Learning Representations of Atomistic Systems with Deep Neural Networks

vorgelegt von
Kristof Schütt, M.Sc.
geb. in Kiel

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Benjamin Blankertz
Gutachter:	Prof. Dr. Klaus-Robert Müller
Gutachter:	Prof. Dr. Alexandre Tkatchenko
Gutachter:	Prof. Dr. Manfred Opper

Tag der wissenschaftlichen Aussprache: 25. Mai 2018

Berlin 2018

Abstract

Learning Representations of Atomistic Systems with Deep Neural Networks Deep Learning has been shown to learn efficient representations for structured data such as image, text or audio. However, with the rise of applying machine learning to quantum chemistry, research has been largely focused on the development of hand-crafted descriptors of atomistic systems. In this thesis, we propose novel neural network architectures that are able to learn efficient representations of molecules and materials. We demonstrate the capabilities of our models by accurately predicting chemical properties across compositional and configurational space on a variety of datasets. Beyond that, we perform a study of the quantum-mechanical properties of C_{20}-fullerene that would not have been computationally feasible with conventional *ab initio* molecular dynamics. Finally, we analyze the trained models to find evidence that they have learned local representations of chemical environments and atom embeddings that agree with basic chemical knowledge.

Zusammenfassung

Lernen von Repräsentationen für Atomistische Systeme mit Tiefen Neuronalen Netzen Tiefes Lernen hat gezeigt, dass es effiziente Repräsentationen für strukturierte Daten wie Bilder, Texte oder Audio lernen kann. Mit der zunehmenden Anwendung von Maschinellem Lernen in der Quantenchemie hat sich die Forschung dort vor allem auf die manuelle Entwicklung von Deskriptoren für atomistische Systeme konzentriert. In dieser Arbeit schlagen wir zwei neuartige Architekturen für Neuronale Netze vor, die in der Lage sind, effiziente Repräsentationen für Moleküle und Materialien zu erlernen. Wir demonstrieren die Fähigkeiten unserer Modelle durch die genaue Vorhersage von chemischen Eigenschaften für Systeme mit verschiedenen Zusammensetzungen sowie verschiedenen Atomanordnungen. Darüber hinaus führen wir eine Studie der quantenmechanischen Eigenschaften von dem Fulleren C_{20} durch, welche mit konventionellen *ab initio* Moleküldynamik-Simulationen nicht möglich gewesen wäre. Schließlich zeigt eine umfassende Analyse der trainierten Modelle deutliche Hinweise darauf, dass sie lokale Repräsentationen von chemischen Umgebungen sowie Atomeinbettungen gelernt haben, die mit chemischem Grundlagenwissen übereinstimmen.

Acknowledgements

First and foremost, I thank Klaus-Robert Müller for his invaluable support and inspiration. Klaus introduced me to the exciting topic of applying machine learning to quantum chemistry and allowed me the freedom to realize my own research ideas, while always being ready to offer scientific advice and encouragement. I equally thank Alexandre Tkatchenko for his advice and continuous strive for perfection. I am especially grateful for the fruitful collaborations with Klaus and Alex.

I thank all my co-authors – independent of whether the shared work made it into the thesis or not – notably Huziel Sauceda, Stefan Chmiela, Henning Glawe, Hardy Gross, Antonio Sanna, Farhad Arbabzadah and Felix Brockherde. I want to especially highlight the work on PatternNet with Pieter-Jan Kindermans, Max Alber and Sven Dähne which was an outstanding experience. Special thanks go to Michael Gastegger for inspiring discussions and proof-reading of this thesis.

I am grateful to the Institute for Pure and Applied Mathematics (IPAM), UCLA for allowing me to take part in two of their long programs in 2013 and 2016. I especially thank the IPAM team for creating a great atmosphere and an outstanding opportunity for initiating interdisciplinary research.

I thank my supervisors and teachers over the years: Sandro Rodriguez-Garzon for sparking my interest in machine learning and supervising my BSc thesis as well as Marius Kloft and Konrad Rieck for teaching me the nuts and bolts of machine learning while supervising my MSc thesis.

Finally, I thank all my colleagues of the ML group at TU Berlin, in particular my office mates over the years Grégoire Montavon, Mihail Bogojeski, Alexander Bauer and Tammo Krüger.

Contents

Chapter 1

Introduction

Chemistry is integral to a wide variety of technologies ranging from food processing and drug design to batteries and solar cells. The discovery of novel molecules and materials with desired properties is crucial to progress in these areas. While quantum-chemical calculations deliver the means to predict such properties for given atomistic systems and simulate their dynamic behavior, the vastness of chemical compound space prevents an exhaustive exploration [Lil13]. To overcome this issue, discoveries in chemistry are guided by databases of experimental and theoretical structures and properties. Those are mined for systems with desired chemical properties using descriptors and fingerprints that aim to encode chemical similarity, e.g. based on the molecular graph [RH10] or quantum-chemical properties obtained from electronic structure calculations [KLK96]. Indeed, high-throughput screening computational methods [Cur+13; Pyz+15], which combine electronic structure calculations with data analysis techniques, have proven to be a powerful tool, e.g. in the discovery of improved batteries [KC09; Hau+13], catalysts [Nør+09] and photovoltaics [Hac+11]. However, the computational cost of accurate quantum-chemical calculations remains the bottleneck of these approaches.

In recent years, there has been increased interest in applying machine learning techniques to model quantum-chemical systems [Lil13]. A significant part of the research has been dedicated to engineering of features that characterize global molecular similarity [Rup+12; Mon+12; Han+13; Han+15] or local chemical environments [BP07; BKC13] based on atomic positions. Then, a non-linear regression method – such as kernel ridge regression or a neural network – is used to correlate these features with the chemical property of interest. In these types of approaches, the representation of an atomistic system is fixed and can not be adapted to the task at hand. While this may be desirable if there is only a limited amount of data available, such an approach struggles to exploit regularities in the data that are not reflected in the descriptor. This is in particular the case if such internal structure is strongly property-specific or can only be approximated based on chemical intuition. E.g., the similarity of

atom types can not be easily encoded, especially if we aim to avoid heuristics that only apply to certain classes of molecules or materials.

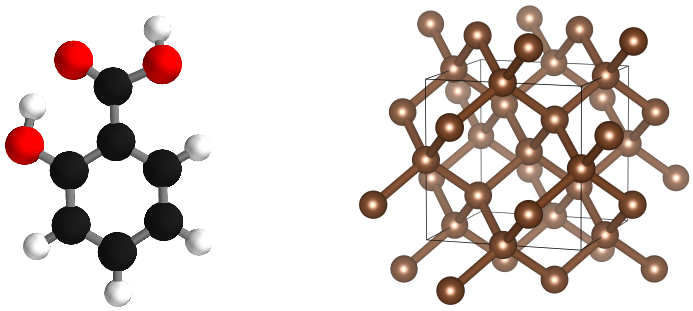
In other applications, such as computer vision and natural language processing, recent breakthroughs in deep neural networks [KSH12; SVL14; Vin+15; Mni+15] have caused a major shift towards end-to-end learning of representations [LBH15; Sch15]. Just like images, text or audio, molecules and materials are highly organized data that show local structure such as chemical bonds or functional groups.

The goal of this thesis is to develop deep neural networks that are capable of learning representations for atomistic systems. Beyond that we aim to provide techniques to extract insights about the obtained representation as well as the underlying data. We will reuse the deep learning architecture in a variety of applications, thus, the resulting representation has to adapt to the task at hand. The predictions obtained from the learned representation should follow fundamental quantum-mechanical principles. Therefore, we will encode important invariances, e.g. towards rotation and translation, directly into the deep learning model and constrain our models to obey physical laws such as energy conservation. Following this principle, we aim to increase the sample efficiency of our models without reducing the generality of the approach, as it would be the case when including chemical intuition and heuristics into manually crafted features.

We will apply the developed deep learning techniques to a variety of tasks ranging from the prediction of various chemical properties across chemical compound space for molecules and solids to accelerating molecular dynamics simulations. This constitutes an important step toward machine learning-driven quantum-chemical exploration. By analyzing the learned representations, we will get a glimpse into the inner working of the neural network in order to validate whether the model has learned known chemical concepts or might even provide novel insights.

1.1 Theoretical background

In this section, we will introduce some necessary background and important terminology that is used throughout the thesis. First, we will define atomistic systems before we introduce the quantum mechanical foundations to illustrate the complexity of electronic structure calculations. Then, we will go on to discuss density functional theory – the electronic structure method providing the reference calculations used in this thesis. Finally, we describe the tasks that the methods in this work are applied to.



(a) Molecule: Salicylic acid **(b)** Bulk crystal: Diamond

Figure 1.1: Examples of atomistic systems.

1.1.1 Atomistic systems

An **atomistic system** S consisting of N atoms can generally be described as a set of tuples

$$S = \{(Z_i, \mathbf{r}_i) | \, i \in [1, n_{\text{atoms}}]\},$$

where Z is the nuclear charge that characterizes the atom type and $\mathbf{r}$ is the position of the atom. We conveniently write interatomic distances $d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$.

In this thesis, we will consider two types of atomistic systems, namely molecules and bulk crystals. **Molecules** consist of a set of atoms that are connected by chemical bonds (Fig. 1.1a). **Crystals** are highly organized atomistic systems where the atoms are located in a **unit cell** that repeats periodically and forms the **Bravais lattice** [AM76]. Fig. 1.1b shows diamond with its cubic unit cell. In an ideal crystal, the cell repeats infinitely in all three directions of the lattice. This is called **periodic boundary condition (PBC)**. Thus, we can write a crystal as a set of tuples

$$S = \{(Z_i, \mathbf{r}_i + n_1\mathbf{l}_1 + n_2\mathbf{l}_2 + n_3\mathbf{l}_3) | \, i \in [1, n_{\text{atoms}}]; n_1, n_2, n_3 \in \mathbb{N}\},$$

where $\mathbf{l}_k$ are the lattice vectors that span the unit cell.

1.1.2 The Schrödinger equation

A significant part of quantum chemistry is concerned with finding approximate solutions to the **time-independent Schrödinger equation**

$$\hat{H}\,\Psi = E\,\Psi$$

of an atomistic system with the **total energy** E and the **wave function** Ψ. The quantum-mechanical **Hamiltonian** operator represents how charged particles (electrons and nuclei) interact among each other and can be written in atomic

units as follows:

$$\hat{H} = \underbrace{-\sum_i \frac{1}{2m_e}\nabla_i^2}_{\substack{\text{kinetic energy}\\ \text{of electrons}}} \underbrace{-\sum_k \frac{1}{2M_k}\nabla_k^2}_{\substack{\text{kinetic energy}\\ \text{of nuclei}}} \underbrace{-\sum_i\sum_k \frac{Z_k}{d_{ik}}}_{\substack{\text{electron-nuclear}\\ \text{attraction}}} + \underbrace{\sum_{i<j}\frac{1}{d_{ij}}}_{\substack{\text{electron-electron}\\ \text{repulsion}}} + \underbrace{\sum_{k<l}\frac{Z_k Z_l}{d_{kl}}}_{\substack{\text{nuclear-nuclear}\\ \text{repulsion}}} \tag{1.1}$$

with electron indices i, j, atom indices k, l, the electron mass m_e, the mass M_k of nucleus k and the Laplacians ∇_i, ∇_k of the electrons and nuclei, respectively [SO96; Cra04]. Within the **Born-Oppenheimer approximation**, the nuclear positions are considered fixed compared to the much faster electrons. Therefore, we obtain the electronic Hamiltonian

$$\hat{H}_{\text{el}} = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_i\sum_k \frac{Z_k}{d_{ik}} + \sum_{i<j}\frac{1}{d_{ij}}, \tag{1.2}$$

while the nuclei are effectively considered point charges which generate the external potential. Still, this constitutes an n-body problem for which there exists no analytic solution for $n > 1$ electrons.

A major reason that the solution to this is much more complex than in classical mechanics is that the electrons obey quantum-mechanical principles, and have to be described by the many-body wave function $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$[1]. This can be represented in a set of basis functions so that all necessary constraints, such as the antisymmetry of the electron wave function

$$\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_i, \ldots, \mathbf{r}_j, \ldots, \mathbf{r}_N) = -\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_j, \ldots, \mathbf{r}_i, \ldots, \mathbf{r}_N), \tag{1.3}$$

are fulfilled. The **variational principle** states that the eigenvalues of the Hamiltonian are bounded from below, thus,

$$E = \frac{\int \Psi \hat{H}_{\text{el}} \Psi d\mathbf{r}}{\int \Psi^2 d\mathbf{r}} \geq E_0 \tag{1.4}$$

where the ground state E_0 is the lowest possible energy of a system. Eq 1.4 allows us to compare the quality representations of the wave function with the criterion of which achieves a lower ground state [SO96; Cra04]. At the same time, this presents a solution to the Schrödinger equation, namely to minimize the energy which involves computing the integrals in Eq. 1.4. This can be achieved by a self-consistent field approach, where the Hamiltonian is applied to a trial wave function to obtain a more accurate set of wave function parameters.

[1] For simplicity, we neglect the electron spin in this introduction.

The choice of how the wave function is parametrized determines the accuracy of the solution as well as the computational cost. While the Hartree-Fock approximation, where the wave function is represented by a single Slater determinant, scales with $O(n^4)$, more accurate calculations like CCSD(T) already scale with $O(n^7)$ [Cra04]. Therefore, an accurate solution becomes soon infeasible with growing system sizes and more complex representations of the wave function.

1.1.3 Density functional theory

As we have seen in the last section, a major scaling issue is that the wave function depends on the positions of all particles. A simpler object is the electron density $\rho(\mathbf{r})$ which corresponds to the probability of finding an electron at position $\mathbf{r}$, and is normalized to the number of electrons

$$N = \int \rho(\mathbf{r}) d\mathbf{r}. \tag{1.5}$$

Hohenberg and Kohn [HK64] showed that there exists a unique mapping from the ground-state electron density $\rho_0(\mathbf{r})$ to the external potential, which implies that it also determines the wave function. Therefore, we can write the ground-state energy as a functional of the density:

$$E_0 = \underbrace{\bar{T}[\rho_0(\mathbf{r})]}_{\substack{\text{kinetic energy} \\ \text{of electrons}}} + \underbrace{\bar{V}_{\text{ne}}[\rho_0(\mathbf{r})]}_{\substack{\text{nuclear-electron} \\ \text{attraction}}} + \underbrace{\bar{V}_{\text{ee}}[\rho_0(\mathbf{r})]}_{\substack{\text{electron-electron} \\ \text{repulsion}}} \tag{1.6}$$

Beyond that, Hohenberg and Kohn [HK64] showed that the density obeys a variational principle, i.e.,

$$\bar{T}[\rho(\mathbf{r})] + \bar{V}_{\text{ne}}[\rho(\mathbf{r})] + \bar{V}_{\text{ee}}[\rho(\mathbf{r})] = \bar{E}[\rho(\mathbf{r})] \geq E_0, \tag{1.7}$$

which would allow us to compute the ground-state energy, if we knew the exact energy functional $\bar{E}[\rho(\mathbf{r})]$ [Cra04]. However, we do not know how the kinetic energy $\bar{T}$ and the electron-electron repulsion $\bar{V}_{ee}$ can be obtained from the density.

Kohn and Sham [KS65] introduced a formalism to rewrite the energy functional as a system of non-interacting electrons

$$\bar{E}[\rho(\mathbf{r})] = \underbrace{\bar{T}_{\text{ni}}[\rho_0(\mathbf{r})]}_{\substack{\text{kinetic energy of} \\ \text{non-interacting electrons}}} + \underbrace{\bar{V}_{\text{ne}}[\rho_0(\mathbf{r})]}_{\substack{\text{nuclear-electron} \\ \text{attraction}}} + \underbrace{\bar{V}_{\text{ee}}[\rho_0(\mathbf{r})]}_{\substack{\text{classic} \\ \text{electron repulsion}}} \tag{1.8}$$

$$+ \underbrace{\Delta\bar{T}[\rho_0(\mathbf{r})]}_{\substack{\text{interaction correction} \\ \text{of kinetic energy}}} + \underbrace{\Delta\bar{V}_{\text{ee}}[\rho_0(\mathbf{r})]}_{\substack{\text{non-classical} \\ \text{electron repulsion}}}, \tag{1.9}$$

where the last two terms are corrections that reintroduce the electron interactions and are summarized as the exchange-correlation energy $\bar{E}_{\text{xc}} = \Delta\bar{T}[\rho_0(\mathbf{r})] +$

$\Delta\bar{V}_{\text{ee}}[\rho_0(\mathbf{r})]$. This leads to the same ground-state energy as the original system, which now can be decomposed in terms of electronic basis functions

$$\rho_0(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2. \tag{1.10}$$

The solution can be obtained through a self-consistent field approach using the Kohn-Sham operator

$$\hat{h}_i = -\frac{1}{2}\nabla_i^2 - \sum_k \frac{Z_k}{r_{ik}} + \int \frac{\rho(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} d\mathbf{r}' + \frac{\partial \bar{E}_{\text{xc}}[\rho(\mathbf{r})]}{\partial \rho(\mathbf{r})}, \tag{1.11}$$

where the last term is the functional derivative of the exchange-correlation functional.

While density functional theory (DFT) is exact in principle, one would have to know the correct $\bar{E}_{\text{xc}}$ to obtain the correct ground state. Since this is not the case, there exist several approximations with varying accuracy and computational cost. The most simple approaches approximate the exchange-correlation locally, i.e., depending on the density at a given location $\mathbf{r}$. These functionals are called local (spin) density approximations (LDA/LSDA) and are in practice derived from the uniform electron gas, where the density is a constant [Cra04; PZ81]. This approach can be extended by using the gradient of the density in so-called generalized gradient approximation (GGA) functionals. Popular GGA functionals include the parameter-free PBE [PBE96] and fitted functionals like B88 [Bec88]. Hybrid functionals such as B3LYP [Bec88; LYP88; Bec93] or PBE_0 [PEB96] include exact exchange from the Hartree-Fock formalism using the Kohn-Sham orbitals.

The data sets employed in this thesis have been calculated using DFT with various functionals (see Appendix A). The computational cost of DFT scales with $O(n^3)$ w.r.t. the number of particles. The exchange-correlation functional can increase the cost. E.g. DFT using hybrid functionals scales with $O(n^4)$ since those require the exchange term from Hartree-Fock.

1.1.4 Typical quantum-chemical tasks for ML

At absolute zero temperature, atomistic systems relax into a state where all atomic forces cancel, which we call **equilibrium**. A common task for ML in quantum chemistry is the prediction of properties for systems at equilibrium across chemical compound space. One important property is the energy needed to break the atomistic system down into single, non-interacting atoms, which is called **atomization energy** or **formation energy** for molecules and materials, respectively.

On the other hand, **molecular dynamics (MD) simulations** approximate the time evolution of a system including, e.g., interactions with the environment. In this case, the data contains not only the equilibrium configuration

of an atomistic system, but also perturbed configurations, often together with the atomic forces. The energy of the system depending on the atomic positions defines its **potential energy surface (PES)** $E(\mathbf{r}_1, \dots \mathbf{r}_{n_{\text{atoms}}})$. The force on atom i can then be obtained as the negative derivative of the energy:

$$\mathbf{F}_i = -\frac{\partial E(\mathbf{r}_1, \dots \mathbf{r}_{n_{\text{atoms}}})}{\partial \mathbf{r}_i}$$

Predicting PESs and the associated force fields is another important application of ML for quantum chemistry which we will tackle in this thesis.

Even though density functional theory is faster than accurate wave function methods, it is still a bottleneck in exploring chemical space and performing large-scale molecular dynamics simulations. This is because these application require huge numbers of calculations. As we will demonstrate in this thesis, machine learning has the potential to speed up these applications, even for small molecules, by several orders of magnitude.

1.2 Description of the chapters

Chapter 2 (Representing atomistic systems) We introduce necessary background on how to represent atomistic systems for machine learning. We review necessary and desirable properties of representations and, with that in mind, analyze a variety of existing descriptors. Finally, we draw conclusions on requirements and constraints for learning a representation.

Chapter 3 (Learning representations of chemical enviroments) Based on the analysis of Chapter 2, we conceive a deep tensor neural network (DTNN) architecture that is able to learn a representation for atomistic systems while exhibiting the previously established necessary constraints and invariances. We use our model to predict molecular energies across chemical compound space as well as molecular dynamics trajectories. Beyond that, we analyze the learned representation to obtain quantum chemical insights.

Chapter 4 (Continuous-filter convolutional neural networks) In this chapter, we revisit the modeling of quantum interactions in DTNNs under the aspect of convolutions. We develop continuous-filter convolutional layers that we use to build SchNet: an improved architecture to learn representations for atomistic systems. This allows us to define filters with periodic boundary conditions that we use for the prediction of formation energies of bulk materials.

Chapter 5 (Potential energy surfaces) In this chapter, we apply SchNet to the prediction of potential energy surfaces (PES) and corresponding force fields.

Specifically, we use a combined loss to obtain a combined model that can accurately predict molecular dynamics trajectories from a set of trajectories of small organic molecules. Beyond that we apply our method to the prediction of a PES with chemical and conformational changes. Finally, we demonstrate the capabilities of SchNet by using it to drive an MD simulation of the fullerene C_{20}.

1.3 Main contributions of this thesis

This thesis provides the following main contributions:

- **Development of Deep Tensor Neural Networks (DTNNs) for predicting molecular energies** We develop a neural network architecture that is able to predict atomization energies using atom types and positions as input in an end-to-end fashion. The model learns atom-wise representations of chemical environments and follows fundamental quantum-chemical principles such as invariance towards rotation, translation and atom indexing. DTNNs provide size-extensive predictions at chemical accuracy ($\leq$ 1 kcal mol^{-1}) in compositional and configurational chemical space.
- **Development of continuous-filter convolutional layers and the SchNet architecture** We develop continuous-filter convolutional layers in order to model quantum interactions of atoms at arbitrary positions. We built upon the DTNN principles to propose SchNet: a continuous-filter convolutional network for molecules and materials. SchNet is able to predict various chemical properties of a benchmark dataset of small, organic molecules as well as formation energies of a diverse set of bulk materials.
- **Analysis of the representations learned by DTNN and SchNet models** We analyze the representations obtained from DTNN and SchNet in order to gain insights about the model and the underlying data. We study the energy partitioning provided by the models in terms of stability. Furthermore, the neural networks generate a local chemical potential which can probed by a test charge in order to analyze the spatial structure of the obtained representations. The sensitivity of chemical environments is analyzed to estimate the range of atomic interactions.
- **Application to potential energy surfaces and force fields** We train SchNet models using a combined objective of energies and forces in order to obtain accurate potential energy surfaces and corresponding conservative force fields. We will use this to perform an path-integral MD simulation on C_{20} fullerene with SchNet, reducing the required computing time from 7 years to less than 7 hours.

1.4 Previously published work

Many results in this thesis have previously been published in conference proceedings and journals. They are taken from the following articles:

- K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K.-R. Müller, and E. Gross. "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties". *Phys. Rev. B* 89 (20), p. 205118, 2014
- K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko. "Quantum-chemical insights from deep tensor neural networks". *Nature Communications* 8, 13890, 2017
- K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". in: *Advances in Neural Information Processing Systems 30*, pp. 992–1002. 2017
- 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* 148 (24), 241722, 2018

Figures and tables that are fully or partially taken from previously published work, cite the original source in the bold caption title.

Chapter 2

Representing atomistic systems

Predicting properties of atomistic system based on previously observed data is an established procedure in chemistry. In chemoinformatics, predictive models are often used to perform fast virtual screening by correlating the structure of a compound to chemical properties (quantitative structure-property relationship, *QSPR*) or the biological activity (quantitative structure-activity relationship, *QSAR*) [KLK95]. This is usually achieved by a regression of a descriptor of the compound to the property of interest. These descriptors can be derived from the composition, topology and geometry of the compound, or even include results from quantum-chemical calculations [KG93; KLK95; SV03]. These approaches are well suited to predict complex properties such as toxicity or solubility. They are not designed for highly accurate predictions of fundamental quantum-chemical properties such as atomization energies or atomic forces. For purposes like finding stable structures or molecular dynamics simulations, where these properties are required, classical (semi-empirical) force fields are fitted to data from experiments or electronic structure calculations. Examples for such force fields are AMBER [Cor+95], CHARMM [Bro+83] or GROMOS [GB87]. However, these approaches are tailored towards a restricted class of systems and properties. Beyond that, they rely on terms incorporating bond lengths, angles and dihedral angles such that they usually do not allow for bond breaking.

The reason for descriptors in chemoinformatics to be painstakingly optimized to specific atomistic systems, is that linear regression methods are applied. Given more powerful non-linear machine learning techniques, such as kernel methods [CV95; Mül+01; SS02] or deep neural networks [Bis95; LeC+98; LBH15], we are able to use more general features. These may even include first principles information, i.e. encode the full information of the quantum Hamiltonian [Lil+15]. Within the Born-Oppenheimer approximation, this amounts to the types and positions of the atoms in an atomistic system. Note that, depending on the property to be predicted, not the full information might be required, e.g. due to invariance of the property w.r.t. rotation and translation.

	First principles *atoms, positions*	**Chemical graphs** *atoms, bonds, rings*	**Classical force fields** *atoms, bond lengths, angles, ...*
virtual screening	✓ *equilibrium required*	✓	✓ *equilibrium required*
molecular dynamics	✓	✗	✓ *no bond breaking*
chemical & configurational space	✓	✗	✗

Table 2.1: Possible applications for machine learning descriptors using first-principles information, chemical graphs or terms from classical force fields. The table shows whether the descriptor are applicable (✓), applicable with limitations (✓) or not applicable (✗).

Depending on the task, machine learning representations can be inspired by the approaches above, i.e., be derived from first principles, chemoinformatics descriptors or classical force fields. An overview of suitable applications is shown in Table 2.1. Note that first-principles representations are the only option that can be applied to all listed applications, even though the equilibrium geometry is required for virtual screening. While this may be prohibitive in certain situations, a computationally cheap force field can be used to obtain the approximate structure before ML is used to accurately predict the desired property.

In this thesis, we aim for machine learning methods that can obtain representations applicable to all kinds of atomistic systems *and* across chemical as well as configurational space. Only with such a representation, it is conceivable to accurately model general quantum interactions and, in doing so, be able to extract quantum-chemical insights. This can only be achieved by a representation derived from first principles (see Table 2.1). In the following sections, we will discuss desirable properties of such a representation and review existing categories of descriptors for molecules and solids.

2.1 Properties of atomistic representations

A machine learning method in combination with well-crafted features should be able to deliver highly accurate predictions while requiring as few training examples as possible. To achieve this, the chosen representation has to fulfill certain requirements in order to generalize well. Beyond that, in quantum chemistry we would like the representation to follow further application-specific requirements. Lilienfeld et al. [Lil+15] have formulated a list of desirable properties of machine learning descriptors. In the following, we review the requirements of this list that we deem most important:

Uniqueness

Obviously, it is crucial that the representation $\mathbf{x}$ contains all relevant information to uniquely describe the atomistic system S up to invariances of the desired property $y = f(\mathbf{x})$, i.e.

$$\mathbf{x}_S = \mathbf{x}_{S'} \implies f(\mathbf{x}_S) = f(\mathbf{x}_{S'}). \tag{2.1}$$

Otherwise, it is not possible to correctly predict a property that is not equal for those two systems. Having each atomistic system uniquely characterized such that

$$\mathbf{x}_S = \mathbf{x}_{S'} \iff S = S', \tag{2.2}$$

possibly up to rotations and translations, would even result in an invertible representation. This is only required for properties without invariances. Lilienfeld et al. [Lil+15] additionally list the "completeness" or "global nature" of the descriptor as a requirement in contrast to local representations that only reflect a local environment. However, we argue that this is already covered by the definition of uniqueness.

Invariances / Equivariances

Invariances of the property to be predicted with respect to input transformations reduce the domain that needs to be covered by the machine learning model. Therefore, less training examples will be required to achieve the same accuracy. However, this is only the case if the representation reflects these invariances. E.g., following the invariances of the total energy of an atomistic system, the representation for this task should be invariant to translation, rotation and permutation. In contrast, a representation for atomic forces should be *equivariant* with respect to rotation and permutation. If an invariance cannot be explicitly incorporated in the descriptor, it can be learned using data augmentation [Mon+12].

Differentiability

Many quantum chemical properties evolve continuously during atom movement. To properly model this behavior, the representation has to be continuous as well. Beyond that, it may be necessary to differentiate a property prediction with respect to the atom positions. For instance, the force $\mathbf{F}_i$ acting on atom i is defined as the negative derivative of the energy w.r.t. the position:

$$\mathbf{F}_i = -\frac{\partial E}{\partial \mathbf{r}_i}. \tag{2.3}$$

This makes it possible to optimize the atom positions in order to obtain the equilibrium structure, or to perform molecular dynamics simulations. In these

	CM	BoB	PRDF / HDAD	SOAP	ACSF
uniqueness	✓	✗	✗	✓	✗
invariant to translation	✓	✓	✓	✓	✓
invariant to rotation	✓	✓	✓	✓	✓
invariant to atom indexing	✗	✗	✓	✓	✓
differentiable	✓	✓	✗	✓	✓
cross-element generalization	↯	✗	✗	✗	✗

Table 2.2: Properties of various descriptors. We list Coulomb matrix (CM) [Rup+12], Bag of Bonds (BoB) [Han+15], partial radial distribution functions (PRDF) [Sch+14], histograms of distances, angles and dihedral angles (HDAD) [HL16], smooth overlaps of atomic potentials (SOAP) and atom-centered symmetry functions (ACSF) [Beh11]. The table shows whether the properties are fulfilled (✓), partially fulfilled (✓) or not fulfilled (✗). Cross-element generalization is theoretically possible with the Coulomb matrix, however, the used similarity measure turns out to be detrimental (↯).

cases, both the representation and the machine learning model need to be (multiple times) differentiable. While this is often the case for the machine learning method, non-differentiable features occur regularly. Examples of this are one-hot encodings of single and double bonds [Duv+15; Kea+16; Gil+17] or external potentials discretized on a grid [Sny+12] which both introduce discontinuities and lead to noisy gradients [Sny+15; Bal+17]. When derivative information such as atomic forces are available in the reference data and supposed to be incorporated in the loss function, at least second order differentiability is required for gradient descent training.

Cross-element generalization

We suggest an additional desired property: the representation should be able to allow for learning from the observed interactions of atoms to interactions of atoms of another type. This requires a sense of similarity between atom types, e.g. by using chemical concepts such as electronegativity or the group of the periodic table. While this can be helpful if the similarity measure is chosen correctly, it can disadvantageous if it does not correlate well with the similarity of the target property. In this case, it is often better to regard different atom types as orthogonal.

2.2 Representations for molecules and solids

After reviewing some desirable properties of representations, we will have a look at some descriptors and evaluate them with the latter in mind. As mentioned above, we will restrict ourselves to first-principles representations, i.e., we will not discuss fingerprints as employed in chemoinformatics that are not able to reflect configurational degrees of freedom. Table 2.2 gives an overview about which of the discussed representations fulfills the previously discussed requirements.

2.2.1 Coulomb matrix

Rupp et al. [Rup+12] proposed the Coulomb matrix (CM) as a representation to predict properties of molecules across chemical compound space. It is an adjacency matrix with nuclear charge rescaled to fit atomic energies on the diagonal and the Coulomb repulsion of atoms i and j on the off-diagonal:

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

This representation is invariant to rotations and translations due to the pairwise distances that are part of the Coulomb term. However, it lacks invariance to atom indexing. Therefore, the eigenspectrum of the Coulomb matrix was initially used to achieve this invariance [Rup+12]. However, this violates the uniqueness requirement as there can be multiple Coulomb matrices with the same eigenspectrum. Montavon et al. [Mon+12] and Hansen et al. [Han+13] achieved the permutational invariance instead by either sorting by column norm or adding training examples augmented by randomly permuting the atoms. However, both of these techniques have drawbacks: Sorting leads to a representation with singularities at the atom configurations where the norms of multiple columns are equal. Therefore, in these cases permutational invariance is not given. Beyond that, this creates discontinuities in the predicted property during atom movement. In the data augmentation approach, the number of permutations grows rapidly with the size of the system, so that this approach becomes less and less effective or even computational infeasible.

Another issue with the Coulomb matrix is that the repulsion of the nuclei, while being part of the Hamiltonian, is not very informative of the chemical similarity and, thus, not useful for cross-element generalization. Chemical similarity depends much more on the valence electrons as reflected in the groups of the periodic table. Hansen et al. [Han+15] propose the *Bag of Bonds* (BoB) model to alleviate this issue. Here, the terms of the Coulomb matrix are reordered into bags of equivalent atom types or pairs of atom types, respectively. This effectively makes atoms and atom interactions of different types orthogonal. While this eliminates the inappropriate measure of chemical similarity, it also prevents learning across atom types.

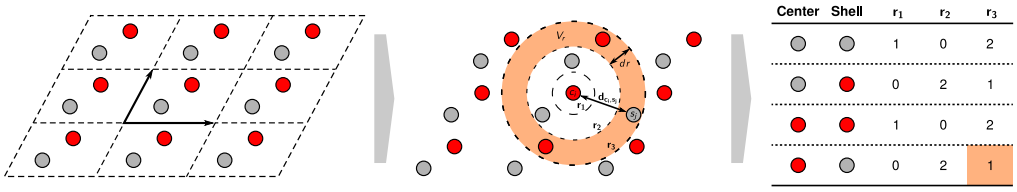


Figure 2.1: Illustration of partial radial distribution function representation. The atom types α, β are color-coded in gray and red. A crystal unit cell with two atoms (left) is replicated such that all distances up to r_3 are covered. The distances lying in a shell r_i (middle) are counted per pair of atom types and put in a histogram (right). Normalizing this by shell volume V_{r_i} and number of atoms per type yields $g_{\alpha\beta}(r)$.

