science or related fields, in

order to help them choose possible starting points for similar optimization problems. For this aim, landscapes of the MAE are generated across the hyperparameter phase space. Visualizing hyperparameter landscapes provides insight into how the prediction performance of the machine learning model changes across a range of possible hyperparameter configurations.

Setup of the study In Publication II, the Bayesian Optimization Structure Search (BOSS) [158] tool is used to optimize the hyperparameters of my KRR machine learning model for the prediction of HOMO energies. The target property is the MAE of the KRR model, which BOSS seeks to minimize. For this aim, BOSS employs BO based on GPR to generate surrogate models of the target property. I first defined a range of hyperparameter values for the hyperparameter search domain, which is applied to BO as input. BO then conducts a fully automated search to find the best combination of hyperparameters. For each new acquisition, the molecular descriptor (either CM or MBTR) is computed and KRR with 5-fold cross validation is performed. The average MAE of the 5-fold cross validation is returned to BO to refine the surrogate model and then the next acquisition begins.

Three different datasets are employed, QM9, AA and OE62, while molecular structures are represented by two different descriptors: the CM and the MBTR. The KRR method itself requires the optimization of the two hyperparameters α and γ . The CM representation has no hyperparameters, thus, when the CM is employed as molecular descriptor, the hyperparameter search space is 2-dimensional. The MBTR on the other hand has many hyperparameters, 14 in total. Through pre-testing it was found that only 2 of the 14 MBTR-internal hyperparameters have a noteworthy influence on the KRR performance, namely the two broadening widths σ_2 and σ_3 (for the definitions of these MBTR parameters please see Section 3.3). Thus, the MBTR introduces 2 hyperparameters to the optimization problem. In combination with the 2 KRR hyperparameters, the search space within this setup is 4-dimensional. The objective function of the optimization problem is the MAE on the test set, which is minimized by tuning the KRR and MBTR hyperparameters.

In grid search, the natural logarithmic grid

$$\{e^i | i = [-10, -9, \dots, 0]\} \quad (4.37)$$

is used as search space for the KRR hyperparameters α and γ . For the MBTR hyperparameters σ_2 and σ_3 , the logarithmic grid

$$\{e^i | i = [-6, -5, \dots, 0]\} \quad (4.38)$$

is employed. In BO, the boundaries of these intervals are used as input for the search domain. The acquisition function can choose any real number between these bounds to evaluate on the objective function.

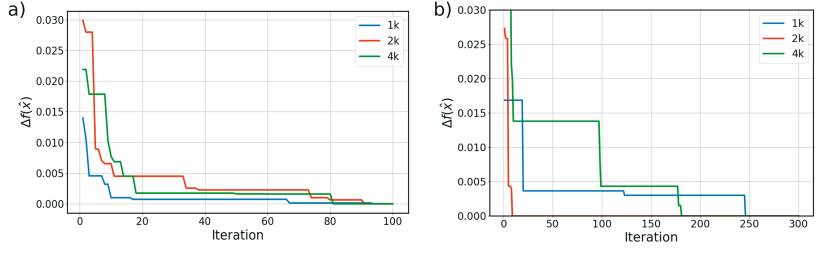


Figure 4.7. Convergence in BO as a function of iterations for three different training set sizes of QM9 data. In a) the CM descriptor and in b) the MBTR descriptor is used. The convergence criterium $\Delta\hat{x}$ describes the difference between the currently lowest true function value and the lowest true function value after the maximum number of iterations. Adapted from Publication II.

BO computational methodology in Publication II During GPR, a surrogate model of the MAE is computed as the posterior mean of a Gaussian process (GP), based on sampled MAE data. The posterior mean fits the MAE data, while the computed posterior variance points to regions in phase space that are less well explored. Given the mean and the variance, the acquisition function is computed. The global minimum of the acquisition function indicates the combination of hyperparameters to be evaluated next on the objective function. Once this point is evaluated, the resulting MAE is added to the dataset and the cycle repeats. With each additional datapoint, the MAE surrogate model is improved.

The choice of the acquisition function defines how exploration and exploitation of the surrogate model is balanced. I chose the exploratory lower confidence bound (eLCB) acquisition function [142], which combines data exploration (searching previously unvisited regions of phase space) with exploitation (searching near known minima) to determine the global minimum with as few iterations as possible.

BO convergence I determine the convergence in BO by monitoring the global minimum location \hat{x} of optimal hyperparameters, as predicted by the surrogate model. The objective function at this minimum location $f(\hat{x})$ is evaluated and the lowest value ever observed is kept track of. The convergence is then computed as the difference between the currently lowest observed true function value, $f_{\min_current}$, and the lowest true function value observed after the maximum number of iterations f_{\min_end} :

$$\Delta f(\hat{x}) = f_{\min_current} - f_{\min_end}. \quad (4.39)$$

The maximum number of iterations in the 2D search case (CM) is 100 iterations and 300 iterations for the 4D search case (MBTR). Within this thesis, the convergence criteria for BO is defined as $\Delta f(\hat{x}) \leq 10^{-2}$. Figure 4.7 shows that $f(\hat{x})$ quickly decreases with progressing iterations. In the 2D optimization case, the surrogate model is already converged in fewer

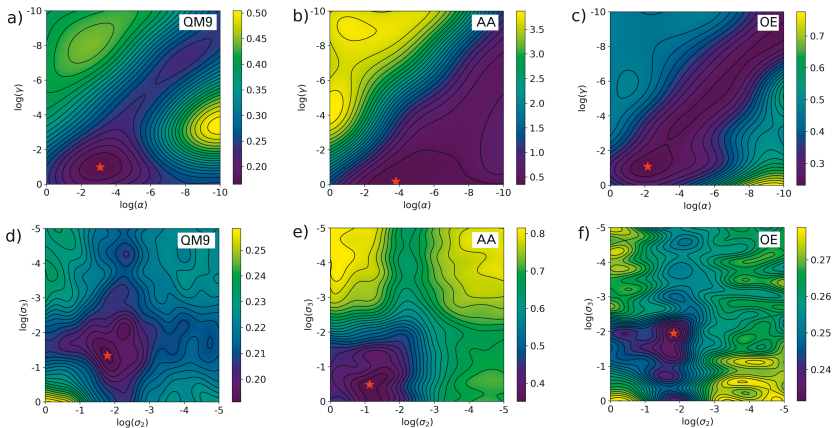


Figure 4.8. MAE landscapes from 4D hyperparameter optimization with BO for three different molecular datasets QM9, AA and OE62, using the MBTR descriptor. In a), b) and c), the predicted MAE $\mu(x)$ is shown as a function of hyperparameters α and γ , evaluated on a logarithmic grid. In d), e) and f), a cut of the MAE landscape through the logarithmic (σ_2, σ_3) plane is provided. From each dataset, a subset of 2k molecules was used for training. Optimal hyperparameters are shown as red stars. Reproduced from Publication II.

than 20 iterations for all training set sizes. In the 4D search case, the surrogate model reaches convergence before iteration 50 for the training set sizes 1k and 2k, while it almost takes 100 iterations to reach convergence for a training set size of 4k. In all cases, global minima are reached in fewer than 100 iterations.

In the following, I will first analyze the MAE landscapes produced by the BO surrogate model $\mu(x)$. Then, I will compare the BO performance with minima found by grid search.

KRR hyperparameters α and γ Figure 4.8 shows BO landscapes resulting from the 4D-optimization problem when the MBTR descriptor is employed. Results are shown for the three datasets QM9, AA and OE62. Two dimensional cross-sections are extracted at the global minimum \hat{x} (marked as a red star). The upper panels a), b) and c) in Figure 4.8 illustrate logarithmic (α, γ) cross-sections for QM9, AA and OE62, while σ_2, σ_3 are held constant at their optimal values. While varying in small details, the qualitative behaviour of the MAE as a function of α and γ is consistent across all three datasets. For QM9, the optimal parameter region can be found on a diagonal and on a horizontal line at the bottom of the map. In the diagonal part, the hyperparameters are mutually dependent, that is, the choice of α and γ equally contributes to the KRR performance. In the one dimensional part at the bottom of the map, only the choice of γ is important, while α can take any value. For AA, the triangle in the landscape is filled, revealing a wide range of optimal hyperparameters. For OE, there is no horizontal line at the bottom. For all three datasets, the location \hat{x} of optimal

hyperparameters lies within the same region.

While Figure 4.8 shows results of the 4D-search when the MBTR is used as molecular descriptor, a qualitatively equivalent MAE landscape is observed when employing the CM (see Figure 4 in Publication II). In this case, the search space is only 2-dimensional since only the two KRR hyperparameters need to be optimized. The overall MAE values are higher than in the 4D case with the MBTR descriptor, since the CM is unable to capture all molecular information necessary for learning. The performance of CM vs. MBTR will be further discussed in Section 5.1.

MBTR hyperparameters σ_2 and σ_3 In the lower panels d), e) and f) in Figure 4.8, the 4D landscapes are cut through the logarithmic (σ_2, σ_3) plane, while α and γ values are held constant at their optimal values. All three datasets feature a cross-like shape of low MAE values. For QM9 and OE62, the optimal MAE values are constrained to a relatively narrow region that roughly lies on the crossover point. In contrast to the KRR hyperparameters, varying the σ_2 and σ_3 values throughout the map does not significantly influence the MAE. Thus, all combinations of σ_2 and σ_3 are reasonably good choices for learning. For AA however, no cross-like shape is observed. The choice of σ_2 and σ_3 significantly affects the MAE for AA.

For both logarithmic hyperparameter planes, (α, γ) and (σ_2, σ_3), the optimal MAE values vary considerably across the three datasets. The MAE values are lowest for the QM9 dataset of small organic molecules, which is easiest to learn among all three datasets. Higher MAEs are observed for the AA and OE62 datasets, which are more difficult to learn due to the more complex molecular structures. A more comprehensive discussion on the predictive power of KRR in dependence on the complexity of the underlying dataset will be given in Chapter 5.

We have seen so far that BO provides easily readable MAE landscapes that enable a deeper analysis of my KRR model and facilitate the choice of possible starting points for KRR model training. Next, the performance of BO in comparison to grid search is discussed, which is the central aspect of Publication I. First, MAE landscapes of BO and grid search are compared to ensure that BO is able to find the same or lower MAE as grid search. Then, the efficiency of BO with respect to grid search is discussed.

