universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Steinhardt Bond Order Parameters for Neural Network Potentials"

verfasst von / submitted by

Igor Kraut, BSc, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2018/ Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 876
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Physik
Betreut von / Supervisor:	Univ.-Prof. Mag. Dr. Christoph Dellago

"In all disciplines in which there is systematic knowledge of things with principles, causes, or elements, it arises from a grasp of those: we think we have knowledge of a thing when we have found its primary causes and principles, and followed it back to its elements. Clearly, then, systematic knowledge of nature must start with an attempt to settle questions about principles. "

Aristotle, Physics Book I

Abstract

In this work the implementation of a new type of symmetry functions as input for a neural network potential (NNP) is studied. Instead of Cartesian coordinates, symmetry functions are utilized since they fulfill certain invariances. Two types of symmetry functions are used for the NNP: radial and angular symmetry functions. In this work the angular symmetry functions are replaced by Steinhardt bond order parameters (SBOP), which are widely applied descriptors of the local molecular environment in physics and chemistry. To date, the used angular symmetry functions have the drawback that the search for the best parameters is tedious and the their functional form varies significantly. This changes when using SBOPs: A generic set of different orders of SBOPs which have the same functional form, is enough to reproduce the local angular distribution of neighbors. It is found that in 3d visualization odd orders of SBOPs show rotational symmetries. This coincides with the fact that the odd order SBOPs vanish in perfect cubic lattices. Nevertheless, it was shown that SBOPs for the application in the NNP are not as effective in the descriptive nature as the angular symmetry functions. The results are consistently associated with larger deviations from the reference model, which is in all cases mW-water. Although, SBOPs are extremely effective in finding ordered structures, i.e. lattices, they lack the ability to efficiently describe liquid phases. This is manifested in the results of molecular dynamics simulations driven by a SBOP NNP where the radial distribution function shows strong deviations from the reference simulation. Furthermore, the measured diffusion coefficient is underestimated by a factor of 2.

Contents

1 Introduction

Neural networks are becoming increasingly interesting for scientific purposes, which includes molecular dynamics simulations (MD). Neural networks are utilized for the calculation of potential energy surfaces and thus can be the driver of MD simulations. This means that a neural network is capable of learning from a reference model and able to calculate the forces on the particles in MD simulations as well as calculating the energy of the configuration. A so-called neural network potential (NNP) can learn from a reference model how the particles interact between each other. Instead of the computationally expensive direct application of the reference model, the neural network is applied for the calculation of the forces. The trained NNP requires information about the atomic configuration, in order to calculate the forces between the particle. It is not possible to use Cartesian coordinates as direct input for the neural network, since special symmetries like rotational invariance, translation invariance and particle permutation invariance have to be fulfilled by the input quantities. For this reason so-called symmetry functions are utilized to describe the local molecular environment. There are two types of symmetry functions: radial symmetry functions and angular symmetry functions. For a good description of the environment of each particle a set of symmetry functions with different parametrization is used. The first type of symmetry functions contains information about the radial distribution of particles around a center particle within a certain cutoff radius. On the other hand, the angular symmetry functions describe the angular distribution of neighboring particles.

The task of this master's thesis is to test the applicability of a new type of symmetry functions, the so-called Steinhardt bond order parameters (SBOP) to replace the angular symmetry functions. This work begins with an introduction to molecular dynamics, which includes presenting algorithms used to solve the Newtonian equations of motion. Additionally, quantities like the diffusion coefficient and the radial distribution function (RDF) are introduced. They will be calculated from MD simulations later in the thesis. In the following a short introduction to ab-initio methods for the calculation of potential energy surfaces is given. A detailed discussion of SBOPs and their application in computational science is portrayed in the subsequent section. The usage of SBOPs in the determination of crystal structures as well as driver of nucleation processes is presented. A detailed review on neural

network potentials (NNP) for the calculation of potential energy surfaces in molecular dynamics provides insight into the methods necessary to understand the purpose of this work. This section is followed by a review of commonly used optimization algorithms in neural networks.
Finally the results of this work are presented which includes the 3d visualization of the SBOPs, which was to the authors knowledge never done before. This visualization indicates that there is a significant difference between SBOPs of even and odd orders. A direct comparison of learning curves where both the angular symmetry functions and SBOPs were applied to the NNP is conducted. Finally, molecular dynamics simulations for both these methods and the reference model are performed The focus lies on the ability to reproduce the radial distribution function and the diffusion coefficient of the reference model.

2 Molecular Dynamics

Molecular dynamics is the a numerical method to reproduce particle dynamic on atomic scale to mimic real experiments by the laws of classical mechanics. Via simplified models the forces of a configuration of N particles are computed and the Newtonian equations of motion are solved.
The observables in experiments are not the positions of the single atoms but averaged thermodynamical quantities. However, statistical mechanics provides us the means to express the thermodynamical quantities as function of the microscopic quantities. A fundamental macroscopic quantity is the temperature T of the particle system, which is in relation to the average kinetic energy of the particles:

$$\left\langle \frac{1}{2} m v^2 \right\rangle = \frac{1}{2} k_B T \tag{1}$$

Here, m is the particles mass, v is the particles absolute velocity and k_B is the Boltzmann constant. This expression can be rearranged to express the temperature of the system

$$T(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{k_k N} \tag{2}$$

This only holds under the assumption that the systems particle number N is high. A model predicts the forces on the particles and the Newtonian equations of motion have to be solved in order to progress the system in time.