2.2.2 Many-body expansions

The many-body expansion decomposes the energy of an atomistic system S into n-body terms [DT07]:

$$E(S) = \sum_{i=1}^{n_{\text{atoms}}} E^{(1)}(\mathbf{r}_i) + \sum_{i<j}^{n_{\text{atoms}}} E^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i<j<k}^{n_{\text{atoms}}} E^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots \tag{2.4}$$

In choosing concrete n-body terms, a variety machine learning descriptors can be designed. These are then no longer an expansions of the energy, but a unique decomposition of the geometry of the atomistic system from which the energy or any other property can be inferred. In practice, one often neglects higher-order terms, sacrificing uniqueness for computational efficiency. Beyond that, these term can often not be fit properly without a huge amount of training data. The previously discussed BoB model can be seen as a many-body expansion with 1- and 2-body terms that model the Coulomb repulsion. Huang and Lilienfeld [HL16] have proposed BAML (bond-angles machine learning) where the idea of BoB was extended to 3- and 4-body terms. Similarly, the MBTR [HR17] is a general framework for building tensors of many-body terms.

The approaches above still require zero-padding of the bags since different atom type compositions result in different bag sizes. This can be circumvented by using a histogram over value ranges of many-body terms instead of bags. The partial radial distribution function (PRDF) representation is a 2-body variant of this idea [Sch+14]. It has been applied to the prediction of the density of states at Fermi level of bulk crystals and was inspired by the radial distribution function as used in x-ray powder diffraction [BT98]. The core idea is to collect statistics about the distribution of distances between atoms of type α and β (see Fig. 2.1). The distances $r_{\alpha_i\beta_j}$ of all atoms α_i and β_j are collected in a normalized histogram bin

$$g_{\alpha\beta}(r) = \frac{1}{n_\alpha n_\beta V_r} \sum_{i=1}^{n_\alpha} \sum_{j=1}^{n_\beta} \mathbb{1}_{r<r_{\alpha_i\beta_j}<r+dr} \tag{2.5}$$

where N_α, N_β are the numbers of atoms of the respective type and V_r is the volume of the shell that corresponds to the bin. Normalization is important

here since there are more atoms situated in a shell with larger radius r due to the increased volume. Just like BoB, this kind of representation can be extended to 3- and 4-body terms as done by Faber et al. [Fab+17] with their HD (histogram of distances), HDA (histogram of distances and angles) and HDAD (histogram of distances, angles and dihedral angles) representations. A disadvantage all of these approaches share is that they are not differentiable. This can be solved by using Gaussian basis functions instead, e.g., as applied in atom-centered symmetry functions [BP07; Beh11] (see Section 2.2.3).

Malshe et al. [Mal+09] proposed an approach for predicting potential energy surfaces that directly uses interatomic distances r_{ij} as input for n-body neural network $f_n : \mathbb{R}^{\frac{n^2-n}{2}} \to \mathbb{R}$ forming the potential:

$$E = \sum_{i<j}^{n_{\text{atoms}}} f_2(r_{ij}) + \sum_{i<j<k}^{n_{\text{atoms}}} f_3(r_{ij}, r_{ik}, r_{jk}) + \dots$$

A drawback of this approach is that the n-body neural networks are not invariant to the order of inputs $r_{ij}, r_{ik}, r_{jk}, \dots$ for $n \geq 3$. Furthermore, for each n-body term, a separate n-body neural network is required that needs to be trained to fit the corresponding energy contribution which limits the expressive power of the whole model to highest explicitly modeled n-body term. In contrast, the previously described representations BoB, PRDF and HDAD contain all n-body terms up to the specified order such that a (non-linear) machine learning method is able to infer some higher-order interactions.

2.2.3 Chemical environments

Instead of decomposing atomistic systems in terms of n-body interactions, an alternative is a partitioning into local, chemical environments. From Fig. 2.1, it may appear that the PRDF does this, however, due to the sum over all atoms of the same type, localizing information is lost. This does not affect the predictability of a global property, such as the energy, if all many-body terms in Eq. 2.4 are included due to the uniqueness of the full expansion. However, a representation may be more efficient, in terms of computational cost and required training data, when localized information is retained.

In terms of the many-body expansion, this amounts to a reordering of the terms in Eq. 2.4 by the atoms i defining the center of the chemical environments:

$$E(S) = \sum_{i=1}^{n_{\text{atoms}}} \left[E^{(1)}(\mathbf{r}_i) + \frac{1}{2} \sum_{j \neq i}^{n_{\text{atoms}}} E^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \frac{1}{3} \sum_{j \neq i}^{n_{\text{atoms}}} \sum_{k \neq i, k \neq j}^{n_{\text{atoms}}} E^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots \right] \tag{2.6}$$

$$= \sum_{i=1}^{n_{\text{atoms}}} E_i(\mathbf{r}_1, \dots, \mathbf{r}_{n_{\text{atoms}}}) \tag{2.7}$$

Now, the energy contributions E_i can either be calculated by many-body energy terms (Eq. 2.6) or inferred from an arbitrary ML representation. This could be based on a many-body decomposition as introduced in Section 2.2.2 for the global case or any other localized version of previously introduced representation such as CM or BoB.

Alternatively, a density function $\rho(\mathbf{r})$ can be defined over the space from which features are derived. This approach has been adopted by Hirn, Poilvert, and Mallat [HPM15] in a global setting using wavelet scattering transforms [HMP17; Eic+17] as well as for chemical environments by the *Smooth Overlap of Atomic Positions* (SOAP) kernel introduced by Bartók, Kondor, and Csányi [BKC13]. Here, a similarity of chemical environments ρ, ρ' is defined as

$$S(\rho, \rho') = \int \rho(\mathbf{r})\rho'(\mathbf{r})d\mathbf{r}$$

which is then used define the rotationally invariant SOAP kernel [BKC13]

$$k(\rho, \rho') = \int \left|S(\rho, \hat{R}\rho')\right|^n d\hat{R},$$

where n is a hyper-parameter. Choosing the neighborhood densities ρ to be Gaussians expanded in spherical harmonics, allows for a smooth overlap of chemical environments. Similarly, moment tensors [Sha16] represent chemical environment through polynomials that are invariant to rotation, translation and atom permutations.

A representation of chemical environments that models the many-body decomposition explicitly are the atom-centered symmetry functions (ACSF) as proposed by Behler and Parrinello [BP07]. Behler [Beh11] has proposed a variety of 2- and 3-body symmetry functions, e.g., the radial 2-body function

$$G_i^2 = \sum_{j \neq i}^{n_{\text{atoms}}} e^{-\eta(r_{ij} - r_s)^2} f_c(r_{ij})$$

for center atom i summed over neighboring atoms j and hyper-parameter r_s that centers the Gaussian on a distance value. The symmetry function shows similarities with the radial distribution function, but only considers 2-body terms including atom i. In contrast to the histogram-based representations introduced in the last section, G_i^2 is differentiable as it uses Gaussian basis functions instead of rectangular bins. The cutoff function

$$f_c(r_{ij}) = \begin{cases} \frac{1}{2}\cos\left(\frac{\pi r_{ij}}{r_c}\right) + \frac{1}{2} & \text{for } r_{ij} \leq r_c \\ 0 & \text{for } r_{ij} > r_c \end{cases}$$

fulfills a similar purpose as the volume normalization in the PRDF representation, i.e. it compensates for more atoms at larger distances and enforces a local environment. Fig 2.2 shows how both representations initially weight atoms at distances r_{ij}. While the PRDF normalization decays faster than the ACSF cutoff, it does not go towards zero but keeps the collective contribution

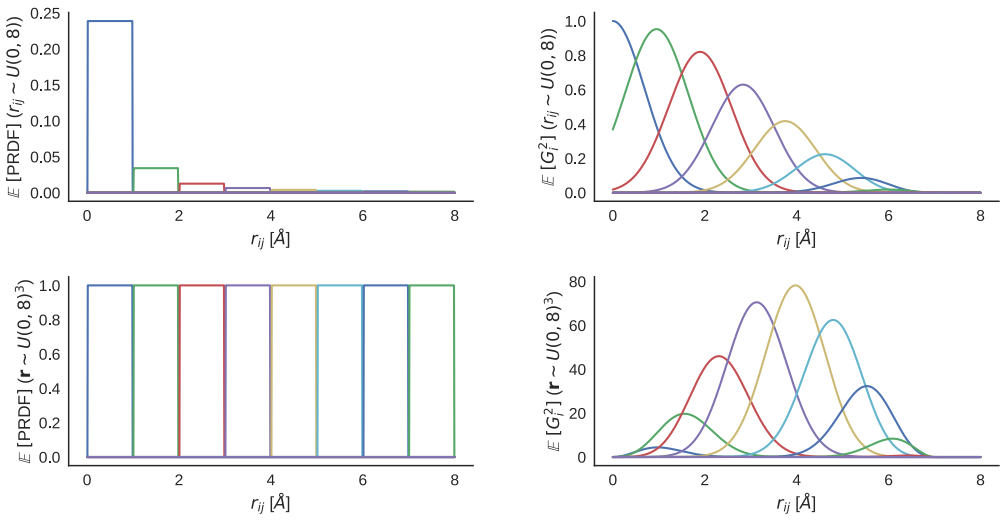


Figure 2.2: Comparison of the effects of cutoff functions of PRDF (left) and atom-centered symmetry function G_i^2 (right). We assume uniform distribution of interatomic distances r_{ij} (top) and uniform distribution of atom positions $\mathbf{r}$ in space (bottom).

of atoms in each radial bin constant, assuming uniform distribution of atoms in space. In contrast, the ACSF cutoff decreases less rapid in the beginning but then decreases smoothly to zero, in effect emphasizing atom contributions at medium distances and localizing the representation by bringing atom contributions smoothly to zero at distances $r_c = 8$ and larger. In a similar fashion, angular symmetry functions are defined, e.g.,

$$G_i^4 = 2^{1-\zeta} \sum_{j,k\neq i}^{n_{\text{atoms}}} (1 + \lambda \cos\theta_{ijk})^{\zeta} e^{-\eta(r_{ij}^2 + r_{ik}^2 + r_{jk}^2)} f_c(r_{ij})\, f_c(r_{ik})\, f_c(r_{jk}).$$

To predict potential energy surfaces, a neural network is used for each chemical environment to predict its local energy contribution before those are summed to obtain the total energy. The energy contributions are latent variables that do not need to be known but are learned during backpropagation [BP07]. ACSFs are used to represent the geometry of the system while the composition is taken into account by neural networks specific to the type of the center atom. The types of the neighboring atoms are not taken into account and generalization across atom types is not possible since for each atom type a separate network is trained. Other approaches that build upon Behler's atom-centered symmetry functions include ANI-1[SIR17] and TensorMol-0.1 [Yao+18].

2.3 Summary and discussion

In this chapter, we have reviewed a set of desired properties of representations for atomistic systems as well as some commonly used machine learning descriptors. None of those fulfills all required properties. While the Coulomb matrix is the only discussed representation that is always unique, it does not fulfill all required invariances and implies an unsuitable chemical similarity based on nuclear charges. All other representation consider different atom types as orthogonal such that cross-element generalization is not possible. We reviewed two important concepts of atomistic representations: many-body expansions and chemical environments. While both have the potential to be unique, in many cases only a finite number of many-body terms, a small cutoff or a limited number of spherical harmonics coefficients are chosen to prevent overfitting or reduce computational cost. However, these methods are able to increase the reproduction accuracy of geometric structure systematically by addition of higher many-body terms [HL16] or tuning of hyperparameters [BKC13].

Having established this foundation, we will use the above concepts to develop deep learning architectures that are capable to learn representations fulfilling all desired properties introduced in this chapter.

Chapter 3

Deep tensor neural networks

In the previous chapter, we have reviewed existing, manually engineered features for molecules and materials. Even with all the discussed possibilities to represent atomistic systems, there are some clear advantages of learning a representation.

Scale adaption The data domain for the machine learning model can widely vary for different applications. E.g., in molecular dynamics the precise positions of atoms have to be reflected in the representation. In contrast, in virtual screening we only deal with equilibrium structures where the positional resolution may be much coarser. On the other hand, here, the ML model has to cover chemical compound space with varying compositions and system sizes.

Task adaption In the previous chapter, we discussed cross-element generalization, i.e., the ability to transfer knowledge from atoms of one atom type to those of another, for which a similarity of atom types would need to be encoded in the descriptor. This is complicated if not infeasible to do in a fixed representation as it would require to know the quantum chemical properties of each atom which is exactly what we attempt to learn. Moreover, a full quantum-mechanical specification may not be required if we only aim to predict specific chemical properties.

Insights In recent years, there have been significant efforts to explain predictions of non-linear ML models [Bae+10; SVZ13; ZF14; Bac+15; Mon+17; Kin+18]. These allow to extract insights about the model as well as the data. Specifying a representation already determines the vocabulary that these approaches can use to explain the prediction. Learning a complex feature space embedding of chemical environment enables us to find patterns in this feature space. Beyond that, learning representations with cross-element generalization allows for more general chemical insights beyond discrete atom types.

In the following, we will successively develop the molecular deep tensor neural network (DTNN): a deep learning architecture based on the insights

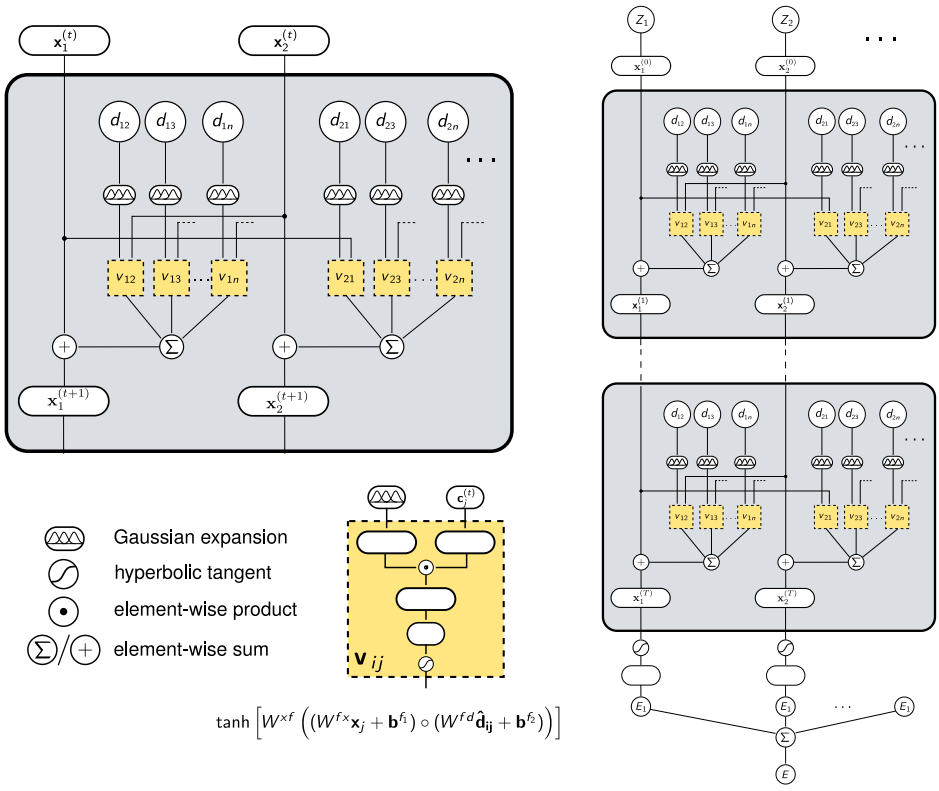


Figure 3.1: Visualization of the deep tensor neural network (DTNN) architecture. Chemical environments centered at atom i are represented by vector $\mathbf{x}_i^{(t)}$ that are repeatedly refined by additive pair-wise interaction corrections (grey). The interaction network (yellow) models the effect of a neighboring chemical environment $\mathbf{x}_j^{(t)}$ at distance d_{ij} on the refined environment. They are implemented using a factorized tensor layer (yellow). After the final representation $\mathbf{x}_i^{(T)}$ of every atom has been obtained, energy contributions are predicted atom-wise using a fully-connected output network with one hidden layer. Finally, these atom-wise energies are summed to yield the molecular energy.

from the last chapter, specifically using concepts from the many-body expansion and the notion of interlinked chemical environments. The proposed end-to-end method will exclusively use first principles information as input, i.e., the atomistic systems is encoded by their atom types and positions. Fig. 3.1 gives a visual overview of the proposed approach. We will limit the scope to the prediction of energies for molecules and discuss other chemical properties as well as materials with periodic boundary conditions in later chapters.

3.1 Embedding chemical environments

As discussed in Section 2.2.3, chemical environments have the advantage that they make the model scalable in terms of system size by decoupling local atom neighborhoods. In case of the energy, this can be written as

$$E(S) = \sum_{i=1}^{n_{\text{atoms}}} E_i((Z_1, \mathbf{r}_1), \ldots, (Z_{\text{atoms}}, \mathbf{r}_{n_{\text{atoms}}})).$$

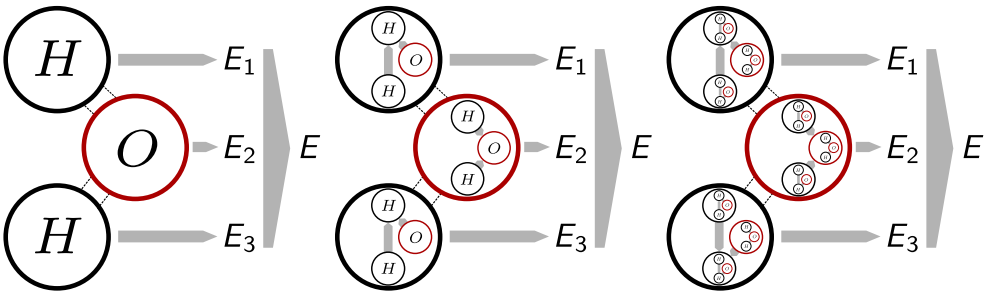
A drawback of the discussed methods was that atom types have been considered orthogonal rendering cross-element generalization impossible. Therefore, we will define an embedding in a feature space that represents a chemical environments consisting of the center atom as well as the interactions with the surrounding atoms.

As a starting point, we choose the most basic chemical environment: the single atom. Atom i of an atomistic system S is defined by its atom type, represented by it nuclear charge Z_i and position $\mathbf{r}_i$. To embed this in a vector space $\mathbb{R}^{n_{\text{feats}}}$, where n_{feats} is the number of features, we only need to consider its atom type Z_i. Since there exist only a limited number of chemical elements, we can simply define a lookup table of atom type embeddings $A \in \mathbb{R}^{n_{\text{types}} \times n_{\text{feats}}}$. Therefore, the initial embedding of the chemical environment,

$$\mathbf{x}_i^{(0)} = A_{[Z_i,:]}, \tag{3.1}$$

is simply the Z_ith row of the embedding matrix A, similar to how word embeddings are used in neural networks for natural language processing [Mik+13a; Mik+13b]. The embedding represents the quantum-chemical properties of an atom and, as such, enables cross-element generalization and can be interpreted as a dressed atom [Han+15]. Embeddings can either be learned in advance or, as in our case, initialized randomly and learned as a parameter of the neural network during back-propagation.

The obtained embedding obviously is invariant to rotation and translation as it does not use positional information at this stage. In the next section, we will introduce positional information through successive interactions of chemical environment to link and refine the defined embeddings in order to obtain more and more complete descriptions of the chemical environments.



(a) T=0 **(b)** T=1 **(c)** T=2

Figure 3.2: Illustration of how chemical environments are successively refined with higher-order interactions at the example of an H_2O molecule. Initially, each chemical environment $\mathbf{x}^{(0)}$ only represents of a single isolated atom (a). In successive, pairwise interaction refinements, increasingly more environment information is aggregated within the atom-wise representations (b,c). This allows for a decoupling of chemical environments and prediction of the energy from atom-wise energy contributions.

3.2 Interactions of chemical environments

In Section 2.2.2 and 2.2.3, we introduced the many-body expansion and how it can be applied to representations of chemical environments. Instead of expanding the energy in many-body terms directly, the decomposition into many-body interactions systematically guides the design of machine learning representations. In the same spirit, we will apply this to the previously defined representation $\mathbf{x}_i^{(1)}$.

A naive option would be to explicitly define explicit n-body neural networks $f^{(n)}$ for each $n \in [2, n_{\text{atoms}}]$:

$$\mathbf{x}_i = \mathbf{x}_i^{(1)} + \sum_{j \neq i} f^{(2)}((\mathbf{x}_i, \mathbf{r}_i), (\mathbf{x}_j, \mathbf{r}_j)) + \sum_{\substack{j,k \neq i \\ k \neq j}} f^{(3)}((\mathbf{x}_i, \mathbf{r}_i), (\mathbf{x}_j, \mathbf{r}_j), (\mathbf{x}_k, \mathbf{r}_k)) + \dots$$

In this approach the n-body networks have to be manually designed as they take more and more inputs with increasing n. Beyond that, rotational and translational invariance need to be either learned from data or manually enforced by using distances, angels, dihedral angles and so on, instead of atomic positions. Finally, this leads to $\sum_{n=1}^{N} \binom{n_{\text{atoms}}}{n}$ terms considering terms up to order N.

Since we do not deal with scalar energies anymore but with potentially complex representations of chemical environments, there is, however, a more efficient approach: Instead of explicitly modeling an n-body neural network, we design an interaction network $\mathbf{v} : \mathbb{R}^F \times \mathbb{R} \to \mathbb{R}^F$ that we use to model

perturbations

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \sum_{j \neq i} \mathbf{v}(\mathbf{x}_j^{(t)}, d_{ij}), \tag{3.2}$$

to the chemical environment $\mathbf{x}_i^{(t)}$ by its neighboring environments $\mathbf{x}_j^{(t)}$ depending on their distance $d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$. Applying this perturbation recursively, successively refines the representation and correlates chemical environments with increasing complexity. This behavior is illustrated in Fig. 3.2 using the water molecule as an example. The recursively applied interaction function has the advantage that only one interaction network has to be trained and evaluated. On top of that, we already incorporate all desired invariances, with respect to rotation, translation and atom indexing, since we only use pairwise distances to describe the geometry of the system.

The proposed approach shows similarities to known concepts from physics and machine learning. E.g., the additive perturbation of the representation is the core principle of residual neural networks [He+16]. Given a suitably defined $\mathbf{v}$ and large enough F, applying Eq. 3.2 repeatedly T times can represent the collection of all walks of length T ending at atom i. In that sense, this is closely related to diffusion kernels [KL02] or the transition function in graph neural networks as proposed by Scarselli et al. [Sca+09]. Several graph neural network architectures following a similar principle have been developed for molecular graphs [Duv+15; Kea+16] and other graph data [Bru+13; HBL15]. According to Gilmer et al. [Gil+17], these neural networks, including DTNNs, can be reformulated within the framework of message-passing neural networks, where the interaction function is considered a message-passing between nodes of a graph. Considering the initial representation as coefficients of atom-centered basis functions, the interaction network $\mathbf{v}$ can also be interpreted as reducing the overlap of those basis functions from two nearby atoms. Ideally, this leads to a final atomistic representation that allows for the additive partitioning of the target property into atom-wise contributions. In this picture, the DTNN learns an atom-centered basis that is adapted to the scale of the input data as well as the property to be predicted.

The embeddings and interaction networks as described above will be used in all model architectures developed throughout this thesis. The differences between those lie in the specific design of interactions $\mathbf{v}$ and output networks o.

3.3 Tensor layers and factorization

Modeling the interaction function $\mathbf{v}(\mathbf{x}_j, d_{ij})$ requires the combination of two inputs of different scale and dimensionality. A simple stacking of inputs, re-

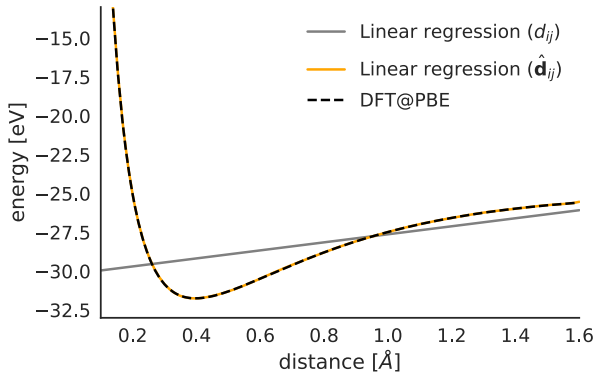


Figure 3.3: Comparison of features for regression of bond stretching energies of H_2. As features, we use scalar distances d_{ij} or distances in a radial basis $\hat{\mathbf{d}}_{ij}$ with $\Delta\mu = 0.1$ and $\gamma = 10$, respectively. The energies were computed by Brockherde et al. [Bro+17] with DFT at the PBE level of theory.

sulting in a fully-connected layer

$$\mathbf{v}(\mathbf{x}_j, d_{ij}) = W \begin{bmatrix} x_{j1} & \cdots & x_{jF} & d_{ij} \end{bmatrix}^\mathsf{T} + \mathbf{b}$$

as the first layer of the interaction network, only allows for additive composition of distance and chemical environment. A more expressive architecture should also allow for multiplicative compositions, as the distance can be seen as a non-linear damping factor: the larger the distance, the weaker we expect the influence of a neighboring chemical environment to be.

A related problem can be observed in neural networks for natural language processing when combining word representations and hidden states in recurrent neural networks [SMH11] or merging word representations in recursive neural networks [Soc+13]. This is solved by introducing tensor layers, where we introduce an additional weight tensor $V \in \mathbb{R}^{n_\text{feats} \times n_\text{feats} \times 1}$ that composes distance and chemical environment through a tensor product. The interaction term for feature k is then

$$v_k(\mathbf{x}_j, d_{ij}) = \mathbf{x}_j^{(t)} V_k d_{ij} + \left(W \begin{bmatrix} x_{j1} & \cdots & x_{jF} & d_{ij} \end{bmatrix}^\mathsf{T} \right)_k + b_k \tag{3.3}$$

with the tensor slice $V_k \in \mathbb{R}^{n_\text{feats} \times 1}$. Similar tensor layers have also been applied in tensor RNNs [SMH11]) and recursive neural tensor networks [Soc+13] in natural language processing as well as deep tensor neural networks for speech recognition [YDS13].

A crucial shortcoming of the interaction function in Eq. 3.3 is that the linear relationship of the scalar distance amounts to a linear scaling of the tensor slices V_k. This does clearly not characterize the non-linear interaction of atoms well. We solve this by representing the distance within a radial basis

$$\hat{\mathbf{d}}_{ij} = \left[\exp(-\gamma(\|\mathbf{r}_i - \mathbf{r}_j\| - k\Delta\mu)^2) \right]_{0 \leq k \leq r_\text{cut}/\Delta\mu} \tag{3.4}$$

with $\Delta\mu$ being the spacing of Gaussians with scale γ on a grid ranging from 0 to the distance cutoff r_{cut}. The radial basis grid is reminiscent of the partial radial distribution function representation [Sch+14] and the atom-centered symmetry function G_i^2 [Beh11] as described in Chapter 2. It decouples the distance regimes by increasing the dimension and serving as a non-linearity.

These effects are demonstrated in Fig. 3.3. A linear regression model taking directly the scalar distances is not able to fit the potential of stretching the bond of an H_2 molecule. However, in the feature space of the radial basis $\hat{\mathbf{d}}_{ij}$, a linear model is flexible enough to fit the potential perfectly. Therefore, it should also be flexible enough to express two-body interaction functions in order to arbitrarily perturb the features of the chemical environment representations. Additionally, we apply the hyperbolic tangent to the interaction function

$$v_{ijk} = \tanh\left(\mathbf{c}_j^{(t)} V_k \hat{\mathbf{d}}_{ij} + (W^c \mathbf{c}_j^{(t)})_k + (W^d \hat{\mathbf{d}}_{ij})_k + b_k\right), \tag{3.5}$$

to allow for further nonlinearity in the interaction perturbation. While neural networks with tanh activation functions tend to suffer from vanishing gradients [BSF94; Hoc98], the shortcut-connection $\mathbf{x}_i^{(t)}$ in Eq. 3.2 alleviates this effect as the gradient can pass through the linear term:

$$\frac{\partial \mathbf{x}_i^{(t+1)}}{\partial \mathbf{x}_j^{(t)}} = \begin{cases} \frac{\partial \mathbf{v}(\mathbf{x}_i^{(t)})}{\partial \mathbf{x}_i^{(t)}} & \text{if } i \neq j \\ \mathbf{1} & \text{if } i = j \end{cases} \tag{3.6}$$

While Eq. 3.5 sufficiently models the interaction function, it has the major drawback that the weight tensor $V \in \mathbb{R}^{n_{\text{feats}} \times n_{\text{feats}} \times n_{\text{rbf}}}$ incorporates many parameters which makes the tensor layer both computationally expensive and prone to overfitting. This can be solved by using a factorization of the tensor, as described by Taylor and Hinton [TH09], yielding

$$\mathbf{v}_{ij} = \tanh\left[W^{xf}\left((W^{fx}\mathbf{x}_j + \mathbf{b}^{f_1}) \circ (W^{fd}\hat{\mathbf{d}}_{\mathbf{ij}} + \mathbf{b}^{f_2})\right)\right], \tag{3.7}$$

where $\circ$ is the Hadamard product while $\mathbf{b}^{f_1}$ and $\mathbf{b}^{f_1}$ are the biases in factor space. The weight matrices $W^{fx} \in \mathbb{R}^{n_{\text{factors}} \times n_{\text{feats}}}$ and $W^{fd} \in \mathbb{R}^{n_{\text{factors}} \times n_{\text{rbf}}}$ map their respective inputs into factor space while $W^{xf} \in \mathbb{R}^{n_{\text{feats}} \times n_{\text{factors}}}$ maps the result of the interaction back into the feature space of chemical environments. Increasing the number of factors lets the factorization converge towards the full tensor product. On the other hand, choosing only a limited number of factors decreases the number of parameters significantly, thus, reducing the computational cost. On top of that, this can serve as a bottleneck to prevent overfitting.

3.4 Output network

After applying a fixed number of interaction perturbations T, we obtain the final atom-wise representation $\mathbf{x}_i^{(T)}$ that describes atom i in its broader chemi-

cal environment. Through this effective decoupling of chemical environments, we can now predict the energy as a sum over atom-wise energy contributions

$$E = \sum_{i=1}^{n_{\text{atoms}}} E_i = \sum_{i=1}^{n_{\text{atoms}}} o(\mathbf{x}_i^{(T)}), \tag{3.8}$$

where $o : \mathbb{R}^{n_{\text{feats}}} \to \mathbb{R}$ is an output network, mapping from representation to the atom-wise energy contributions. We model the output network using one hidden layer with tanh activation, predicting a scaled energy contribution $\hat{E}_i$. We obtain the final energy contribution

$$E_i = E_\sigma \hat{E}_i + E_\mu,$$

where E_μ is the mean and E_σ is the standard deviation of the energy per atom. These can be estimated before training from the training set using

$$E_\mu = \frac{1}{n_{\text{struct}}} \sum_{s=1}^{n_{\text{struct}}} E^{(m)} / n_{\text{atoms}}^{(m)}$$

$$E_\sigma = \sqrt{\frac{1}{n_{\text{struct}} - 1} \sum_{s=1}^{n_{\text{struct}}} (E^{(m)} / n_{\text{atoms}}^{(m)} - E_\mu)^2}$$

reference energy $E^{(m)}$ of a training example $n_{\text{atoms}}^{(m)}$ atoms. This constitutes a good starting point for training at the per-atom mean predictor.

3.5 Results

In the following, we demonstrate the versatility of DTNNs in learning representations of chemical environments in molecules. We will use our model to predict accurate energies for datasets with compositional as well as configurational degrees of freedom. For this, we will train DTNNs on data sets that include a diverse set of molecules across chemical compound space as well as on molecular dynamics trajectories of single molecules.

We employ DTNN models with up to $T = 3$ interaction refinements and consistently use $n_{\text{feats}} = 30$ features to represent chemical environments and $n_{\text{factors}} = 60$ in the factorized tensor layers for all trained models. All DTNN models are trained by minimizing the squared loss using stochastic gradient descent with momentum set to 0.9. We split the data into subsets for training validation and testing. We train all models for 3,000 epochs, where we validate for early stopping after every epoch. The final results are taken from the model with best validation error. The reported errors are averages over five repetitions of random subsampling on the respective test set.

Table 3.1: Mean abs. errors and standard errors over five repetitions of DTNNs in chemical compound space [Sch+17a]. The evaluated model use $T \in \{1, 2, 3\}$ interaction passes and are trained on te QM7b and QM9 data set with the given number of reference calculations N used for training. Errors are given in kcal mol^{-1}. Best results in **bold**.