BO vs. grid search Figure 4.9 shows a comparison of the MAE landscapes produced by grid search and BO for the QM9 dataset. The grid search landscape is made up of discrete points since the search was performed on a grid of 10×10 points for α and γ and of 6×6 points for σ_2 and σ_3 . The BO search on the other hand is not constrained to a grid but interpolates the MAE between individual acquisitions. BO and grid search produce qualitatively and quantitatively consistent MAE landscapes and optimal solutions \hat{x} , $f(\hat{x})$ and $\mu(\hat{x})$. Moreover, the optimal hyperparameter solution

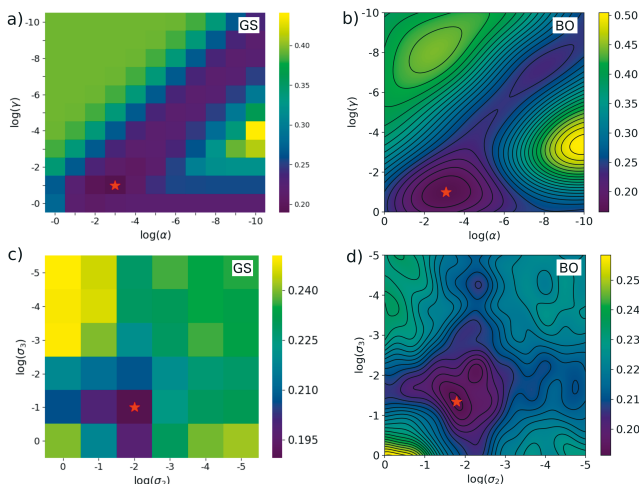


Figure 4.9. MAE landscapes for the 4D optimization problem with the MBTR descriptor. Panels a) and b) show 2D slices through the logarithmic (α, γ) plane, while panels c) and d) show 2D slices through the logarithmic (σ_2, σ_3) plane. In a) and c), the MAE from grid search and in b) and d), the MAE from the BO surrogate model $\mu(x)$ are presented. Both grid search and BO were applied to a subset of 2k molecules taken from the QM9 dataset. Optimal hyperparameters are shown as red stars. Reproduced from Publication II.

\hat{x} found by BO and grid search are located in the same hyperparameter region of low $\log(\alpha)$ values and high $\log(\gamma)$ values. Thus, BO is capable of reproducing grid search solutions in terms of accuracy. But can it find the optimal solution more efficiently in terms of time and computational resources than grid search?

To answer this question, we take a look at Figure 4.10, which shows the total computing time for grid search and BO as a function of training set size. The total time mainly depends on two quantities: The number of times that the descriptor has to be built (n_{desc}) and the number of times that cross-validated KRR needs to be performed (n_{KRR}) in order to find the optimal solution. While n_{desc} scales linearly with training set size, the crucial quantity is n_{KRR} , which scales cubically with training set size due to the inversion of the kernel matrix that is performed during KRR model training. Panel a) shows the total time for the 2D search case, when the CM is used as molecular descriptor. Grid search clearly outperforms BO, especially for large training set sizes. The reason for this is that in BO, the molecular descriptor has to be built and cross-validated KRR has to be performed every single time the objective function is evaluated (see Algorithm 3 in Publication I). That is, the cost of the BO approach depends on the number of iterations necessary to reach convergence, which is in this case 100. Contrary, in grid search, the effort depends on the type of descriptor and on the size of the hyperparameter grid. When the CM is used in grid search, the descriptor needs to be built only once at the

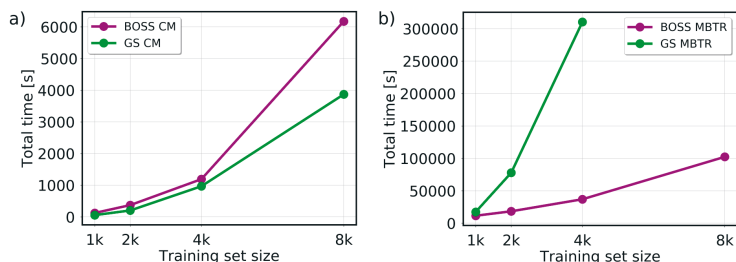


Figure 4.10. Total times for hyperparameter optimization by BO and grid search. In a) the CM is used (2D search) and in b) the MBTR is used (4D search). Timings are shown for optimization on the QM9 dataset. Adapted from Publication II.

beginning of the search routine (see Algorithm 1 in Publication I), since the CM does not have any hyperparameters. Cross-validated KRR is then performed 121 times, for each combination of α and γ .

However, in the 4D search case, when the MBTR is used as molecular descriptor, BO is significantly faster than grid search, as shown in panel b). Now, grid search needs to compute the MBTR for each combination of σ_2 and σ_3 , i.e. 36 times and perform cross-validated KRR for each possible combination of α , γ , σ_2 and σ_3 , i.e. 4,356 times (see Algorithm 2 in Publication I). This is so costly that already for a training set size of 8k, it is not feasible to perform a 4D-grid search. BO, in contrast, requires only 300 iterations to reach convergence and clearly outperforms grid search.

In summary, Publication II shows that for hyperparameter optimization in machine learning, BO is many times more efficient than grid search when the search space is larger than two dimensions, while delivering the same, or superior accuracy. BO is not limited to a grid, but can instead pick any continuous value to evaluate on the objective function. Moreover, BO provides an efficient way of generating easily readable score landscapes that enable a deeper analysis of machine learning performance.

4.5 Summary

In this chapter, I gave an introduction to machine learning and can now answer the third research question

RQ3 Which machine learning methods should I choose and how can I determine the hyperparameters of my models?

I choose three methods for the machine learning framework within this thesis: Kernel ridge regression (KRR), artificial neural networks (ANNs) and Bayesian optimization (BO).

KRR is able to map multi-dimensional inputs to scalar outputs and will be used to predict the molecular HOMO energy, given molecular input structures. ANNs are more flexible and are able to map from multi-dimensional inputs to multi-dimensional outputs. I will use them to predict photoemission spectra consisting of multiple data points.

Before KRR or ANN models are able to make reliable predictions for orbital energies or photoemission spectra, they need to be fine-tuned by optimizing problem-specific hyperparameters of the models. These are parameters that the machine learning model can not learn by itself during training, and thus they need to be specified separately, which is often a burden. Optimization algorithms can facilitate and automate the selection process, but commonly used methods such as brute-force grid search are expensive in terms of time and resources. Tools based on BO find minima in fewer evaluations, since they are able to pick promising hyperparameters in an informed manner based on past results. For the problem setup of predicting HOMO energies with KRR, I employed the Bayesian optimization tool BOSS to tune two KRR hyperparameters and two MBTR hyperparameters. All four hyperparameters are optimized within reasonable effort, while grid search can efficiently tune only two hyperparameters simultaneously. These findings are published in Publication II.

Now that the hyperparameters are tuned for my KRR model, it can be applied for large-scale predictions of HOMO energies. ANNs include even more hyperparameters than KRR due to the large freedom in the design of neural network architectures. In Publication IV, BO is also used to optimize ANN hyperparameters when predicting photoemission spectra. Results of KRR and ANN predictions are presented in the following chapter.

5. Machine learning application

With the first four steps of the machine learning workflow completed – the acquisition of data, the numerical representation of these data, the selection of suitable machine learning methods and the optimization of hyperparameters – it is now time to put the whole machinery into motion. This chapter elaborates on the application of machine learning for the prediction of photoemission spectra and molecular orbital energies. Results from Publications III and IV are presented. In Publication III, a KRR model is developed to predict molecular orbital energies and in Publication IV, three types of neural networks are applied for the prediction of photoemission spectra. After presenting and discussing the results from these studies, I will showcase the applicability of these developed KRR and neural network models to explore chemical space, addressing the final research question of the thesis,

RQ4 How can the chosen machine learning methods be applied in practice to predict molecular orbital energies and photoemission spectra and to explore chemical space?

5.1 Prediction of molecular orbital energies with kernel ridge regression

In Publication III, the energy of the highest occupied molecular orbital (HOMO) is predicted with KRR for three different molecular datasets, using the CM and the MBTR descriptors. Orbital energies of organic molecules play an important role in many technological applications, such as OLEDs, and have thus become the target of many machine learning studies in materials science [58, 62, 105, 108, 159–168]. All of these studies utilize either the QM7/QM7b collection of small organic molecules, or the expanded version QM9. Both are subsets of the GDB database enumerating billions of combinatorial possible molecules of small size and with similar structures. While machine learning models trained on QM7 and

QM9 are reported to perform well when predicting properties for similar molecules, it is not obvious how these methods would perform when trained on other, more diverse molecular datasets. With regard to optoelectronic applications, I am interested in predicting HOMO energies for larger and more diverse organic molecules, such as the molecules covered in the OE62 dataset. As described in Section 3.2, OE62 is a diverse collection of 62k organic molecules with complex aromatic backbones and diverse functional groups, which differ remarkably from the QM9 molecules. Another, more diverse dataset than QM9 is the AA dataset of 44k conformers of proteinogenic amino acids.

In Publication III, these two less known datasets – OE62 and AA – are used for KRR training in order to predict HOMO energies. The objective in Publication III is to demonstrate learning performance of KRR on these more complex and diverse datasets that contain more realistic structures than offered in QM9. In addition, I also include the standard QM9 benchmark dataset of 134k small organic molecules into this study to enable the comparison of KRR performance across these three datasets of different chemical diversity. The inclusion of QM9 furthermore facilitates the comparison with findings from other machine learning studies. For all three datasets, pre-calculated reference HOMO energies with DFT are used for model training and testing. The desired accuracy in terms of MAE for my HOMO energy predictions is 0.1 eV or lower. In experiments, HOMO energies are commonly determined with a resolution of several tenth of eV, while errors of state-of-the-art computational spectroscopy methods typically range between 0.1 and 0.3 eV.

In addition to three different datasets, I also compare KRR performance based on two molecular descriptors. The first one is the widely-used CM, which is a simple and easy to compute representation of molecular structures, usually yielding fast predictions at low cost. The second descriptor is the constant-size MBTR representation based on interatomic many-body terms including bonding and angular information. While previous work already found that the CM descriptor can easily be surpassed in terms of machine learning performance by more sophisticated representations [105, 159, 165], my aim is to specifically quantify the accuracy achieved with the CM in comparison to the costlier MBTR.

In the following, I first describe the learning curves resulting from KRR training on the different datasets. Then, I quantify the accuracy of my HOMO energy predictions with respect to the desired accuracy of 0.1 eV. After that, I compare the performance between MBTR and CM. Finally, I discuss KRR performance with respect to chemical diversity of the three different datasets.