2.1 Equations of Motion

The forces can be calculated for each particle by forming the negative gradient of the potential ($\mathbf{F} = -\nabla U$), which is determined by the used model. A typical potential often encountered in physics is the Lennard-Jones potential ([13]):

$$U(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{3}$$

Here, r is the distance between a particle pair, σ is the distance for which the potential is zero and ϵ is the depth of the potential. Newton's second law (equation 4) provides the link between the force on the particle and its acceleration $\mathbf{a}$:

$$\mathbf{F}(t) = m\mathbf{a}(t) = m\dot{\mathbf{v}}(t) \tag{4}$$

Hence, the set of equations, which has to be solved in order to progress the system in time, is given by equations 5 and 6:

$$\dot{\mathbf{v}}_i(t) = \frac{\mathbf{F}_i(t)}{m_i} \tag{5}$$

$$\dot{\mathbf{r}}_i(t) = \mathbf{v}_i(t) \tag{6}$$

This equations have to be integrated in parallel to advance the system in time.
A further way of finding the equations of motion is via the Hamilton equations. The Hamiltonian of a N-body system is given by the system's total energy:

$$H(\mathbf{r}^N, \mathbf{p}^N) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}^N). \tag{7}$$

Here N is the number of particles in the system, $\mathbf{r}^N$ are the positions of all particles, $\mathbf{p}^N$ are the momenta of all particles and $U(\mathbf{r}^N)$ is the potential energy. The Hamiltonian equations of motion can be formed as partial derivatives of the coordinates and canonical momenta:

$$\dot{\mathbf{r}}_i = \frac{\partial H}{\partial \mathbf{p}_i} \tag{8}$$

$$\dot{\mathbf{q}}_i = -\frac{\partial H}{\partial \mathbf{q}_i} \tag{9}$$

This leads to the exact same set of equations as the Newtonian equations of motion (equations 5 and 6). The Hamiltonian is an integral of motion, which is equivalent to energy conservation:

$$\frac{dH}{dt} = \sum_{i=1}^{N} \frac{\partial H}{\partial \mathbf{r}_i}\mathbf{r}_i + \frac{\partial H}{\partial \mathbf{p}_i}\mathbf{p}_i = \sum_{i=1}^{N} \frac{\partial H}{\partial \mathbf{r}_i}\frac{\partial H}{\partial \mathbf{p}_i} - \frac{\partial H}{\partial \mathbf{p}_i}\frac{\partial H}{\partial \mathbf{r}_i} = 0 \tag{10}$$

2.2 Integration of Equations of Motion

In general, it is not possible to solve the equations of motions for the N-body system analytically. For this reason numerical finite difference methods can be utilized, where successive integrations for a small time step Δt are applied subsequently. This only works, if the error of this approximations is reduced with a smaller time step Δt. The approaches of solving this problem are called integration algorithms.

2.2.1 Verlet Algorithm

The Verlet algorithm which is named after Loup Verlet ([35]) can be derived by a Taylor expansion of the particles position at time t:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{F}(t)}{2m}\Delta t^2 + \frac{\dddot{r}}{3!}\Delta t^3 + O(\Delta t^4) \tag{11}$$

Analogically the reversed time step can be expressed via a Taylor series:

$$\mathbf{r}(t-\Delta t) = \mathbf{r}(t) - \mathbf{v}(t)\Delta t + \frac{\mathbf{F}(t)}{2m}\Delta t^2 - \frac{\dddot{r}}{3!}\Delta t^3 + O(\Delta t^4) \tag{12}$$

The sum of the equations 11 and 12 gives following expression:

$$\mathbf{r}(t+\Delta t) + \mathbf{r}(t-\Delta t) = 2\mathbf{r}(t) + \frac{\mathbf{F}(t)}{2m}\Delta t^2 + O(\Delta t^4) \tag{13}$$

Rearranging to $\mathbf{r}(t+\Delta t)$ gives the Verlet algorithm:

$$\mathbf{r}(t+\Delta t) \approx 2\mathbf{r}(t) - \mathbf{r}(t-\Delta t) + \frac{\mathbf{F}(t)}{2m}\Delta t^2 \tag{14}$$

The Verlet algorithm is accurate up to the order Δt^3. This means that the error is of the order $O(\Delta t^4)$. Furthermore, the velocities are not needed for the computation of the next positions. However, the velocities can be calculated by subtracting equation 12 from equation 11:

$$\mathbf{r}(t+\Delta t) - \mathbf{r}(t-\Delta t) = 2\mathbf{v}(t)\Delta t + O(\Delta t^3) \tag{15}$$

which can be rewritten in terms of the velocity:

$$\mathbf{v}(t) = \frac{\mathbf{r}(t+\Delta t) - \mathbf{r}(t-\Delta t)}{2\Delta t} + O(\Delta t^2) \tag{16}$$

The velocitie's accuracy is reduced by one order due to the division by Δt.
The advantages of the Verlet algorithm are its simplicity, that only the forces and positions are required for the calculation of the next time step and that there is no long term energy drift. Also, time reversibility is provided by the Verlet algorithm. On the other hand, the Verlet algorithm is not self starting since for the first time step also the prior position of the particle are required.