Data set	N	T=1	T=2	T=3
QM7b (E_{PBE0})	5,768	1.28 ± 0.04	$\mathbf{1.04 \pm 0.02}$	$\mathbf{1.04 \pm 0.01}$
QM9 (U_0)	25,000	1.61 ± 0.02	1.09 ± 0.01	$\mathbf{1.04 \pm 0.02}$
	50,000	1.49 ± 0.02	0.96 ± 0.01	$\mathbf{0.94 \pm 0.01}$
	100,000	1.54 ± 0.03	0.93 ± 0.02	$\mathbf{0.84 \pm 0.02}$

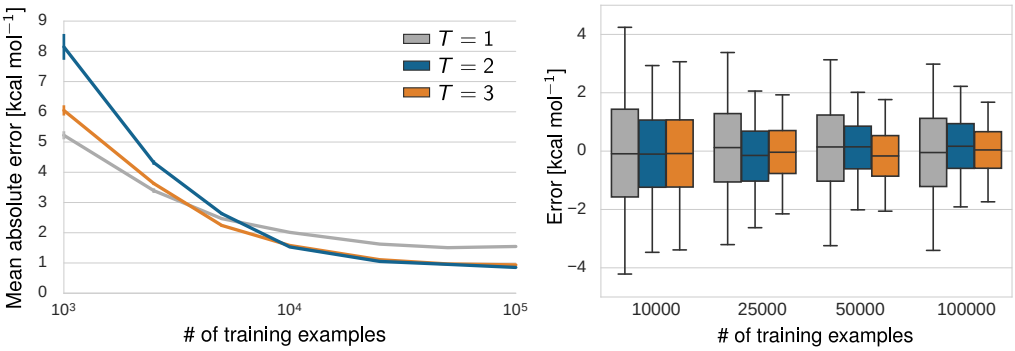


Figure 3.4: Learning curves and error distribution for DTNNs trained on QM9 with $T \in \{1, 2, 3\}$ [Sch+17a]. Left: mean abs. errors and standard errors as error bars depending on number of training examples. Right: error distribution with the box spanning from the 25% to the 75% quantile and the whiskers marking the 5% and 95% quantiles.

3.5.1 Chemical compound space

As a first challenge to our model, we evaluate its performance on the accurate prediction of energies from density functional theory (DFT) for equilibrium molecules across chemical compound space. In order to achieve this, DTNNs have to be able to generalize over molecules of different structures, compositions and sizes. For this purpose, we employ two datasets – QM7b and QM9 – of small organic molecules with up to 7 or 9 heavy atoms, respectively. Further details on the data are available in Appendix A. We use a validation set of 721 examples for QM7b (10%) and of 1,000 examples for QM9. The learning rate is set to 10^{-6} for both datasets.

Table 3.1 lists the performances of DTNN models with up to $T = 3$ interaction refinements for both data sets and varying training set sizes. We observe that the addition of refinement steps consistently improves the performance of the neural networks. DTNNs reach the chemical accuracy of

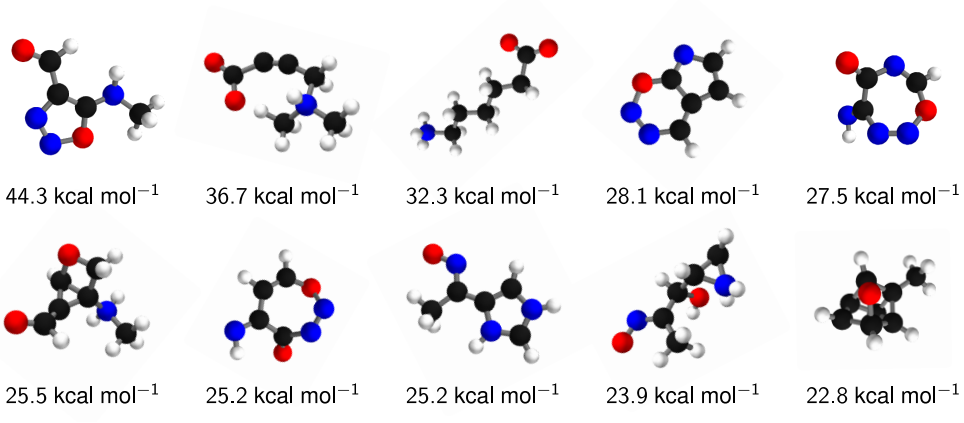


Figure 3.5: Molecules with the top-10 largest prediction errors of DTNN with $T = 2$ trained on 50k examples [Sch+17a].

1.0 kcal mol^{-1}when training on 5.8k examples from QM7b or 25k examples from QM9. Fig. 3.4 shows the learning curves for models with one, two and three interaction passes. While $T = 1$ only performs best for small datasets with up to about 2k-3k training examples, there is only a small difference for larger training sets with more than 10k reference calculations. The right side of Fig. 3.4 shows the distribution of errors for this regime. We see that with increasing amount of data and number of interactions, the error distributions get narrower. However, the plot does not give information about the extent of examples with extreme errors. Fig. 3.5 illustrates the molecules corresponding to those outliers, exhibiting errors up to 44.3 kcal mol^{-1}. While these errors seem disastrous, it is important to notice that the shown molecules exhibit unconventional bonding. Therefore, it is plausible that these molecules are not sufficiently represented by the training data.

A desirable property of predictions in chemical compound space is that the machine learning method is able to generalize across various system sizes. Fig. 3.6 shows mean abs. errors depending on the number of atoms of the test molecule. While small molecules exhibit on average errors larger than 1 kcal mol^{-1}, mean abs. errors of molecules with more than 18 atoms reach chemical accuracy. This behavior seems surprising at first since one might suspect that the prediction errors per atom accumulate due to the energy partitioning performed by our model. However, there are more large molecules in the dataset due to the rapidly increasing possibilities to combine atoms into valid molecules. Since the performed energy loss is not weighted by the number of atoms, this leads to an emphasis on large molecules, possibly at the cost of small ones. On the other hand, one could argue that predictions of larger molecules can be improved by knowledge about their local structure learned from smaller molecules.

In order to test this hypothesis, we train a DTNN on a set of 5k molecules with more than 20 atoms drawn from QM9. On an independent test set that includes the same kind of large molecules, the DTNN achieves a MAE of

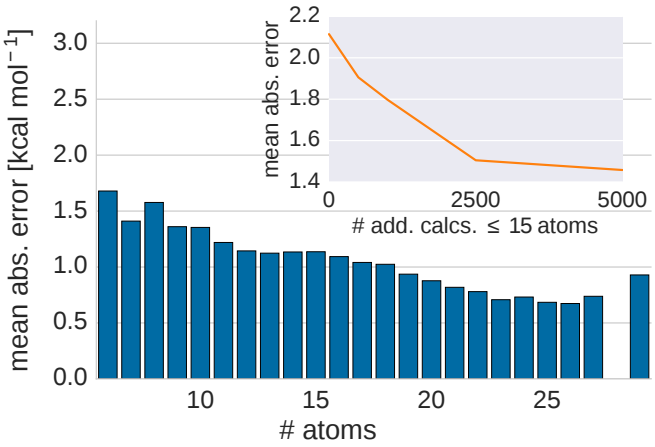


Figure 3.6: Dependence of energy prediction errors on the number of atoms with DTNN trained on QM9 [Sch+17a]. The mean absolute errors are shown for each molecule size separately indicating that larger molecules exhibit smaller errors. The inset shows the test error on large molecules ($\geq$ 20 atoms) for a DTNN trained on a set of 5,000 separate, large molecules ($\geq$ 20 atoms) while adding an increasing number of small molecules ($\leq$ 15 atoms).

2.1 kcal mol^{-1}. Next, we start to add smaller molecules with less than 15 atoms to the training set. As expected, the test error decreases to less than 1.5 kcal mol^{-1}. Therefore, we conclude that the DTNN model is able to generalize well from smaller to larger molecules.

While a large part of the variance of the energy in QM9 can be explained by the composition of molecules, DTNN clearly takes the geometry of the molecule into account. This can be demonstrated by the prediction results on the largest set of isomers in QM9 with the composition $C_7O_2H_{10}$ as they only differ in the positions of atoms. Fig. 3.7 shows the performance of DTNN trained on QM9 on the isomer subset. The distribution of predicted energies matches that of the reference calculation with the exception of a small bump at about -1840 kcal mol^{-1}. Looking at the scatter plot of the inset, this is likely caused by a couple of underestimated outliers at that energy level. Overall, the mean abs. error measured on the isomer subset is 0.89 kcal mol^{-1}. Another important aspect of energy prediction beyond accuracy is that the ranking of local energy minima is correct. Our model performs with a Kendall rank correlation coefficient $\tau = 0.969$ on the isomers ($\tau = 1$ for perfect agreement, $\tau = 0$ for statistical independence). This makes our model applicable to a stability ranking of these compounds.

3.5.2 Molecular dynamics

After we have demonstrated that DTNNs are able to represent compositional as well as structural changes and predict the associated energies with high

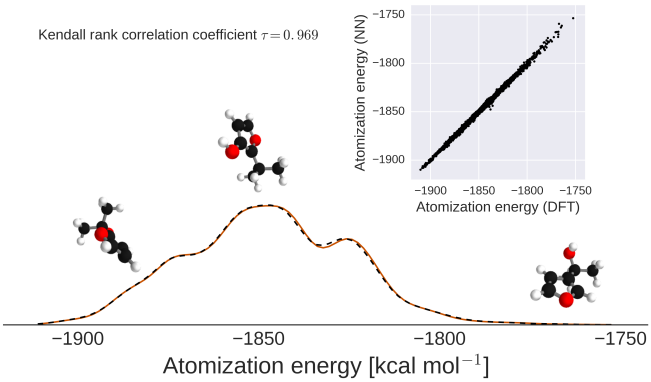


Figure 3.7: Prediction of $C_7O_2H_{10}$ isomer atomization energies [Sch+17a]. The DTNN was trained on the full QM9 database. The energy distribution was generated using kernel density estimation. The inset shows a scatter plot of DFT vs. predicted atomization energies.

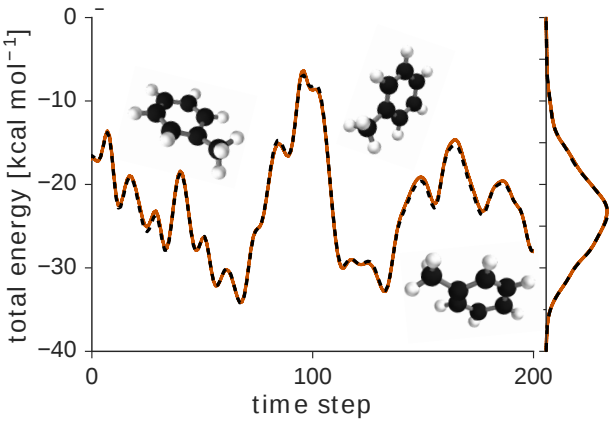


Figure 3.8: Short excerpt of the MD trajectory and associated energy distribution of toluene [Sch+17a]. The DFT energies (black) are plotted against the energy predictions of DTNN (orange).

accuracy, we go on to examine whether our model is also able to resolve small configurational changes. We will test this setting on molecular dynamics trajectories of single molecules. This presents a radically different challenge to the chemical compound space setting: While the composition stays constant, the datasets contain a much more diverse set of configurations as the MD simulation explores beyond the typical bond distances and angles exhibited by equilibrium molecules.

Table 3.2 shows the performance of DTNN on MD trajectories of four small organic molecules taken from the MD17 data collection. Details on this data is given in Appendix A. The learning rate is set to 10^{-4} and the validation sets consist of 1,000 examples for all MD trajectories. The mean absolute errors of all molecular trajectories are well below 1 kcal mol^{-1}. This is because the majority of the energy variation in QM9 comes from the composition and major structural changes, while there are only comparatively small conforma-

Table 3.2: Mean abs. errors and standard errors over five repetitions of DTNNs for molecular dynamics trajectories [Sch+17a]. The evaluated models use $T \in \{1, 2, 3\}$ interaction passes and are trained on MD trajectories of small organic molecules with the given number of reference calculations N used for training. The mean predictor is given as a baseline. Errors are given in kcal mol^{-1}. Best results in **bold**.

Dataset	N	mean pred.	T=1	T=2	T=3
Benzene	25k	1.86 ± 0.00	0.07 ± 0.00	0.05 ± 0.00	$\mathbf{0.04 \pm 0.00}$
	50k	1.86 ± 0.00	0.06 ± 0.00	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.04 \pm 0.00}$
	100k	1.86 ± 0.00	0.07 ± 0.00	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$
Toluene	25k	4.05 ± 0.00	0.48 ± 0.01	$\mathbf{0.20 \pm 0.00}$	0.23 ± 0.00
	50k	4.05 ± 0.00	0.44 ± 0.00	$\mathbf{0.18 \pm 0.00}$	$\mathbf{0.18 \pm 0.00}$
	100k	4.05 ± 0.00	0.42 ± 0.01	$\mathbf{0.16 \pm 0.00}$	0.17 ± 0.00
Malonaldehyde	25k	3.27 ± 0.00	0.54 ± 0.00	$\mathbf{0.23 \pm 0.00}$	$\mathbf{0.23 \pm 0.00}$
	50k	3.27 ± 0.00	0.49 ± 0.01	0.20 ± 0.00	$\mathbf{0.19 \pm 0.00}$
	100k	3.27 ± 0.00	0.51 ± 0.01	0.18 ± 0.00	$\mathbf{0.17 \pm 0.00}$
Salicylic acid	25k	4.30 ± 0.00	0.80 ± 0.02	$\mathbf{0.54 \pm 0.02}$	0.79 ± 0.02
	50k	4.30 ± 0.00	0.73 ± 0.01	$\mathbf{0.41 \pm 0.00}$	0.50 ± 0.01
	100k	4.30 ± 0.00	0.67 ± 0.01	$\mathbf{0.39 \pm 0.01}$	0.42 ± 0.01

tional perturbations caused by the MD simulation in the MD17 datasets. In this scenario, we observe that for three out of four molecules, best results are obtained using two interaction passes. This indicates that DTNN is not able to correctly extract the higher-order interactions beyond $T = 2$ with the given amount of data. To get an intuition of the obtained accuracy, Fig. 3.8 visualizes the prediction of a short fraction of the trajectory of toluene. All major features of the trajectory are well reproduced by the DTNN model. However, the low and high spikes tend to be slightly over- or underestimated, respectively.

For comparison, Table 3.3 shows results of a kernel ridge regression model with the Coulomb matrix as input features and the Matérn kernel taken from Chmiela et al. [Chm+17]. At first, it seems surprising that such a simple descriptor as the Coulomb matrix outperforms DTNN in this setting, moreover while using less training data. However, many of the weaknesses of the Coulomb matrix discussed in Chapter 2 do not apply here. Due to the fixed composition, there is no need for correct cross-element generalization or invariance to atom permutations. We just have to enumerate the atoms consistently across the whole trajectory. Thus, each atom is uniquely identified and the type information is encoded in the feature dimension, similar to the atom type ordering of bag-of-bonds. On the other hand, the Coulomb matrix has the advantage that the molecule is represented uniquely. In this case, this

Table 3.3: Mean abs. errors of kernel ridge regression using the Matérn kernel and the Coulomb matrix. Kernel ridge regression results are taken from Chmiela et al. [Chm+17]. The DTNN with the best performing T on 50,000 training examples is listed. Errors are given in kcal mol^{-1}. Best results in **bold**.

Dataset	KRR with CM		DTNN (best T)	
	N	mean abs. error	N	mean abs. error
Benzene	36,000	**0.04**	50,000	**0.04**
Toluene	45,000	**0.06**	50,000	0.18
Malonaldehyde	27,000	**0.11**	50,000	0.19
Salicylic acid	48,000	**0.10**	50,000	0.41

proves to be the deciding factor. Since DTNN is not able to learn higher-order interactions beyond $T = 2$, as discussed above, it cannot uniquely represent all details of the molecular geometry. Therefore, it lacks accuracy in some conformations, e.g., the spikes in Fig. 3.8.

Given larger molecules or a potential energy surface with reactions as a task, we expect that the ability of DTNN to decompose molecules into local environments would give DTNN an advantage over the Coulomb matrix. Similar chemical environments within the molecule can then be recognized by DTNN which improves generalization. Moreover, this scenario is more similar to the chemical compound space setting since atom assignment becomes ambiguous with greater atom movement or even changes in the bond structure of the molecule. In this case, the drawbacks of the Coulomb matrix apply again.

3.6 Analysis

After demonstrating that deep tensor neural networks are able to accurately predict energies for compositional and configurational degrees of freedom, we go on to analyze the obtained representation. In particular, we examine how the interaction passes of deep tensor neural networks influence the representation and whether chemically meaningful insights can be extracted. Beyond that we study the behavior of the model outside the training manifold for the special case of alchemical pathways.

3.6.1 Energy partitioning

The first aspect of our model we will take a closer look at is the implicit energy partitioning it provides. Having a consistent energy partitioning scheme presents a long-standing challenge in quantum-chemistry. Many alternative

schemes have been suggested that partition space, e.g. using Voronoi polyhedra [Fon+04], topological features of the electron density *Atoms in Molecules* [BB72] or *Hirshfeld surfaces* [Hir77; SB97].

As the existence of such variety suggests, there is no unique partitioning of a molecule into atom environments or its energy into atomic contributions. This applies in particular in our setting, where we have no information about the electron density, but can only infer terms from the many-body expansion based on our dataset of molecular geometries and energies. Given two distinct atoms A and B, we can write the energy in terms of the many-body expansion

$$E = E^{(1)}(A) + E^{(1)}(B) + E^{(2)}(A, B)$$

where $E^{(1)}$ and $E^{(2)}$ correspond to the 1- and 2-body energies. Based on this, we are able to partition the energy in terms of atomic contributions E_A, E_B as

$$E = \underbrace{E^{(1)}(A) + \alpha E^{(2)}(A, B)}_{E_A} + \underbrace{E^{(1)}(B) + (1 - \alpha)E^{(2)}(A, B)}_{E_B} \tag{3.9}$$

with $0 \leq \alpha \leq 1$. It is easy to see that there is no way to determine α uniquely in general, independent of the number of training examples we have available to fit the many-body terms. Only for the case that atoms A and B are of the same type, we can assume symmetry, i.e., $\alpha = 0.5$. Adding a third atom C already results in

$$\begin{aligned} E = &\underbrace{E^{(1)}(A) + \alpha E^{(2)}(A, B) + \beta E^{(2)}(A, C) + \lambda_1 E^{(3)}(A, B, C)}_{E_A} \\ &+ \underbrace{E^{(1)}(B) + (1 - \alpha)E^{(2)}(A, B) + \gamma E^{(2)}(A, C) + \lambda_2 E^{(3)}(A, B, C)}_{E_B} \\ &+ \underbrace{E^{(1)}(C) + (1 - \beta)E^{(2)}(A, C) + (1 - \gamma)E^{(2)}(A, C) + \lambda_3 E^{(3)}(A, B, C)}_{E_C}, \end{aligned} \tag{3.10}$$

with $0 \leq \beta \leq 1$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Since all many-body terms are essentially projected to one atom and the coefficients are independent of the n-body terms $E^{(n)}$, the non-uniqueness becomes more and more opaque as we keep adding distinct atoms to the system.

While the partition schemes above have to introduce additional constraints, the DTNN finds an energy partitioning by design in a data-driven fashion. Since the training of neural networks is a non-convex optimization problem, the learned representation may be different after each training, even if all hyper-parameters of the model such as the size of the atom representation and the number of interaction refinements is kept constant. Because of this and the discussed non-uniqueness, different partitionings may be obtained when training repeatedly. However, there still might be a preferred partitioning of energies enforced by the DTNN. Fig. 3.9 shows that this is not the case for the QM9 dataset. The distributions of atomic energy contributions per atom type

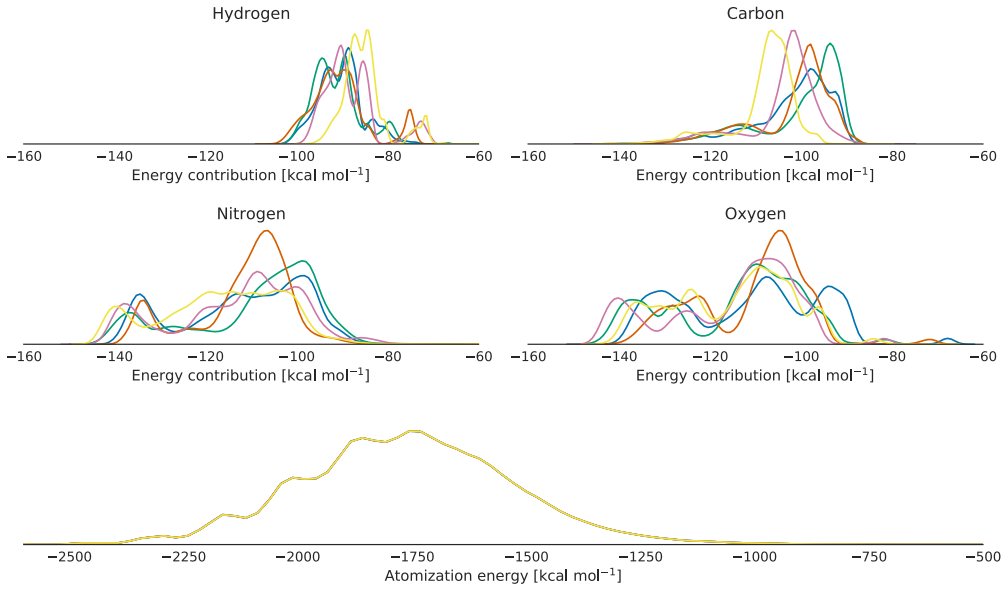


Figure 3.9: Distribution of energy contributions for atoms of types H, C, N, O and atomization energies from QM9 molecules predicted by DTNN models. The models were trained on 100k examples and use three interaction blocks. Each color corresponds to a model trained on a different subset. The distributions of atomization energy predictions agree across models (bottom).

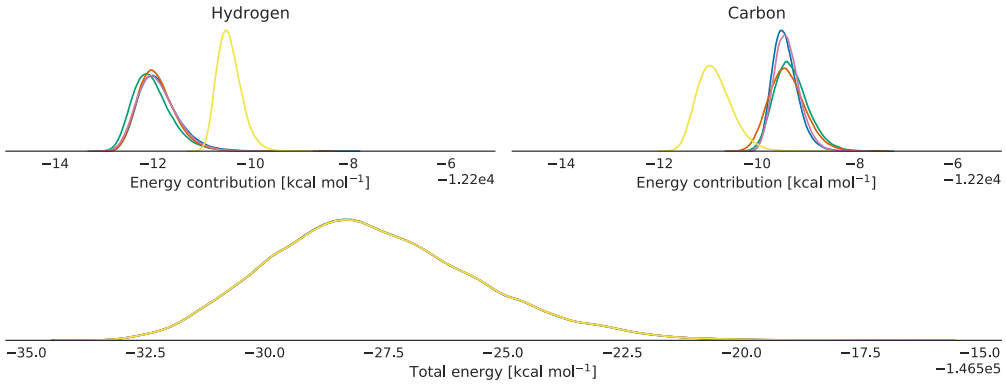


Figure 3.10: Distribution of energy contributions for atoms of types H, C and total energy predictions of DTNN models trained on benzene (C_6H_6). The models were trained on 50k examples and apply two interaction passes. Each color corresponds to a model trained on a different subset. The distributions of total energy predictions agree across models (bottom).

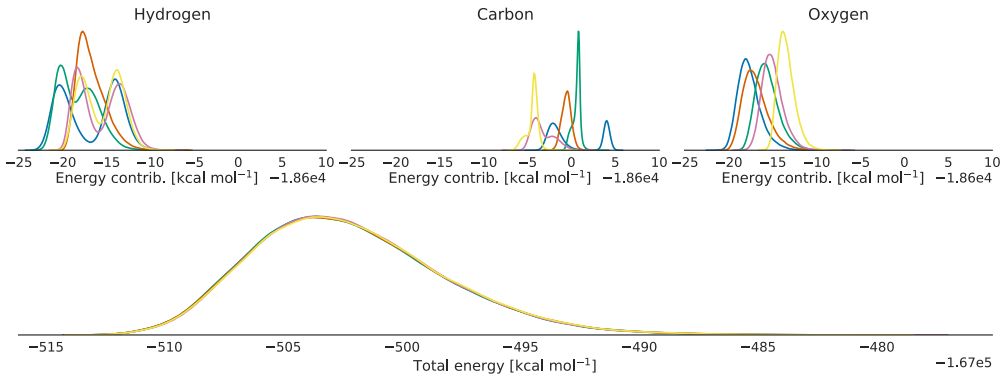


Figure 3.11: Distribution of energy contributions for atoms of types H, C, O and total energy predictions of DTNN models trained on malonaldehyde ($C_3H_4O_2$). The models were trained on 50k examples and apply two interaction passes. Each color corresponds to a model trained on a different subset. The distributions of total energy predictions agree across models (bottom).

predicted by DTNNs for multiple training repetitions vary significantly across models. On the other hand, the DTNNs share highly similar distributions of molecular energies that are shown at the bottom of Fig. 3.9. Complementary figures in Appendix B.1 show this in greater detail for the energy contributions of two models from Fig 3.9, plotted against each other in scatter plots.

As QM9 only contains equilibrium configurations, this might give the model too much flexibility to assign the energy contributions since the space of possible atom configurations is only sampled discretely. Therefore, we additionally examine DTNN models trained on molecular dynamics trajectories of benzene (Fig. 3.10) and malonaldehyde (Fig. 3.11). While the distribution of energy contributions of atoms in benzene are quite similar for four out of five repetitions, the distributions in the malonaldehyde dataset are more diverse again, which is similar to what we observed in QM9. Thus, this behavior does not appear to depend on the range of conformations present in the data set, but rather on the number of distinct atom types in the data. This is also supported by our theoretical argument in Eqs. 3.9 and 3.10. We conclude that deep tensor neural networks learn different energy partitioning schemes, that are equivalent in prediction accuracy.

3.6.2 Local chemical potentials

After having discussed the non-uniqueness of atom-wise energy contributions, we go on to examine the representation regarding spatial changes and interactions. To this end, we introduce a test charge p to the atomistic system which we will use to probe the space surrounding the atoms. Since we can only represent atoms in our model, the test charge is bound to be an atom in our model. This brings the problem that the molecule would be drastically

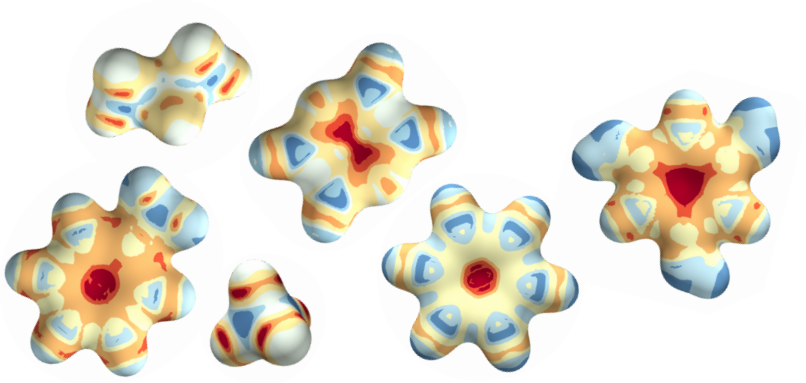


Figure 3.12: Local chemical potentials $\Omega_H^M(\mathbf{r})$ of various molecules from QM9 [Sch+17a]. We have used a hydrogen probe atom and a DTNN model with two interaction passes. The potential is plotted on an isosurface with $\sum_i \|\mathbf{r} - \mathbf{r}_i\|^{-2} = 3.8$ Å^{-2}.

influenced by adding another atom and, moreover, that the resulting molecule is bound to leave the training manifold if we trained the neural network only on equilibrium configuration or single molecular dynamics trajectories with a fixed number of atoms. We solve this by letting the probe atom feel the influence of the molecule, but not vice versa. This allows us to define a local chemical potential $\Omega_A^M(\mathbf{r})$ of the molecules M as the energy of the test charge of atom type A located at position $\mathbf{r}$. It is important to note that this potential does not correspond to the actual potential of the molecule, but is a tool for us to visualize the spatial structure of the representation.

Fig. 3.12 visualizes such potentials for a DTNN trained on QM9 with two interaction passes on a smooth isosurface with constant $\sum_i \|\mathbf{r} - \mathbf{r}_i\|^{-2}$ around a selection of molecules from the dataset. The shown potentials clearly reflect the expected symmetries that stem from the rotational and translational invariance of DTNN, and even chemical concepts such as bond saturation and different degrees of aromaticity.

Fig. 3.13 illustrates how the DTNN architecture has to be modified in order to predict the local potential $\Omega_A^M(\mathbf{r})$. First, we represent the test charge by a virtual probe atom with charge Z_p at position $\mathbf{r}_p$ which gives us an initial embedding

$$\mathbf{x}_p = A_{[Z_p,:]} \tag{3.11}$$

from the embedding matrix A learned by DTNN. Analogue to the interaction refinements defined in Eq. 3.2, we let the atoms of molecule act on the probe:

$$\mathbf{x}_p^{(t+1)} = \mathbf{x}_p^{(t)} + \sum_{j=1}^{n_{\text{atoms}}} \mathbf{v}(\mathbf{x}_j^{(t)}, d_{ij}), \tag{3.12}$$

Finally, we obtain the probe energy by applying the output network to the

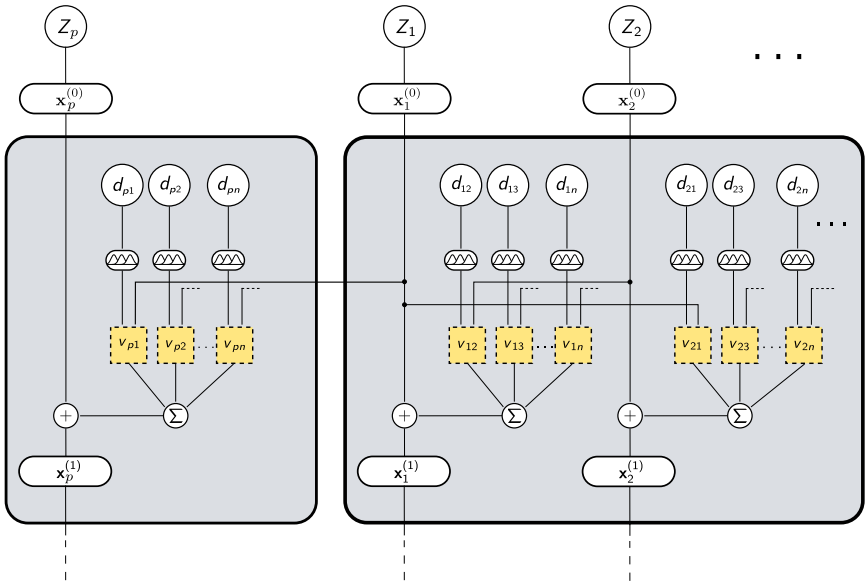


Figure 3.13: Visualization of how local chemical potentials are calculated [Sch+17a]. The left part represents the probe atom that acts as a test charge

probe representation

$$\Omega_A^M(\mathbf{r}) = o(\mathbf{x}_p^{(T)}). \tag{3.13}$$

Fig. 3.14 demonstrates the influence of the probe atom on the local potential. Even though the probe does not influence the molecule, each probe atom representation reacts differently to the presence of the molecular atoms in terms of the predicted energy. While all probe atoms yield structurally similar potentials, there are differences in the energy ranges as well as sensitivity towards interatomic interactions. E.g., the hydrogen probe has a compact energy range of 60 kcal mol^{-1}and shows fine-grained features such as low energy near hydrogen sites and at the center of the ring and high energies near sites of carbon and oxygen. On the other hand, the energy of the carbon probe decreases much quicker.