Learning curves Figure 5.1 shows the MAE on the test set as a function of training set size – the learning curves – for the three different datasets QM9, AA and OE62, using both the CM and MBTR descriptors. For each

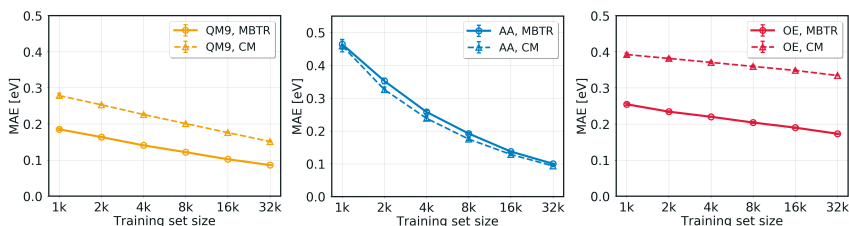


Figure 5.1. MAE as a function of training set size ("learning curves") for molecular orbital energy predictions with KRR trained on three different datasets QM9, AA and OE62. The MAE predictions are reported on 10k out-of-sample molecules from the test set of the respective dataset. Two different molecular representations CM and MBTR are used. Reproduced from Publication III.

dataset, a KRR model is trained on differently sized training sets (1k, 2k, 4k, 8k, 16k and 32k). For each training size, a 5-fold cross-validation is performed to determine the KRR hyperparameters, whose optimal values are in accordance with results from Publication I. For simplicity, MBTR hyperparameters are fixed to their optimal values found in Publication I and therefore are not part of the cross-validation routine within this work. After model training, HOMO energies are predicted for 10k molecules from the test set of the respective dataset, which were not used for training. These out-of-sample predictions on the test set are performed for each training set size, yielding the MAE between predicted and DFT-computed reference HOMO energies. This procedure of cross-validated training and out-of-sample testing is repeated 10 times for each training set size and the average MAE over these 10 runs is reported in the learning curves.

As expected, the MAE decreases with increasing training set size for all datasets. QM9 exhibits the lowest MAE, followed by AA for slightly larger training set sizes. Contrary, the MAEs for OE62 are roughly twice as high as those for QM9, throughout all training set sizes. The learning curves of AA have the steepest slope ("fast learning"), while the curves for OE62 are almost flat.

Quantifying the accuracy of HOMO energy predictions The highest accuracy for HOMO energy predictions is achieved for the QM9 dataset. For a training set size of 32k molecules, an MAE of 0.086 eV is obtained with the MBTR descriptor and an MAE of 0.151 eV is obtained with the CM descriptor. Thus, my objective of accomplishing a prediction error below 0.1 eV is fulfilled for the QM9 dataset when using the MBTR descriptor. This is also true for the AA dataset, where a prediction error of 0.100 eV is obtained (training size of 32k, MBTR descriptor). However, for the OE62 dataset, the prediction error amounts to 0.173 eV for the largest training set size of 32k in combination with the MBTR descriptor. Thus, my goal of MAEs below 0.1 eV is not yet achieved for OE62. Larger training set sizes might be necessary to improve the KRR performance on this more complex dataset. Future work might also consider the use of more

sophisticated methods, such as MBTR descriptors that include torsional angle information of the molecule (four-body terms), or artificial neural networks.

The results achieved on the QM9 dataset allow a comparison with previous work. Recently, Faber *et al.* [105] reported HOMO energy predictions with an out-of-sample MAE of 0.095 eV, using a molecular representation based on interatomic many-body expansions including bonding, angular and higher-order terms, [165], which is very similar to the here applied MBTR. The MAE reported in [105] was achieved with a KRR model trained on 118k molecules from the QM9 dataset. For the CM descriptor, an MAE of 0.133 eV was reported in the same study. My results with the QM9-trained KRR model are in excellent agreement with this work, while the training set sizes employed in my study are notably smaller.

CM vs. MBTR Next, I compare the performance of the two molecular descriptors. It can be seen in Figure 5.1 that for QM9 and OE62, the learning curves associated with the MBTR descriptor exhibit significantly lower MAEs than those associated with the CM. This is true throughout all training set sizes. Hence, the MBTR clearly outperforms the CM for these two datasets. This is in line with results from Publication II and can be explained by the high amount of information on atom types, their bond lengths and angles encoded in the MBTR. Contrary, for AA the MBTR and CM learning curves lie within the same statistical errors. The complexity in AA is dominated by the torsional angles, however, the MBTR employed in Publication III lacks torsional angle information (four-body terms). Meanwhile, the redundant chemical information of similar bonding patterns in AA benefits the performance of the CM. Also for the other two datasets, the qualitative performance of the CM is acceptable, considering that CM-training is performed at a fraction of the cost necessary for MBTR-based training (CM-based training with cross-validation takes about 2 hours for a training set size of 2k, vs. 12 hours for MBTR-based training). The CM representation is simple to compute, provides benchmark results that can be compared to related studies and constitutes a convenient tool to preliminarily study large unknown datasets.

Chemical diversity of the datasets Based on a first diversity analysis of the datasets in Chapter 3, I will now further study the chemical diversity governing the three datasets. Figure 5.2 illustrates the dataset diversity in input and output space. Panel a) represents the diversity in input space and shows a histogram of pairwise Euclidean distances between 2k randomly chosen molecules within each dataset. The molecules are represented by their MBTRs. Euclidean distances between molecules within QM9 and AA are centered around small values, indicating high molecular similarity within these two datasets. Contrary, the OE62 distances are distributed evenly over a large and wide range of values, indicating a high

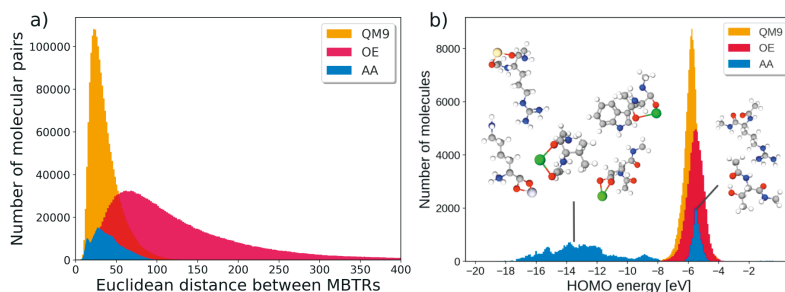


Figure 5.2. Illustration of dataset diversity within QM9, AA and OE62. a) Dataset diversity in input space, as seen by the MBTR descriptor. Shown is the distribution of pairwise Euclidean distances within each dataset between 2k randomly chosen molecules. The molecules are represented by their MBTRs. Euclidean distances within QM9 and AA are confined to small values, while distances within OE62 are distributed evenly over a larger and wider range. b) Dataset diversity in target space. Shown are distributions of DFT-computed reference HOMO energies. Almost all energies of QM9 and OE62 are centered around -6 eV, while the majority of HOMO energies in AA are distributed over a wider range of values. Example molecules from AA are shown together with their energy location. HOMO energies with large negative values correspond to amino acids or dipeptides with one of six added metal cations Ca^{2+} , Ba^{2+} , Sr^{2+} , Cd^{2+} , Pb^{2+} or Hg^{2+} . HOMO energies centered around -6 eV represent structures with no added cation. Reproduced from Publication III.

degree of dissimilarity between the molecules. This is consistent with the findings in Figure 3.3 in Section 3, where the diversity of molecular input space is visualized by the *t*-SNE tool, which is also based on pairwise similarities between molecules. Moreover, the distribution of Euclidean distances in Figure 5.2 resembles the distributions of molecular size and element diversity shown in Figure 3.2. Hence, there here employed analysis of the input space confirms that OE62 is the most diverse of all three datasets. OE62 consists of large and structurally diverse molecules, offering a variety of different backbones and functional groups. QM9 and AA both contain similar structures of smaller-sized molecules.

Panel b) of Figure 5.2 reveals the dataset diversity in target space by showing distributions of the DFT pre-computed HOMO energies. The HOMO energies in QM9 and OE62 are centered around the same value at -6 eV, while HOMO energies in AA are widely distributed over negative values. These energies correspond to amino acids or dipeptides with one of six metal cations Ca^{2+} , Ba^{2+} , Sr^{2+} , Cd^{2+} , Pb^{2+} or Hg^{2+} added to the bare structure. The metal ions are responsible for the shift towards lower energies. Only a fraction of the HOMO energies in AA is located around -6 eV. These correspond to bare structures without cations.

Dependence of KRR performance on dataset diversity With information about the diversity of the three different datasets, I will now elaborate on the KRR performance previously seen in Figure 5.1. The chemical diversity of the datasets is reflected in the KRR learning curves. The MAE for QM9 is low, even for small training set sizes. Given the high similarity

of the QM9 input and output space, KRR benefits from mapping similar molecular structures to similar HOMO energies.

The AA dataset exhibits high MAEs for small training set sizes, which quickly decrease with increasing training set size. This can be explained by the fact that AA contains many similar amino acid sidechains, but with different attached metal cations. For small training set sizes the prediction error is high because there are not enough similar structures per cation available for learning. With increasing training set size, more similar structures with the same cation are presented to the learning algorithm and the error decreases quickly.

The prediction errors for the OE62 dataset are high for all training set sizes. OE62 exhibits high similarity in target space, but the input structures are widely spread throughout chemical space. It is therefore challenging for KRR to map from the diverse chemical space of input structures to its confined target space of HOMO energies. Thus, the learning is slow. For a training set size of 32k the MAE on the test set is still as high as 0.173 eV, which is above the earlier mentioned desired accuracy of 0.1 eV for orbital energy predictions.

In summary, Publication III demonstrates that the KRR prediction error crucially depends on the diversity of the dataset it was trained on. QM9 has become the standard benchmark for machine learning predictions of various molecular properties and yields models with very low prediction errors. However, this dataset exhibits low chemical diversity. When developing KRR models on other datasets of real molecules with more complex structures, one should not expect a similar good model performance. Recently, another study [168] came to a similar conclusion. In particular, the performance of QM9-trained machine learning models were assessed on a subset taken from the more diverse PubChem database. Then, PubChem-trained models were used to predict properties from QM9. It was found that the latter case yields considerably better predictions accuracies. For future work, it would thus be interesting to examine the generalization ability of my OE62-trained KRR models. That is, one could use my OE62-trained model to predict HOMO energies for QM9 and AA, and vice versa. Since OE62 is the most diverse dataset among them, I expect that my OE62-trained KRR model will be able to predict properties of QM9 and AA molecules with satisfying accuracy.