2.2.2 Leapfrog Algorithm

A further popular integration scheme in molecular dynamics is the leapfrog algorithm. The Verlet algorithm has the disadvantage that the velocities are only accurate up to order Δt. The accuracy can be improved by using the leapfrog algorithm, witch calculates the velocities at half integer steps. The velocities at this half time steps are given by equations 17 and 18.

$$\mathbf{v}\left(t - \frac{\Delta t}{2}\right) \approx \frac{\mathbf{r}(t) - \mathbf{r}(t-\Delta t)}{\Delta t} \tag{17}$$

$$\mathbf{v}(t + \frac{\Delta t}{2}) \approx \frac{\mathbf{r}(t+\Delta t) - \mathbf{r}(t)}{\Delta t} \tag{18}$$

The positions can now be expressed with the help of this half time step:

$$\mathbf{r}\left(t + \Delta t\right) \approx \mathbf{r}(t) + \mathbf{v}\left(t + \frac{\Delta t}{2}\right)\Delta t \tag{19}$$

$$\mathbf{r}(t - \Delta t) \approx \mathbf{r}(t) + \mathbf{v}\left(t - \frac{\Delta t}{2}\right)\Delta t \tag{20}$$

Finally the Verlet algorithm (equation 14) is utilized and the position differences are substituted with the newly derived equations:

$$\mathbf{r}(t+\Delta t) - \mathbf{r}(t) \approx \mathbf{r}(t) - \mathbf{r}(t-\Delta t) + \frac{\mathbf{F}(t)}{m}\Delta t^2 \tag{21}$$

Applying equations 19 and 20 results in the expression for the velocities of the leapfrog algorithm:

$$\mathbf{v}\left(t + \frac{\Delta t}{2}\right) \approx \mathbf{v}\left(t - \frac{\Delta t}{2}\right) + \frac{\mathbf{F}(t)}{m}\Delta t^2 \tag{22}$$

2.2.3 Velocity Verlet Algorithm

Another important integration algorithm is the velocity Verlet algorithm. Again, the next time step of the position $\mathbf{r}(t + \Delta t)$ is expressed via a Taylor expansion at the time t:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{F}(t)}{2m}\Delta t^2 + O(\Delta t^3) \tag{23}$$

In addition $\mathbf{r}(t)$ can be written as follows:

$$\mathbf{r}(t) = \mathbf{r}(t + \Delta t) - \mathbf{v}(t + \Delta t)\Delta t + \frac{\mathbf{F}(t + \Delta t)}{2m}\Delta t^2 + O(\Delta t^3) \tag{24}$$

When equation 23 is substituted by equation 24 one yields:

$$\mathbf{r}(t) = \mathbf{r}(t) + \mathbf{v}(t) + \frac{\mathbf{F}(t)}{2m} - \mathbf{v}(t + \Delta t)\Delta t + \frac{\mathbf{F}(t + \Delta t)}{2m}\Delta t^2 \tag{25}$$

The terms $\mathbf{r}(t)$ cancel out and the velocity can be written as follows:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{F}(t) + \mathbf{F}(t + \Delta t)}{2m}\Delta t + O(\Delta t^3) \tag{26}$$

In the velocity Verlet algorithm the positions are accurate up to the order Δt^3 while the velocities are accurate up to the order Δt^2.

2.3 Nosé-Hoover Thermostat

Typically, molecular dynamics simulations operate in the micro-canonical ensemble where the energy is constant. So-called thermostats can be applied in order to simulate a canonical ensemble with constant temperature. Popular methods are the Anderson thermostat ([1]), Langevin dynamics ([31], p 480), isokinetic molecular dynamics ([23]) and the Nosé thermostat ([27]) or its modification the Nosé-Hoover thermostat ([27]).

Here the Nosé-Hoover thermostat is reviewed. The idea of Nosé was to modify the

equations of motion is such a way, that the canonical ensemble is sampled. The modified Hamiltonian has the following form:

$$H_N(p^N, r^N, p_s, s) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i s^2} + U(r^N) + \frac{p_s^2}{2Q} + g k_B T \ln(s) \tag{27}$$

Here, s is scaling factor treated as additional dynamical variable, p_s is the corresponding conjugate momentum, Q can be interpreted as reactivity parameter and g is a free parameter chosen in such a way that the canonical ensemble is sampled. The microcanonical distribution of the Nosé Hamiltonian $H_N(p^N, r^N, p_s, s)$ is corresponding to the canonical distribution with the modified variables p^N/s and r^N for the Hamiltonian $H(p^N/s, r^N)$. The corresponding equations of motion are:

$$\dot{\mathbf{r}}_i = \frac{\partial H_N}{\partial \mathbf{p}_i} = \frac{\mathbf{p}_i}{m_i s^2} \tag{28}$$

$$\dot{\mathbf{p}}_i = -\frac{\partial H_N}{\partial \mathbf{r}_i} = \mathbf{F}_i \tag{29}$$

$$\dot{s} = \frac{\partial H_N}{\partial p_s} = \frac{p_s}{Q} \tag{30}$$

$$\dot{p}_s = -\frac{\partial H_N}{\partial s} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i s^3} - \frac{g k_B T}{s} = \frac{1}{s}\left[\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i s^2} - g k_B T\right] \tag{31}$$