We will focus on using the acquired visualizations to further understanding of the inner workings of DTNNs. Therefore, we observe how the local potentials change with the number of interaction passes and the used training set. Fig. 3.15 shows a comparison between benzene potentials using a hydrogen probe of DTNNs trained on QM9 and, respectively, the MD17 trajectory of benzene. For each training dataset, we show models with $T \in \{1, 2, 3\}$ interaction passes. Since the MD17 models are trained on total energies instead of atomization energies, the energies are significantly lower. However, we focus exclusively on the structural features of $\Omega_A^M(\mathbf{r})$. The models trained on MD17 show much clearer distinguished regions corresponding to low and high ener-

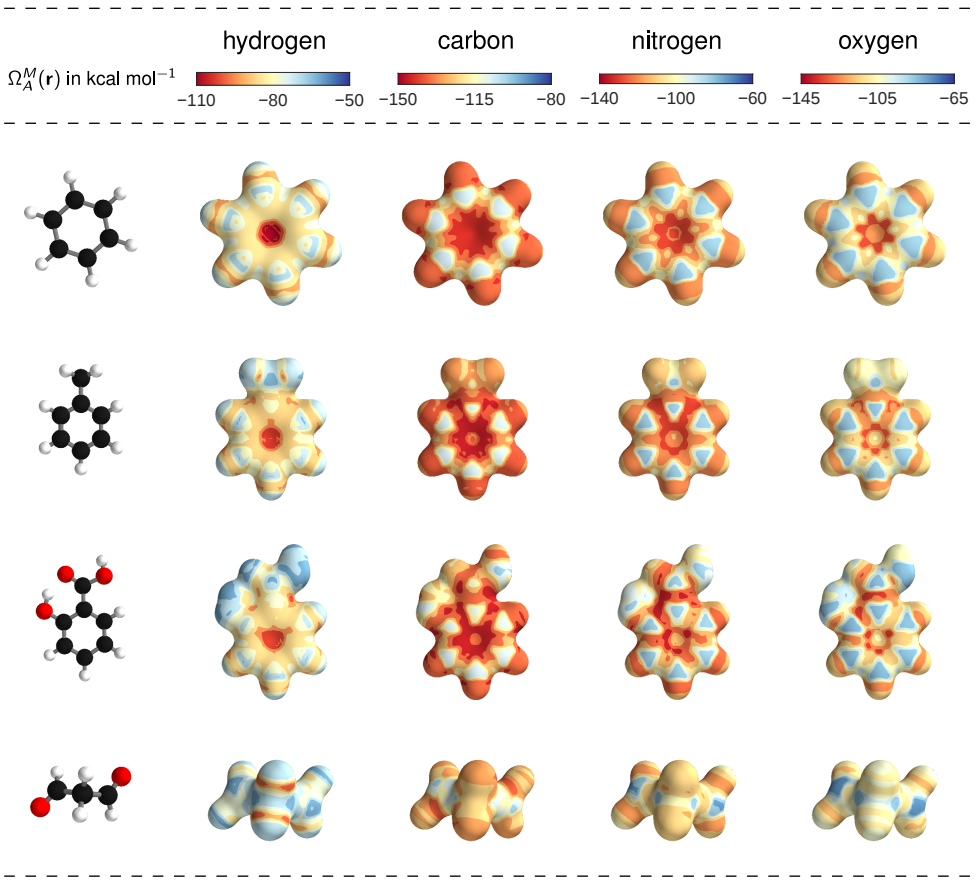


Figure 3.14: Local chemical potentials $\Omega_A^M(\mathbf{r})$ for benzene, toluene, salicylic acid and malonaldehyde with probe atoms of type A $\in \{$H, C, N, O$\}$ [Sch+17a]. All potentials are plotted on an isosurface with $\sum_i \|\mathbf{r} - \mathbf{r}_i\|^{-2} = 3.8$ Å^{-2} and energy ranges are adjusted per column.

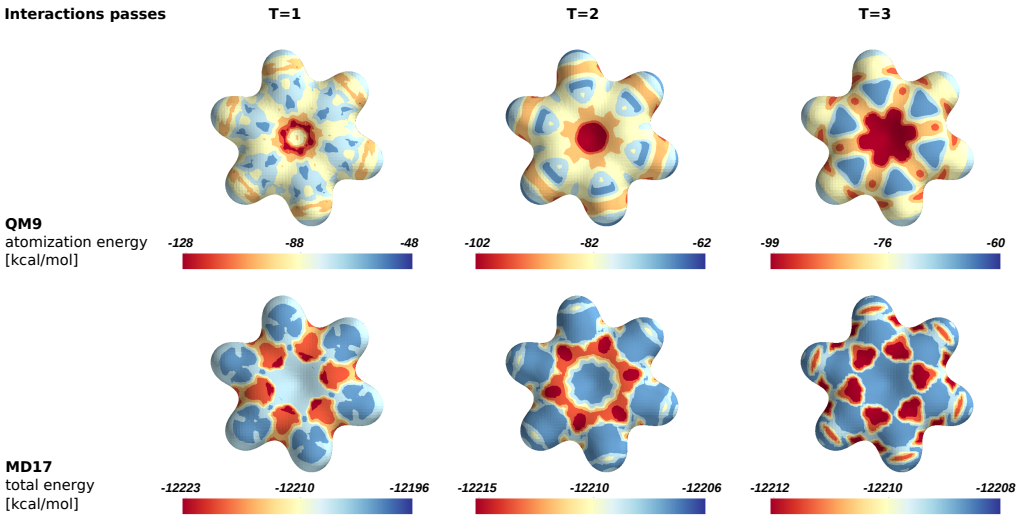


Figure 3.15: Local chemical potentials $\Omega_H^M(\mathbf{r})$ for benzene using DTNNs trained on QM9 (top) and an MD17 trajectory of benzene (bottom) using T interaction passes. All potentials are plotted on an isosurface with $\sum_i \|\mathbf{r} - \mathbf{r}_i\|^{-2} = 3.8$ Å^{-2} and energy ranges are adjusted per molecule.

gies. Since the MD17 model was trained on a single MD trajectory, it includes only the set of interactions present in benzene, however, is able to model local deformations. This can only be achieved by a smoother interaction function, while the large variety of interactions only covers typical bond lengths.

Another aspect to examine is the change of the local chemical potentials in Fig. 3.15 with increasing interaction passes T. For both QM9 and MD17 models, the models with $T = 1$ exhibit sharp features that appear to be artifacts from the insufficient pair-wise interactions that these DTNNs are able to represent. With higher number of interaction passes, the potentials become smoother. This can be explained as the DTNN is modeled similar to a diffusion process [KL02]. The effect of this can be observed in particular for the MD17 model: while the low-energy areas for $T = 1$ are concentrated at the carbon ring, they are partially propagated to the hydrogens for two and three interaction passes. This leads not only to low-energy areas near the hydrogen sites but also in a compression of the energy range. In an extreme scenario, one could think of a representation where the energy contributions are completely delocalized and equally distributed between the atoms. Therefore, the DTNN architecture is ideally suited for the energy, however, might not be suited for properties that require localized structural information.

3.6.3 Alchemical pathways

An important application of energy prediction with ML is the discovery of stable, low-energy compounds. A DTNN model trained on QM9 could be used for this task as it is defined for the complete chemical space and not just for

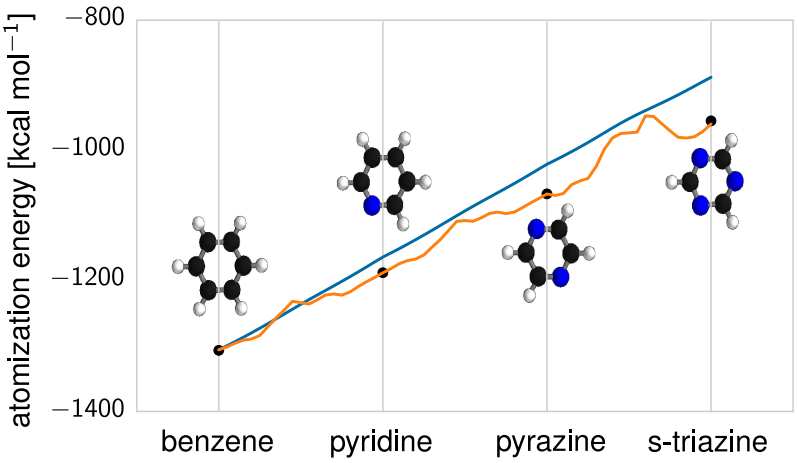


Figure 3.16: Alchemical path from benzene to s-triazine [Sch+17a]. The path was generated with fixed atom position (blue) as well as relaxed atom positions (orange).

discrete chemical graphs. For this, the model has to behave rather smoothly outside of its training domain. As QM9 did not include non-equilibrium configurations that would produce energy barriers, this might indeed be the case. To be able to smoothly optimize in chemical compound space, we also have to be able to blend atoms in and out as well as morph between atom types. This is called an alchemical reaction [Lil13]. While it does not reflect nature, it opens up reaction pathways for our search. Therefore, one has to force the search to arrive at a chemically valid setting at the end of the optimization.

To generate a chemical path, we morph atom types by interpolating linearly between atom type representations. Given two nuclear charges $Z_a, Z_b \in \mathbb{N}$, we define the embedding for any charge $Z_i = \alpha_i Z_a + (1-\alpha)Z_b$ with $0 \leq \alpha \leq 1$ as

$$\mathbf{x}_i^{(0)} = \alpha_i A_{[Z_a,:]} + (1-\alpha_i) A_{[Z_b,:]}. \tag{3.14}$$

Similarly, in order to add or remove atoms, we introduce fading factors $\beta_1, \ldots, \beta_n \in [0,1]$ for each atom. This way, interactions with other atoms

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \sum_{j \neq i} \beta_j v(\mathbf{x}_j^{(t)}, d_{ij}) \tag{3.15}$$

as well as energy contributions to the molecular energy $E = \sum_{i=1}^{n} \beta_i E_i$ can be faded out smoothly.

Using this, we show alchemical reactions from benzene over pyridine and pyrazine to s-triazine in Fig. 3.16. If we retain the geometry of benzene and only morph and blend atoms to reach the target composition (blue), we observe a virtually linear rise in energy from benzene to s-triazine. When adding linear interpolation of atom positions to the reaction (orange), the energy profile gets rougher due to suboptimal atom distances during the path, but stays below the composition-only reaction. This is in agreement with chemistry, since the non-equilibrium configurations are expected to have higher energies.

Note that we show only one possible alchemical path that was easy to generate manually. When performing an alchemical optimization even smoother paths might be found. On the other hand, the optimization might also be led astray by unnatural minima, similar to those causing adversarial examples [Sze+14; GSS15]. Furthermore, in order to arrive at equilibrium configurations, a model that is trained also on non-relaxed molecules is required to correctly model the energy barriers between equilibriums. In this case, the alchemical pathways are needed even more to circumvent these barriers. Performing an alchemical optimization using a suitable training set with both compositional and configurational degrees of freedom is subject to future work.

3.7 Summary and discussion

In this chapter, we have introduced a general framework to learn representations of atomistic systems from first-principles information. Starting from embeddings of single atoms, we have systematically constructed complex atom-wise representations of chemical environments by modeling repeated pairwise interactions. As a concrete implementation of such interactions, we have proposed a neural network architecture, where the perturbation by a neighboring environment is modeled using a factorized tensor layer. We could show that these deep tensor neural networks are able to predict chemically accurate energies throughout chemical and configurational space. Furthermore, we have analyzed the obtained representations regarding the learned energy partitioning, the spatial structure of the learned interactions as well as the smoothness of the obtained potential energy surface outside of the training domain.

The intrinsic non-uniqueness of the energy partitioning could not be resolved by DTNNs. The existence of many equivalent representations is analog to solutions of the electronic problem in different basis sets. However, this must not be the case in other neural networks since there might be a simple, preferred solution when using a different approach. This would also be a strong indicator for a more suitable neural network architecture.

We have hinted at possible applications such as virtual screening of molecular properties or modeling of potential energy surfaces for molecular dynamics simulations which we will explore further in later chapters. Another application that is subject to future work is the optimization of molecular properties in alchemical space. In the next chapter, we will build upon the introduced framework in order to further improve the prediction accuracy and extend the scope of the architecture to other chemical properties as well as atomistic systems with periodic boundary conditions.

Chapter 4

Continuous-filter convolutional neural networks

In the last chapter, we have established the deep tensor neural network framework. An important design decision is how to model the quantum interactions between atoms. While DTNNs have used factorized tensor layers, we will employ convolutions in this chapter. In particular, we will examine how convolutional layers can model atoms at arbitrary positions instead of uniformly sampled data such as pixels on a grid or discrete time series.

An issue of the DTNN implementation presented in Chapter 3 is its lack of separation between learning atom-wise representations and interactions. Both subtasks are essentially handled in the interaction function

$$\mathbf{v}_i = \sum_{j \neq i} \tanh \left[W^{xf} \left((W^{fx} \mathbf{x}_j + \mathbf{b}^{f_1}) \circ (W^{fd} \hat{\mathbf{d}}_{\mathbf{ij}} + \mathbf{b}^{f_2}) \right) \right],$$

i.e., atom and distance information are directly merged in the factorized tensor layer. In contrast, the SchNet architecture, which we will introduce in this chapter, uses *filter-generating networks* to learn the interaction function which in turn will then modulate the atom-wise representations linearly. As a beneficial side-effect, this will allow us to define periodic filters for materials.

First, we will introduce *continuous-filter convolutional* (cfconv) layers, which are generalizations of discrete convolutional layers that are commonly used in deep learning. Then, we will apply these to modeling of the interaction function in the DTNN framework. Finally, we evaluate the prediction performance of the new neural network architecture and analyze how the learned representations have changed compared to those of DTNNs.

4.1 Convolutional layers

Convolutional neural networks [LeC+89] have led to major breakthroughs applying machine learning to images [KSH12], videos [Kar+14] or audio data [Oor+16]. Given a two-dimensional neuron layer, e.g. for the hidden activations within a convolutional neural network for images,

$$X^l = \begin{bmatrix} \mathbf{x}_{11} & \dots & \mathbf{x}_{1L} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{K1} & \dots & \mathbf{x}_{KL} \end{bmatrix}$$

with size $K = 2k_{\text{cut}} + 1$ by $L = 2k_{\text{cut}} + 1$ and each entry $\mathbf{x}_{i,j} \in \mathbb{R}^{F_{\text{in}}}$ having F_{in} features, the output of convolutional layer l is defined as

$$\mathbf{x}_{i,j}^{l+1} = (X^l * W)(i,j) = \sum_{k=-k_{\text{cut}}}^{k_{\text{cut}}} \sum_{l=-l_{\text{cut}}}^{l_{\text{cut}}} W_{kl} \mathbf{x}_{i-k,j-l}^{l} + \mathbf{b}. \tag{4.1}$$

The star symbol "$*$" represents the convolution, the filter tensor $W \in \mathbb{R}^{K \times L \times F_{\text{out}} \times F_{\text{in}}}$ and the bias $b \in \mathbb{R}^{F_{\text{out}}}$) are learned during training.

This leads to some favorable properties for learning of structured data: First, the finite impulse responses of small filters, which can also be interpreted as locally-connected neurons, enable the neural network to recognize local patterns [FM82; LeC+89]. Second, the weight sharing across the data dimensions i, j leads to translational invariance of these patterns [LB+95]. This makes convolutional neural networks very efficient at recognizing local structure with a relatively small set of weights compared to fully-connected layers.

Due to these advantages, they should also be ideally suited to model quantum interactions in atomistic systems. In this application, we also have strong local interactions and require translational invariance of the system.

4.2 Continuous-filter convolutions

The commonly used convolutional layers, as presented above, employ discrete filter tensors since they are usually applied to uniformly sampled data, e.g. digital images, video and audio. However, this is not applicable for atomistic systems, because the atoms can be located at arbitrary positions. E.g. when predicting a potential energy surface, the output of a convolutional layer will change rapidly when an atom moves from one grid cell to the next. Fig 4.1 (left) illustrates how this results in a discontinuous energy surface. Especially when we require a correct derivative of the energy prediction, e.g. for the prediction of atomic forces (see Chapter 5), this is not a viable solution.

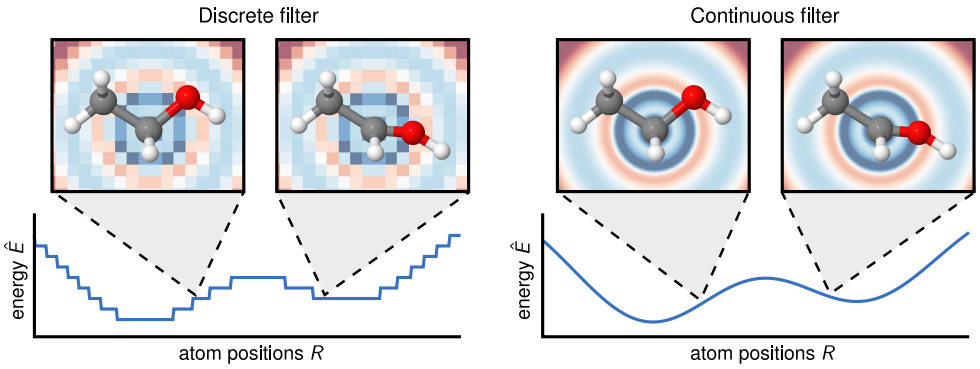


Figure 4.1: Discrete vs. continuous convolution filters [Sch+17b]. The discrete filter (left) is not able to capture the subtle positional changes of the atoms resulting in discontinuous energy predictions $\hat{E}$ (bottom left). The continuous filter captures these changes and yields smooth energy predictions (bottom right).

Even though discrete convolutions are commonly used in deep learning and signal processing, the convolution is defined for continuous functions. E.g., for data in 3-dimensional space, we can convolve arbitrary functions $\rho : \mathbb{R}^3 \to \mathbb{R}^F$ and $W : \mathbb{R}^3 \to \mathbb{R}^F$ as follows:

$$(\rho * W)(\mathbf{r}) = \int_{\mathbf{r}_a \in \mathbb{R}^3} \rho(\mathbf{r}_a) \circ W(\mathbf{r} - \mathbf{r}_a) d\mathbf{r}_a. \tag{4.2}$$

Here, "$\circ$" is the element-wise product, i.e. we apply convolutions to all F feature dimensions separately. Now let's assume that ρ describes an atomistic system by atom-wise features at discrete position

$$\rho^l(\mathbf{r}) = \sum_{i=1}^{n_{\text{atoms}}} \mathbb{1}_{\{\mathbf{r}=\mathbf{r}_i\}} \mathbf{x}_i^l, \tag{4.3}$$

where $\mathbf{x}_i^l$ is the representation of the chemical environment of atom i at layer l, analogous to how this was defined in the DTNN. On the other hand, the filter function W describes the interaction of feature maps with an atom at the relative position $\mathbf{r} - \mathbf{r}_i$. The filter functions can be modeled by a *filter-generating* neural network similar to those used in dynamic filter networks [Jia+16]. Plugging this into Eq. 4.2, we get

$$\begin{aligned}
(\rho^l * W)(\mathbf{r}) &= \int_{\mathbf{r}_a \in \mathbb{R}^3} \left(\sum_{j=1}^{n_{\text{atoms}}} \mathbb{1}_{\{\mathbf{r}_a=\mathbf{r}_j\}} \mathbf{x}_j^l \right) \circ W(\mathbf{r} - \mathbf{r}_a) d\mathbf{r}_a \\
&= \sum_{j=1}^{n_{\text{atoms}}} \int_{\mathbf{r}_a \in \mathbb{R}^3} \mathbb{1}_{\{\mathbf{r}_a=\mathbf{r}_j\}} \mathbf{x}_j^l \circ W(\mathbf{r} - \mathbf{r}_a) d\mathbf{r}_a \\
&= \sum_{j=1}^{n_{\text{atoms}}} \mathbf{x}_j^l \circ W(\mathbf{r} - \mathbf{r}_j)
\end{aligned} \tag{4.4}$$

This gives us a continuous function in space which represents how the atoms of the system act on another location in space. To obtain the influence of the

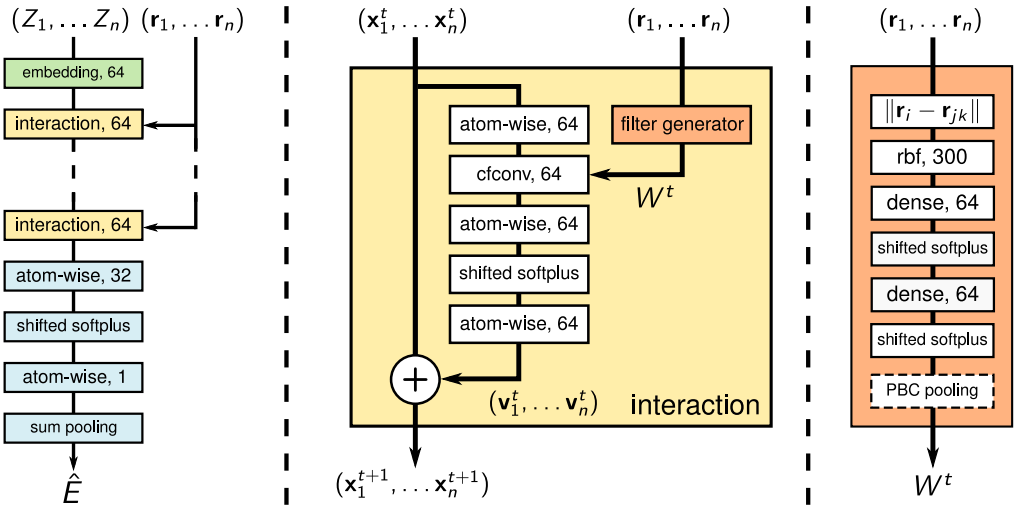


Figure 4.2: The SchNet architecture [Sch+17b; Sch+18]. The illustration shows an architectural overview (left), the interaction block (middle) and the filter-generating network (right). The shifted softplus activation function is defined as $\mathrm{ssp}(x) = \ln(0.5e^x + 0.5)$. The number of neurons used in the employed SchNet models, if not specified otherwise, is given for each parameterized layer.

atoms on each other, we only need to calculate this at the atom positions

$$\mathbf{x}_i^{l+1} = (X^l * W^l)_i = \sum_{j=1}^{n_{\text{atoms}}} \mathbf{x}_j^l \circ W(\mathbf{r}_i - \mathbf{r}_j), \tag{4.5}$$

i.e., we perform a convolution at discrete locations in space using a continuous filter function W.

In the following, we will develop an improved deep learning architecture using such continuous-filter convolutional (*cfconv*) layers to model quantum interactions. In particular, we will discuss how to design the filter-generating networks in order to guarantee all required invariances.

4.3 SchNet

Building upon the principles of the previously described DTNNs, we propose SchNet as an improved neural network architecture for learning representations for molecules and materials. Both methods share a number of their essential building blocks, such as atom-wise embeddings, additive interaction refinements and atom-wise contributions to the property to be predicted. Due to the similarities to the DTNN, we will shortly describe the general structure of SchNet, recapitulate reoccurring building blocks and point out noteworthy differences.

Fig. 4.2 illustrates the proposed model architecture, which exhibits the same overall structure as DTNNs. First, the representations of the chemical

environments are initialized using an embedding lookup layer

$$\mathbf{x}_i^{(0)} = A_{[Z_i,:]},$$

just like in the DTNN, depicted in green in the left panel of Fig. 4.2. Next, we apply several interaction blocks to these atom-wise representations, depicted in yellow. While they serve the same purpose as the interaction passes of DTNN, they present the largest change in the architecture. Most importantly, the tensor layers of DTNNs are replaced by continuous-filter convolutions and respective filter-generating networks in the interaction blocks. We will describe these in detail in Section 4.3.1. Finally, an output network (blue in Fig. 4.2) obtains the final prediction from the atomic environments using atom-wise layers l, i.e., fully-connected layers

$$\mathbf{x}_i^{(l+1)} = W^{(l)}\mathbf{x}_i^{(l)} + \mathbf{b}^{(l)} \tag{4.6}$$

that are applied separately to each atom i with tied weights $W^{(l)}$.

4.3.1 Interaction blocks

Analogous to the interaction passes of DTNNs, each interaction block of SchNet models pair-wise interactions of chemical environments, thereby distributing many-body information across the molecule. In contrast to DTNNs, there is not one single interaction function $\mathbf{v}(\mathbf{x}_j, d_{ij})$ that is repeatedly applied, but we use a different convolution filter and untied weights in the atom-wise layers of each block. We perturb the representations of atomic environments by an interaction refinement modeled as a residual building block [He+16]

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t)}, \tag{4.7}$$

where $\mathbf{v}_i^{(t)}$ is the residual mapping of atom i.

The middle panel of Fig. 4.2 illustrates how this residual is obtained. Most importantly, we use a cfconv layer to convolve the chemical environments $\mathbf{x}_i^t$ with continuous filters $W^t(\mathbf{r}_i - \mathbf{r}_j)$ following Eq. 4.5. Since our cfconv layers are applied feature-wise, we achieve the cross-talk between feature maps by atom-wise layers before and after the convolution. This is analogous to depth-wise separable convolutional layers in Xception nets [Cho17] which could outperform the architecturally similar InceptionV3 [Sze+16] on the ImageNet dataset [Den+09] while having slighly less parameters. Beyond a potential gain in accuracy, feature-wise convolutional layers reduce the number of filters. This reduces the computational cost, in particular for continuous-filter convolutions, where each filter has to be computed by the filter-generating network.

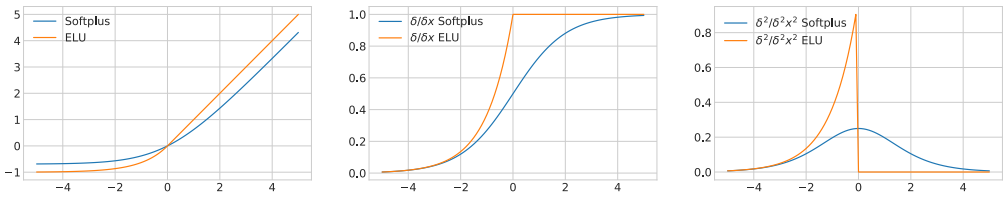


Figure 4.3: Comparison of shifted softplus and ELU activation function. We show plots of the activation functions (left), and their first (middle) and second derivatives (right).

Activation function

We use a softplus activation function [Dug+01] that was shifted to cross the origin:

$$f(x) = \ln\left(\frac{1}{2}e^x + \frac{1}{2}\right). \tag{4.8}$$

Fig. 4.3 shows the similarity of this activation function to the recently popular exponential linear units (ELU) [CUH15] non-linearity

$$f(x) = \begin{cases} e^x - 1 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases} \tag{4.9}$$

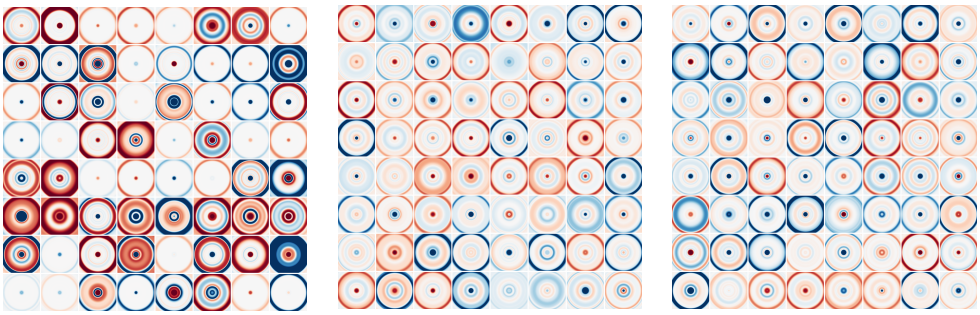
The first and second derivatives for ELU and softplus are shown in the middle and right panel of Fig. 4.3, respectively. A crucial difference is that the shifted softplus has infinite order of continuity while ELUs have a discontinuity starting with the 2nd derivative. As discussed in Chapter 2, the differentiability of the model, and therefore also of the employed activation functions is crucial for the prediction of atomic forces. As shown in Fig. 4.3, the first derivative of the softplus activation function is the sigmoid – a common activation function itself – which makes it an ideal choice for the training of forces (see Chapter 5).

Comparison of SchNet and DTNN

Before moving on to detail the filter-generating networks used by SchNet, we compare the interaction blocks and the factorized tensor layers of DTNN. Recalling the DTNN interaction refinements

$$\mathbf{v}_i^{(t)} = \sum_{j \neq i} \tanh\left[W^{xf}\left((W^{fx}\mathbf{x}_j + \mathbf{b}^{f_1}) \circ (W^{fd}\hat{\mathbf{d}}_{\mathbf{ij}} + \mathbf{b}^{f_2})\right)\right], \tag{4.10}$$

we recognize that the crucial change here is the replacement of the hyperbolic tangent *within* the sum over neighbors, with the softplus outside of the sum. If we ignore the activation function in Eq. 4.10, we can reformulate a linear



(a) 1st interaction block **(b)** 2nd interaction block **(c)** 3rd interaction block

Figure 4.4: Continuous convolution filters of SchNet [Sch+17b]. 10x10 Å cuts through all 64 radial, three-dimensional filters in each interaction block. The model has been trained on a molecular dynamics trajectory of ethanol. Negative values are blue, positive values are red.

variant $\tilde{\mathbf{v}}_i^{(t)}$ of DTNN interactions as

$$\mathbf{h}_1 = W^{fx}\mathbf{x}_j + \mathbf{b}^{f_1} \tag{4.11}$$

$$W(\mathbf{r}_i - \mathbf{r}_j) = \begin{cases} W^{fd}\hat{\mathbf{d}}_{\mathbf{ij}} + \mathbf{b}^{f_2} & \text{if } \mathbf{r}_i - \mathbf{r}_j > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

$$\tilde{\mathbf{v}}_i^{(t)} = W^{xf} \sum_{j=1}^{n_{\text{atoms}}} \mathbf{h}_1 \circ W(\mathbf{r}_i - \mathbf{r}_j) \tag{4.13}$$

which corresponds to the first three layers of the SchNet interaction block (i.e., atom-wise $\rightarrow$ cfconv $\rightarrow$ atoms-wise). Therefore, the factorized tensor layer of DTNNs can be interpreted as a generalized continuous-filter convolution with a non-linearity within the sum. On the other hand, the interaction block of SchNet is more general than the tensor layers of DTNN, since the filter function $W(\mathbf{r}_i - \mathbf{r}_j)$ can be freely chosen and another atom-wise layer has been added. Finally, placing the activation function outside the sum keeps the convolution linear, which will be important for defining periodic filters for materials in the next section.

4.3.2 Filter-generating networks

In the interaction blocks of SchNet, filter-generating networks have to model the interactions of feature maps depending on interatomic distances. Fig. 4.2 (right) shows the architecture of the filter-generating networks used in SchNet. The convolution and architecture of SchNet already guarantee invariance with respect to translation and atom indexing. Rotational invariances and properties have to be achieved by the design of the filter. In the following, we will therefore discuss the design choices of the filter-generating network under this aspect.

Self-interaction

In an interatomic potential, we aim to avoid self-interaction of atoms, as reflected in the many-body expansion:

$$E(S) = \sum_{i=1}^{n_{\text{atoms}}} E^{(1)}(\mathbf{r}_i) + \sum_{i<j}^{n_{\text{atoms}}} E^{(2)}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i<j<k}^{n_{\text{atoms}}} E^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots$$

DTNNs achieve this by restricting the sum to neighboring atoms $j \neq i$. An equivalent formulation of this is to define the filter-network such that $W(\mathbf{r}_i - \mathbf{r}_j) = 0$ for $\mathbf{r}_i = \mathbf{r}_j$ as we did in Eq. 4.12. Since there are never two atoms at the same position, this is a unambiguous condition to exclude self-interaction.

Rotational invariance

As the input to the filter $W(\mathbf{r}_i - \mathbf{r}_j) : \mathbb{R}^3 \to \mathbb{R}$ is already invariant to translations of the molecule, we only need to consider rotational invariance. Analogue to the DTNN, this can easily achieved here by using only the interatomic distances instead, resulting in a radial filter $W(\|\mathbf{r}_i - \mathbf{r}_j\|) : \mathbb{R} \to \mathbb{R}$. Here, we also use the radial basis of Gaussians (Eq. 3.4), we already employed in the DTNN. Beyond the reasoning given for DTNNs, the entries of filter tensors in discrete convolutional layers are initialized independently. However, if we initialize a neural network with the usual weight distributions and nonlinearities, the resulting function is almost linear as the neuron activations are close to zero. Therefore, the filter values would be strongly correlated, leading a plateauing cost function at the beginning of training. The radial basis functions alleviate this problem by decorrelating the various distance regimes.

Fig. 4.4 shows 10x10 Å cuts through all 64 radial, three-dimensional filters of each interaction block for a SchNet model trained on a molecular dynamics trajectory of ethanol. In contrast to DTNN, we do not tie the weights across interaction blocks, so the filters will change for each interaction.

Periodic boundary conditions

Bulk crystals are characterized by their periodic boundary conditions (PBCs), i.e. a unit cell repeats infinitely in space on a lattice. Therefore, periodic images of atoms have an identical chemical environment, and thus, should also have an identical representation. This is already guaranteed in the SchNet architecture, as we obtain the atom-wise representations from the chemical enviroments. Due to the linearity of the convolution, we can make this more efficient by moving the sum over periodic images into the filter. Considering that representations $\mathbf{x}_i = \mathbf{x}_{ia} = \mathbf{x}_{ib}$ are identical for repeated unit cells a and b,

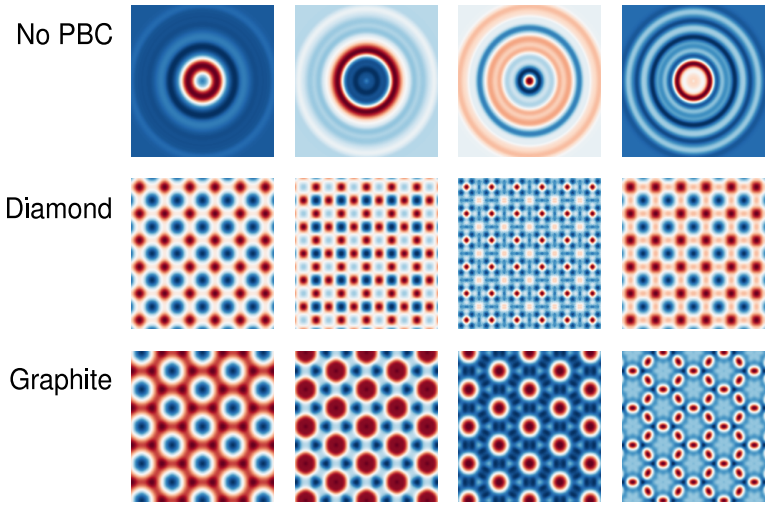


Figure 4.5: Dependence of convolutional filters on the employed periodic boundary conditions [Sch+18]. 5Å x 5Å cuts through generated filters from the same filter-generating networks (columns) under different periodic bounding conditions (rows). Each filter is learned from data and represents the effect of an interaction on a given feature of an atom representation located in the center of the filter.

we obtain

$$\begin{aligned}\mathbf{x}_i^{l+1} = \mathbf{x}_{im}^{l+1} &= \frac{1}{n_{\text{neighbors}}} \sum_{j=0}^{n_{\text{atoms}}} \sum_{b=0}^{n_{\text{cells}}} \mathbf{x}_{jb}^{l} \circ \tilde{W}^l(\mathbf{r}_{jb} - \mathbf{r}_{ia}) \\ &= \frac{1}{n_{\text{neighbors}}} \sum_{j=0}^{n_{\text{atoms}}} \mathbf{x}_j^l \circ \underbrace{\left(\sum_{b=0}^{n_{\text{cells}}} \tilde{W}^l(\mathbf{r}_{jb} - \mathbf{r}_{ia}) \right)}_{W} . \end{aligned} \tag{4.14}$$

Note that we average over neighbors in contrast to the filter for molecules since we potentially have a large number of neighbors in a periodic system. We compute the periodic filter before convolving only with the atomic representations of one unit cell. Since they only depend on the atom positions, all filters can be computed independently and potentially in parallel with the atomic representations.