5.2 Prediction of photoemission spectra with deep neural networks

In Publication IV, I participated in predicting photoemission spectra with deep neural networks. In the past years, several studies in materials science have addressed the prediction of spectra or spectral properties with

machine learning. Examples include the prediction of spectral features, such as peak intensities of MS^2 spectra with ANNs [169] or the prediction of absorption spectra based on images of materials [170] with deep neural networks. Recently, a study was published that employs surface-enhanced Raman scattering (SERS) spectra of DNA molecules as input for a deep neural network in order to identify specific DNA targets [171]. Other studies applied bundles of SERS spectra to a CNN model in order to quantify the concentration of single molecules [172, 173]. However, to the best of my knowledge, the mapping of molecular structures to photoemission spectra has not yet been attempted.

In Publication IV, three types of neural networks are compared for the prediction of DFT-based photoemission spectra: A simple multilayer perceptron (MLP) network, a convolutional neural network (CNN) and a deep tensor neural network (DTNN). The architectures of these networks are described in Section 4.3. The task of each network is to map molecular structures – using the QM7 and QM9 datasets – to their photoemission spectra. While the MLP and the CNN use the CM descriptor as input, the DTNN learns its own internal representation based on nuclear charges and atomic interactions, which is similar to the MBTR. Thus, the DTNN only requires Cartesian coordinates of the atomic positions as input. To numerically represent photoemission spectra, the two approaches introduced in Section 2.2.3 are employed. The first type of spectra representation constitutes a discrete spectrum of 16 KS energies. The task of the network is to predict these 16 energies simultaneously. The second representation is obtained by broadening these 16 KS energies into a continuous curve made up of 300 discretized points on the energy range [-40, 0] eV. The task of the neural network is to predict these 300 points simultaneously.

Prediction of discrete energy states For the prediction of 16 discrete energies, we tested all three network types. However, the MLP is only applied to the QM7 dataset of 6k molecules, since training the fully-connected MLP on the QM9 dataset of 134k molecules was computationally too demanding. Figure 5.3 shows the resulting root mean squared errors (RMSE) of the three networks for each energy state. For QM9, the DTNN performs uniformly well for all energy states, with an average RMSE of 0.186 eV over all 16 states. The CNN exhibits low RMEs for energy states in the middle region, while high and low states exhibit larger errors. For the QM7 dataset, the DTNN performs worse than for QM9, especially for energy levels with high and low state numbers. The same dependence of model performance on the state location is true for the other two networks. This trend can be explained by the fact that for small molecules, the deeper states (11 to 16) correspond to electronic core states, which have a significantly higher absolute energy value than valence states. Due to the small size of QM7, not enough reference data are available to learn these core

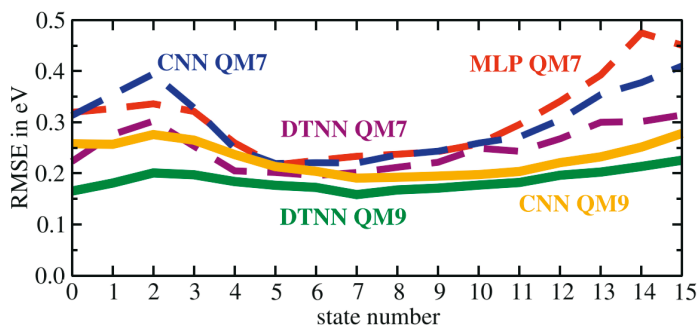


Figure 5.3. Performance of the DTNN, CNN and MLP for the prediction of 16 discrete molecular energy states. Shown are the root mean square error (RMSE) for each of the 16 states and for the two datasets QM9 and QM7. The energy states are labeled in descending order beginning from the HOMO (state number 0). Adapted from Publication IV.

states properly. The DTNN performs notably better for all QM7-states than the CNN and MLP. Another finding worth noting is that all neural networks perform better on the larger QM9 dataset than on QM7. This confirms our previous findings from Publication III, where the prediction error decreases when more training data is available. However, the DTNN trained on QM7 almost outperforms the CNN trained on QM9, proving that a carefully designed network architecture can learn from very little data.

Spectra prediction For the prediction of continuous spectra, only the CNN and the DTNN were employed. The MLP was left out, because its architecture would require 300 output neurons, which is too large for training on QM7 or QM9. For the CNN and the DTNN, 90% of the QM7 and QM9 data are used for training and the rest for testing. Figure 5.4 shows the spectra predicted by the CNN and DTNN on a test set of 13k QM9 molecules. The left panels show the relative spectral error (RSE) between the predicted and the reference spectrum. The RSE distribution is narrow, with an average RSE of 4% for the CNN and 3% for the DTNN. For both networks, three predicted spectra (orange) are shown with respect to the reference spectrum (green). These spectra represent the best, average and worst predictions made by the CNN and DTNN. The best predictions are able to capture all features of the reference spectrum. The average prediction of the CNN misses spectral features, but captures the average shape of the spectrum well, while the worst CNN prediction does not resemble the reference spectrum well. In contrast, the DTNN is able to capture most spectral features in the average and even worst prediction, while averaging through some features.

The DTNN outperforms the CNN in the prediction of 16 discrete energy states as well as in the prediction of continuous photoemission spectra. The reason for this is that the DTNN builds its own internal molecular

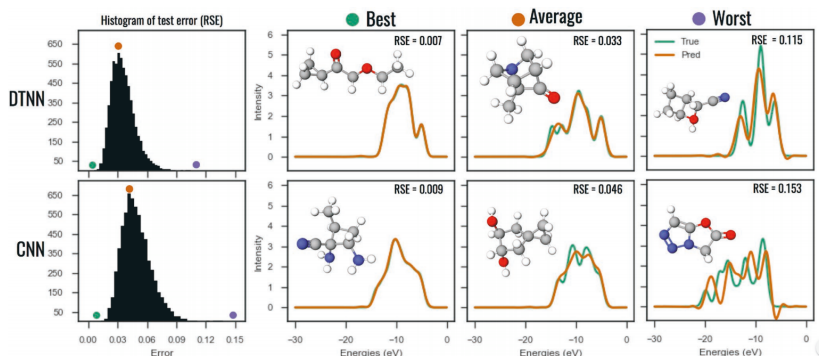


Figure 5.4. Comparison of CNN and DTNN performance on spectra prediction. The first column shows RSE histograms for 13k test molecules of QM9. The three following columns show spectra for the best, an average and the worst predictions and the corresponding DFT-based reference spectrum. The colored circles mark positions of selected molecules in the RSE histogram. Reproduced from Publication IV.

representation, which is similar to the sophisticated MBTR descriptor. Contrary, the CNN uses the CM descriptor as input. We saw in the previous Section 5.1 that for machine learning with KRR, the MBTR typically leads to a better prediction performance than the CM. I expect the same to be true for the CNN. Therefore, for future work, it would be interesting to employ the CNN in combination with the MBTR descriptor and to compare its performance again to the DTNN.

5.3 KRR and ANN predictions for materials discovery

The discovery of new materials would be greatly facilitated if one could scan through a large list of possible candidate compounds and filter this list based on pre-computed properties that give information about the usability of the material for a certain application. In Publications III and IV, such lists containing spectral properties and energies were generated by applying the KRR and DTNN models to a dataset of 10k diastereomers (in the following referred to as '10k dataset'), which is introduced in Section 3.1. Made up of 9,868 isomers of 6,095 parent $C_7H_{10}O_2$ isomers from the QM9, the 10k dataset is a special dataset of same-sized molecules and highly similar structures. Hence, this dataset is used only to illustrate the principle of how the developed methodology could be used to gain instant energy predictions for a collection of new molecules with unknown spectra, but not to discover any new or useful compounds.

The 10k dataset contains only molecular structures, but no pre-computed orbital energies or spectra. Computing the orbital energies with DFT would require significant time and effort. In Publication III, I therefore use my KRR model – trained on 32k QM9 molecules with the MBTR descriptor

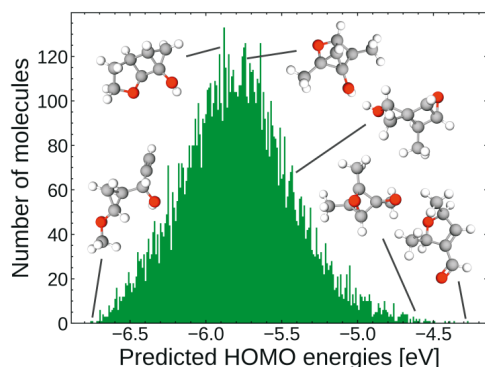


Figure 5.5. Distribution of predicted HOMO energies for 10k diastereomers, produced by the QM9-trained KRR model. MBTR descriptors of the 10k diastereomers were used as input for the KRR model. Molecules falling within a certain energy range can easily be detected and further studied for potential applications, e.g. organic electronic applications. Reproduced from Publication III.

– to predict the HOMO energies of all molecules in the 10k dataset.

In this case, it is a good choice to use my QM9-trained KRR model because the target molecules in the 10k dataset – while not part of QM9 – are highly similar to their 6,095 parent isomers from QM9. Hence, my QM9-trained KRR model will likely deliver reliable HOMO energy predictions for the 10k isomers, especially since it yields very low MAEs on the QM9 test set. To predict the HOMO energies for the 10k dataset, all that needs to be done is compute the input MBTRs for the 10k molecules and use them as input for my KRR model.

As a result, we instantly obtain a spread of predicted HOMO energies, as shown in Figure 5.5. The distribution of HOMO energies gives an instant overview over the energetic characteristics of the new 10k dataset. By scanning the distribution, one can quickly identify potentially interesting molecules based on a pre-defined range of energy values. These molecules could subsequently be further studied with more accurate computational or experimental methods to better assess their applicability for a certain desired application. This shows that my KRR model can produce fast and cheap energy predictions serving as a preliminary scan to discover promising structures within a certain energy range.

In a similar fashion, the DTNN from Publication IV trained on 120k QM9-molecules is employed to produce continuous photoemission spectra for all 10k diastereomers. Since the DTNN builds its own molecular representation, only the x, y, z -molecular structures are needed as input, which are already available in the dataset. Instant spectra predictions are obtained at no further computational cost. Figure 5.6a) shows a scan of spectral weights for all 10k molecules. The spectral weights indicate at which energy location the peaks of a predicted molecular spectrum occur.