The only difference in the equation of motion is the factor s^2 in the positions. The variables p^N, r^N, p_s and s are corresponding to canonical variables for transformed momenta:

$$\mathbf{p}_i' = \frac{\mathbf{p}_i}{s}, \tag{32}$$

$$p_s' = \frac{p_s}{s}. \tag{33}$$

Applying the transformations leads to the new equations of motion:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{s}\frac{1}{m_i s} = \frac{\mathbf{p}_i'}{m_i s} \tag{34}$$

$$\begin{aligned} \dot{\mathbf{p}}_i = \frac{d}{dt}\frac{\mathbf{p}_i'}{s} &= \frac{1}{s}\frac{d\mathbf{p}_i}{dt} + \mathbf{p}_i \frac{d}{dt}\frac{1}{s} = \frac{\mathbf{F}_i}{s} - \frac{\mathbf{p}_i}{s^2}\dot{s} \\ &= \frac{\mathbf{F}_i}{s} - \frac{\mathbf{p}_i}{s^2}\frac{p_s}{Q} = \frac{\mathbf{F}_i}{s} - \mathbf{p}_i'\frac{p_s'}{Q} \end{aligned} \tag{35}$$

$$\dot{s} = \frac{p_s}{Q} = \frac{s p_s'}{Q} \tag{36}$$

$$
\begin{aligned}
\dot{p}'_s = \frac{d}{dt}\frac{p_s}{s} &= \frac{1}{s}\frac{dp_s}{dt} - p_s\frac{\dot{s}}{s^2} \\
&= \frac{1}{s^2}\left[\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i s^2} - \frac{gk_BT}{s}\right] - \frac{p_s}{s}\frac{sp'_s}{Q} \\
&= \frac{1}{s^2}\left[\sum_{i=1}^{N}\frac{\mathbf{p}_i'^2}{2m_i} - gk_BT\right] - \frac{p_s'^2}{Q}
\end{aligned}
\tag{37}
$$

These are the Nosé equations of motion. However, they can be simplified by introducing a friction variable $\zeta = \frac{sp'_s}{Q}$ which leads to the Nosé-Hoover equations of motion:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} \tag{38}$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \zeta\mathbf{p}_i \tag{39}$$

$$\dot{\zeta} = \frac{1}{Q}\left[\sum_i \frac{\mathbf{p}_i^2}{m_i} - 3Nk_BT\right] \tag{40}$$

$$\dot{\ln s} = \zeta \tag{41}$$

The associated Hamiltonian is given by:

$$H_{NH}(r^N, p^N, \zeta, s) = \sum_i \frac{\mathbf{p}_i^2}{2m_i} + U(r^N) + \frac{\zeta^2 Q}{2} + 3Nk_BT\ln s \tag{42}$$

Since the Hamiltonian is a conserved quantity it can be used to test the correct implementation of the thermostat.

2.4 Radial Distribution Function

The radial distribution function (RDF) or pair correlation function $g(r)$ describes the distribution of distances between particles in an atomic system. It can be defined via the number of particles $dn(r)$ found in an infinitesimally small spherical shell centered around a particle witch thickness dr at the radius r.

$$dn(r) = \frac{N}{V}g(r)4\pi r^2 dr \tag{43}$$

The term N/V represents the average density of the system. Figure 1 shows the typical form of the RDFs for three different phases of matter:

- Solid phase: The main characteristics of the RDF of solid matter are strong peaks and valleys. This is due to the ordered structure of solids. The positions of the atoms in crystals are fixed and only small fluctuations around their average position occur.

- Liquid phase: The RDF of liquids shows typically only several peaks quickly decaying with distance. The first peak occurs at $R \approx \sigma$ and further ones approximately at multiples of σ. The RDF quickly reaches the mean density with the value $g(r) = 1$.

- Gas phase: Typical for a gas RDF is that only a single peak can be found which decays to the value $g(r) = 1$ without deceeding it again. No further peaks occur since no regular structure exists in the gas phase.

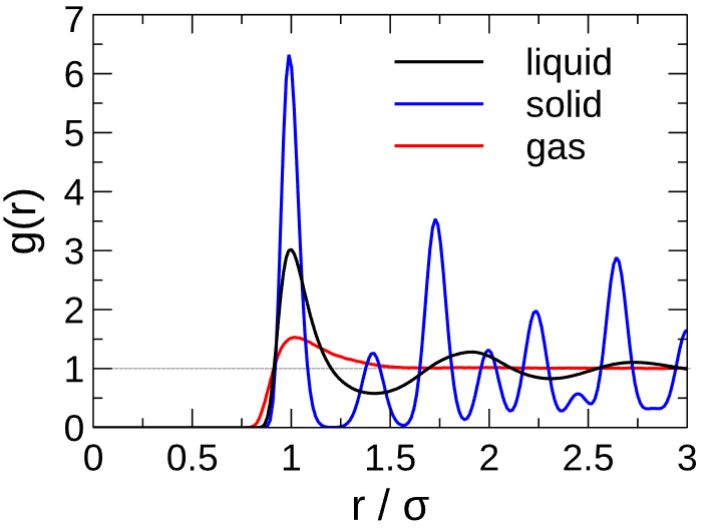


Figure 1: The form of the radial distribution function $g(r)$ is depending on the phase of the system. The depicted system is Argon. Typically, the solid phase shows very distinctive peaks, whereas for liquids only several unremarkable peaks are found, which are quickly decaying with larger radii. The RDF of a gas shows only a single peak. Figure taken from: `https://bit.ly/2zitV1e`, last time accessed on 28-07-2018.