Fig. 4.5 shows four filters under different periodic boundary conditions. While the filters without PBCs are radial, the filters with the PBCs of diamond and graphite are superpositions of radial filters on the respective lattice.

4.4 Results

In the following, we will evaluate SchNet for the prediction of various molecular properties across chemical compound space as well as formation energies of bulk crystals. We use SchNet models with up to $T = 6$ interaction refine-

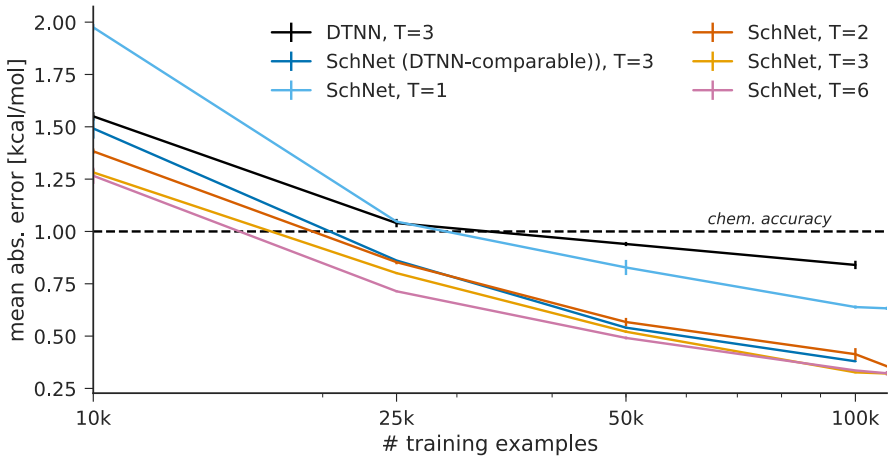


Figure 4.6: Learning curves for DTNN and SchNet models [Sch+18]. Mean absolute error in kcal $\text{mol}^{-1}$of energy predictions (U_0) on the QM9 dataset [Ram+14; BR09; Rey15] depending on the number of interaction blocks and reference calculations used for training are given. We give the best performing DTNN models as well as a SchNet model with comparable hyper-parameters, using 30 features and 60 filters.

ments and consistently use $n_{\text{feats}} = 64$ features to represent chemical environments. For a full specification of the network, see Fig. 4.2.

In each experiment, we split the data into a training set of the sizes given below and use a validation set for early stopping. All models are trained by minimizing the squared loss using the ADAM optimizer [KB15] with 32 examples per mini-batch with an initial learning rate of 10^{-3} and an exponential learning rate decay with ratio 0.96 per 100,000 steps. We train all models for up to 10M parameter update steps and select the one that performs best on the validation set. The remaining data is used for computing test errors. The reported errors are averages over three repetitions of random subsampling.

4.4.1 Molecular properties across chemical compound space

In Section 3.5.1, we have used deep tensor neural networks to predict energies for the QM9 benchmark dataset. Fig. 4.6 shows the performance of SchNet with $T \in \{1,2,3,6\}$ compared to the best-performing DTNN ($T = 3$). Just like in the DTNN model, we do not use a distance cutoff due to the relatively small molecules in QM9. We use 10,000 examples for validation on the QM9 benchmark, following Faber et al. [Fab+17] and Gilmer et al. [Gil+17]. We train a SchNet model with comparable settings to DTNN: we use 30-dimensional atom-wise representation and 60 convolutional filters wich correspond to the 60-dimensional factor space in the DTNN. SchNet drastically improves over DTNN in terms of mean absolute errors for all training set sizes. For training sets larger than 25k examples, the SchNet model with one interaction block even surpasses the DTNN with three interaction passes. This can be attributed

Table 4.1: Number of parameter updates until model with lowest validation error in early stopping. All models were trained for 10M iterations before the best models were selected. Lowest number of required updates in **bold**.

Training examples	T=1	T=2	T=3	T=6
10k	3.40M	1.77M	1.68M	**0.93M**
50k	5.72M	3.89M	4.55M	**2.87M**
100k	9.47M	7.09M	7.91M	**5.96M**

to the interaction blocks, in particular the filter-generating networks, that allow for a more flexible interaction potential.

Comparing the SchNet models with varying numbers of interaction blocks trained on 100k examples, we observe that more than two interaction blocks reduce the error only slightly from 0.35 kcal mol^{-1} with $T = 2$ interaction blocks to 0.32 kcal mol^{-1} for $T \in \{3, 6\}$. For smaller training sets, the differences become more apparent. Here, the model with six interaction blocks shows the lowest errors even though it has the most parameters. Additionally, the model requires much less parameter updates to converge as shown in Table 4.1. This indicates that the larger models can easier fit the interactions and might yield a more suitable representation for the learning problem. Therefore, we use SchNet models setting $F = 64$ and $T = 6$ in the following, if not specified otherwise.

Up until now, we have only predicted the property U_0 of the QM9 dataset, i.e. the total energy at 0K. A full description for all properties can be found in Appendix A. We have used the sum pooling of atomic contributions for all properties except for the intensive properties ϵ_{HOMO}, ϵ_{LUMO} and $\Delta\epsilon$, for which we have used mean pooling.

Table 4.2 shows mean absolute errors also for properties other than the energy for SchNet and the message-passing neural network enn-s2s [Gil+17]. Gilmer et al. [Gil+17] have proposed the notion of message-passing neural networks (MPNNs), under which they also categorize DTNN. They have developed the MPNN enn-s2s, which uses first-principles as well as information about structural chemical features such as bonds and aromatic rings. In contrast to DTNN and SchNet, the output network uses a set2set approach that results in a single representation for the molecule [VBK16].

SchNet outperforms enn-s2s for 8 of 12 properties and even achieves comparable performance with the ensemble for the properties U_0, U and G. However, SchNet can not reach the performance of the message passing neural networks for the dipole moment, polarizability and electronic spatial extent. We conjecture this is due to the strong dependence of these properties to the structure of the molecule such that they can not be as easily decomposed into atomic contributions as the energy. Here, the set2set readout function of the

Table 4.2: Mean absolute errors for energy predictions on the QM9 data set using 110k training examples [Sch+18]. We give error for SchNet, the message-passing neural network enn-s2s as well as an ensemble of enn-s2s models [Gil+17] in kcal mol^{-1}. For SchNet, we give the average over three repetitions as well as standard errors. Best single models in **bold**.

Property	Unit	SchNet ($T = 6$)	enn-s2s	enn-s2s-ens5
ϵ_{HOMO}	kcal mol^{-1}	**0.95 ± 0.02**	0.99	0.71
ϵ_{LUMO}	kcal mol^{-1}	**0.78 ± 0.00**	0.85	0.65
$\Delta\epsilon$	kcal mol^{-1}	**1.45 ± 0.00**	1.59	1.22
ZPVE	kcal mol^{-1}	0.039 ± 0.001	**0.035**	0.030
μ	Debye	0.033 ± 0.001	**0.030**	0.020
α	Bohr^3	0.235 ± 0.061	**0.092**	0.068
$\langle R^2 \rangle$	Bohr^2	**0.073 ± 0.002**	0.180	0.168
U_0	kcal mol^{-1}	**0.32 ± 0.02**	0.45	0.33
U	kcal mol^{-1}	**0.44 ± 0.14**	**0.45**	0.34
H	kcal mol^{-1}	**0.32 ± 0.02**	0.39	0.30
G	kcal mol^{-1}	**0.32 ± 0.00**	0.44	0.34
C_v	cal / molK	**0.033 ± 0.000**	0.040	0.031

enn-s2s has more expressive power as it produces a graph-level embedding which is then used to predict the property. Another possibility is to predict physically meaningful terms as a proxy, e.g. a latent charges $\hat{q}_i$ which is then used to calculate the dipole moment [GBM17]:

$$\boldsymbol{\mu} = \sum_{i=1}^{n_{\text{atoms}}} \hat{q}_i \mathbf{r}_i$$

Adding such property-specific output networks to SchNet is subject to future work.

4.4.2 Formation energies of bulk crystals

Beyond predicting molecular properties, we are able to use filters with periodic boundary conditions to predict properties of materials. We predict formation energies of equilibrium bulk crystals from the Material Project repository [Jai+13]. Further details about the dataset are listed in Appendix A. As detailed in Section 4.3.2, we obtain a filter with PBCs by summing over non-periodic filters for each periodic repetition and normalizing by the number of neighboring atoms within the chosen cutoff. Here, the choice of the cutoff is important since the number of neighbors rises fast with the cutoff. We have chosen to use a cutoff of 5Å which is a compromise of keeping computa-

Table 4.3: Mean absolute errors for formation energy predictions in eV/atom on the Materials Project data set [Sch+18]. For SchNet, we give the average error over three repetitions as well as standard errors of the mean. Best models in **bold**.

Model	$N = 3,000$	$N = 60,000$
ext. Coulomb matrix [Fab+15]	0.64	–
Ewald sum matrix [Fab+15]	0.49	–
sine matrix [Fab+15]	0.37	–
SchNet ($T = 6$)	**0.127 ± 0.001**	**0.035 ± 0.000**

tion time reasonably low while capturing the short-range interactions between atoms directly.

The Materials Project dataset is much more diverse than QM9 in terms of atom types but includes less than half the amount of training data. Table 4.3 shows mean absolute errors for the prediction of formation energies per atom by SchNet for 3,000 and 60,000 training examples. We use 1,000 and 4,500 additional examples for early stopping, respectively, corresponding to the Materials Project subsets. For the smaller training set, we list the performances of several descriptors for materials proposed by Faber et al. [Fab+15] that were used as features for kernel ridge regression. These descriptor are similar to the Coulomb matrix in that they consist of pairwise interaction terms organized in an adjacency matrix. They differ in how they specifically include the periodicity of the material. E.g., the sine matrix, which yields the lowest error out of the reference descriptors, is defined as

$$x_{ij} = \begin{cases} 0.5 Z_i^{2.4} & \text{for } i = j \\ Z_i Z_j \bar{\phi}(\mathbf{r}_i - \mathbf{r}_j) & \text{for } i \neq j \end{cases}$$

with the periodicity being included by

$$\bar{\phi}(\mathbf{r}_{ij}) = \left\| \mathbf{B} \cdot \sum_{k=1}^{3} \hat{e}_k \sin^2(\pi \hat{e}_k \mathbf{B}^{-1} \cdot \mathbf{r}ij) \right\|_2^{-1} .$$

SchNet significantly improves over the best hand-crafted features, reducing the mean absolute error from 0.37 eV/atom of the sine matrix to 0.13 eV/atom[1]. With the large data set of 60,000 examples, the error can be reduced even further to 0.035 eV/atom. Fig. 4.7 shows the distribution of errors of SchNet. While there are a considerable number of examples with high errors, most materials are predicted well. Less than 10% of the materials are predicted with absolute errors above 0.1eV.

[1] 1 kcal mol^{-1}≈ 0.043 eV

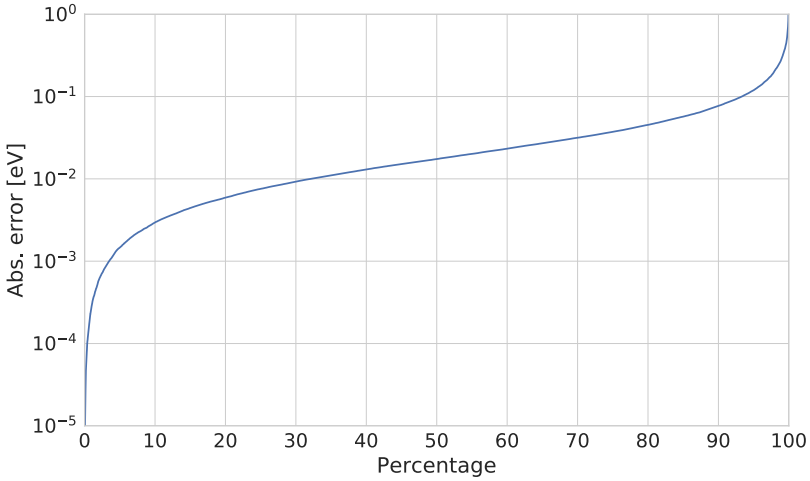


Figure 4.7: Distribution of absolute errors for the predictions of formation energies per atom for the Materials Project dataset. The plot shows the percentage of materials predicted with lower than given test errors. SchNet was trained with $T = 6$ interaction blocks on 50k training examples.

4.5 Analysis

We continue to analyze the obtained representations. Given the similarities between the two neural network architectures, we adopt some of the analysis methods from DTNNs and observe how the representation has changed. Additionally, we apply these methods to atomistic systems with periodic boundary conditions.

4.5.1 Energy contributions

In Section 3.6.1, we have discussed the non-uniqueness of energy partitioning in general and evaluated the energy contributions of chemical environments learned by DTNN models in particular. We concluded that DTNNs obtain a different energy partitioning in each training run, i.e., different representations that yield equivalent results in terms of prediction error. Here, we will examine whether this is also the case for SchNet.

Fig. 4.8 compares the energy partitioning of DTNN and SchNet models with $T \in \{3, 6\}$ interaction blocks. We show the distributions of atom-wise energy contributions of each architecture for three training runs on different training sets. The distributions of atomization energies agree across all repetitions and models. However, the atom-wise energy contributions vary significantly between model architectures. We observe that the distributions of SchNet show a narrower range of energy contributions than those of DTNN,

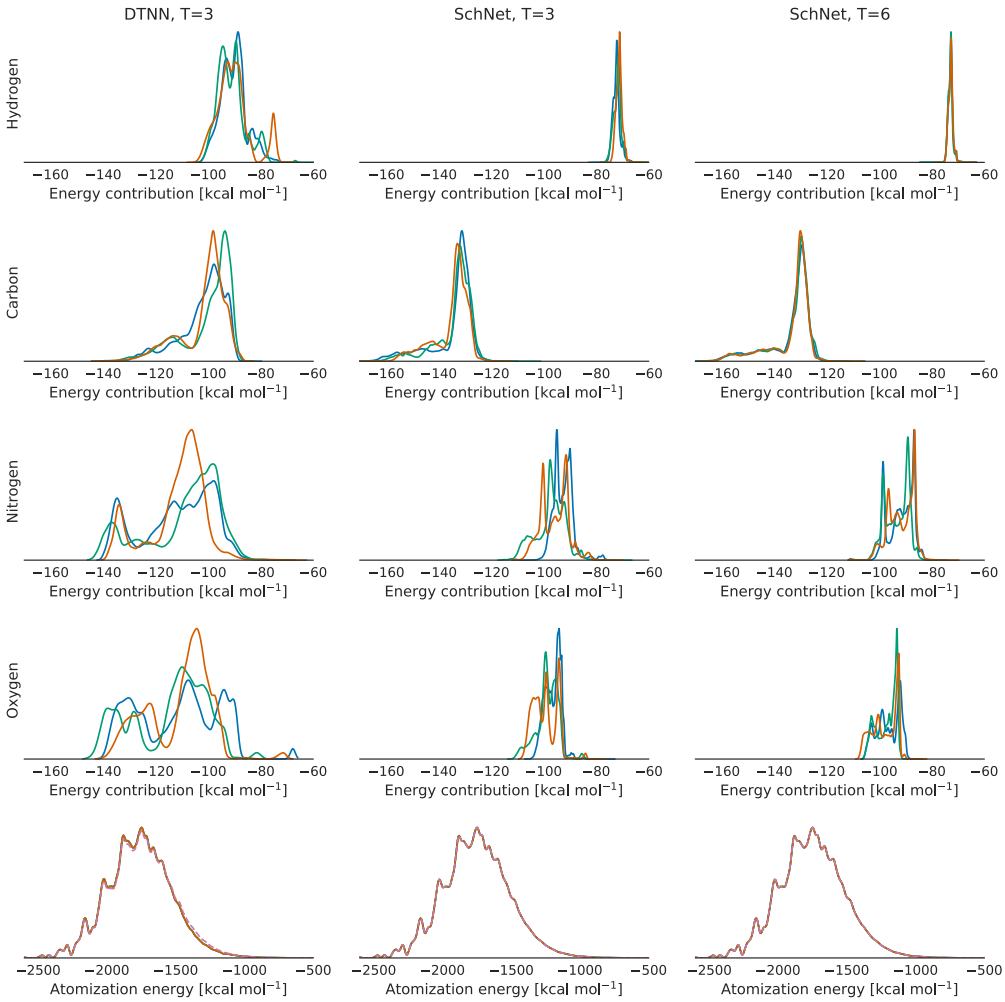


Figure 4.8: Distribution of energy contributions for atoms of types H, C, N, O and atomization energies from QM9 molecules predicted by DTNN and SchNet models. The models were trained on 100k examples. Each color corresponds to a model trained on a different subset. The distributions of atomization energy predictions agree across models (bottom).

especially for hydrogen which has reduced to a peak with a width of approximately 10 kcal mol^{-1}. While the DTNN energy contributions occur most often around -100 kcal mol^{-1}which is close to the mean energy per atom of -97.8 kcal mol^{-1}, SchNet exhibits distinct peaks in the distributions for hydrogen and carbon at about -75 kcal $\text{mol}^{-1}$and -130 kcal mol^{-1}, respectively. Complementary figures in Appendix B.1 show the convergence in greater detail for the energy contributions of pairs of equivalent models from Fig 4.8, plotted against each other in scatter plots.

Most importantly, the distributions seem to converge from DTNN over SchNet with three interaction blocks to SchNet with six interaction blocks towards a unique solution. This is especially noticeable for carbon and hydrogen, where the obtained energy partitionings for SchNet ($T = 6$) are qualitatively equivalent. While a convergence for the distributions of oxygen and nitrogen can be observed, too, they are still more diverse across training runs than those of hydrogen and carbon. A likely reason is the lower number of these atom types in the training data.

We conclude that the models attempt to solve the learning problem while minimizing the deviation of the interaction energy within atom types to obtain a simple solution. This is most successful with the SchNet (T=6) model with the sharpest peaks in the distribution, i.e., learning characteristic energies for atom types. This conclusion also agrees with Table 4.1, where we have shown that more interaction blocks lead to less required parameter updates in early stopping to obtain the best model.

4.5.2 Local chemical potentials

In Section 3.6.2, we have defined local chemical potentials for the DTNN by using a virtual probe atom as a test charge. To achieve this, we can pass the probe into the network like any atom of the molecule and only have to consider how to handle the continuous-filter convolutions. This can be derived straight-forward from the definition of the continuous-filter convolutional layer in Eq. 4.4, which is defined for arbitrary positions in space. Thus, the continuous-filter convolution for a probe atom can be calculated as

$$\mathbf{x}_{\text{probe}} = (\rho^l * W)(\mathbf{r}_{\text{probe}}) = \sum_{j=1}^{n_{\text{atoms}}} \mathbf{x}_j^l \circ W(\mathbf{r}_{\text{probe}} - \mathbf{r}_j) \tag{4.15}$$

All other layers are applied atom-wise and, thus, can be applied to the probe atom unchanged.

Fig. 4.9 shows local chemical potentials for bulk crystals of SchNet models with six interaction blocks trained on the Materials Project dataset. We show cuts through the potentials of graphite and diamond using a carbon test charge. They reflect the symmetry and periodicity of each system. Fig. 4.10 shows a comparison of the local chemical potentials of SchNet with those of

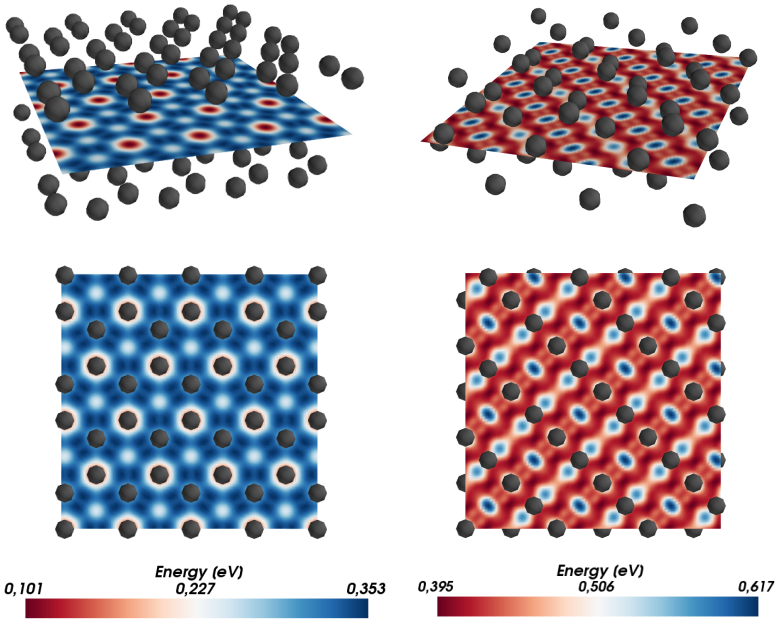


Figure 4.9: Cuts through local chemical potentials $\Omega_C(\mathbf{r})$ of SchNet The analyzed SchNet (T=6) model was trained on the Materials Project dataset. Local potentials using a carbon test charge are shown for graphite (left) and diamond (right).

the DTNN. For both architectures, we use a carbon atom to probe the generated potential and plot it on an isosurface with constant $\sum_i \|\mathbf{r} - \mathbf{r}_i\|^{-2} = 3.7\text{Å}$. The general structure of the local chemical potentials of both models is similar. In particular, both models reflect symmetries of the molecules in the potential.

The energy range of the local chemical potential $\Omega_C(\mathbf{r})$ of SchNet is compressed for all molecules which corresponds to what we have observed for the energy contributions in the last section. Moreover, the low- and high-energy regions are separated more clearly in the SchNet model, indicating a more localized representation. Again, this agrees with the results on atom-wise energy contributions in Fig. 4.8, where the distributions converge for the SchNet models to a simpler model with minimal deviation. One way for the model to achieve this is to localize the interaction refinements, which we will examine in the next section.

4.5.3 Interaction analysis

Both the energy contributions as well as the local chemical potentials suggest that SchNet achieves more accurate prediction by learning more local models than the DTNN. To test this hypothesis, we study the interaction corrections of SchNet and DTNN. Recall that the atom-wise representations $\mathbf{x}$ are modified multiple times by additive corrections in both architectures. This leads to a

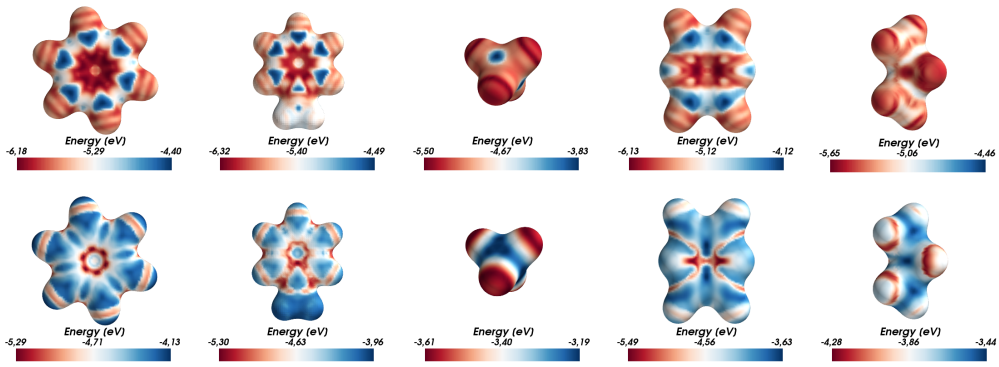


Figure 4.10: Local chemical potentials $\Omega_C(\mathbf{r})$ of DTNN (top) and SchNet (bottom) [Sch+18]. Potentials using a carbon probe on a $\sum_i \|\mathbf{r} - \mathbf{r}_i\|^{-2} = 3.7\text{Å}^{-2}$ isosurface are shown for benzene, toluene, methane, pyrazine and propane.

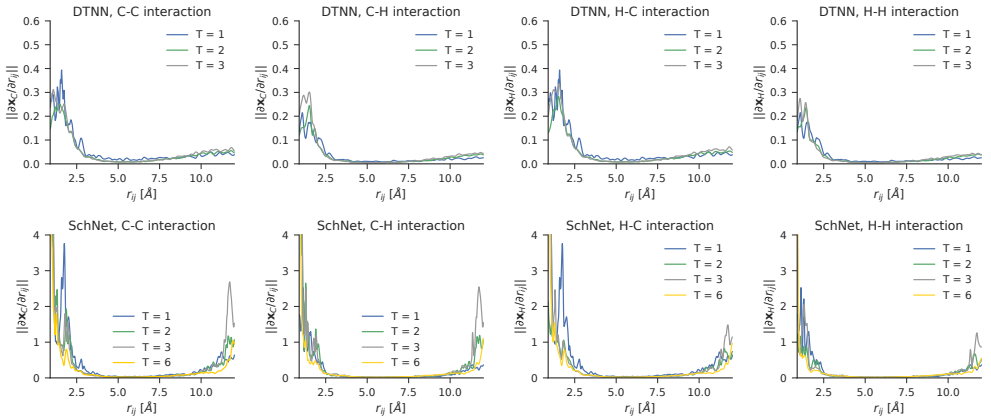


Figure 4.11: Change of the representation during bond breaking. We increase the distance between two atoms while observing the sensitivity of the representations for carbon and hydrogen atoms. The analyzed DTNN and SchNet models use T interactions as given in the legend. All models were trained on 100k training examples of QM9.

final representation

$$\mathbf{x}_i^{(T)} = \mathbf{x}^0 + \sum_{t=1}^{T} \mathbf{v}^{(t)}.$$

If our model is local, we expect this representation to converge while moving two atoms apart from each other. In this case, the sensitivity of the representation to atom movement

$$\frac{\partial \mathbf{x}_i^{(T)}}{\partial r_{ij}}$$

approaches zero. The faster this happens, the more local we consider our representation. Note, that locality can be enforced by choosing a small distance cutoff. However, in our molecule models, we set the cutoff such that all occurring distances are covered. Note, that the representations of different models vary on different scales due to differences in the architecture of the model and

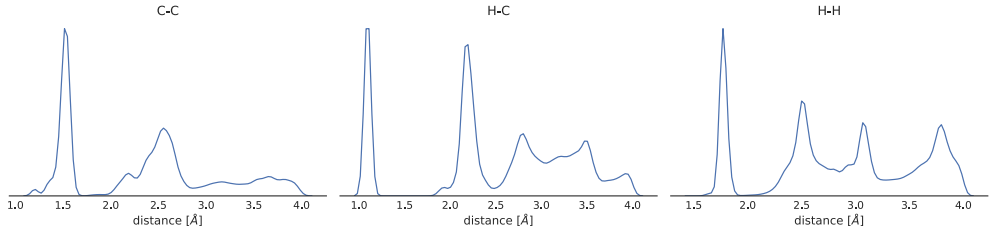


Figure 4.12: Pairwise distributions of carbon and hydrogen in QM9 up to 4.0Å.

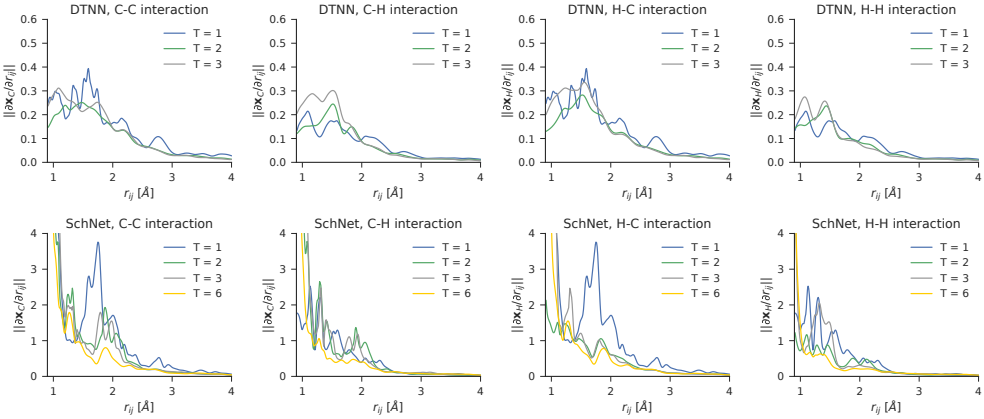


Figure 4.13: Change of the representation during bond breaking for $r_{ij} < 4.0$Å. We increase the distance between two atoms while observing the sensitivity of the representations for carbon and hydrogen atoms. The analyzed DTNN and SchNet models use T interactions as given in the legend. All models were trained on 100k training examples of QM9.

dimensionality of the representation. Therefore, we have to evaluate our locality measure separately for each model, i.e., how it develops with respect to the pair-wise atom distance r_{ij}.

Fig. 4.11 shows how our locality measure behaves for moving carbon and hydrogen atoms apart from each other in various combinations (C-C, H-H, C-H)). The models were trained on the QM9 dataset which covers distance up to about 12.0 Å. For both models, we see large changes for nearby atoms up to distances of about 4.0 Å across all interaction types. This agrees with chemical intuition since this corresponds to the distance regime relevant for chemical bonds and short-range non-bonded interactions. Another distance regime the representation is sensitive to is beyond 10.0 Å. This can be explained by the lack of data in that region, since only a few larger molecules cover this regime. Therefore, this region is either used to identify large molecules or is noisy due to the lack of training data.

Fig. 4.12 shows the pairwise distribution of carbon and hydrogen atoms. We can recognize the bonds as the first peaks in the carbon-carbon and hydrogen-carbon pairs at approximately 1.5Å and 1.1Å, respectively. As there are no

hydrogen-hydrogen bonds in the data set, the first peak in the H-H plot corresponds to hydrogens that are bonded with the same carbon atom. Similarly, the H-C and C-C show peaks for these kinds of interactions at 2.0-3.0Å, which indicates bonding with common neighbors.

We can use this as a reference to identify some learned interactions in Fig. 4.13 which shows the sensitivity profiles for distances up to 4.0Å. Comparing the sensitivity profiles of DTNN and SchNet, we observe that SchNet puts more emphasis on the < 1.5Å regime than the DTNN models. It follows that SchNet is more sensitive to the bonds while DTNN incorporates more non-bonded interactions. Based on these observations, we have trained a SchNet ($T = 6$) model using a distance cutoff of 4Å on 110,000 examples from QM9. As expected from our analysis, the obtained accuracy is equivalent to that of the model including all distance (MAE 0.32 kcal mol^{-1}).

Within the SchNet models, the model with $T = 6$ interaction blocks decreases faster and smoother than those with less interaction blocks. In contrast, SchNet (T=1) has a large spike at 1.5-2.0 Å. Since this model is only able to incorporate pair-wise interactions to construct the representation, it needs to make use of non-bonded interactions when attempting to uniquely represent a molecule. On the other hand, more interaction blocks enable SchNet to decompose the geometry into complex interactions of more localized chemical environments. This also serves as a plausible explanation as to why SchNet is able to generalize better and learn faster with more interaction blocks. Note that this does not necessarily restrict SchNet to a fully local representation: indirect interactions over multiple neighbors can still play a role for molecules with more than two atoms.