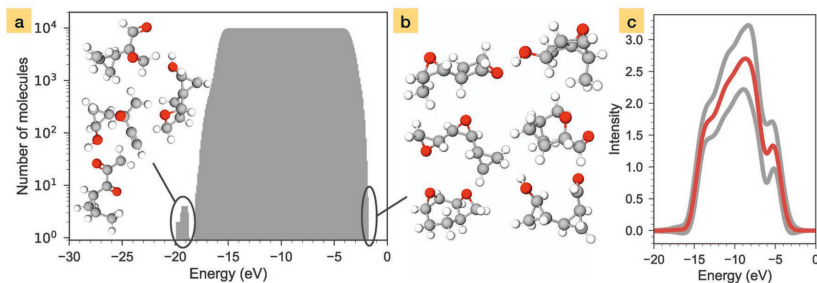


Figure 5.6. Scan of predicted spectra for the dataset of 10k diastereomers, produced by the DTNN. XYZ structures were used as input for the DTNN. Panel a) shows the distribution of spectral intensities for the 10k molecules. The four molecules with spectral weight around -20 eV are outliers. Panel b) shows six molecules that have the highest spectral intensity. Panel c) depicts the average predicted spectrum of all 10k molecules (red line). The grey lines represent the deviations from this average spectrum. Reproduced from Publication IV.

For the 10k dataset, spectral weights are uniformly distributed between -18 and -2 eV. Individual molecules can be easily detected, for example, the four outlier molecules with lowest spectral weights, whose peaks in the predicted spectrum lie below -18 eV, or the six molecules shown in panel b) with highest spectral weight values. Numerous promising structures, e.g. molecules with peaks in a particular region of its spectrum, can be further studied to determine their functional properties for future applications. The average predicted spectrum of all molecules in the dataset is shown in panel c). This is the typical photoemission spectrum to expect for molecules of this dataset.

While the size of the 10k dataset is by far not large and diverse enough to explore chemical space, one can easily transfer this idea to larger and more diverse datasets that contain more interesting molecules. My KRR model trained on the OE62 dataset would potentially be able to produce insightful energy scans for large collections of organic molecules that might be of interest for optoelectronic devices. However, the current prediction error of my OE62-trained KRR model is too high to apply it to new datasets. Alternatively, the DTNN might be able to achieve lower prediction errors on OE62. Hence, future work may focus on improving the accuracy of my OE62-trained KRR model, or on training the DTNN or other types of deep neural networks on OE62. To further increase the accuracy of future machine learning models, the subset of 5k molecules from OE62, whose orbital energies were computed with the *GW* method, might prove especially valuable. While KS eigenvalues from DFT computations were used for simplicity so far, the KRR and ANN methods presented in this dissertation are in principle transferable to datasets of better computational accuracy.

However, due to the high computational cost of *GW*, there will always be more data available from lower fidelity methods such as DFT, which can produce large datasets of KS eigenvalues at manageable cost. Future

work might consider Δ -learning techniques that are able to utilize data computed at different fidelity levels, by learning the difference between these data [59, 64, 70].

5.4 Summary

In this chapter, results of Publications III and IV were presented, providing an answer to the final research question in this thesis,

RQ4 How can the chosen machine learning methods be applied in practice to predict molecular orbital energies and photoemission spectra and to explore chemical space?

The success of machine learning for molecular spectra and energy predictions is an interplay between many factors: One needs to choose a suitable molecular descriptor, acquire large, diverse and consistent data, employ a machine learning method that fits the problem and tune the model hyperparameters.

Publication III, which applies KRR to three molecular datasets of different chemical composition using two types of molecular descriptors, emphasizes the interdependence of different factors in machine learning. The predictive performance of KRR not only depends on the training set size and molecular representation, but also on the chemical diversity within the dataset. Fast decreasing learning curves and low MAEs are observed for the standard QM9 dataset of small organic molecules and simple bonding patterns. This is also true for the AA dataset which comprises a limited collection of amino acids and dipeptides. OE62, on the contrary, is more difficult to learn. This dataset consists of comparably large molecules with complex electronic structures and rare functional groups. As a result, nearly flat learning curves and high MAEs are observed. Publication IV employs three types of neural networks for photoemission spectra prediction. It is shown that, besides the training set size, the type of neural network critically determines the predictive accuracy. The DTNN achieves an accuracy of 97% on the QM9 dataset and is able to capture most features of individual spectra, while the CNN typically misses certain spectral characteristics, such as peaks.

Once trained, both the KRR and the DTNN models have the potential to explore chemical space and facilitate the discovery of new materials. Within the blink of an eye and at no further computational cost, predictions can be made for large collections of new molecules for which spectral information is not yet available.

6. Summary and Outlook

This chapter summarizes my research objectives and achieved results. An outlook to future work concludes this dissertation.

6.1 Summary

The objective of this dissertation was the implementation of machine learning models to advance the discovery of new materials and to facilitate the exploration of chemical space. The desired models learn to map molecular structures to their corresponding photoemission spectra and orbital energies.

The first chapter of the thesis motivated the importance of photoemission spectra and frontier orbital energies. Spectroscopy is an important concept in the natural sciences and one of the primary techniques to characterize materials. Photoemission spectra and frontier orbital energies play an important role in the discovery and development of materials for optoelectronic applications, such as solar cells or OLEDs, which are made of organic molecules or polymers. Experimental measurements of photoemission spectra in synchrotron facilities are laborous and time-consuming. The same is true for state-of-the-art quantum-mechanical calculations with *GW* or density functional theory (DFT). While constituting indispensable parts of scientific research, experimental and computational spectroscopy are not (yet) suitable to explore chemical space on a large scale. Due to the unmanageable size of the chemical compound space, it is impossible to explore all structures case by case.

The key idea of this thesis was to exploit redundancy in DFT reference calculations by using machine learning. DFT calculations performed for a series of related molecules often contain redundant information. This redundancy can be utilized by performing only a limited number of expensive reference DFT computations and interpolating between them to obtain approximate results for the remaining molecules. This approach of interpolating between given data and inferring estimated solutions for the

rest is what machine learning is based on.

Motivated by this idea, I followed a 5-step machine learning workflow: (i) acquiring reference data, (ii) representing them so that the machine has all necessary information about the molecule, (iii) choosing a suitable machine learning method, (iv) optimizing the model hyperparameters and finally (v) train the model on the acquired reference data to predict spectra and energies. For each step I formulated a research question, which I answered throughout the dissertation.

The first step – acquiring and curating reference data – was one of the most time-consuming steps of the entire workflow. I first reviewed existing datasets in the materials science field and then defined criteria that a dataset should fulfill to be of use for my work. The three chosen datasets – QM9, QM7 and AA – all include molecular structures in form of Cartesian coordinates of atomic positions, but not all provide molecular spectra or orbital energies. I therefore produced my own reference data by optimizing molecular structures and computing orbital energies with DFT, using the same method and settings for all datasets to ensure consistency. While DFT is not able to correctly describe excited states, I used the KS eigenvalues from DFT to approximate molecular orbital energies. For large datasets of tens of thousands of molecules, *GW* would be too expensive. Within Publication I, I generated a new diverse spectroscopy dataset, OE62, that contains 62k organic molecules including orbital energy calculations with DFT, as well as *GW* results of higher numerical accuracy for a subset of 5k molecules. This 5k subset may be used in future work to refine the methodology presented in this thesis.

The second step was to represent the spectra and molecules in such a way that the machine is able to relate molecular inputs to spectral outputs. In this thesis, two descriptors of the molecular structure were used, i.e. the simple and cheap Coulomb matrix (CM), encoding atom types and positions, and the more sophisticated many-body tensor representation (MBTR), encoding atom types, pairwise interactions and angular information. Spectra were represented by either 16 discrete DFT-based KS energies or by broadening these discrete energy values into a continuous curve of 300 points.

Next, I presented the machine learning methods applied in my work: kernel ridge regression (KRR), artificial neural networks (ANNs) and Bayesian optimization (BO). KRR can map multi-dimensional inputs to one-dimensional outputs, which is the reason that I chose this method for the prediction of scalar HOMO energies. ANNs are able to map multiple inputs to multiple outputs simultaneously and were therefore a natural choice for the prediction of photoemission spectra. Before applying these methods in practice, an important step is the choice of the model hyperparameters. This can often be a burden, since it is not always obvious which hyperparameters perform best for a given problem setting. In Publication

II, I applied the BO tool BOSS to optimize up to 4 hyperparameters of my KRR model and compared the efficiency of BOSS with the commonly used grid search method. In the 4D search, BOSS was significantly faster in finding the optimal hyperparameter solution, requiring only a fraction of iterations necessary in grid search. Moreover, BOSS provided scoring landscapes in hyperparameter space that enabled a deeper analysis of my KRR model and that may facilitate the choice of starting points for related problems in the future.

In the final step, I applied KRR and ANNs to the generated reference datasets for the prediction of HOMO energies and photoemission spectra. In Publication III, KRR was used to predict HOMO energies of QM9, AA and OE62 molecules. Due to the different chemical diversities within these datasets, KRR performance varied notably. The standard QM9 dataset and the AA dataset of amino acids are easy to learn due to their relatively simple and small molecular structures. KRR models trained on these two datasets are able to achieve MAEs below or equal to 0.1 eV, which is the desirable accuracy for orbital energy predictions in this thesis. Contrary, the OE62 dataset of complex and technologically relevant molecules is more difficult to learn. This underlines the effect of a dataset's diversity on machine learning performance. While QM9 is the golden standard for many studies in materials science, it is relatively easy to achieve good machine learning results with models trained and tested on QM9. However, it is not realistic to achieve the same or a comparably good performance on other, more complex molecular datasets. To further improve the KRR performance on the OE62 set, larger training set sizes, a more sophisticated molecular descriptor or the employment of ANNs might prove helpful.

In Publication IV, two types of deep neural networks and one feed-forward neural network were employed to predict photoemission spectra. The deep tensor neural network (DTNN) is the most sophisticated network and is able to predict QM9 spectra with an accuracy of 97%, capturing all essential characteristics of the reference spectra.

In the final step of the thesis, I also showcased the applicability of my machine learning models for chemical space exploration. For a dataset of previously unseen molecules, for which no spectra or energies are available, the KRR and DTNN models were employed to produce instant predictions and negligible computational cost. Within seconds, distributions of HOMO energies and spectral intensities were produced, from which interesting molecules that fell within a certain energy range could be easily identified. These structures could be studied in more detail with electronic structure methods or experiments. This example showcased that structures no longer need to be randomly picked from a pool of candidates, but can be selected in an informed manner based on approximate energy predictions. This closes the circle to the initial idea of advancing chemical space exploration with machine learning.