2.5 Diffusion Coefficient

The process of equalization of concentration differences is called diffusion. The process is described by Fick's first law (equation 44), which states that the diffusion constant D is the proportionality factor between the flux j and the concentration gradient ∇c. The following derivation is based on [9].

$$\mathbf{j} = -D\nabla c \tag{44}$$

Fick's law and additional utilization of the continuity equation (45) leads to the diffusion equation (46).

$$\frac{\partial c}{\partial t} + \nabla \cdot \mathbf{j} = 0 \tag{45}$$

$$\frac{\partial c}{\partial t} - D\nabla^2 c = 0 \tag{46}$$

It holds that the squared displacement can be expressed via the following equation if the concentration is normalized (equation 48):

$$\left\langle r^2(t) \right\rangle = \int d\mathbf{r} c r^2 \tag{47}$$

$$\int d\mathbf{r} c = 1 \tag{48}$$

The multiplication of the diffusion equation with r^2 and an integration over $d\mathbf{r}$ yields:

$$\frac{\partial}{\partial t} \int d\mathbf{r} c(\mathbf{r}, t) r^2 = D \int d\mathbf{r} r^2 \nabla^2 c(\mathbf{r}, t) \tag{49}$$

The left-hand side of the equation 49 can be rewritten with equation 47 and the right-hand side can be solved directly.

$$\begin{aligned}
\frac{\partial \left\langle r^2(t) \right\rangle}{\partial t} &= D \int d\mathbf{r} r^2 \nabla^2 c(\mathbf{r}, t) \\
&= D \int d\mathbf{r} \nabla \cdot (r^2 \nabla c(\mathbf{r}, t) - D \int d\mathbf{r} \nabla r^2 \cdot \nabla c(\mathbf{r}, t) \\
&= D \int d\mathbf{S} (r^2 \nabla c(\mathbf{r}, t)) - 2D \int d\mathbf{r} \nabla c(\mathbf{r}, t) \\
&= 0 - 2D \int d\mathbf{r} (\nabla \cdot \mathbf{r} c(\mathbf{r}, t)) + 2D \int d\mathbf{r} (\nabla \cdot \mathbf{r}) c(\mathbf{r}, t) \\
&= 0 + 6D \int d\mathbf{r} c(\mathbf{r}, t) \\
&= 6D
\end{aligned} \tag{50}$$

The expression $\langle \Delta \mathbf{r}(t)^2 \rangle$ can be interpreted as the mean squared distance that the molecules move for a time interval of length t. Microscopically this quantity is

expressed by following equation:

$$\left\langle \Delta\mathbf{r}(t)^2 \right\rangle = \frac{1}{N}\sum_{i=1}^{N} \Delta\mathbf{r}(t)^2 \tag{51}$$

The displacement can be written as integral of the particles velocity over the simulation time:

$$\Delta\mathbf{r}(t) = \int_0^t \mathbf{v}(t')dt' \tag{52}$$

Let consider only one dimension which implies that $\frac{\partial\left\langle x^2(t)\right\rangle}{\partial t} = 2D$ holds. We can now express the position x as integrals of the velocity v_x and obtain following equation:

$$\begin{aligned}
\left\langle x^2(t)\right\rangle &= \left\langle \left(\int_0^t v_x(t')dt' \right)^2 \right\rangle \\
&= \int_0^t \int_0^t \left\langle v_x(dt')v_x(dt'')\right\rangle dt'dt'' \\
&= 2\int_0^t \int_0^{t'} \left\langle v_x(dt')v_x(dt'')\right\rangle dt'dt''
\end{aligned} \tag{53}$$

Hence, the mean squared displacement can be written as integral of the velocity auto correlation function. The autocorrelation function is depending only on the time differences:

$$\left\langle v_x(t')v_x(t'')\right\rangle = \left\langle v_x(t'-t'')v_x(0)\right\rangle, \tag{54}$$

which means that the diffusion coefficient takes following form:

$$D = \lim_{t\to\inf}\int_0^t \left\langle v_x(t-t'')v_x(0)\right\rangle. \tag{55}$$

The expression $t'-t''$ can be substituted by τ:

$$D = \int_0^{\inf} d\tau \left\langle v_x(\tau)v_x(0)\right\rangle \tag{56}$$

Equation 56 is a so-called Green-Kubo relation ([11], [17]) which connects a transport coefficient with the correlation functions of microscopic quantities of the system.

3 Ab-initio Methods

Molecular dynamics is utilizing simplified models for the calculation of the forces. These approximative classical models give good estimates for many phenomena in the classical domain. However, quantum effects play a great role in N-body particle interaction. Thus, electronic structure calculations have great importance since they are capable of accurately describing non-classical phenomena like chemical bonding processes. This section follows the review article of Kühne [18].
In ab-initio molecular dynamics the forces (see equation 57) are calculated by finding the ground state eigen-fuctions $|\psi_0\rangle$ of the N-body Hamiltonian for the electrons. The only information provided are the positions and the charge of the atoms in the configuration.

$$\mathbf{F}_i = -\nabla_{\mathbf{r}}\Phi(\mathbf{r}) \tag{57}$$

In the Born-Oppenheimer approximation the potential $\Phi(\mathbf{r})$ can be written as follows:

$$\Phi(\mathbf{r}) = \langle\psi_0|H_e(\{\mathbf{R}_i\};\mathbf{r})|\psi_0\rangle + E_{II}(\mathbf{r}) \tag{58}$$

In this equation $H_e(\{\mathbf{R}_i\};\mathbf{r})$ is the electronic many body Hamiltonian, which is a function of the configuration $\{\mathbf{R}_i\}$, as well as the position $\mathbf{r}$ and E_{II} is the nuclear potential. The mass difference between the electrons and nuclei allows to assume an instantaneous equilibrium of the electrons in respect to the nuclei.Therefore, it possible to write an eigen-equation in the following form:

$$H_e(\{\mathbf{R}_i\};\mathbf{r})\psi_0(\{\mathbf{R}_i\}) = \epsilon_0(\mathbf{R})\psi_0(\{\mathbf{R}_i\}) \tag{59}$$

Here, $\psi_0(\{\mathbf{R}_i\})$ are the eigen-functions and $\epsilon_0(\mathbf{R})$ are the eigen-values. However, to solve this eigenvalue-problem typically further simplifications are needed.