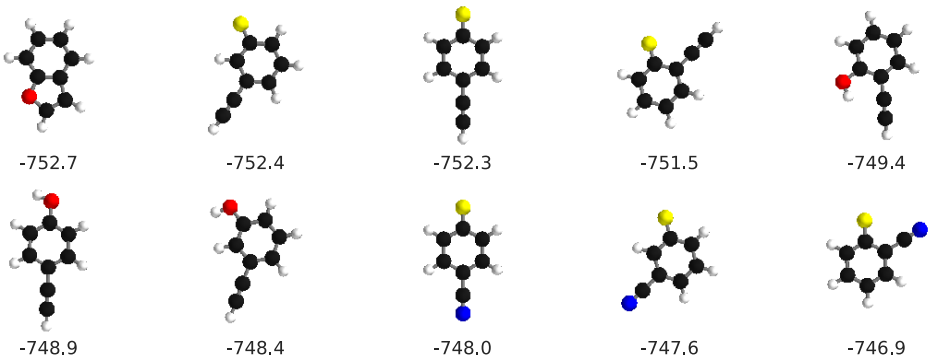
4.5.4 Ranking of molecular carbon ring stability

In the previous sections, we have established that SchNet learns a local representation yielding an energy partitioning that is largely consistent across retrained models. This also allows us to assign atomization energy contributions to substructures of molecules, which can be interpreted as a measure of local stability. A particular interesting substructure in this regard are aromatic rings,

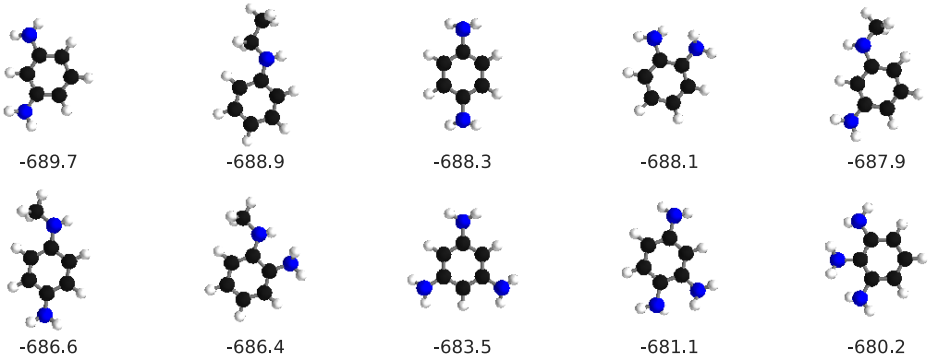
$$E_{\text{ring}} = \sum_{i \in \text{ring}(S)} E_i \tag{4.16}$$

where the set $\text{ring}(S)$ contains the atoms that belong to a particular ring of molecule S.

Fig 4.14 shows the 10 molecules with most and least stable 6-membered carbon rings yielded by SchNet ($T = 6$). We observe that nitrogen atoms that are directly connected to the ring increase the ring energy such that the 10 least stable rings are all bonded with nitrogen. Beyond that, we observe that nitrogen and fluorine atoms that are close to each other or other carbon



(a) Top-10 most stable 6-membered carbon rings from most to least stable.



(b) Top-10 least stable 6-membered carbon rings from most to least stable.

Figure 4.14: Ranking of molecular carbon ring stability. We show molecules with the highest and lowest energy contributions from 6-membered carbon rings. The ring energies were calculated with SchNet ($T = 6$) trained on 110k molecules of QM9.

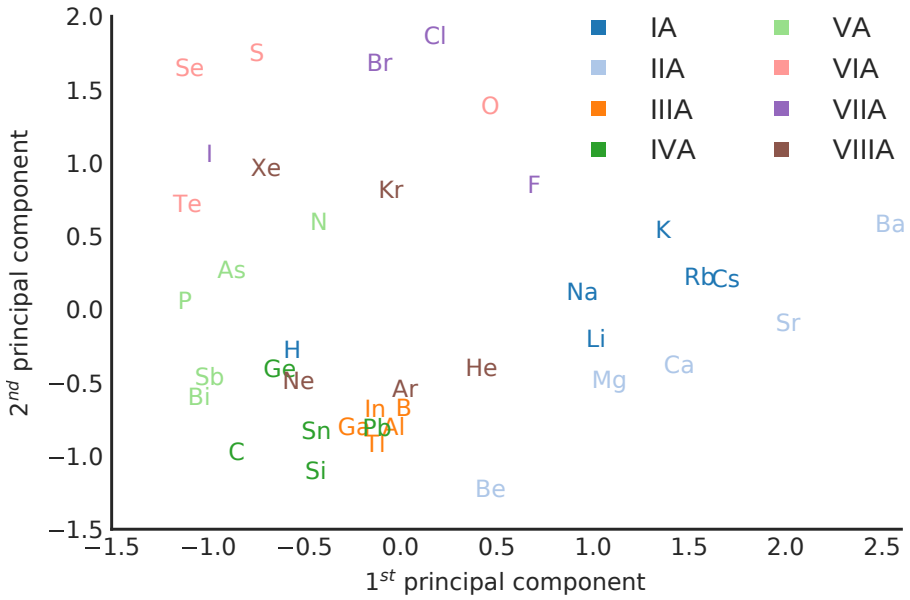


Figure 4.15: Two leading principal components of the learned embeddings $\mathbf{x}^0$ of sp atoms learned by SchNet from the Materials Project dataset [Sch+18]. We recognize a structure in the embedding space according to the groups of the periodic table (color-coded) as well as an ordering from lighter to heavier elements within the groups, e.g., in groups IA and IIA from light atoms (left) to heavier atoms (right).

atoms connected to the ring reduce its relative stability. E.g. in Fig. 4.14a the 4th molecule only differs from the 2nd and 3rd most stable molecules by having a fluorine atom connecting to the ring next to the carbon chain, or in Fig. 4.14b the least stable molecules differ only in the distances between connected nitrogens. A full stability ranking of the 6-membered carbon rings in QM9 is listed in Appendix B.2.

4.5.5 Atom type embeddings

Having extensively analyzed what SchNet has learned about atom interactions, we go on to take a look at how atom types are represented. In Chapter 2, we have introduced cross-element generalization as a desirable property of representations of atomistic systems. While most descriptors consider different atom types orthogonal, DTNN and allow for cross-element generalization through the initial embeddings $\mathbf{x}_i^{(0)}$. If the trained models learn to efficiently make use of this possibility, we should be able to extract atom similarities from the embeddings that resemble chemical intuition. Since QM9 only contains five atom types (H, C, N, O, F), we will perform this analysis on the Materials Project dataset as it includes 89 atom types ranging across the periodic table.

Fig. 4.15 shows the two leading principal components of the atom type embeddings of sp-atoms, i.e. the main group elements of the periodic table.

The projection explains only about 20% of the variance, therefore atom types might appear closer than they are in the high-dimensional space. However, we see that atoms belonging to the same group tend to form clusters. This is especially apparent for main groups 1-5, while groups 6-8 appear to be slightly more scattered. In group 1, hydrogen lies further apart from the other members which coincides with its special status, being the element without core electrons. Beyond that, there are partial orderings of elements according to their period within some of the groups. There are orderings from light to heavier elements, e.g. in group 1 (left to right: H - [Na,Li] - [K, Rb, Cs]), group 2 (left to right: Be - Mg - Ca - Sr - Ba) and group 5 (top to bottom: N-[As, P]-[Sb,Bi]).

Note that these extracted chemical insights were not imposed by the SchNet architecture onto the embeddings as they were initialized randomly before training. It had to be inferred by the model based on the co-occurrence of atoms in the bulk systems of the training data.

4.6 Summary and discussion

In this chapter, we have developed SchNet which is constructed using the same principles as deep tensor neural networks. The crucial change is how SchNet models quantum interactions. To that end, we have proposed continuous-filter convolutional layers for non-uniformly sampled data. We use them in combination with filter-generating networks [Jia+16] to obtain smooth convolutional filters that model the interactions between atoms. Most importantly, we have incorporated periodic boundary conditions into the filter, making efficient predictions for materials possible.

SchNet is able to reduce the mean absolute error to 0.32 kcal $\text{mol}^{-1}$for the prediction of atomization energies at 0K of the QM9 benchmark dataset. Beyond that, we have applied SchNet successfully to the accurate prediction of other chemical properties from QM9 as well as formation energies of bulk crystals. We have identified problems with the prediction of dipole moments and polarizabilities due to their strong dependence on the global spatial structure of the molecule. An extension of SchNet with output networks for the dipole moment [GBM17], polarizability tensor and further properties is subject to future research.

We have continued with the analysis of the obtained representations in comparison with those yielded by DTNNs. The results have shown evidence that SchNet learns representations that agree with chemical intuition. While DTNN models obtain wildly different energy partitioning, the distribution of energy contributions in SchNet stabilizes and characteristic energies of atom types are found. These results indicate that SchNet is able to make better use of the training data, in particular for hydrogen and carbon, such that a

partitioning, which requires smaller perturbations of atomic energy contributions, can be found. This finding agrees with the visualization of the local chemical potentials of exemplary molecules as well as the sensitivity analysis of the atom-wise representations with respect to the distance between two atoms during bond breaking. In particular, we have found that both DTNN and SchNet learn that the distance regime below 4Å is most important for the prediction of molecular energies. However, SchNet focuses even more on the regime of bonds and the first sphere of non-bonded interactions up to 2Å. In conclusion, SchNet learns a less complex, more localized representation which helps to drastically improve prediction accuracy.

Chapter 5

Potential energy surfaces

We have spent a large part of this thesis evaluating and analyzing our developed neural network architectures using benchmark datasets such as QM9 [Ram+14] and the Materials Project [Jai+13]. Those datasets have been created by performing density functional theory computations [HK64] of candidate molecules and crystals, in order to relax them into equilibrium structures. Subsequently, we have predicted properties, in particular atomization and formation energies, of these structures. While this has been a good benchmark to test our architectures, it is not a realistic setting, since we have no way to obtain these structures without calculating the energies first.

In order to solve this, we either need machine learning algorithms that do not require the exact atom positions, or extend the training domain of our models to include non-equilibrium geometries. The first possibility is chosen by a lot of virtual screening methods that operate on molecular graphs [Ram+15; Duv+15; WDA16; Góm+16] or approaches that learn from approximate equilibrium geometries obtained by less accurate methods, e.g. semi-empirical force fields [Bro+83; GB87; Cor+95]. We choose the second possibility of learning an interatomic potential that is applicable for chemical and configurational degrees of freedom. Thus, we will perform several intermediate steps towards such a general model in this chapter.

Beyond the prediction of energies, we need to accurately predict atomic forces. Therefore, we will first describe a common model for energies and forces and how to incorporate forces into the training of the network. In Chapter 3, we have already used DTNN to predict energies of molecular dynamics trajectories. We will extend these experiments by predicting both energies and forces using single-trajectory SchNet models. As an application, we will perform a molecular dynamics simulation with a SchNet model for the fullerene C_{20}. Finally, we will train a model with chemical and configurational degrees of freedom for a set of $C_7O_2H_{10}$ isomers.

5.1 Training with energies and forces

The atomic forces can be obtained from the potential energy E of the atomistic system as the gradient

$$\mathbf{F}(\mathbf{r}) = -\frac{\partial E(\mathbf{r})}{\partial \mathbf{r}}. \tag{5.1}$$

Using this knowledge allows us to constrain the possible solutions for the force model. Chmiela et al. [Chm+17] proposed such a procedure for kernel learning called *gradient-domain machine learning (GDML)*. This method differentiates a kernel model for energies w.r.t the atom positions to obtain a force model

$$\hat{\mathbf{F}}(\mathbf{r}) = \sum_{i=1}^{n_{\text{train}}} \sum_{j=1}^{3n_{\text{atoms}}} (\boldsymbol{\alpha}_i)_j \frac{\partial}{\partial(\mathbf{r})_j} \nabla\kappa(\mathbf{r}, \mathbf{r}_i) \tag{5.2}$$

with a vector valued kernel $\nabla\kappa(\mathbf{x}, \mathbf{x}_i)$ and the parameter vector $\boldsymbol{\alpha}_i$ corresponding to training example i. Chmiela et al. [Chm+17] use a Matérn kernel over Coulomb matrices for the energy kernel κ. In the case of neural networks, this can be achieved by directly defining the force model as the derivative of the energy model $\hat{E}$ analog to Eq. 5.1. This is can be obtained easily by performing a full backward pass to the input layer.

Chmiela et al. [Chm+17] have shown that this procedure drastically improves the predictions, even if only forces and no energies are used for training. This is because the force field $\hat{\mathbf{F}}$ is constrained to be conservative, i.e. it is guaranteed to have the scalar potential $\hat{E}(\mathbf{r})$. In physical terms, this means that the force field is energy conserving, i.e. the energy difference

$$\Delta E = -\int_S \hat{\mathbf{F}}(\mathbf{r}) \cdot d\mathbf{r} \tag{5.3}$$

is independent of the choice of path S from $\mathbf{r}_1$ to $\mathbf{r}_2$. This is bound to be the case since $\mathbf{F}$ is integrable (see Eq. 5.1), such that

$$\Delta E = \hat{E}(\mathbf{r}_2) - \hat{E}(\mathbf{r}_1). \tag{5.4}$$

Since SchNet uses filter-generating networks that produce radial filters, the energy prediction is rotationally invariant, i.e.,

$$\hat{E}(\mathbf{r}) = \hat{E}(R\mathbf{r}), \tag{5.5}$$

where $R \in \mathbb{R}^{3\times3}$ is a rotation matrix, the derived force model is rotationally equivariant:

$$\begin{aligned}
\hat{\mathbf{F}}(R\mathbf{r}) &= -\frac{\partial \hat{E}(R\mathbf{r})}{\partial R\mathbf{r}} \overset{RR^T=I}{=} -RR^T \frac{\partial \hat{E}(R\mathbf{r})}{\partial R\mathbf{r}} \\
&= -R\frac{\partial R\mathbf{r}}{\partial \mathbf{r}} \frac{\partial \hat{E}(R\mathbf{r})}{\partial R\mathbf{r}} \overset{\hat{E}(\mathbf{r})=\hat{E}(R\mathbf{r})}{=} -R\frac{\partial \hat{E}(\mathbf{r})}{\partial \mathbf{r}} = R\hat{\mathbf{F}}(\mathbf{r}).
\end{aligned} \tag{5.6}$$

That is, a rotation of the molecule results in an equivalent rotation of the predicted force by design.

Until now, we have only shown how to derive a force prediction from an energy model. In order to also use known force targets during training, we have to modify our loss function. We use a combined loss of energies and forces inspired by Pukrittayakamee et al. [Puk+09]:

$$\ell((\hat{E}, \hat{\mathbf{F}}_1, \dots, \hat{\mathbf{F}}_{n_{\text{atoms}}})), (E, \mathbf{F}_1, \dots, \mathbf{F}_{n_{\text{atoms}}})) = \rho \quad \|E - \hat{E}\|^2 + \frac{1}{n_{\text{atoms}}} \sum_{i=0}^{n_{\text{atoms}}} \left\| \mathbf{F}_i - \left(-\frac{\partial \hat{E}}{\partial \mathbf{R}_i} \right) \right\|^2 \quad (5.7)$$

where ρ is a trade-off between energy and force loss.

It is crucial for the model to have at least 2nd order of continuity since we require second derivatives for the gradient descent of the force loss. Therefore, we made sure in the definition of the filter-generating networks and the choice of activation function that our model has infinite order of continuity. In SchNet, this is achieved by Gaussians in the basis expansion of distances and shifted softplus activation functions.

5.2 Prediction of total energies and atomic forces

In the following, we will apply SchNet to the prediction of potential energy surfaces and corresponding energy-conserving force-fields. We will first perform this on MD trajectories of single molecules and then go on to train a combined model for multiple trajectories of various isomers. All models have been trained using mini-batch stochastic gradient descent using the ADAM optimizer [KB15] with a batch size of 32 training examples.

5.2.1 Single-trajectory predictions

In Chapter 3, we have already demonstrated that a DTNN is able to represent configurational degrees of freedom for small molecules and predict the corresponding energies. Here, we will extend this to force predictions and all molecules from the MD17 dataset [Chm+17]. Beyond training SchNet on 50k reference calculation, we will study predictions on trajectories with a small subset of 1k training examples. This setting is especially relevant when we aim to predict more accurate and therefore more computationally expensive quantum calculations, e.g. using coupled-cluster theory [BM07], in future work.

Learning from few data points is a challenging task for SchNet since the representation has to be learned, in contrast to GDML or other methods with a fixed descriptor. Therefore, end-to-end learning usually requires more training

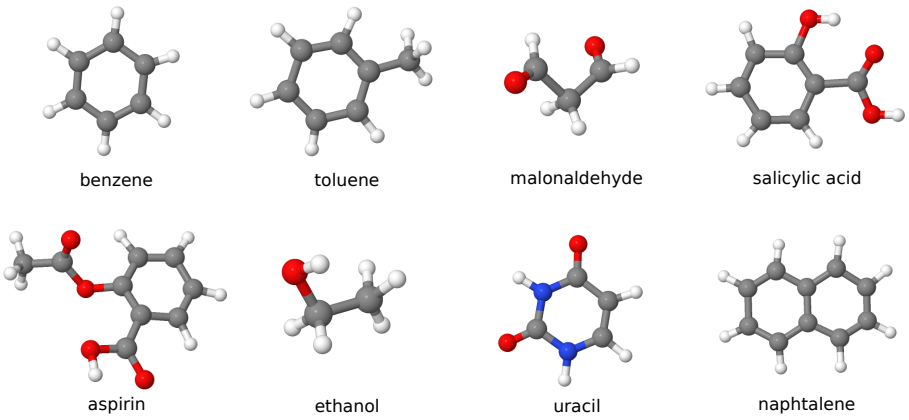


Figure 5.1: Illustrations of the molecules in the MD17 collection of molecular dynamics trajectories.

data. Beyond that, the SchNet architecture is build to learn general atomistic systems, while GDML is designed for single-trajectory data. While the ability to learn arbitrary chemical environments is an advantage for diverse data sets, it makes learning from similar configuration from MD trajectories harder. The main difference is that GDML uniquely identifies each atom while SchNet has to recognize them by their neighboring atoms.

We use ethanol and benzene as two representative molecules for model selection as they represent different aspects of the MD17 collection (see Fig. 5.1). While ethanol is small and flexible with a rotating O-H group, benzene consists of a 6-membered aromatic carbon ring which is quite stable up to the fast movements of the hydrogens. We have trained SchNet model with $T \in \{1, 2, 3\}$ interaction blocks. In all models, we set the energy-force trade-off $\rho = 0.01$, since we have empirically found this setting to be a good compromise between energy and force accuracy. A more detailed discussion of the trade-off between energy and force prediction will follow in Section 5.3. We give mean absolute errors and root mean squared errors for the prediction of energies and atomic forces. The force errors are given component-wise, i.e.,

$$\ell_{\text{force}}(\mathbf{F_i}, \hat{\mathbf{F}}_i) = \frac{1}{n_{\text{atoms}}} \sum_{i=1}^{n_{\text{atoms}}} \left\| \mathbf{F}_i - \hat{\mathbf{F}}_i \right\|,$$

analog to the force term in the training loss.

Tables 5.1 and 5.2 show the performance of SchNet trained on subsets of ethanol and benzene MD trajectories, respectively, from the MD17 data collection. When using 1k reference calculations for training, we observe that the models with two and three interaction blocks perform similarly well in terms of energy and force errors, but significantly better than SchNet with $T = 1$. For the large training sets, SchNet with $T = 3$ performs slightly better in terms of force errors then with less interaction blocks. The models with $T = 6$ interactions blocks perform similar to $T = 3$ on the large dataset, however, tend to overfit on the subset with 1,000 training examples.

Table 5.1: Test errors of SchNet trained on ethanol trajectory with $T \in \{1, 2, 3\}$. interaction blocks. We evaluate the effect of shared and unshared filter-generating networks on ethanol and benzene data sets trained on energies and forces using 50k examples.

Ethanol			*Energy [kcal/mol]*		*Force [kcal/mol/Å]*	
		T	*MAE*	*RMSE*	*MAE*	*RMSE*
N = 1,000		1	0.43	0.57	1.72	2.52
		2	**0.08**	0.13	0.40	0.70
		3	**0.08**	0.14	**0.39**	0.72
		6	0.09	0.14	0.42	0.69
	shared	2	**0.08**	0.16	0.43	0.82
		3	**0.08**	**0.12**	**0.39**	**0.68**
		6	**0.08**	0.13	0.40	0.69
N = 50,000		1	0.34	0.45	1.34	1.94
		2	**0.05**	**0.06**	0.07	0.11
		3	**0.05**	**0.06**	**0.05**	**0.08**
		6	**0.05**	**0.06**	**0.05**	**0.08**
	shared	2	**0.05**	**0.06**	0.09	0.14
		3	**0.05**	**0.06**	0.08	0.13
		6	**0.05**	**0.06**	0.10	0.15

Next, we study the effect of sharing the same filter-generating network across all interaction blocks. Similarly to the shared interaction functions in the DTNN architecture, this reduces the number of model parameters which might improve generalization, in particular on the small training set. In contrast to DTNN, this still allows for varying emphasis on different distance regimes throughout the network, since we do *not* share the full interaction blocks. While we observe minor improvement in energies and forces for the ethanol trajectory using 1,000 training examples, there are no improvements for benzene and the large training set. For the datasets with 50,000 training examples, sharing the filters deteriorates the force predictions, in particular for models with $T = 6$. The error is even higher than for the smaller training set, which indicates that sharing convolutional filters leads to a too constraint model which can get stuck in a local minimum.

Overall, the results are reasonably robust to the choice of number and sharing of interaction blocks for $T \geq 2$. The second interaction block is crucial for SchNet to be able to incorporate important angle information in the atom-wise representations. Based on this, we use SchNet with three interaction blocks and separate filter-generating networks in the following. Additional results with $T = 6$ are listed in Appendix B.3.

Table 5.2: Test errors of SchNet trained on benzene trajectory with $T \in \{1, 2, 3\}$. interaction blocks. We evaluate the effect of shared and unshared filter-generating networks on ethanol and benzene data sets trained on energies and forces using 50k examples.

Benzene			*Energy [kcal/mol]*		*Force [kcal/mol/Å]*	
		T	*MAE*	*RMSE*	*MAE*	*RMSE*
N = 1,000		1	**0.08**	0.11	0.35	0.66
		2	**0.08**	**0.10**	**0.30**	**0.46**
		3	**0.08**	**0.10**	0.31	0.47
		6	0.20	0.22	0.37	0.53
	shared	2	**0.08**	**0.10**	0.31	0.52
		3	**0.08**	**0.10**	0.31	0.47
		6	0.09	0.11	0.33	0.51
N = 50,000		1	0.08	0.10	0.31	0.48
		2	**0.07**	**0.09**	0.20	0.30
		3	**0.07**	**0.09**	**0.17**	**0.27**
		6	0.08	0.10	0.18	0.28
	shared	2	0.08	0.10	0.23	0.35
		3	**0.07**	**0.09**	0.18	**0.27**
		6	**0.07**	**0.09**	0.61	0.84

Table 5.3 shows the performance of SchNet in terms of energy prediction on all eight MD trajectories of the MD17 collection. We have trained SchNet only on energies as well as using the combined loss with energy-force trade-off $\rho = 0.01$. In the experimental setting using 1,000 training examples, we compare to GDML models trained on atomic forces [Chm+17]. Note that the training examples for the GDML models were sampled uniformly according to the energy distribution of the corresponding MD trajectory while SchNet was trained on randomly sampled training data. We observe that the energy predictions of SchNet greatly benefit from the added gradient information of the atomic forces. The mean absolute errors can be reduced by more than one order of magnitude consistently. The predictions of atomic forces in Table 5.4 show a similar picture. The improvement by using forces in training is even more drastic here with improvements of 1-2 orders of magnitude.

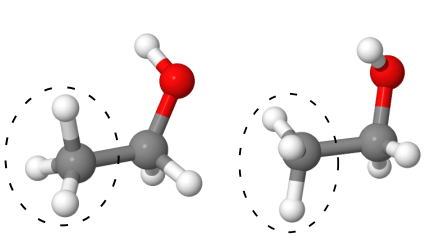
SchNet improves over GDML in terms of energy and force predictions for two out of eight molecules: malonaldehyde and ethanol. Fig. 5.1 shows that these are the two molecules in MD17 that do not include aromatic rings. These molecules have many symmetries such as the rotating hydroxyl and methyl groups in ethanol and aldehyde groups in malonaldehyde (see Fig. 5.2). SchNet, being implicitly invariant to atom indexing, can make use of these and, thereby, represent the molecules in a smaller feature space. On the other

Table 5.3: Mean absolute errors for total energies of MD17 trajectories in kcal/mol. GDML [Chm+17] and SchNet (T=3) [Sch+17b] test errors for N=1,000 and N=50,000 reference calculations of molecular dynamics simulations of small, organic molecules are shown. Best results are given in **bold**.

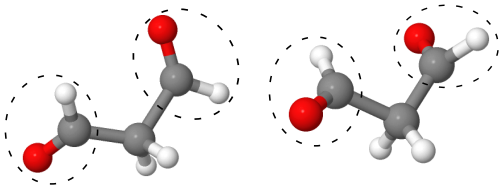
	N = 1,000			N = 50,000	
	GDML	**SchNet**		**SchNet**	
trained on	*forces*	*energy*	*energy+forces*	*energy*	*energy+forces*
Benzene	**0.07**	1.19	0.08	0.08	0.07
Toluene	**0.12**	2.95	**0.12**	0.16	**0.09**
Malonaldehyde	0.16	2.03	**0.13**	0.13	**0.08**
Salicylic acid	**0.12**	3.27	0.20	0.25	**0.10**
Aspirin	**0.27**	4.20	0.37	0.25	**0.12**
Ethanol	0.15	0.93	**0.08**	0.07	**0.05**
Uracil	**0.11**	2.26	0.14	0.13	**0.10**
Naphtalene	**0.12**	3.58	0.16	0.20	**0.11**

Table 5.4: Mean absolute errors for atomic forces of MD17 trajectories in kcal/mol/Å. GDML [Chm+17] and SchNet (T=3) [Sch+17b] test errors for N=1,000 and N=50,000 reference calculations of molecular dynamics simulations of small, organic molecules are shown. Best results are given in **bold**.

	N = 1,000			N = 50,000	
	GDML	**SchNet**		**SchNet**	
trained on	*forces*	*energy*	*energy+forces*	*energy*	*energy+forces*
Benzene	**0.23**	14.12	0.31	1.23	**0.17**
Toluene	**0.24**	22.31	0.57	1.79	**0.09**
Malonaldehyde	0.80	20.41	**0.66**	1.51	**0.08**
Salicylic acid	**0.28**	23.21	0.85	3.72	**0.19**
Aspirin	**0.99**	23.54	1.35	7.36	**0.33**
Ethanol	0.79	6.56	**0.39**	0.76	**0.05**
Uracil	**0.24**	20.08	0.56	3.28	**0.11**
Naphtalene	**0.23**	25.36	0.58	2.58	**0.11**



(a) Ethanol with methyl group.



(b) Malonaldehyde with aldehyde groups.

Figure 5.2: Two configurations from the MD trajectories of ethanol and malonaldehyde each. The rotating functional groups of the molecules are marked with dashed lines.

hand, these symmetries do not correspond well to the assignment of unique identifiers for the atoms in the molecules, which is performed implicitly by GDML due to the use of the second derivative of the Coulomb matrix as descriptor. While similar symmetries can also be observed for molecules with aromatic rings, SchNet appears to require more data in order to distinguish between locally similar atom environments at distinct positions in the molecules. E.g., SchNet performs worse than GDML on toluene even though it possesses a rotating methyl group.

GDML has limited ability to scale due to the kernel matrix scaling quadratically with the total number of atoms in the training set. Since SchNet can easily scale to larger training sets, we also train on a set of 50,000 training examples. Again, the force information helps significantly for energy and force predictions, however, the improvements have become smaller. This is because of the increased likelihood of redundant information about the local environment of training examples in the energy gradients and the added training examples. SchNet is now reaching or surpassing the performance of GDML on the small dataset for all molecules (see Tables 5.3 and 5.4). We conclude that SchNet has the expressive power and necessary scalability to represent the configurations in the MD trajectories, however GDML is more data-efficient up to highly symmetric molecules.

5.2.2 PES for $C_7O_2H_{10}$ isomers

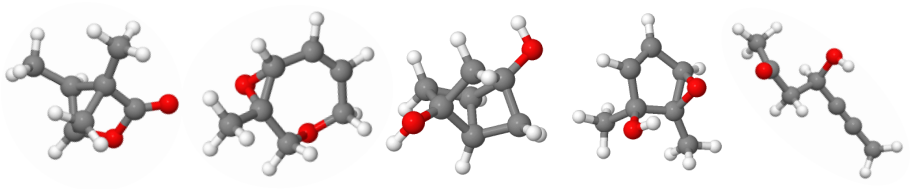


Figure 5.3: Selection of $C_7O_2H_{10}$ isomers from ISO17 dataset.

Having predicted energies and forces for single MD trajectories of small

organic molecules from MD17, the next challenge is to use SchNet to learn a more general potential energy surface. While the ultimate goal is a model for compositional and configurational degrees of freedom, we will take an intermediate step here, training a common model for molecular dynamics of various isomers. For this, we employ a dataset of short MD trajectories of 129 molecules that are randomly sampled from the largest set of isomers in QM9 with the composition $C_7O_2H_{10}$. With each trajectory consisting of 5,000 steps, the data set consists of $129 \times 5,000 = 645,000$ labeled examples with calculated energies and atomic forces. While these molecules have the same composition, they represent diverse structures with different chemical bonding. This can be seen in Fig. 5.3, where we have plotted five molecules from the ISO17 dataset. Specifics about how the dataset was generated are listed in Appendix A.

We split the data according to the following scheme: First, we split the data into 80% known and 20% unknown MD trajectories. Then, we split the known trajectories into 80% known and 20% unknown configurations. This leaves us with a set for training and validation consisting of 80% of configurations of 80% of the MD trajectories, a test set of the remaining 20% unseen configurations within known MD trajectories (test-within) as well as a test set of the unseen 20% of the MD trajectories (test-other). While the first test set serves to estimate how well the model can represent multiple trajectories, the second test set will be used to evaluate how well the model has learned to generalize to other molecules.

Table 5.5: Mean absolute errors on $C_7O_2H_{10}$ isomers of energy and force predictions in kcal mol^{-1}and kcal mol^{-1} Å^{-1}, respectively [Sch+17b]. SchNet was trained using three interaction layers using only energies as well as with energies and forces ($\rho = 0.01$). We give the mean predictor for reference.

		mean predictor	**SchNet**	
			energy	*energy+forces*
known molecules / unknown conformation	*energy*	14.89	0.52	**0.36**
	forces	19.56	4.13	**1.00**
unknown molecules / unknown conformation	*energy*	15.54	3.11	**2.40**
	forces	19.15	5.71	**2.18**

Table 5.5 shows the performance of SchNet using three interaction layers on both test sets. When predicting the remaining conformations of the known trajectories, SchNet reaches chemical accuracy. While this is not enough to perform an MD simulation, it shows that SchNet is able to represent geometries of a more general potential energy surface. For the setting with unknown MD trajectories, SchNet does still reach 2.40 kcal mol^{-1}and 2.18 kcal mol^{-1} Å^{-1}, respectively.

In both settings, training including atomic forces improves both energy and force predictions. This demonstrates that force information does not only

help with the prediction of very similar configurations, but also helps with generalization across chemical compound space.

5.3 Molecular dynamics study of C_{20} fullerene

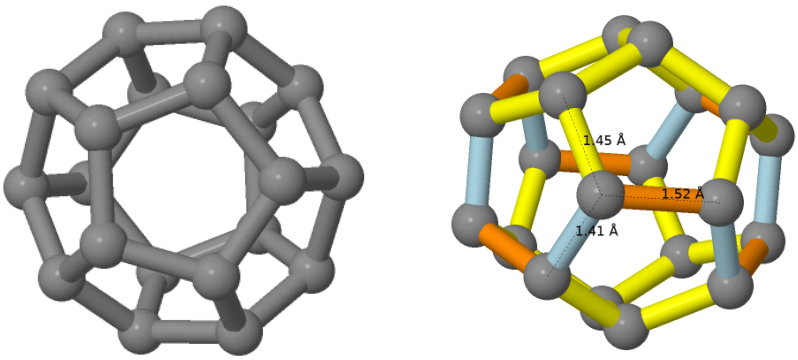


Figure 5.4: Two perspectives of the fullerene C_{20}. The geometry was optimized using the predicted forces of SchNet. On the right, equal bond lengths are color-coded and annotated in Ångstrom.

While we have demonstrated that SchNet can deliver accurate predictions of energies and forces, we still need to show that this can practically be used to drive a molecular dynamics simulation. We have selected the fullerene C_{20} for an exemplary study of whether this is feasible and how much speedup we can gain for a small molecule of this size. Fig. 5.4 depicts the molecule in its equilibrium configuration, which is a cage of carbon atoms.

The reference data was generated using a classical MD simulation at 500K for 29,689 time steps at the PBE+vdW^{TS} level of theory [PBE96; TS09]. Further details on the data generation are listed in Chapter A.

We perform a two-step model selection where we evaluate SchNet models with $T \in \{3, 6\}$ interaction blocks and $F \in \{64, 128\}$ feature dimensions of the atom-wise representations. In a first step, we set the energy-force trade-off to $\rho = 0.01$ which proved to be a good compromise in our experiments for the MD17 and ISO17 datasets. Table 5.6 (upper half) shows the results of the model selection in terms of mean absolute errors. The best results could be obtained with the largest model with $T = 6$ interaction blocks and $F = 128$ feature dimensions.