6.2 Outlook

As new materials data continue to emerge in public databases, the developed machine learning methodology in this dissertation will be a valuable tool for future work. Many of my co-workers already employ my KRR model developed in this thesis for their own research. I here discuss open questions that might be worth addressing and potential future research based on the results of this dissertation.

First, the KRR and ANN models employed in this thesis were trained on DFT-computed KS orbital energies, since they are cheaper to produce for large datasets compared to *GW*. This proved a convenient way to build "prototype" machine learning models and to develop my methodology. However, future work might consider training KRR and ANNs on energies computed with a more appropriate method, such as *GW*. Orbital energies computed with *GW* were made available for a 5k-subset of OE62 within this thesis. While this dataset is not large in size, one could combine its high-fidelity data with lower-fidelity data from DFT computations, which are available for the entire 62k molecules. This approach is known as multi-fidelity learning.

This hints to the second open point, which is the low predictive KRR performance on the OE62 dataset. It might help to include high-fidelity data of the 5k subset into the KRR training in order to improve the performance of the model. As mentioned before, other options are using different molecular descriptors that include torsional angle information or to training a deep neural network on OE62 data.

Another interesting question is how well machine learning models trained on OE62 would perform on a simpler and less diverse dataset, such as QM9 or AA. I would expect that due to the high diversity of OE62, KRR models trained on OE62 data will be able to predict the energies of QM9 and AA molecules easily.

References

- [1] C. Dobson, “Chemical space and biology,” *Nature*, vol. 432, pp. 824–8, 01 2005.
- [2] J.-L. Reymond, R. van Deursen, L. C. Blum, and L. Ruddigkeit, “Chemical space as a source for new drugs,” *Med. Chem. Commun.*, vol. 1, pp. 30–38, 2010.
- [3] R. S. Bohacek, C. McMartin, and W. C. Guida, “The art and practice of structure-based drug design: A molecular modeling perspective,” *Medicinal Research Reviews*, vol. 16, no. 1, pp. 3–50, 1996.
- [4] J. Clayden, N. Greeves, and S. Warren, *Organic Chemistry*. Oxford University Press, 2012.
- [5] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton, “PubChem 2019 update: improved access to chemical data,” *Nucleic Acids Research*, vol. 47, pp. D1102–D1109, 10 2018.
- [6] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, “The high-throughput highway to computational materials design,” *Nature materials*, vol. 12 3, pp. 191–201, 2013.
- [7] L. Himanen, A. Geurts, A. S. Foster, and P. Rinke, “Data-driven materials science: Status, challenges, and perspectives,” *Advanced Science*, vol. 6, no. 21, p. 1900808, 2019.
- [8] “The NOMAD Laboratory.” <https://nomad-repository.eu>. Accessed: 2020-03-01.
- [9] “The Materials Project.” <https://materialsproject.org>. Accessed: 2020-03-01.
- [10] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation,” *APL Materials*, vol. 1, no. 1, p. 011002, 2013.
- [11] F. H. Allen, “The Cambridge Structural Database: a quarter of a million crystal structures and rising,” *Acta Crystallogr. B*, vol. 58, no. 3 Part 1, pp. 380–388, 2002.
- [12] “The Cambridge Crystallographic Data Centre (CCDC).” <https://www.ccdc.cam.ac.uk/>. Accessed: 2020-03-01.
- [13] “Materials Cloud.” <https://www.materialscloud.org>. Accessed: 2020-03-01.
- [14] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

- [15] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, “Recent advances and applications of machine learning in solid-state materials science,” *npj Computational Materials*, vol. 5, pp. 1–36, 2019.
- [16] T. Müller, A. Kusne, and R. Ramprasad, *Machine Learning in Materials Science: Recent Progress and Emerging Applications*, pp. 186–273. 05 2016.
- [17] K. T. Butler, D. W. Davies, H. M. Cartwright, O. Isayev, and A. Walsh, “Machine learning for molecular and materials science,” *Nature*, vol. 559, pp. 547–555, 2018.
- [18] Y. Liu, T. Zhao, W. Ju, and S. Shi, “Materials discovery and design using machine learning,” *Journal of Materiomics*, vol. 3, no. 3, pp. 159 – 177, 2017. High-throughput Experimental and Modeling Research toward Advanced Batteries.
- [19] G. Ravanhani Schleder, A. C. Padilha, C. Acosta, M. Costa, and A. Fazzio, “From dft to machine learning: recent approaches to materials science – a review,” *Journal of Physics: Materials*, 02 2019.
- [20] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, “Machine learning and materials informatics: Recent applications and prospects,” *npj Computational Materials*, vol. 3, 07 2017.
- [21] J. E. Gubernatis and T. Lookman, “Machine learning in materials design and discovery: Examples from the present and suggestions for the future,” *Phys. Rev. Materials*, vol. 2, p. 120301, Dec 2018.
- [22] A. Agrawal and A. Choudhary, “Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science,” *APL Materials*, vol. 4, no. 5, p. 053208, 2016.
- [23] J. Hill, G. Mulholland, K. Persson, R. Seshadri, C. Wolverton, and B. Meredig, “Materials science with large-scale data and informatics: Unlocking new opportunities,” *MRS Bulletin*, vol. 41, no. 5, p. 399–409, 2016.
- [24] A. Jain, G. Hautier, S. P. Ong, and K. Persson, “New opportunities for materials informatics: Resources and data mining techniques for uncovering hidden relationships,” *Journal of Materials Research*, vol. 31, no. 8, p. 977–994, 2016.
- [25] D. Golze, M. Dvorak, and P. Rinke, “The GW compendium: A practical guide to theoretical photoemission spectroscopy,” *Frontiers in Chemistry*, vol. 7, p. 377, 2019.
- [26] A. Saeki and K. Kranthiraja, “A high throughput molecular screening for organic electronics via machine learning: present status and perspective,” *Japanese Journal of Applied Physics*, vol. 59, p. SD0801, dec 2019.
- [27] R. Ma, *Organic Light-Emitting Diodes (OLEDs)*, pp. 1–22. 01 2016.
- [28] H. Klauk, *Organic electronics: materials, manufacturing, and applications*. John Wiley & Sons, 2006.
- [29] H. Klauk, *Organic electronics II: more materials and applications*, vol. 2. John Wiley & Sons, 2012.
- [30] R. Farchioni and G. Grosso, *Organic electronic materials: conjugated polymers and low molecular weight organic solids*, vol. 41. Springer Science & Business Media, 2013.
- [31] W. Brütting, “Introduction to the physics of organic semiconductors,” *Physics of organic semiconductors*, pp. 1–14, 2005.

- [32] W. Hu, F. Bai, X. Gong, X. Zhan, H. Fu, and T. Bjornholm, *Organic optoelectronics*. John Wiley & Sons, 2012.
- [33] S. Ogawa, *Organic Electronics Materials and Devices*. Springer, 2015.
- [34] D. Golze, J. Wilhelm, M. J. van Setten, and P. Rinke, “Core-level binding energies from gw: An efficient full-frequency approach within a localized basis,” *Journal of Chemical Theory and Computation*, vol. 14, no. 9, pp. 4856–4869, 2018. PMID: 30092140.
- [35] R. M. Martin, *Electronic structure*. Cambridge University Press, 2008.
- [36] R. G. Parr and Y. Weitao, *Density functional theory of atoms and molecules*. Oxford Science Publications, 1994.
- [37] V. Brázdová and D. R. Bowler, *Atomistic computer simulations: a practical guide*. Wiley-VCH, 2013.
- [38] J. P. Perdew, K. Burke, and M. Ernzerhof, “Generalized gradient approximation made simple,” *Phys. Rev. Lett.*, vol. 77, pp. 3865–3868, 1996.
- [39] T. Stein, L. Kronik, and R. Baer, “Reliable prediction of charge transfer excitations in molecular complexes using time-dependent density functional theory,” *Journal of the American Chemical Society*, vol. 131, no. 8, pp. 2818–2820, 2009. PMID: 19239266.
- [40] R. Baer, E. Livshits, and U. Salzner, “Tuned range-separated hybrids in density functional theory,” *Annual Review of Physical Chemistry*, vol. 61, no. 1, pp. 85–109, 2010. PMID: 20055678.
- [41] H. R. Eisenberg and R. Baer, “A new generalized kohn–sham method for fundamental band-gaps in solids,” *Phys. Chem. Chem. Phys.*, vol. 11, pp. 4674–4680, 2009.
- [42] S. Pittalis and E. Räsänen, “Exchange-correlation potential with a proper long-range behavior for harmonically confined electron droplets,” *Phys. Rev. B*, vol. 82, p. 195124, Nov 2010.
- [43] J. Gräfenstein and D. Cremer, “The self-interaction error and the description of non-dynamic electron correlation in density functional theory,” *Theoretical Chemistry Accounts*, vol. 123, pp. 171–182, 06 2009.
- [44] E. Da Como and E. von Hauff, “The wspan reference on organic electronics: Organic semiconductors. materials and energy series. volume 1: Basic concepts, volume 2: Fundamental aspects of materials and applications. edited by jean-luc brédas and seth r. marder,” *Angewandte Chemie International Edition*, vol. 56, no. 18, pp. 4915–4916, 2017.
- [45] L. Hedin, “New method for calculating the one-particle green’s function with application to the electron-gas problem,” *Phys. Rev.*, vol. 139, pp. A796–A823, Aug 1965.
- [46] L. Rudigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, “Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17,” *J. Chem. Inf. Model.*, vol. 52, no. 11, pp. 2864–2875, 2012.
- [47] T. Fink, H. Bruggesser, and J.-L. Reymond, “Virtual exploration of the small-molecule chemical universe below 160 daltons,” *Angewandte Chemie International Edition*, vol. 44, no. 10, pp. 1504–1508, 2005.