3.1 Density Functional Theory

The density functional theory provides the means to reduce the problems complexity by using the Hohenberg-Kohn theorem (HW) which proves that the potential $v(\mathbf{r})$ and the wave functions are unambiguously related to the electronic density $\rho(\mathbf{r})$. The N-body Hamiltonian as seen in equation 60 consists of three contributions:

$$H_e = \frac{1}{2}\sum_{i=1}^{N_e}\nabla_i^2 + \sum_{i<j}^{N_e}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{I,i}^{N,N_e}\frac{Z_I}{|\mathbf{r}_I - \mathbf{r}_i|} = \hat{T} + \hat{U} + \hat{V} \tag{60}$$

N_e is the number of electrons, $\hat{T}$ is the kinetic energy operator of the electrons, $\hat{U}$ is the electron-electron interaction operator and $\hat{V}$ is the electron-nucleus interaction operator. Z_I is the proton number of the I-th nucleus. To simplify the equations atomic units are used. The energy can now be written as:

$$E^{DFT}[\rho(\mathbf{r})] = \langle\psi|H_e|\psi\rangle = T[\rho(\mathbf{r})] + U[\rho(\mathbf{r})] + V[\rho(\mathbf{r})]. \tag{61}$$

In the next step the Thomas-Fermi approximation is applied, which estimated the interaction energy between the electrons with the Hartree energy as seen in equation 62.

$$U[\rho(\mathbf{r})] \approx U_H[\rho(\mathbf{r})] = \frac{1}{2}\int d\mathbf{r}\int d\mathbf{r}'\frac{\rho(\mathbf{r})\rho(\mathbf{r})'}{\mathbf{r}-\mathbf{r}'} \tag{62}$$

To this point many body correlation effects are neglected. Nevertheless the corrections can be expressed analytically as density functionals, $E_{XC} = E_{XC}[\rho(\mathbf{r})]$.

4 Steinhardt Bond Order Parameter

One necessary mathematical function should be discussed- the spherical harmonics - before introducing the main tool of this work, the Steinhardt bond order parameters.

4.1 Spherical Harmonics

Spherical harmonics are special functions which form a complete set of orthogonal functions and can therefore be applied to describe functions on a sphere's surface. Spherical harmonics are the solutions of the angular part of the Laplace equation (equation 63) in spherical coordinates.

$$\Delta f(r,\theta,\varphi) = 0 \tag{63}$$

The Laplace operator can be separated into a radial part and an angular part:

$$\Delta = \frac{\partial^2}{\partial r^2} + \frac{2}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\left(\frac{\partial^2}{\partial\theta^2} + \frac{\cos\theta}{\sin\theta}\frac{\partial}{\partial\theta} + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\varphi^2}\right) = \Delta_r + \frac{1}{r^2}\Delta_{\theta,\varphi} \tag{64}$$

The eigenvalue equation of the Laplace equation's angular part is solved by the spherical harmonics:

$$\left(\frac{\partial^2}{\partial\theta^2} + \frac{\cos\theta}{\sin\theta}\frac{\partial}{\partial\theta} + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\varphi^2}\right)Y_{lm}(\theta,\varphi) = -l(l+1)Y_{lm}(\theta,\varphi) \tag{65}$$

The spherical harmonics are defined as follows:

$$Y_{lm}(\theta,\varphi) = \frac{1}{\sqrt{2\pi}} N_{lm} P_{lm}(\cos\theta) e^{im\varphi}, \tag{66}$$

where

$$N_{lm} = \sqrt{\frac{2l+1}{2}\frac{(l-m)!}{(l+m)!}} \tag{67}$$

and P_{lm} are the associated Legendre polynomials defined in equation 68.

$$P_{ml}(x) = \frac{(-1)^m}{2^l l!}(1-x^2)^{m/2}\frac{d^{l+m}}{dx^{l+m}}(x^2-1)^l \tag{68}$$

4.1.1 Properties of the Spherical Harmonics

Spherical harmonics have unique properties, which make them very convenient to handle:

- Completeness:

$$\sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y^*_{lm}(\theta',\varphi')Y^*_{lm}(\theta,\varphi) = \delta(\theta-\theta')\delta(\varphi-\varphi') \tag{69}$$

- Orthogonality:

$$\int_0^{2\pi}\int_0^{\pi} Y^*_{lm}(\theta,\varphi)Y^*_{l'm'}(\theta,\varphi) = \delta_{ll'}\delta_{mm'} \tag{70}$$

- Parity:

$$Y_{lm}(\pi-\theta,\pi+\varphi) = (-1)^l Y_{lm}(\theta,\varphi) \tag{71}$$

- Addition Theorem: Consider two unit vectors in spherical coordinates. The direction of this vectors is given by the angles θ,φ and θ',φ'. The angle between the vectors is denoted by γ. There is trigonometric addition identity: $\cos(\gamma) = \cos\theta'\cos\theta + \sin\theta\sin\theta\cos(\varphi-\varphi')$. Analogically there is an addition theorem for spherical harmonics:

$$P_l(cos(\gamma)) = \frac{4\pi}{2l+1}\sum_{m=-l}^{l} Y_{lm}(\theta,\varphi)Y^*_{lm}(\theta',\varphi'). \tag{72}$$

  P_l is the Legendre polynomial of order l. In the special case of $\gamma = 0$ this leads to

$$\sum_{m=-l}^{l} |Y_{lm}(\theta,\varphi)|^2 = \frac{2l+1}{4\pi}. \tag{73}$$

  For details see [2].