While we have aimed for a compromise between energy and force predictions in previous sections, it might make sense to train separate models for energies and forces. This is because errors can be shifted between the energy and force prediction accuracy depending on the trade-off. By choosing the

Table 5.6: Model selection for C_{20} molecular dynamics study [Sch+18]. Mean absolute errors for energy and force predictions of C_{20}-fullerene in kcal mol^{-1}and kcal mol^{-1}Å^{-1}, respectively. We compare SchNet models with varying number of interaction blocks T, feature dimensions F and energy-force tradeoff ρ. For force-only training ($\rho = 0$), the integration constant is fitted separately. Best models in **bold**.

interactions T	**features** F	**energy loss scale** ρ	**energy**	**forces**
3	64	0.010	0.228	0.401
6	64	0.010	0.202	0.217
3	128	0.010	0.188	0.197
6	128	0.010	**0.1002**	**0.120**
6	128	0.100	**0.027**	0.171
6	128	0.010	0.100	0.120
6	128	0.001	0.238	0.061
6	128	0.000	0.260	**0.058**

correct trade-off, we can obtain optimal force and energy models

$$\hat{\mathbf{F}}_{\rho_F} = -\nabla \tilde{E}_{\rho_F} \tag{5.8}$$

$$\tilde{\mathbf{F}}_{\rho_E} = -\nabla \hat{E}_{\rho_E}, \tag{5.9}$$

where ρ_E and ρ_F are the optimal trade-offs for energy and force prediction, respectively. Each model has a corresponding suboptimal force field $\tilde{\mathbf{F}}_{\rho_E}$ or potential $\tilde{E}_{\rho_F}$. Given the ground truth force $\mathbf{F}$ and potential E, the errors fulfill

$$\|\hat{\mathbf{F}}_{\rho_F} - \mathbf{F}\|_{L_2}^2 \leq \|\tilde{\mathbf{F}}_{\rho_E} - \mathbf{F}\|_{L_2}^2 \tag{5.10}$$

$$\|\hat{E}_{\rho_E} - E\|_{L_2}^2 \leq \|\tilde{E}_{\rho_F} - E\|_{L_2}^2. \tag{5.11}$$

Each force field is conservative w.r.t. its corresponding potential. However, the MD simulation might still leak energy with respect to the optimal energy model or the ground truth energy. For the following results, we only need accurate forces and do not require energies.

We use the previously selected settings for interaction blocks and number of features and train models with trade-offs $\rho \in \{10^{-1}, 10^{-2}, 10^{-3}, 0.0\}$. The last setting corresponds to a model that is exclusively trained using forces, however, even here we use the differentiated energy model to guarantee energy conservation [Chm+17]. For the energy prediction, we have to additionally fit the bias of the last layer as it corresponds to the integration constant. Table 5.6 (bottom half) shows the influence of the trade-off. By training specialized models for energies and forces, we are able to improve energy prediction from mean absolute errors of 0.1002 kcal mol^{-1}to 0.027 kcal mol^{-1}and force prediction from 0.12 kcal mol^{-1} Å^{-1}to 0.058 kcal mol^{-1} Å^{-1}.

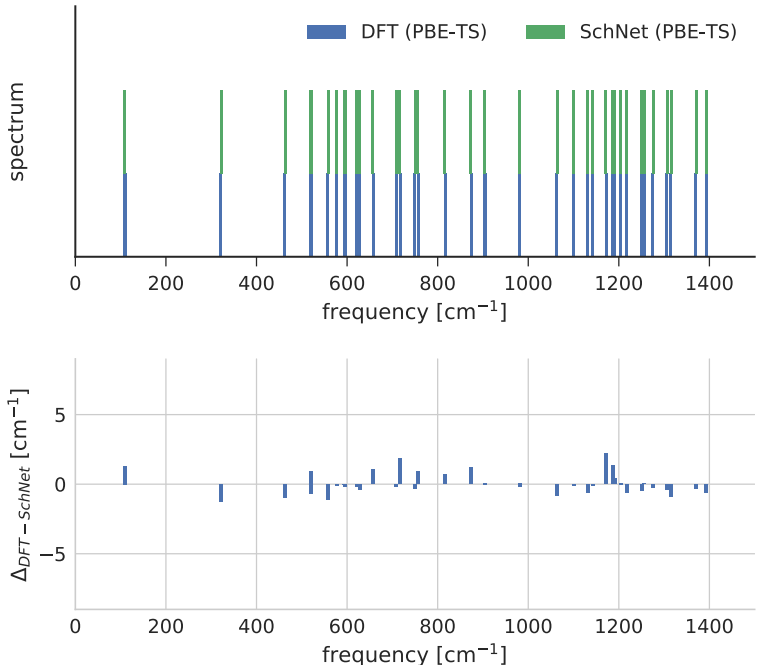


Figure 5.5: Normal mode analysis of the fullerene C_{20} dynamics comparing SchNet and DFT results [Sch+18].

As a first step to validate the force model, we apply to relaxation of the fullerene C_{20} geometry. Fig. 5.4 shows the relaxed structure of C_{20} where the relaxation has been converged up to a maximum force of 10^{-4} kcal mol^{-1} Å^{-1}. The molecule is not a perfect dodecahedron, but possesses three distinct bond lengths [PA91]. These are color-coded on the right side of Fig. 5.4 and are for our model 1.41Å, 1.45Å and 1.52Å, which agrees with the relaxed structure using DFT at PBE+vdW^{TS} level of theory. Fig. 5.5 shows a comparison of the vibrational spectrum of DFT and our model. The frequencies in the vibrational spectrum correspond to the eigenvalues of the mass-weighted Hessian

$$\mu_{ij} = \frac{1}{\sqrt{m_i m_j}} \frac{\partial^2 E}{\partial \mathbf{r}_i \partial \mathbf{r}_j}(\mathbf{r}_1, \ldots, \mathbf{r}_{20})$$

at the equilibrium configuration. The largest error in the frequencies is $\sim$1% of the corresponding DFT reference energy. This analysis as well as the results in Table 5.6 demonstrate that SchNet is able to accurately reconstruct the potential energy surface and its symmetries.

We perform the MD simulation using SchNet at 300K with classical MD as well as path-integral MD (PIMD) using 8 beads, which introduces nuclear quantum effects. Fig 5.6 shows the distributions of nearest neighboring atom distances and the diameter of C_{20} as well as the radial distribution function for both MD trajectories. The addition of nuclear quantum effects widens the distribution of nearest neighbor distances which agrees with recently reported PIMD results on graphene [PDT18].

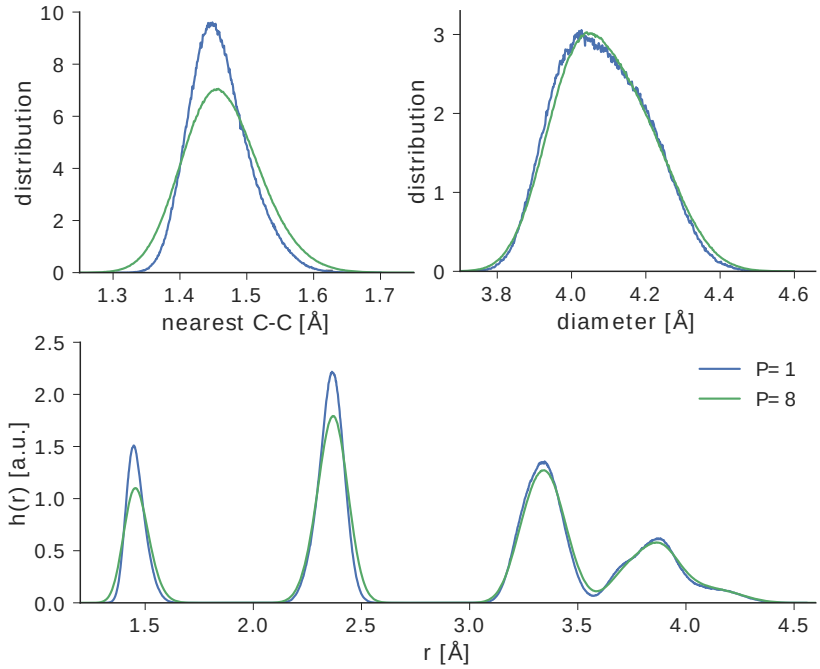


Figure 5.6: Analysis of the fullerene C_{20} dynamics at 300K using SchNet [Sch+18]. Distribution functions for nearest neighbours, diameter of the fullerene and the radial distribution function using classical MD (blue) and PIMD with 8 beads (green).

Each single-point DFT calculation of C_{20} requires a computation time of 11 seconds using 32 CPU cores. Using SchNet, this could be reduced to 10 ms for a single prediction on an NVIDIA GTX1080 GPU. Since PIMD requires multiple calculations per time step, this runtime can be further reduced by predicting the forces of the batch in parallel without much overhead. This speedup has made it possible to perform 1.25 ns of PIMD by reducing the runtime by 3-4 orders of magnitude: from about 7 years to less than 7 hours.

5.4 Summary and discussion

In this chapter, we have applied SchNet to the prediction of potential energy surfaces and energy-conserving force fields. We have used a combined loss with energy and force terms to obtain models that improve upon pure energy models and reduce the required amount of reference calculations. SchNet accurately predicts energies and forces of MD trajectories of small organic molecules, even on a small training set of 1,000 molecules. We have explored the prediction across chemical and configurational space and obtained encouraging results on a set of $C_7O_2H_{10}$ isomers. In future research, such models may be used in combination with active learning strategies to build data-efficient predictors for reaction paths and catalysis.

Finally, we have applied SchNet to study of the dynamics of C_{20} fullerene

at the PBE+vdW^{TS} level of theory. We have validated our model by demonstrating accurate predictions of energies and forces as well as good agreement in the vibrational spectrum compared to ab initio DFT calculations. Then, we have used SchNet to generate a 1.25 ns PIMD trajectory including nuclear quantum effects. This would not have been computationally feasible with ab initio DFT calculations which would have taken years instead of hours. In future work, we will apply SchNet to MD simulation studies of other molecules and validate our models against an expanded range of properties. Further research is also necessary to evaluate the best strategy for cases where both accurate energies and forces are required, in particular, the prediction of thermodynamical properties such as thermal energy or specific heat.

Chapter 6

Conclusions and outlook

The goal of this thesis has been to develop end-to-end machine learning techniques capable of learning representations for atomistic systems directly from atom types and positions. Based on the analysis of hand-crafted machine learning descriptors for molecules and materials in Chapter 2, we have proposed two neural network architectures that learn atom-wise representations of chemical environments. They guarantee the fundamental invariances towards translation, rotation and atom indexing. Both neural networks obtain embeddings of atom types that allow for cross-element generalization and apply repeated pair-wise interactions between atoms to incorporate environment information into the atom-wise features. The crucial difference between the two architectures is how the interactions are modeled. In Chapter 3, we have proposed the deep tensor neural network (DTNN) for molecules that use factorized tensor layers to model the interaction function. In Chapter 4, we have introduced continuous-filter convolutional layers, which we use within the interaction blocks of our second architecture SchNet.

Both neural networks yield chemically accurate predictions of energies in compositional and configurational space. SchNet improves over DTNN consistently, in particular, reducing the error on the benchmark dataset QM9 by more than 50%. DTNN and SchNet decompose the property of interest into atom-wise contributions, such that we can obtain a partitioning of the atomization energy. In our analysis, we have found that SchNet learns more stable partitionings than DTNN that have narrower energy ranges. On top of that we have defined local chemical potentials that visualize the spatial structure of the interactions. Finally, we have conducted a sensitivity analysis of the atom-wise features towards bond breaking. All three experiments have shown evidence that the representations of SchNet models are more local than those of DTNN models. This presents a plausible explanation of the improved performance.

We have encoded periodic boundary conditions into the filter-generating networks of SchNet to directly obtain filters that reflect the periodicity of bulk crystals. This allowed us to efficiently predict formation energies for a diverse

set of crystals from the Materials Project repository. Due to the wide variety of atom types in this data, we were able to show that SchNet is indeed able generalizes across elements: We have shown that the atom type embeddings learned by SchNet agree with chemical intuition: they cluster based to their main group and partially order from light to heavy elements.

Due to a careful choice of activation function and distance basis, SchNet has been designed to be smooth such that second derivatives can be obtained. On this basis, we use a combined loss for energies and forces to improve the accuracy of the model without requiring more reference calculations. We have used this to apply SchNet to the prediction of potential energy surfaces in configurational and chemical space. Most notably, we have performed a molecular dynamics study of fullerene C_{20}. SchNet has been able to accurately reproduce the vibrational spectrum and has been used to generate a 1.25ns path-integral MD trajectory at the PBE+vdW^{TS} level of theory, which would not have been feasible with conventional ab initio methods.

Several avenues of future research remain for improving and extending the SchNet and DTNN architectures. A major concern is the improvement of data efficiency in order to go to larger system sizes and higher levels of theory, where less training data is available. This may be achieved by semi-supervised learning of the representation, transfer learning from less accurate calculations to higher levels of theory or active learning. Another issue is the reliability of the prediction accuracy, in particular during molecular dynamics simulations, where configurations that are not well represented by the training data might be encountered at some point in the trajectory. Here, uncertainty measures are crucial so that such a situation can be detected. Finally, we will need to study whether and how the architecture can be extended to delocalized properties and long-range quantum interactions. In conclusion, SchNet and DTNN present flexible deep learning frameworks for atomistic systems that we expect to facilitate further developments towards interpretable deep learning architectures to assist chemistry research.

Appendix A

Datasets

In the following, we briefly describe the employed datasets. All reference calculations were performed using density functional theory [HK64] employing various levels of theory as given per dataset.

A.1 Chemical compound space

Datasets in this section contain diverse sets of molecules at equilibrium across chemical compound space.

QM7b [Mon+13] This dataset consists of all possible 7211 organic molecules with up to seven heavy atoms from the set {C, N, O, S, Cl} and saturated with hydrogen. It is a subset of the GDB-13 [BR09] enumeration of organic molecules and includes geometries as well as 13 properties at different levels of theory. In this thesis, we only use the atomization energy calculated with FHI-AIMS [Blu+09] at PBE0 [PEB96] level of theory. The data is available at `www.quantum-machine.org`.

QM9 [Ram+14] This dataset constitutes a subset of the GDB-17 database [BR09; Rey15] consisting of all 133,885 molecules with up to nine heavy atoms from the set {C, N, O, F}. It includes 15 quantum-chemical properties calculated at the B3LYP/6-31G(2df,p) [Bec88; LYP88; Bec93] level of theory with Gaussian 09 [Fri+09]. The properties are described in Table A.1. For the properties U_0, U, H, G and C_v, QM9 provides single-atoms references, which can be used to obtain a better starting point for the neural networks when predicting only the contributions due to the interactions. E.g., instead of predicting

Table A.1: Properties of QM9 and the units as used in this thesis. For further details, see Ramakrishnan et al. [Ram+14].

Symbol	Unit	Description
ϵ_{HOMO}	kcal mol^{-1}	The **energy of the highest occupied molecular orbital** is the highest energy level which is occupied with electrons.
ϵ_{LUMO}	kcal mol^{-1}	The **energy of the lowest unoccupied molecular orbital** is the energy level above ϵ_{HOMO}, which is the unoccupied level.
$\Delta\epsilon$	kcal mol^{-1}	The **HOMO-LUMO gap** is the energy difference $\epsilon_{\text{LUMO}} - \epsilon_{\text{HOMO}}$ which determines how much energy is required to reach an excited state.
ZPVE	kcal mol^{-1}	The **zero-point vibrational energy** corresponds to the motion of the molecule at 0K caused by Heisenberg's uncertainty principle.
μ	Debye	The **magnitude of the dipole moment** describes the polarity of the molecule.
α	Bohr^3	The **isotropic polarizability** describes to what degree an external field can induce a dipole moment in the molecule.
$\langle R^2 \rangle$	Bohr^2	The **electronic spatial extent** is the second moment of the charge distribution.
U_0	kcal mol^{-1}	The **internal energy** of the molecule at 0K.
U	kcal mol^{-1}	The **internal energy** of the molecule at 298.15K.
H	kcal mol^{-1}	The **enthalpy** of the molecule at 298.15K.
G	kcal mol^{-1}	The **free energy** of the molecule at 298.15K.
C_v	cal / molK	The **heat capacity** of the molecule at 298.15K.

Table A.2: Overview about datasets in MD17 collection.

Molecule	Formula	n_{data}
Benzene	C_6H_6	627,000
Uracil	$C_4H_4N_2O_2$	133,000
Naphthalene	$C_{10}H_8$	326,000
Aspirin	$C_9H_8O_4$	211,000
Salicylic acid	$C_7H_6O_3$	320,000
Malonaldehyde	$C_3H_4O_2$	993,000
Ethanol	C_2H_6O	555,000
Toluene	C_7H_8	442,000

the internal energy U_0, we predict the atomization energy

$$U_{0,\text{at}} = U_0 - \sum_{i=1}^{n_{\text{atoms}}} U_{0,Z_i},$$

where U_{0,Z_i} is the internal energy of atoms with nuclear charge Z_i. Since this is only an offset, prediction of both atomization and internal energy can be obtained with the same accuracy. The QM9 dataset is available at:
`https://doi.org/10.6084/m9.figshare.978904`

A.2 Molecular dynamics trajectories

MD17 [Chm+17] This is a collection of path-integral molecular dynamics trajectories of small organic molecules at the PBE-TS [PBE96; TS09] level of theory using the FHI-aims code [Blu+09]. The MD simulations were performed at 500K with a time of 0.5 fs using the i-PI code [CMM14]. Energies and forces were calculated at the PBE+vdW$^{\text{TS}}$ [PBE96; TS09] level of theory. Table A.2 gives an overview about the molecules in the collection as well as the size of the datasets. The data is available at `www.quantum-machine.org`.

Fullerene C_{20} [Sch+18] This dataset consists of a short MD trajectory of $\sim$30k configurations of the fullerene C_{20} generated by a classical MD at 500K with a step size of 1fs using DFT at the PBE+vdW$^{\text{TS}}$ [PBE96; TS09] level of theory using the FHI-aims code [Blu+09].

ISO17 [Sch+17a; Sch+17b] This dataset was generated from molecular dynamics simulations at the PBE+vdW$^{\text{TS}}$ [PBE96; TS09] level of theory. It consists of 129 molecules each containing 5,000 conformational geometries, energies

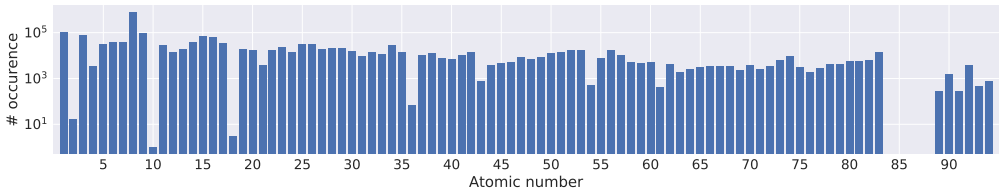


Figure A.1: Histogram of atom types in the Materials Project dataset. The dataset includes 89 atom types ranging across the periodic table.

and forces with a step size of 1 fs. The molecules were randomly drawn from the largest set of isomers in the QM9 dataset ($C_7O_2H_{10}$). The data is available at `www.quantum-machine.org`.

A.3 Materials

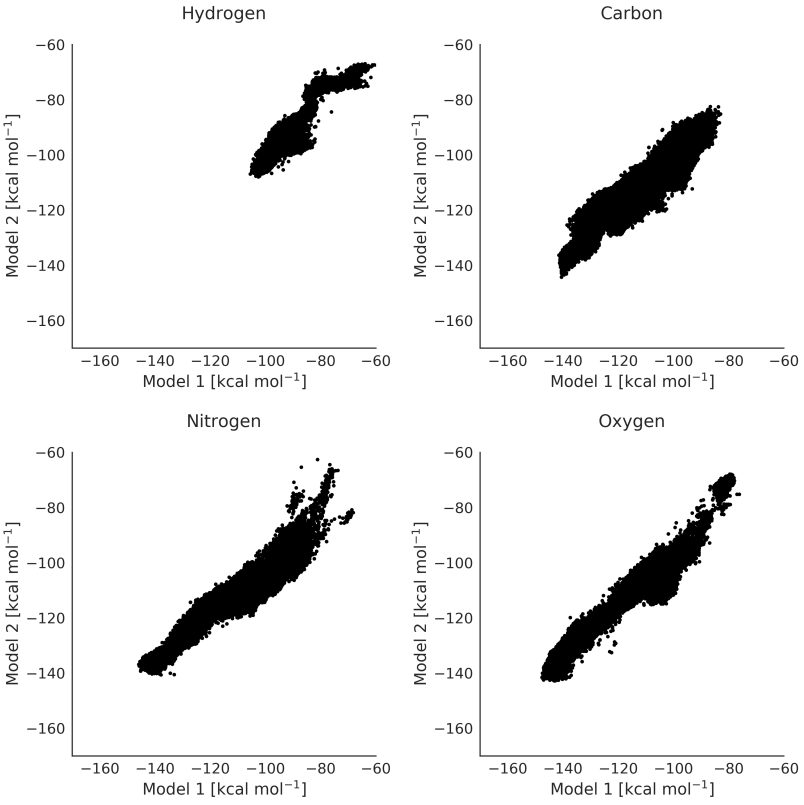
Materials Project [Jai+13] The Materials Project is a repository of bulk crystals and their electronic properties calculated with VASP [KF96] at the GGA+U level of theory [Jai+11]. We use a snapshot of the repository downloaded on August 14th, 2017 including 69,640 structures and reference calculations of formation energies. The crystal unit cells contain up to 296 atoms from 89 different atom types. Fig. A.1 shows a histogram over atom types in the Materials Project dataset. The data is available at `www.materialsproject.org`.

Appendix B

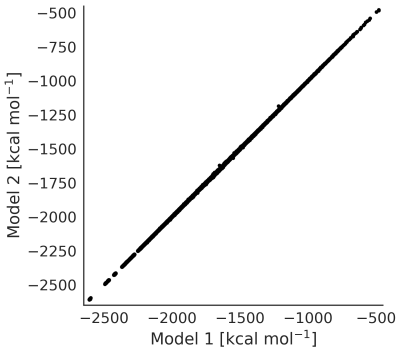
Supplemental results

B.1 Scatter plots of energy contributions

We show scatter plots of energy contributions for DTNN and SchNet, which are complementary to the distribution plots in Sections 3.6.1 and 4.5.1.

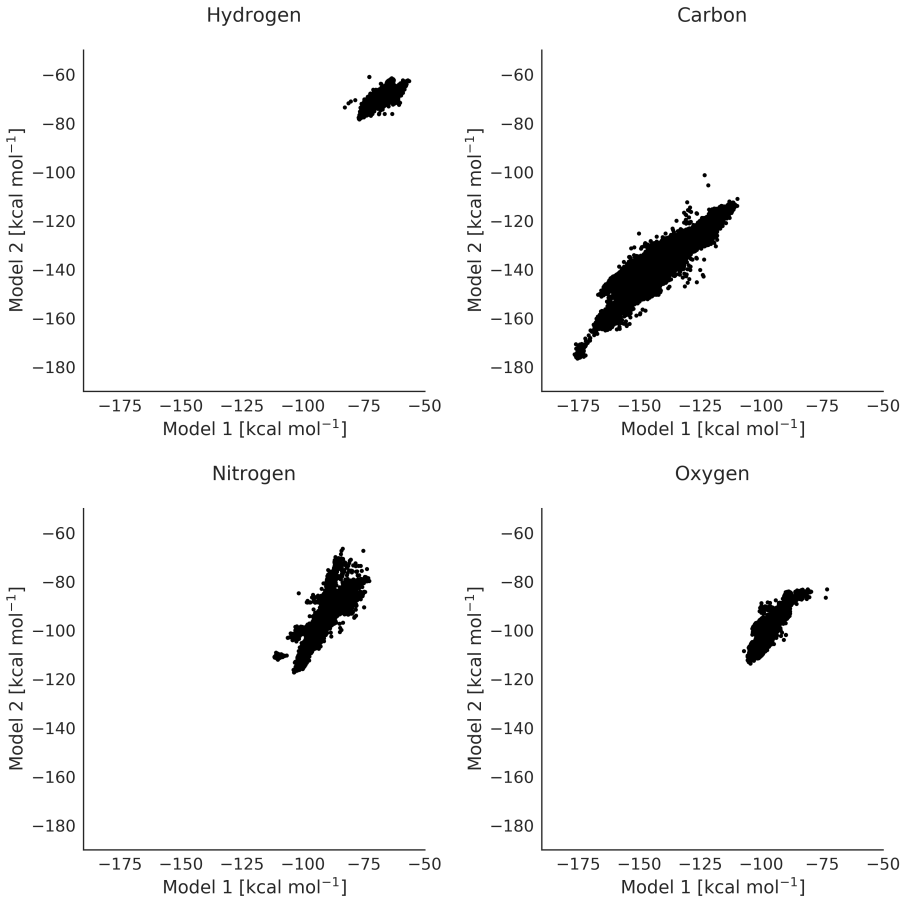


(a) Energy contributions

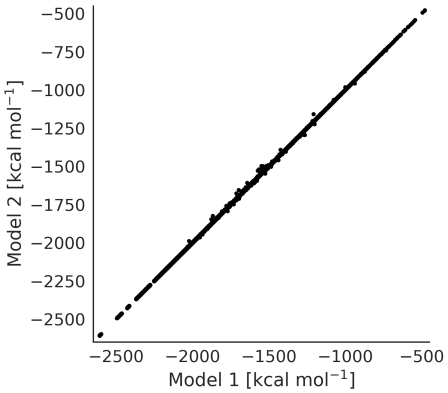


(b) Atomization Energies

Figure B.1: Scatter plots of energy contributions for atoms of types H, C, N, O and atomization energies from QM9 molecules predicted by two DTNN (T=3) models. The models were trained on 100k examples. Model 1 and 2 were trained on different subsets.

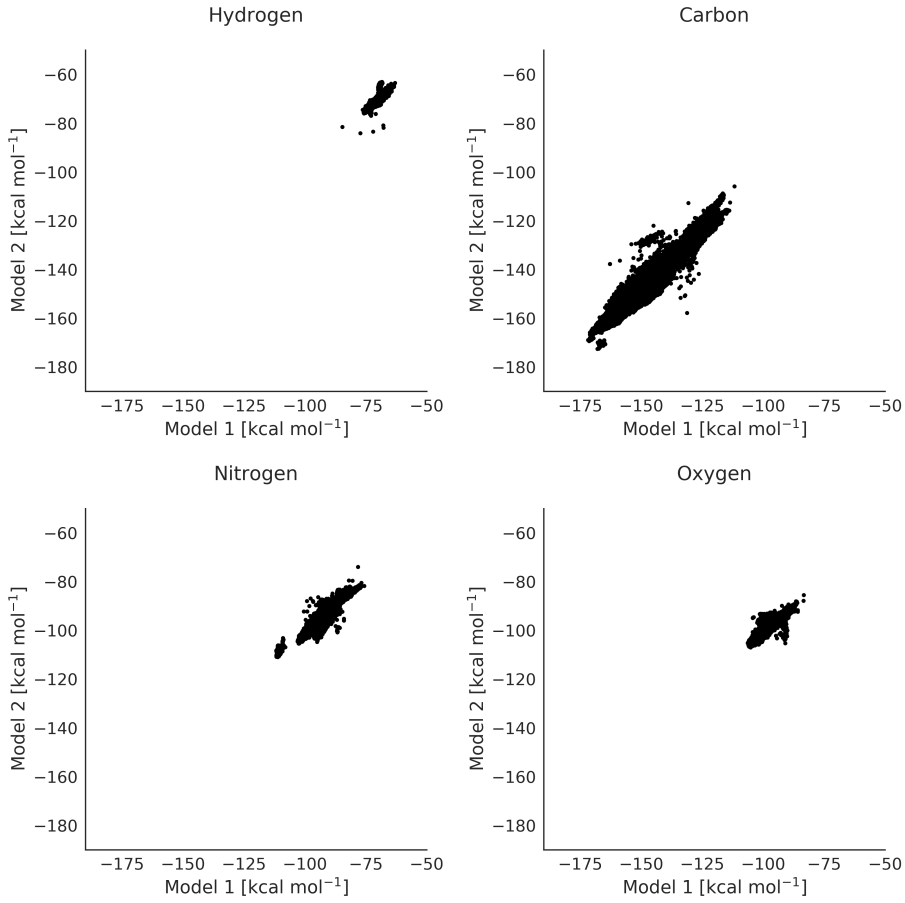


(a) Energy contributions

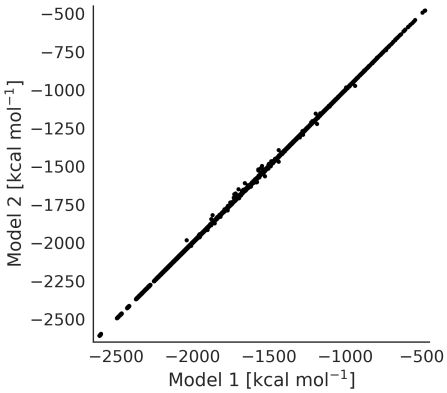


(b) Atomization Energies

Figure B.2: Scatter plots of energy contributions for atoms of types H, C, N, O and atomization energies from QM9 molecules predicted by two SchNet (T=3) models. The models were trained on 100k examples. Model 1 and 2 were trained on different subsets.