- [48] T. Fink and J.-L. Reymond, "Virtual exploration of the chemical universe up to 11 atoms of c, n, o, f: Assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery," *Journal of Chemical Information and Modeling*, vol. 47, no. 2, pp. 342–353, 2007.
- [49] L. C. Blum and J.-L. Reymond, "970 million druglike small molecules for virtual screening in the chemical universe database gdb-13," *J. Am. Chem. Soc.*, vol. 131, no. 25, pp. 8732–8733, 2009.
- [50] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Sci. Data*, vol. 1:140022, 2014.
- [51] M. Rupp, "Machine learning for quantum mechanics in a nutshell," *Int. J. Quantum Chem.*, vol. 115, no. 16, pp. 1058–1073, 2015.
- [52] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, and A. Knoll, "Fourier series of atomic radial distribution functions: A molecular fingerprint for machine learning models of quantum chemical properties," *Int. J. quantum Chem.*, vol. 115, pp. 1084–1093, 2015.
- [53] R. Ramakrishnan and O. A. von Lilienfeld, "Many molecular properties from one kernel in chemical space," *CHIMIA International Journal for Chemistry*, vol. 69, no. 4, pp. 182–186, 2015.
- [54] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Phys. Rev. Lett.*, vol. 108, no. 5, p. 058301, 2012.
- [55] G. Montavon, K. Hansen, S. Fazli, M. Rupp, F. Biegler, A. Ziehe, A. Tkatchenko, O. A. von Lilienfeld, and K.-R. Müller, "Learning invariant representations of molecules for atomization energy prediction," *Advances in Neural Information Processing Systems*, pp. 440–448, 2012.
- [56] J. Barker, J. Bulin, J. Hamaekers, and S. Mathias, "Localized coulomb descriptors for the gaussian approximation potential," *arXiv:1611.05126*, 2016.
- [57] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, "Machine learning predictions of molecular properties: accurate many-body potentials and nonlocality in chemical space," *J. Phys. Chem. Lett.*, vol. 6, no. 12, pp. 2326–2331, 2015.
- [58] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagiota, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Machine learning of molecular electronic properties in chemical compound space," *New J. Phys.*, vol. 15, p. 095003, 2013.
- [59] R. Ramakrishnan, M. Hartmann, E. Tapavicza, and O. A. von Lilienfeld, "Electronic spectra from tddft and machine learning in chemical space," *The Journal of Chemical Physics*, vol. 143, no. 8, p. 084111, 2015.
- [60] M. Ropo, M. Schneider, C. Baldauf, and V. Blum, "First-principles data set of 45,892 isolated and cation-coordinated conformers of 20 proteinogenic amino acids," *Scientific Data*, vol. 3, 2 2016.
- [61] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, "Machine learning unifies the modeling of materials and molecules," *Science Advances*, vol. 3, no. 12, 2017.

- [62] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, “Comparing molecules and solids across structural and alchemical space,” vol. 18, pp. 13754–13769, May 2016.
- [63] N. Artrith, A. Urban, and G. Ceder, “Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species,” *Phys. Rev. B*, vol. 96, p. 014112, Jul 2017.
- [64] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, “Big data meets quantum chemistry approximations: The δ -machine learning approach,” *Journal of Chemical Theory and Computation*, vol. 11, no. 5, pp. 2087–2096, 2015. PMID: 26574412.
- [65] “The Harvard Clean Energy Project Database.” <https://www.re3data.org/repository/r3d100010708>. Accessed: 2020-03-21.
- [66] J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. M. Brockway, and A. Aspuru-Guzik, “The Harvard Clean Energy Project: Large-scale computational screening and design of organic photovoltaics on the world community grid,” *J. Phys. Chem. Lett.*, vol. 2, no. 17, pp. 2241–2251, 2011.
- [67] S. Lopez, E. Pyzer-Knapp, G. Simm, T. Lutzow, K. Li, L. Seress, J. Hachmann, and A. Aspuru-Guzik, “The harvard organic photovoltaic dataset,” *Scientific Data*, vol. 3, 09 2016.
- [68] F. M. Paruzzo, A. Hofstetter, F. Musil, S. De, M. Ceriotti, and L. Emsley, “Chemical shifts in molecular solids by machine learning,” *Nat. Comm.*, vol. 9, no. 4501, pp. 2041–1723, 2018.
- [69] F. M. Paruzzo, A. Hofstetter, F. Musil, S. De, M. Ceriotti, and L. Emsley, “Chemical shifts in molecular solids by machine learning datasets.” <https://archive.materialscloud.org/2019.0023/v1>. Accessed: 2020-03-21.
- [70] G. Pilania, J. Gubernatis, and T. Lookman, “Multi-fidelity machine learning models for accurate bandgap predictions of solids,” *Computational Materials Science*, vol. 129, pp. 156 – 163, 2017.
- [71] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, and D. Morgan, “Aflow: An automatic framework for high-throughput materials discovery,” *Computational Materials Science*, vol. 58, pp. 218 – 226, 2012.
- [72] S. Gorsse, M. Nguyen, O. Senkov, and D. Miracle, “Database on the mechanical properties of high entropy alloys and complex concentrated alloys,” *Data in Brief*, vol. 21, pp. 2664 – 2678, 2018.
- [73] A. Zakutayev, N. Wunder, M. Schwarting, J. Perkins, R. White, K. Munch, W. Tumas, and C. Phillips, “An open experimental database for exploring inorganic materials,” *Scientific data*, vol. 5, 4 2018.
- [74] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, “Ab initio molecular simulations with numeric atom-centered orbitals,” *Comp. Phys. Comm.*, vol. 180, pp. 2175–2196, 2009.
- [75] C. Schober, K. Reuter, and H. Oberhofer, “Virtual screening for high carrier mobility in organic semiconductors,” *The Journal of Physical Chemistry Letters*, vol. 7, no. 19, pp. 3973–3977, 2016. PMID: 27661442.
- [76] C. Kunkel, C. Schober, H. Oberhofer, and K. Reuter, “Knowledge discovery through chemical space networks: the case of organic electronics,” *Journal of molecular modeling*, vol. 25, no. 4, p. 87, 2019.

- [77] C. Kunkel, C. Schober, J. T. Margraf, K. Reuter, and H. Oberhofer, "Finding the right bricks for molecular legos: A data mining approach to organic semiconductor design," *Chemistry of Materials*, vol. 31, no. 3, pp. 969–978, 2019.
- [78] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [79] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Phys. Rev. Lett.*, vol. 104, p. 136403, Apr 2010.
- [80] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Phys. Rev. B*, vol. 87, p. 184115, May 2013.
- [81] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," *The Journal of Chemical Physics*, vol. 134, no. 7, p. 074106, 2011.
- [82] B. Huang and O. A. von Lilienfeld, "Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity," *The Journal of Chemical Physics*, vol. 145, no. 16, p. 161102, 2016.
- [83] H. Huo and M. Rupp, "Unified Representation for Machine Learning of Molecules and Crystals," *arXiv:1704.06439 [cond-mat, physics:physics]*, Apr. 2017. arXiv: 1704.06439.
- [84] W. J. Dunn, "Handbook of molecular descriptors. methods and principles in medicinal chemistry series. volume 11 by roberto Todeschini and viviana Consonni (universita degli studi di milano-bicocca). edited by r. Mannold, h. Kubinyi, and h. Timmerman. Wiley-VCH: Weinheim and New York. 2000. xxi + 668 pp. 498 dm. isbn 3-527-29913-0," *Journal of the American Chemical Society*, vol. 123, no. 29, pp. 7198–7198, 2001.
- [85] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties," *Phys. Rev. B*, vol. 89, p. 205118, May 2014.
- [86] Z. Li, J. R. Kermode, and A. De Vita, "Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces," *Phys. Rev. Lett.*, vol. 114, p. 096405, Mar 2015.
- [87] P. Simon, *Too big to ignore: the business case for big data*, vol. 72. John Wiley & Sons, 2013.
- [88] P. Langley, *Elements of Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995.
- [89] Brindha Priyadarshini Jeyaraman, Ludvig Renbo Olsen, and Monica Wambugu, *Practical Machine Learning with R*. PACKT PUBLISHING LIMITED, 1 ed., 2019.
- [90] E. Alpaydin, *Introduction to Machine Learning*. Adaptive Computation and Machine Learning, Cambridge, MA: MIT Press, 3 ed., 2014.
- [91] H. J. Escalante, M. Montes, and L. E. Sucar, "Particle swarm model selection," *Journal of Machine Learning Research*, vol. 10, no. 2, 2009.
- [92] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

- [93] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.
- [94] S. Heinen, M. Schwilk, G. Rudorff, and A. von Lilienfeld, "Machine learning the computational cost of quantum chemistry," *arXiv:1908.06714 [physics.chem-ph]*, 08 2019.
- [95] K. Müller, M. Finke, N. Murata, K. Schulten, and S. Amari, "A numerical study on learning curves in stochastic multilayer feedforward networks," *Neural Computation*, vol. 8, pp. 1085–1106, 7 1996.
- [96] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, pp. 1171–1220, 06 2008.
- [97] V. Vovk, *Kernel Ridge Regression*, pp. 105–116. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [98] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
- [99] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 12, pp. 181–201, 02 2001.
- [100] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [101] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [102] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [103] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory* (D. Helmbold and B. Williamson, eds.), (Berlin, Heidelberg), pp. 416–426, Springer Berlin Heidelberg, 2001.
- [104] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, and K.-R. Müller, "Assessment and validation of machine learning methods for predicting molecular atomization energies," *J. Chem. Theory Comput.*, vol. 9, no. 8, pp. 3404–3419, 2013.
- [105] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. von Lilienfeld, "Prediction errors of molecular machine learning models lower than hybrid dft error," *Journal of Chemical Theory and Computation*, vol. 13, no. 11, pp. 5255–5264, 2017. PMID: 28926232.
- [106] G. Montavon, K. Hansen, S. Fazli, M. Rupp, F. Biegler, A. Ziehe, A. Tkatchenko, A. V. Lilienfeld, and K.-R. Müller, "Learning invariant representations of molecules for atomization energy prediction," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 440–448, Curran Associates, Inc., 2012.
- [107] F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, "Machine learning energies of 2 million elpasolite (abC_2D_6) crystals," *Phys. Rev. Lett.*, vol. 117, p. 135502, Sep 2016.