4.2 Steinhardt Bond Order Parameter

Steinhardt et al. ([32]) developed a method for measuring the local and extended orientational symmetries in computational simulations. The idea is to assign a set of spherical harmonics to every bond of an atom to its neighbors. A bond is defined as a line between an atom and its neighbor, not to be confused with the actual chemical bond. The value $q_{lm}(\mathbf{r}_i)$ (equation 74) is formed via the spherical harmonics, where the orientational angles θ and ϕ describe the orientation of aforementioned bond with the i-th neighbor atom.

$$q_{lm}(\mathbf{r}_i) = Y_{lm}(\theta(\mathbf{r}_i), \phi(\mathbf{r}_i)) \tag{74}$$

The order parameter q_{lm} defined in equation 75 sums up the contribution of all neighbors. Here, N_n denotes the number of neighbors and q_{lmi} is the bond order parameter for the i-th neighbor.

$$q_{lm} = \frac{1}{N_n} \sum_{i=1}^{N_n} q_{lm}(\mathbf{r}_i) \tag{75}$$

One major problem with q_{lm} is that the value itself depends on the coordinate system to which one the angles of the bond are measured to relatively. For this reason, it is useful to form rotationally invariant linear combinations of the q_{lm}s. One possible combination is seen in equation 76, which is the so-called Steinhardt bond order parameter (SBOP).

$$q_l = \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |q_{lm}|^2 \right)^{1/2} \tag{76}$$

For certain applications the averaged Steinhardt bond order parameter $\overline{q}_l$ ([19]) can be useful, which is defined in equation 77.

$$\overline{q}_l = \frac{1}{N} \sum_{k=0}^{N} q_{lm}(k) \tag{77}$$

The averaged quantity $\overline{q}_l$ holds not only information of the first shell of neighbors like q_l, but also takes the second shell into account. This property is advantageous when long-range molecular structures are of interest.
A further invariant combination of bond order parameters is seen in equation 78.

$$w_l = \frac{\sum_{m_1+m_2+m_3=0} \begin{bmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{bmatrix} q_{lm_1} q_{lm_2} q_{lm_3}}{\left(\sum_{m=-l}^{l} |q_{lm}|^2 \right)^{3/2}} \tag{78}$$

Also this quantity can be formed with the averaged values of q_{lm}:

$$\overline{w}_l = \frac{\sum_{m_1+m_2+m_3=0} \begin{bmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{bmatrix} \overline{q}_{lm_1} \overline{q}_{lm_2} \overline{q}_{lm_3}}{\left(\sum_{m=-l}^{l} |\overline{q}_{lm}|^2 \right)^{3/2}} \tag{79}$$

The values m_1, m_2 and m_3 in the sum have a range from $-l$ up to l and only terms fulfilling $m_1 + m_2 + m_3 = 0$ are considered. Furthermore, the denominator ensures that the result is independent of the magnitude of q_{lm}.
Via the Landau theory is even possible to express the structure factor $S(\mathbf{q})$ with the help of bond order parameters as seen in equation 80. Here, r_T is a temperature dependent parameter, γ describes the strength of the coupling and $\mathbf{q}$ describes the lattice vector,

$$S(\mathbf{q}) = \frac{k_B T}{r_T + 2\gamma \sum_m Y^*_{6m}(|\theta_\mathbf{q}, \phi_\mathbf{q}) \langle Q_{6m} \rangle}. \tag{80}$$

4.3 Determination of Crystal Structure

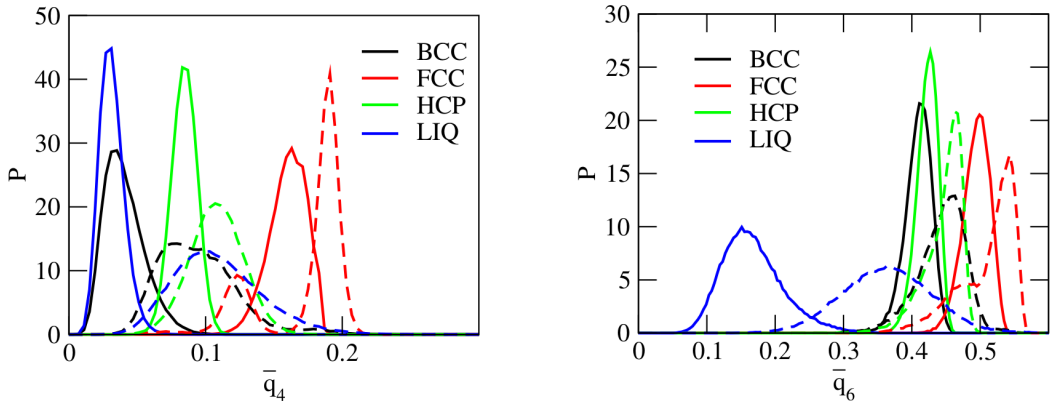
A possible application of the Steinhardt bond order parameters is the determination of crystal structures in molecular dynamical simulations. This method introduced by Steinhardt et al. ([32]) has been improved by Lechner and Dellago ([19]). The differentiation of solid or liquid phases in simulations can be achieved by introducing the quantity S_{ij} (equation 81) ([19]).

$$S_{ij} = \sum_{m=-6}^{6} q_{6mi} q^*_{6mj} \tag{81}$$

One can define two bodies to be bound, if a certain condition for the value, e.g. $S_{ij} > 0.5$, is fulfilled. A particle is then considered to be solid if a certain number of neighbors is connected, typically 6 or 8. If not enough neighbors fulfill the connection criterion, the particle is considered liquid.