(a) Energy contributions

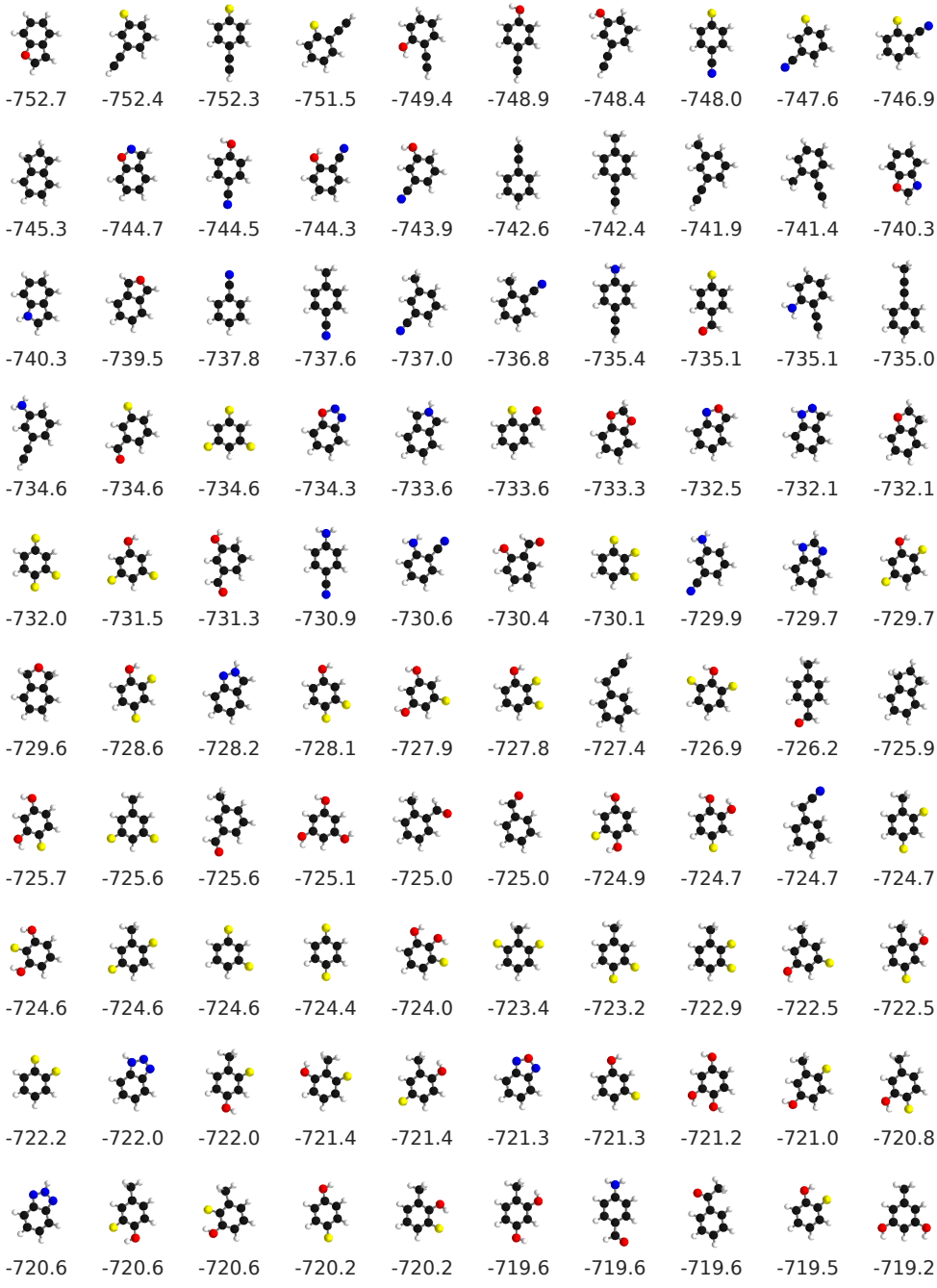


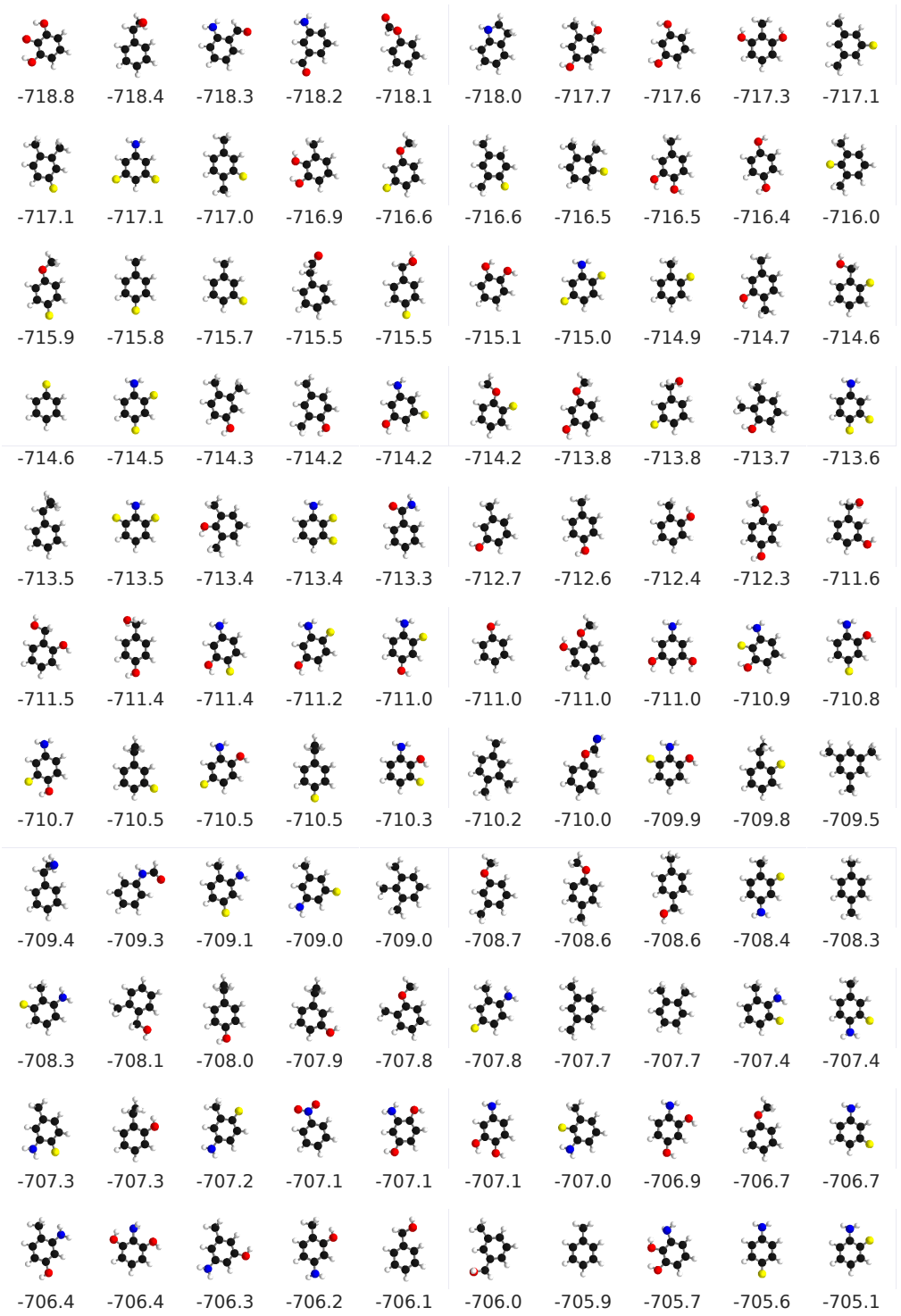
(b) Atomization Energies

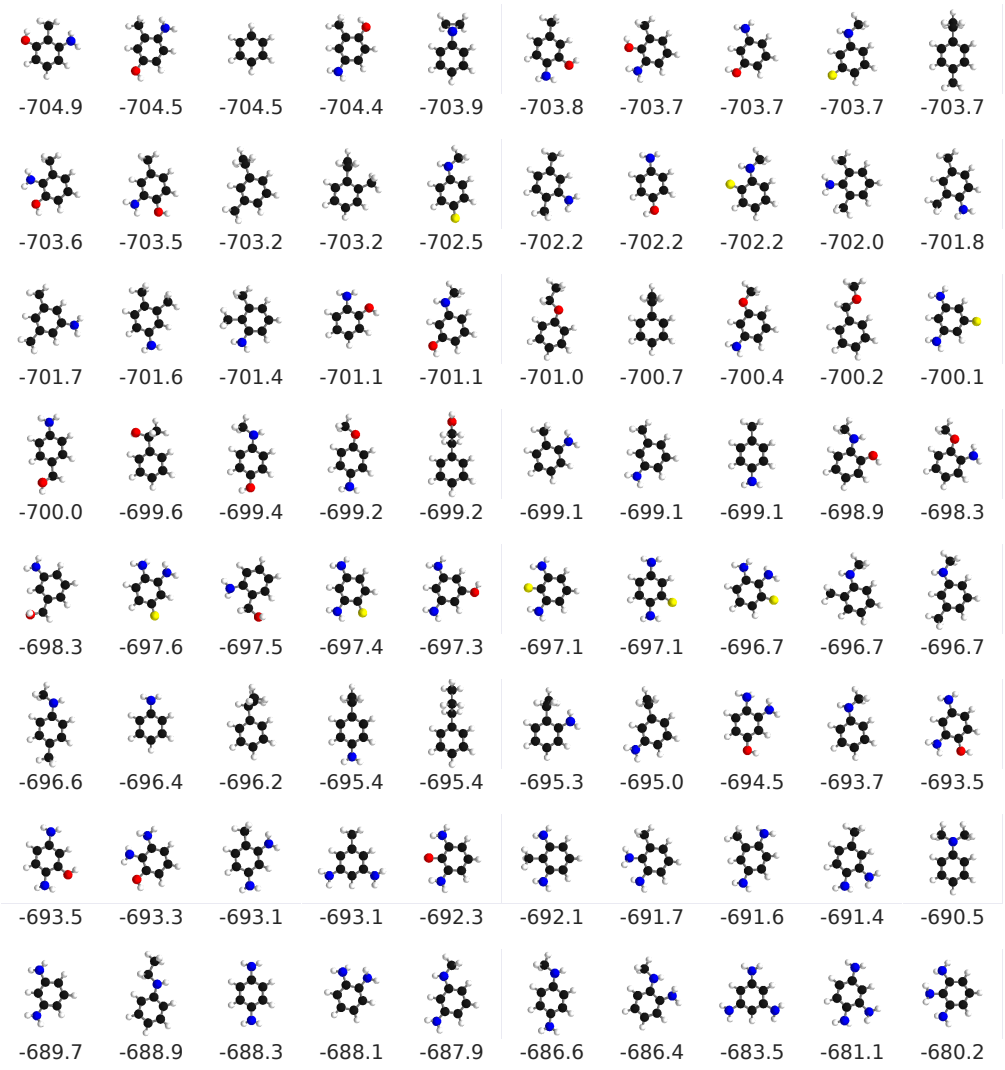
Figure B.3: Scatter plots of energy contributions for atoms of types H, C, N, O and atomization energies from QM9 molecules predicted by two SchNet (T=6) models. The models were trained on 100k examples. Model 1 and 2 were trained on different subsets.

B.2 Stability ranking of 6-membered carbon rings

Here,we show the full list of molecules for the stability ranking of 6-membered carbon rings from Section 4.5.4:




-718.8 -718.4 -718.3 -718.2 -718.1 -718.0 -717.7 -717.6 -717.3 -717.1
-717.1 -717.1 -717.0 -716.9 -716.6 -716.6 -716.5 -716.5 -716.4 -716.0
-715.9 -715.8 -715.7 -715.5 -715.5 -715.1 -715.0 -714.9 -714.7 -714.6
-714.6 -714.5 -714.3 -714.2 -714.2 -714.2 -713.8 -713.8 -713.7 -713.6
-713.5 -713.5 -713.4 -713.4 -713.3 -712.7 -712.6 -712.4 -712.3 -711.6
-711.5 -711.4 -711.4 -711.2 -711.0 -711.0 -711.0 -711.0 -710.9 -710.8
-710.7 -710.5 -710.5 -710.5 -710.3 -710.2 -710.0 -709.9 -709.8 -709.5
-709.4 -709.3 -709.1 -709.0 -709.0 -708.7 -708.6 -708.6 -708.4 -708.3
-708.3 -708.1 -708.0 -707.9 -707.8 -707.8 -707.7 -707.7 -707.4 -707.4
-707.3 -707.3 -707.2 -707.1 -707.1 -707.1 -707.0 -706.9 -706.7 -706.7
-706.4 -706.4 -706.3 -706.2 -706.1 -706.0 -705.9 -705.7 -705.6 -705.1


-704.9 -704.5 -704.5 -704.4 -703.9 -703.8 -703.7 -703.7 -703.7 -703.7
-703.6 -703.5 -703.2 -703.2 -702.5 -702.2 -702.2 -702.2 -702.0 -701.8
-701.7 -701.6 -701.4 -701.1 -701.1 -701.0 -700.7 -700.4 -700.2 -700.1
-700.0 -699.6 -699.4 -699.2 -699.2 -699.1 -699.1 -699.1 -698.9 -698.3
-698.3 -697.6 -697.5 -697.4 -697.3 -697.1 -697.1 -696.7 -696.7 -696.7
-696.6 -696.4 -696.2 -695.4 -695.4 -695.3 -695.0 -694.5 -693.7 -693.5
-693.5 -693.3 -693.1 -693.1 -692.3 -692.1 -691.7 -691.6 -691.4 -690.5
-689.7 -688.9 -688.3 -688.1 -687.9 -686.6 -686.4 -683.5 -681.1 -680.2

B.3 MD17 predictions with T=6 interaction blocks

The following tables contain supplementary results for Chapter 5 using larger models with six instead of three interaction blocks. As expected from our model selection (see Tables 5.1 and 5.2), SchNet ($T = 3$) achieves the lower prediction errors in most cases.

Table B.1: Mean absolute errors for total energies of MD17 trajectories in kcal/mol. SchNet (T=6) test errors (results for $T = 3$ in brackets) for N=1,000 and N=50,000 reference calculations of molecular dynamics simulations of small, organic molecules are shown. Improved results in **bold**.

	N = 1,000		N = 50,000	
trained on	*energy*	*energy+forces*	*energy*	*energy+forces*
Benzene	1.24 (1.19)	0.20 (0.08)	**0.07** (0.08)	0.08 (0.07)
Toluene	**2.82** (2.95)	0.14 (0.12)	0.17 (0.16)	0.09 (0.09)
Malonaldehyde	2.15 (2.03)	0.17 (0.13)	0.13 (0.13)	0.09 (0.08)
Salicylic acid	3.42 (3.27)	0.24 (0.20)	0.25 (0.25)	0.10 (0.10)
Aspirin	4.25 (4.20)	0.43 (0.37)	0.25 (0.25)	**0.11** (0.12)
Ethanol	1.11 (0.93)	0.09 (0.08)	**0.05** (0.07)	0.05 (0.05)
Uracil	**2.22** (2.26)	0.18 (0.14)	0.13 (0.13)	0.10 (0.10)
Naphtalene	**3.44** (3.58)	0.20 (0.16)	0.21 (0.20)	0.10 (0.11)

Table B.2: Mean absolute errors for atomic forces of MD17 trajectories in kcal/mol/Å. SchNet (T=6) test errors (results for $T = 3$ in brackets) for N=1,000 and N=50,000 reference calculations of molecular dynamics simulations of small, organic molecules are shown. Improved results in **bold**.

	N = 1,000		N = 50,000	
trained on	*energy*	*energy+forces*	*energy*	*energy+forces*
Benzene	11.62 (14.12)	0.37 (0.31)	1.41 (1.23)	0.18 (0.17)
Toluene	16.94 (22.31)	0.59 (0.57)	1.82 (1.79)	0.09 (0.09)
Malonaldehyde	**19.65** (20.41)	0.70 (0.66)	**1.38** (1.51)	**0.07** (0.08)
Salicylic acid	**19.56** (23.21)	0.91 (0.85)	**3.59** (3.72)	**0.16** (0.19)
Aspirin	**21.42** (23.54)	1.60 (1.35)	7.84 (7.36)	**0.25** (0.33)
Ethanol	8.16 (6.56)	0.42 (0.39)	**0.62** (0.76)	0.05 (0.05)
Uracil	**16.56** (20.08)	0.63 (0.56)	3.44 (3.28)	**0.10** (0.11)
Naphtalene	**20.47** (25.36)	0.64 (0.58)	2.70 (2.58)	0.10 (0.11)

References

[AM76] N. Ashcroft and N. Mermin. *Solid State Physics*. Belmond, California, USA: Brooks/Cole, 1976.

[Bac+15] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". *PloS one* 10 (7), e0130140, 2015.

[Bae+10] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. "How to explain individual classification decisions". *Journal of Machine Learning Research* 11, pp. 1803–1831, 2010.

[Bal+17] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams. "The Shattered Gradients Problem: If resnets are the answer, then what is the question?" *arXiv preprint arXiv:1702.08591*, 2017.

[BB72] R. F. Bader and P. Beddall. "Virial field relationship for molecular charge distributions and the spatial partitioning of molecular properties". *The Journal of Chemical Physics* 56 (7), pp. 3320–3329, 1972.

[Bec88] A. D. Becke. "Density-functional exchange-energy approximation with correct asymptotic behavior". *Physical review A* 38 (6), p. 3098, 1988.

[Bec93] A. D. Becke. "A new mixing of Hartree–Fock and local density-functional theories". *The Journal of chemical physics* 98 (2), pp. 1372–1377, 1993.

[Beh11] J. Behler. "Atom-centered symmetry functions for constructing high-dimensional neural network potentials". *J. Chem. Phys.* 134 (7), p. 074106, 2011.

[Bis95] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[BKC13] A. P. Bartók, R. Kondor, and G. Csányi. "On representing chemical environments". *Phys. Rev. B* 87 (18), p. 184115, 2013.

[Blu+09] 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". *Computer Physics Communications* 180 (11), pp. 2175–2196, 2009.

[BM07] R. J. Bartlett and M. Musiał. "Coupled-cluster theory in quantum chemistry". *Reviews of Modern Physics* 79 (1), p. 291, 2007.

[BP07] J. Behler and M. Parrinello. "Generalized neural-network representation of high-dimensional potential-energy surfaces". *Phys. Rev. Lett.* 98 (14), p. 146401, 2007.

[BR09] 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.* 131, p. 8732, 2009.

[Bro+17] F. Brockherde, L. Voigt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller. "Bypassing the Kohn-Sham equations with machine learning". *Nature Communications* 8, p. 872, 2017.

[Bro+83] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. "CHARMM: a program for macromolecular energy, minimization, and dynamics calculations". *Journal of computational chemistry* 4 (2), pp. 187–217, 1983.

[Bru+13] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. "Spectral networks and locally connected networks on graphs". *Proceedings of the 2nd International Conference on Learning Representations*, 2013.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". *IEEE transactions on neural networks* 5 (2), pp. 157–166, 1994.

[BT98] S. J. Billinge and M. Thorpe. *Local structure from diffraction*. Springer, 1998.

[Chm+17] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. "Machine Learning of Accurate Energy-Conserving Molecular Force Fields". *Science Advances* 3 (5), e1603015, 2017.

[Cho17] F. Chollet. "Xception: Deep Learning With Depthwise Separable Convolutions". in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[CMM14] M. Ceriotti, J. More, and D. E. Manolopoulos. "i-PI: A Python interface for ab initio path integral molecular dynamics simulations". *Computer Physics Communications* 185 (3), pp. 1019–1026, 2014.

[Cor+95] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules". *Journal of the American Chemical Society* 117 (19), pp. 5179–5197, 1995.

[Cra04] C. J. Cramer. *Essentials of computational chemistry: theories and models*. John Wiley & Sons, 2004.

[CUH15] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. "Fast and accurate deep network learning by exponential linear units (ELUs)". *arXiv preprint arXiv:1511.07289*, 2015.

[Cur+13] S. Curtarolo, G. L. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy. "The high-throughput highway to computational materials design". *Nature materials* 12 (3), p. 191, 2013.

[CV95] C. Cortes and V. Vapnik. "Support-vector networks". *Machine learning* 20 (3), pp. 273–297, 1995.

[Den+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". in: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 248–255. 2009.

[DT07] E. E. Dahlke and D. G. Truhlar. "Electrostatically embedded many-body expansion for large systems, with applications to water clusters". *Journal of chemical theory and computation* 3 (1), pp. 46–53, 2007.

[Dug+01] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. "Incorporating second-order functional knowledge for better option pricing". in: *Advances in neural information processing systems*, pp. 472–478. 2001.

[Duv+15] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. "Convolutional Networks on Graphs for Learning Molecular Fingerprints". in: *NIPS*. ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, pp. 2224–2232. 2015.

[Eic+17] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat. "Solid Harmonic Wavelet Scattering: Predicting Quantum Molecular Energy from Invariant Descriptors of 3D Electronic Densities". in: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 6543–6552.

[Fab+15] 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* 115 (16), pp. 1094–1101, 2015.

[Fab+17] 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. "Fast machine learning models of electronic and energetic properties consistently reach approximation errors better than DFT accuracy". *arXiv preprint arXiv:1702.05532*, 2017.

[FM82] K. Fukushima and S. Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition". in: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[Fon+04] C. Fonseca Guerra, J.-W. Handgraaf, E. J. Baerends, and F. M. Bickelhaupt. "Voronoi deformation density (VDD) charges: Assessment of the Mulliken, Bader, Hirshfeld, Weinhold, and VDD methods for charge analysis". *Journal of computational chemistry* 25 (2), pp. 189–210, 2004.

[Fri+09] M. Frisch, G. Trucks, H. B. Schlegel, G. Scuseria, M. Robb, J. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. Petersson, et al. *Gaussian 09, revision D. 01*. 2009.

[GB87] W. F. van Gunsteren and H. J. Berendsen. "Groningen molecular simulation (GROMOS) library manual". *Biomos, Groningen* 24 (682704), p. 13, 1987.

[GBM17] M. Gastegger, J. Behler, and P. Marquetand. "Machine learning molecular dynamics for the simulation of infrared spectra". *Chemical science* 8 (10), pp. 6924–6935, 2017.

[Gil+17] 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*, pp. 1263–1272. 2017.

[Góm+16] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. "Automatic chemical design using a data-driven continuous representation of molecules". *ACS Central Science*, 2016.

[GSS15] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". in: *International Conference on Learning Representations*. 2015.

[Hac+11] 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". *The Journal of Physical Chemistry Letters* 2 (17), pp. 2241–2251, 2011.

[Han+13] 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.* 9 (8), pp. 3404–3419, 2013.

[Han+15] 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.* 6, p. 2326, 2015.

[Hau+13] G. Hautier, A. Jain, T. Mueller, C. Moore, S. P. Ong, and G. Ceder. "Designing Multielectron Lithium-Ion Phosphate Cathodes by Mixing Transition Metals". *Chemistry of Materials* 25 (10), pp. 2064–2074, 2013.

[HBL15] M. Henaff, J. Bruna, and Y. LeCun. "Deep convolutional networks on graph-structured data". *arXiv preprint arXiv:1506.05163*, 2015.

[He+16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. 2016.

[Hir77] F. L. Hirshfeld. "Bonded-atom fragments for describing molecular charge densities". *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* 44 (2), pp. 129–138, 1977.

[HK64] P. Hohenberg and W. Kohn. "Inhomogeneous electron gas". *Physical review* 136 (3B), B864, 1964.

[HL16] B. Huang and O. A. von Lilienfeld. *Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity*. 2016.

[HMP17] M. Hirn, S. Mallat, and N. Poilvert. "Wavelet scattering regression of quantum chemical energies". *Multiscale Modeling & Simulation* 15 (2), pp. 827–863, 2017.

[Hoc98] S. Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (02), pp. 107–116, 1998.

[HPM15] M. Hirn, N. Poilvert, and S. Mallat. "Quantum energy regression using scattering transforms". *arXiv preprint arXiv:1502.02077*, 2015.

[HR17] H. Huo and M. Rupp. "Unified representation for machine learning of molecules and crystals". *arXiv preprint arXiv:1704.06439*, 2017.

[Jai+11] A. Jain, G. Hautier, S. P. Ong, C. J. Moore, C. C. Fischer, K. A. Persson, and G. Ceder. "Formation enthalpies by mixing GGA and GGA+ U calculations". *Physical Review B* 84 (4), p. 045115, 2011.

[Jai+13] 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. "The Materials Project: A materials genome approach to accelerating materials innovation". *APL Materials* 1 (1), p. 011002, 2013. ISSN: 2166532X. DOI: 10.1063/1.4812323.

[Jia+16] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. "Dynamic Filter Networks". in: *Advances in Neural Information Processing Systems 29*. ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, pp. 667–675. 2016.

[Kar+14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. "Large-scale video classification with convolutional neural networks". in: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732. 2014.

[KB15] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". in: *International Conference on Learning Representations (ICLR)*. 2015.

[KC09] B. Kang and G. Ceder. "Battery materials for ultrafast charging and discharging". *Nature* 458 (7235), p. 190, 2009.

[Kea+16] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. F. Riley. "Molecular graph convolutions: moving beyond fingerprints". *Journal of Computer-Aided Molecular Design* 30 (8), pp. 595–608, 2016.

[KF96] G. Kresse and J. Furthmüller. "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set". *Physical review B* 54 (16), p. 11169, 1996.

[KG93] A. R. Katritzky and E. V. Gordeeva. "Traditional topological indexes vs electronic, geometrical, and combined molecular descriptors in QSAR/QSPR research". *Journal of chemical information and computer sciences* 33 (6), pp. 835–857, 1993.

[Kin+18] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne. "Learning how to explain neural networks: PatternNet and PatternAttribution". in: *International Conference on Learning Representations (ICLR)*. 2018.

[KL02] R. I. Kondor and J. D. Lafferty. "Diffusion Kernels on Graphs and Other Discrete Input Spaces". in: *Proceedings of the 19th International Conference on Machine Learning*. ICML '02, pp. 315–322. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[KLK95] A. R. Katritzky, V. S. Lobanov, and M. Karelson. "QSPR: the correlation and quantitative prediction of chemical and physical properties from structure". *Chemical Society Reviews* 24 (4), pp. 279–287, 1995.

[KLK96] M. Karelson, V. S. Lobanov, and A. R. Katritzky. "Quantum-chemical descriptors in QSAR/QSPR studies". *Chemical reviews* 96 (3), pp. 1027–1044, 1996.

[KS65] W. Kohn and L. J. Sham. "Self-consistent equations including exchange and correlation effects". *Physical review* 140 (4A), A1133, 1965.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". in: *Advances in neural information processing systems*, pp. 1097–1105. 2012.

[LB+95] Y. LeCun, Y. Bengio, et al. "Convolutional networks for images, speech, and time series". *The handbook of brain theory and neural networks* 3361 (10), p. 1995, 1995.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". *Nature* 521 (7553), pp. 436–444, 2015.

[LeC+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition". *Neural computation* 1 (4), pp. 541–551, 1989.

[LeC+98] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. "Efficient backprop". in: *Neural networks: Tricks of the trade*. Springer, 1998, pp. 9–50.

[Lil+15] 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". *International Journal of Quantum Chemistry* 115 (16), pp. 1084–1093, 2015.

[Lil13] O. A. von Lilienfeld. "First principles view on chemical compound space: Gaining rigorous atomistic control of molecular properties". *International Journal of Quantum Chemistry* 113 (12), pp. 1676–1689, 2013.

[LYP88] C. Lee, W. Yang, and R. G. Parr. "Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density". *Physical review B* 37 (2), p. 785, 1988.

[Mal+09] R. Malshe M .and Narulkar, L. M. Raff, M. Hagan, S. Bukkapatnam, P. M. Agrawal, and R. Komanduri. "Development of generalized potential-energy surfaces using many-body expansions, neural networks, and moiety energy approximations". *J. Chem. Phys.* 130 (18), p. 184102, 2009.

[Mik+13a] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space". *ICLR Workshop*, 2013.

[Mik+13b] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. "Distributed representations of words and phrases and their compositionality". in: *Advances in Neural Information Processing Systems*, pp. 3111–3119. 2013.

[Mni+15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. "Human-level control through deep reinforcement learning". *Nature* 518 (7540), p. 529, 2015.

[Mon+12] 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*. ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 440–448.

[Mon+13] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, 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.* 15 (9), p. 095003, 2013.

[Mon+17] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. "Explaining nonlinear classification decisions with deep taylor decomposition". *Pattern Recognition* 65, pp. 211–222, 2017.

[Mül+01] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. "An introduction to kernel-based learning algorithms". *IEEE transactions on neural networks* 12 (2), pp. 181–201, 2001.

[Nør+09] J. K. Nørskov, T. Bligaard, J. Rossmeisl, and C. H. Christensen. "Towards the computational design of solid catalysts". *Nature chemistry* 1 (1), p. 37, 2009.

[Oor+16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". in: *9th ISCA Speech Synthesis Workshop*, pp. 125–125. 2016.

[PA91] V. Parasuk and J. Almlöf. "C20: the smallest fullerene?" *Chemical physics letters* 184 (1-3), pp. 187–190, 1991.

[PBE96] J. P. Perdew, K. Burke, and M. Ernzerhof. "Generalized gradient approximation made simple". *Physical review letters* 77 (18), p. 3865, 1996.

[PDT18] I. Poltavsky, R. A. DiStasio Jr., and A. Tkatchenko. "Perturbed path integrals in imaginary time: Efficiently modeling nuclear quantum effects in molecules and materials". *J. Chem. Phys.* 148 (10), p. 102325, 2018.

[PEB96] J. P. Perdew, M. Ernzerhof, and K. Burke. "Rationale for mixing exact exchange with density functional approximations". *J. Chem. Phys.* 105 (22), pp. 9982–9985, 1996.

[Puk+09] A. Pukrittayakamee, M. Malshe, M. Hagan, L. Raff, R. Narulkar, S. Bukkapatnum, and R. Komanduri. "Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks". *The Journal of chemical physics* 130 (13), p. 134101, 2009.

[Pyz+15] E. O. Pyzer-Knapp, C. Suh, R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, and A. Aspuru-Guzik. "What is high-throughput virtual screening? A perspective from organic materials discovery". *Annual Review of Materials Research* 45, pp. 195–216, 2015.

[PZ81] J. P. Perdew and A. Zunger. "Self-interaction correction to density-functional approximations for many-electron systems". *Phys. Rev. B* 23, pp. 5048–5079, 10 1981.

[Ram+14] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. "Quantum chemistry structures and properties of 134 kilo molecules". *Scientific Data* 1, 2014.

[Ram+15] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. "Massively multitask networks for drug discovery". *arXiv preprint arXiv:1502.02072*, 2015.

[Rey15] J.-L. Reymond. "The chemical space project". *Acc. Chem. Res.* 48 (3), pp. 722–730, 2015.

[RH10] D. Rogers and M. Hahn. "Extended-connectivity fingerprints". *Journal of chemical information and modeling* 50 (5), pp. 742–754, 2010.

[Rup+12] 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.* 108 (5), p. 058301, 2012.

[SB97] M. A. Spackman and P. G. Byrom. "A novel definition of a molecule in a crystal". *Chemical physics letters* 267 (3-4), pp. 215–220, 1997.

[Sca+09] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The graph neural network model". *IEEE Trans. Neural Netw.* 20 (1), pp. 61–80, 2009.

[Sch+14] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K.-R. Müller, and E. Gross. "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties". *Phys. Rev. B* 89 (20), p. 205118, 2014.

[Sch+17a] K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko. "Quantum-chemical insights from deep tensor neural networks". *Nature Communications* 8, 13890, 2017.

[Sch+17b] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". in: *Advances in Neural Information Processing Systems 30*, pp. 992–1002. 2017.

[Sch+18] 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* 148 (24), 241722, 2018.

[Sch15] J. Schmidhuber. "Deep learning in neural networks: An overview". *Neural networks* 61, pp. 85–117, 2015.

[Sha16] A. V. Shapeev. "Moment tensor potentials: A class of systematically improvable interatomic potentials". *Multiscale Modeling & Simulation* 14 (3), pp. 1153–1173, 2016.

[SIR17] J. S. Smith, O. Isayev, and A. E. Roitberg. "ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost". *Chemical science* 8 (4), pp. 3192–3203, 2017.

[SMH11] I. Sutskever, J. Martens, and G. E. Hinton. "Generating text with recurrent neural networks". in: *Proceedings of the 28th International Conference on Machine Learning*, pp. 1017–1024. 2011.

[Sny+12] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke. "Finding density functionals with machine learning". *Physical review letters* 108 (25), p. 253002, 2012.

[Sny+15] J. C. Snyder, M. Rupp, K.-R. Müller, and K. Burke. "Nonlinear gradient denoising: Finding accurate extrema from inaccurate functional derivatives". *International Journal of Quantum Chemistry* 115 (16), pp. 1102–1114, 2015.

[SO96] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Books on Chemistry, 1996.

[Soc+13] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. "Recursive deep models for semantic compositionality over a sentiment treebank". in: *EMNLP*. vol. 1631, p. 1642. 2013.

[SS02] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[SV03] C. Selassie and R. P. Verma. "History of quantitative structure–activity relationships". *Burger's Medicinal Chemistry and Drug Discovery*, 2003.

[SVL14] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to sequence learning with neural networks". in: *Advances in neural information processing systems*, pp. 3104–3112. 2014.

[SVZ13] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". *arXiv preprint arXiv:1312.6034*, 2013.

[Sze+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". in: *International Conference on Learning Representations*. 2014.

[Sze+16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision". in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826. 2016.

[TH09] G. W. Taylor and G. E. Hinton. "Factored conditional restricted Boltzmann machines for modeling motion style". in: *Proceedings of the 26th annual international conference on machine learning*. ACM, pp. 1025–1032. 2009.

[TS09] A. Tkatchenko and M. Scheffler. "Accurate molecular van der Waals interactions from ground-state electron density and free-atom reference data". *Physical review letters* 102 (7), p. 073005, 2009.

[VBK16] O. Vinyals, S. Bengio, and M. Kudlur. "Order matters: Sequence to sequence for sets". in: *International Conference on Learning Representations (ICLR)*. 2016.

[Vin+15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. "Show and tell: A neural image caption generator". in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 3156–3164. 2015.

[WDA16] J. N. Wei, D. Duvenaud, and A. Aspuru-Guzik. "Neural networks for the prediction of organic chemistry reactions". *ACS central science* 2 (10), pp. 725–732, 2016.

[Yao+18] K. Yao, J. E. Herr, D. W. Toth, R. Mckintyre, and J. Parkhill. "The TensorMol-0.1 Model Chemistry: a Neural Network Augmented with Long-Range Physics". *Chemical Science*, 2018.

[YDS13] D. Yu, L. Deng, and F. Seide. "The deep tensor neural network with applications to large vocabulary speech recognition". *IEEE Transactions on Audio, Speech, and Language Processing* 21 (2), pp. 388–396, 2013.

[ZF14] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". in: *European Conference on Computer Vision*. Springer, pp. 818–833. 2014.

Vorveröffentlichungen und Eigenanteile

Publikation:
K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K.-R. Müller und E. Gross. "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties". *Phys. Rev. B* 89 (20), S. 205118, 2014

Die Hauptbeiträge zu diesem Artikel stammen zu gleichem Anteil von mir und Henning Glawe, der die Daten generiert und den Physikhintergrund beigetragen hat. Ich habe den PRDF-Deskriptor entwickelt, der in diesem Artikel eingeführt wurde, die Modelle trainiert und ausgewertet sowie einen Teil der Abbildungen erstellt. Alle Autoren haben die Ergebnisse diskutiert und zum finalen Text beigetragen.

Publikation:
K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller und A. Tkatchenko. "Quantum-chemical insights from deep tensor neural networks". *Nature Communications* 8, 13890, 2017

Ich habe die *Deep Tensor Neural Network*-Architektur für Molekülvorhersagen entwickelt, die Modelle trainiert und ausgewertet sowie die weiteren Analysen ausgeführt. Die Theorie wie die Netzarchitektur die Quantenchemie reflektiert habe ich gemeinsam mit F. Arbabzadah, A. Tkatchenko und K.-R. Müller erarbeitet. Weiterhin habe ich die Abbildungen erstellt und große Teile des Textes geschrieben. Alle Autoren haben die Ergebnisse diskutiert und zum finalen Text beigetragen.

Publikation:
K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko und K.-R. Müller. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". in: *Advances in Neural Information Processing Systems 30*, S. 992–1002. 2017

Ich habe das Neuronale Netz *SchNet* und die *Continuous-Filter Convolutional Layers* entwickelt, die Experimente durchgeführt sowie die Abbildungen erstellt. Weiterhin habe ich einen Großteil des Textes geschrieben. Den Aufbau des Artikels sowie die Auswahl der Datensätze und Experimente habe ich gemeinsam mit P.-J. Kindermans erarbeitet. H. E. Sauceda hat den ISO17-Datensatz für diesen Artikel erstellt. Alle Autoren haben die Ergebnisse diskutiert und zum finalen Text beigetragen.

Publikation:
K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko und K.-R. Müller. "SchNet - a deep learning architecture for molecules and materials". *The Journal of Chemical Physics* 148 (24), 241722, 2018

Dieser Artikel ergänzt den NIPS-Artikel zur *SchNet*-Architektur um weitere Experimente und Analysen. Ich habe alle Modelle trainiert sowie die Ergebnisse für die Molekül- und Materialvorsagen ausgewertet. Mithilfe einer von mir entwickelten Python-Schnittstelle für Moleküldynamiksimulationen (MD) konnte H. E. Sauceda auf SchNet basierende MD-Trajektorien von dem Fulleren C_{20} generieren und auswerten. Weiterhin habe ich große Teile des Textes geschrieben sowie die Abbildungen erstellt. Alle Autoren haben die Ergebnisse diskutiert und zum finalen Text beigetragen.