- [108] F. A. Faber, A. S. Christensen, B. Huang, and O. A. von Lilienfeld, “Alchemical and structural distribution based representation for universal quantum machine learning,” *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241717, 2018.
- [109] F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, “Crystal structure representations for machine learning models of formation energies,” *International Journal of Quantum Chemistry*, vol. 115, no. 16, pp. 1094–1101, 2015.
- [110] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, “A survey on multi-output regression,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 5, pp. 216–233, 2015.
- [111] E. S. Xioufis, G. Tsoumakas, W. Groves, and I. P. Vlahavas, “Multi-target regression via input space expansion: treating targets as inputs,” *Machine Learning*, vol. 104, pp. 55–98, 2016.
- [112] J. Westermayr, F. A. Faber, A. S. Christensen, O. A. von Lilienfeld, and P. Marquetand, “Neural networks and kernel ridge regression for excited states dynamics of CH_2NH_2^+ : From single-state to multi-state representations and multi-property machine learning models,” *arXiv preprint arXiv:1912.08484*, 2019.
- [113] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
- [114] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, “Learning semantic representations using convolutional neural networks for web search,” in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374, 2014.
- [115] Y. LeCun, Y. Bengio, *et al.*, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
- [116] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.
- [117] J. Fu, H. Zheng, and T. Mei, “Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4438–4446, 2017.
- [118] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.
- [119] L. Deng, G. Hinton, and B. Kingsbury, “New types of deep neural network learning for speech recognition and related applications: An overview,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8599–8603, IEEE, 2013.
- [120] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [121] C. M. Bishop, *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc., 1996.
- [122] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

- [123] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [124] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, pp. 303–314, Dec. 1989.
- [125] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [126] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, *Efficient BackProp*, pp. 9–48. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [127] J. Behler, "Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations," *Phys. Chem. Chem. Phys.*, vol. 13, pp. 17930–17955, 2011.
- [128] J. Behler, "Representing potential energy surfaces by high-dimensional neural network potentials.," *Journal of physics. Condensed matter : an Institute of Physics journal*, vol. 26 18, p. 183001, 2014.
- [129] J. Behler, "Constructing high-dimensional neural network potentials: A tutorial review," *International Journal of Quantum Chemistry*, vol. 115, no. 16, pp. 1032–1050, 2015.
- [130] T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, "Neural network models of potential energy surfaces," *The Journal of Chemical Physics*, vol. 103, no. 10, pp. 4129–4137, 1995.
- [131] G. Magela E Silva, P. H. Acioli, and A. C. Pedroza, "Estimating correlation energy of diatomic molecules and atoms with neural networks," *Journal of Computational Chemistry*, vol. 18, no. 11, pp. 1407–1414, 1997.
- [132] S. Urata, A. Takada, T. Uchimaru, A. K. Chandra, and A. Sekiya, "Artificial neural network study for the estimation of the c–h bond dissociation enthalpies," *Journal of Fluorine Chemistry*, vol. 116, no. 2, pp. 163 – 171, 2002.
- [133] S. S. Haykin, *Neural networks and learning machines*. Upper Saddle River, NJ: Pearson Education, third ed., 2009.
- [134] Y. Le Cun, L. Bottou, G. B. Orr, and K.-R. Müller, *Neural networks: tricks of the trade (Lecture notes in Computer Science vol 1524)*. Springer, 1998.
- [135] K. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Communications*, vol. 8, 01 2017.
- [136] A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, "Insightful classification of crystal structures using deep learning," *Nature communications*, vol. 9, no. 1, pp. 1–10, 2018.
- [137] M. Ziatdinov, A. Maksov, and S. V. Kalinin, "Learning surface molecular structures via machine vision," *npj Computational Materials*, vol. 3, no. 1, pp. 1–9, 2017.
- [138] B. Alldritt, P. Hapala, N. Oinonen, F. Urtev, O. Krejci, F. F. Canova, J. Kanala, F. Schulz, P. Liljeroth, and A. S. Foster, "Automated structure discovery in atomic force microscopy," *Science Advances*, vol. 6, no. 9, p. eaay6913, 2020.

- [139] P. Lerman, “Fitting segmented regression models by grid search,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 29, no. 1, pp. 77–84, 1980.
- [140] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.
- [141] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 ed., 2009.
- [142] E. Brochu, V. M. Cora, and N. De Freitas, “A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning,” *arXiv preprint arXiv:1012.2599*, 2010.
- [143] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.
- [144] J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander, “Fundamentals and recent developments in approximate bayesian computation,” *Systematic biology*, vol. 66, no. 1, pp. e66–e82, 2017.
- [145] A. Törn and A. Žilinskas, *Global optimization*, vol. 350. Springer, 1989.
- [146] H. Kushner and G. Yin, “Stochastic approximation algorithms and applications, vol. 35 of stoch. modelling and appl,” *Prob., Springer-Verlag, New York*, 1997.
- [147] V. S. Borkar and S. P. Meyn, “The ode method for convergence of stochastic approximation and reinforcement learning,” *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 447–469, 2000.
- [148] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.
- [149] L. Younes, “Parametric inference for imperfectly observed gibbsian fields,” *Probability theory and related fields*, vol. 82, no. 4, pp. 625–645, 1989.
- [150] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [151] H. Wang, G. Wang, G. Li, J. Peng, and Y. Liu, “Deep belief network based deterministic and probabilistic wind speed forecasting approach,” *Applied Energy*, vol. 182, pp. 80–93, 2016.
- [152] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, “Hyperparameter optimization for machine learning models based on bayesian optimization,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26 – 40, 2019.
- [153] D. Yogatama and G. Mann, “Efficient transfer learning method for automatic hyperparameter tuning,” in *AISTATS*, 2014.
- [154] V. Perrone, H. Shen, M. Seeger, C. Archambeau, and R. Jenatton, “Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning,” in *NeurIPS*, 2019.
- [155] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, “Evaluation of a tree-based pipeline optimization tool for automating data science,” in *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO ’16*, (New York, NY, USA), p. 485–492, Association for Computing Machinery, 2016.

- [156] M. T. Young, J. Hinkle, A. Ramanathan, and R. Kannan, “[h]yperspace: Distributed bayesian hyperparameter optimization”, in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pp. 339–347, 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), 2018.
- [157] D. Dua and C. Graff, 2017.
- [158] M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, “Bayesian inference of atomistic structure in functional materials,” *npj Comp. Mat.*, vol. 5, no. 35, 2019.
- [159] C. R. Collins, G. J. Gordon, O. A. von Lilienfeld, and D. J. Yaron, “Constant size descriptors for accurate machine learning models of molecular properties,” *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241718, 2018.
- [160] K. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. Maurer, “Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions,” *Nature Communications*, vol. 10, p. 5024, 11 2019.
- [161] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller, “A universal density matrix functional from molecular orbital-based machine learning: Transferability across organic molecules,” *The Journal of Chemical Physics*, vol. 150, no. 13, p. 131103, 2019.
- [162] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “Schnet – a deep learning architecture for molecules and materials,” *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241722, 2018.
- [163] E. O. Pyzer-Knapp, K. Li, and A. Aspuru-Guzik, “Learning from the Harvard Clean Energy Project: The use of neural networks to accelerate materials discovery,” *Advanced Functional Materials*, vol. 25, no. 41, pp. 6495–6502.
- [164] F. Pereira, K. Xiao, D. A. R. S. Latino, C. Wu, Q. Zhang, and J. Aires-de Sousa, “Machine learning methods to predict density functional theory b3lyp energies of homo and lumo orbitals,” *Journal of Chemical Information and Modeling*, vol. 57, no. 1, pp. 11–21, 2017. PMID: 28033004.
- [165] B. Huang and O. A. von Lilienfeld, “Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity,” *The Journal of Chemical Physics*, vol. 145, no. 16, p. 161102, 2016.
- [166] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, p. 1263–1272, JMLR.org, 2017.
- [167] F. Hou, Z. Wu, Z. Hu, Z. Xiao, L. Wang, X. Zhang, and G. Li, “Comparison study on the prediction of multiple molecular properties by various neural networks,” *The Journal of Physical Chemistry A*, vol. 122, no. 46, pp. 9128–9134, 2018. PMID: 30285444.
- [168] M. Glavatskikh, J. Leguy, G. Hunault, T. Cauchy, and B. Da Mota, “Dataset’s chemical diversity limits the generalizability of machine learning predictions,” *Journal of Cheminformatics*, vol. 11, 12 2019.
- [169] Y.-M. Lin, C.-T. Chen, and J.-M. Chang, “MS2cnn: Predicting MS/MS spectrum based on protein sequence using deep convolutional neural networks,” *BMC Genomics*, vol. 20, 12 2019.

References

- [170] H. S. Stein, D. Guevarra, P. F. Newhouse, E. Soedarmadji, and J. Gregoire, "Machine Learning of Optical Properties of Materials - Predicting Spectra from Images and Images from Spectra," 7 2018.
- [171] H. Shi, H. Wang, X. Meng, R. Chen, Y. Zhang, Y. Su, and Y. He, "Setting up a surface-enhanced raman scattering database for artificial-intelligence-based label-free discrimination of tumor suppressor genes," *Analytical chemistry*, vol. 90, no. 24, pp. 14216–14221, 2018.
- [172] W. J. Thrift and R. Ragan, "Quantification of analyte concentration in the single molecule regime using convolutional neural networks," *Analytical chemistry*, vol. 91, no. 21, pp. 13337–13342, 2019.
- [173] W. J. Thrift, C. Q. Nguyen, J. Wang, J. E. Kahn, R. Dong, A. B. Laird, and R. Ragan, "Improved regressions with convolutional neural networks for surface enhanced raman scattering sensing of metabolite biomarkers," in *Nanoengineering: Fabrication, Properties, Optics, Thin Films, and Devices XVI*, vol. 11089, p. 1108907, International Society for Optics and Photonics, 2019.

Appendices

Appendix A: FHI-aims package

Within the scope of this dissertation, the DFT package FHI-aims [74] is used to produce spectroscopic reference data that serve to develop our machine learning models. FHI-aims is an all-electron electronic structure code based on numeric atom-centered orbitals. It is suitable to perform efficient modeling of molecules, clusters, surfaces, interfaces and bulk materials. Properties like total energies, band structures, electron densities, KS orbitals and density of states can be efficiently modeled. Basis functions are organized in 'tiers' to form different basis sets, where each tier contains a different number and different types of basis functions (mostly hydrogen-like basis functions are used). FHI-aims is equipped with three different pre-defined grid settings for all atomic species which govern the numerical accuracy. Those settings are light, tight and really tight. While tight settings yield results with a high level of accuracy, the computational time to converge the KS equation usually takes long. Light settings are the right choice for fast prerelaxations. In order to optimize the positions of atoms in a molecule or molecular cluster towards the energy minimum, calculations known as structure relaxation can be carried out.



ISBN 978-952-60-3966-4 (printed)
ISBN 978-952-60-3967-1 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Applied Physics
www.aalto.fi

**BUSINESS +
ECONOMY**

**ART +
DESIGN +
ARCHITECTURE**

**SCIENCE +
TECHNOLOGY**

CROSSOVER

**DOCTORAL
DISSERTATIONS**