This is a very effective method of identifying solid structures in molecular simulations. Nevertheless, this method lacks the ability of identifying the specific crystal structure.

Different crystal structures exhibit differently pronounced values of Steinhardt bond order parameters depending on the order l, which can be seen in figure 2. The probability densities of the Steinhardt bond order parameter changes drastically for different orders l, in this case $l = 4$ and $l = 6$.



(a) Probability density of SBOP of order $l = 4$ for different crystal structures

(b) [Probability density of SBOP of order $l = 6$ for different crystal structures

Figure 2: The SBOPs have a different sensitivity for distinct crystal structures depending on the order l. The left chart shows the probability density of q_4 (dashed line) and $\overline{q}_4$ (solid line), whereas the right chart depicts the probability density of q_6 (dashed line) and $\overline{q}_6$ (solid line). In this case the molecular system used is the Lennard-Jones system. Figure taken from: [19].

Generally, the peaks in the probability density for the averaged values $\overline{q}_l$ are more pronounced than for the single values q_l. This indicates that the averaged values $\overline{q}_l$ are better applicable for the task of structure analysis. Furthermore, the averaged values are of smaller magnitude.

Another observation is that the probability densities in figure 2 does overlap for different crystal structures, which makes it hard to assign a structure without reasonable doubt. In the case for the q_4 it is almost impossible to distinguish between BCC and the liquid phase of the system. The same is found for the averaged values $\overline{q}_4$. On the other hand, having a look on the probability density of q_6 shows that

the liquid and the BCC phase are well separated and distinguishable.
This motivates the idea of using a plane of two bond order parameters in order to characterize the system. Different combinations of bond order parameters are depicted in figure 3. One observation is that the averaged values on the figures right-hand side provide a better distinction between the different structures. The most difficult structures to distinguish are the pairs BCC- HCP and again BCC-liquid phase. The best results are provided by the usage of the $\overline{q}_4 - \overline{q}_6$-plane, where all structures are well separated.

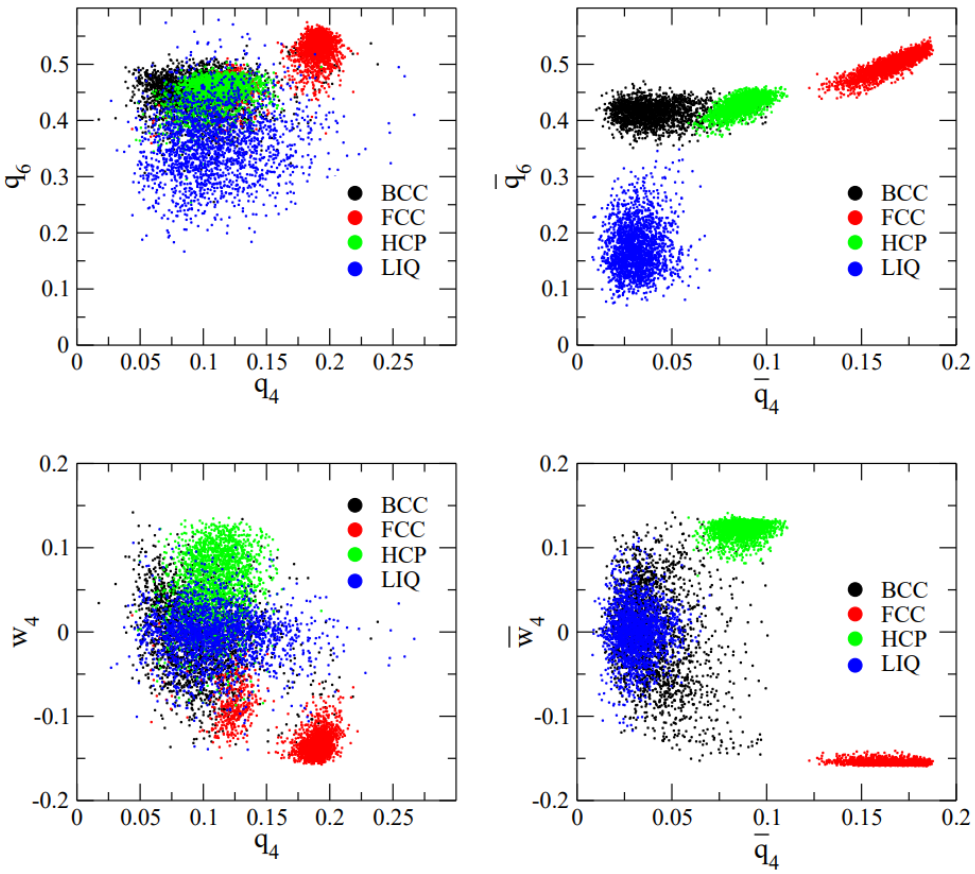


Figure 3: This figure shows different combinations of bond order parameters in order to make crystal structure distinguishable. 2000 particles from a Lenard-Jones system were chosen randomly. The two left panels show the $q_4 - q_6$-plane as well as the $q_4 - W_4$-plane, whereas the right panels show the averaged versions of the planes. Figure taken from: [19].

The conclusion is that the usage of two bond order parameters provides a reliable way to characterize the crystal structure. It is advantageous to apply the averaged values of the bond order parameters. The averaged values take not only into account the first neighbor-shell but also include information about the second shell, thus being more efficient in differentiating between liquid and solid sates.

4.4 SBOP Applied for Driving Homogeneous Ice Nucleation

A further application of the Steinhardt bond order parameters is both, the tracking and driving of homogeneous ice nucleation in molecular dynamics simulations. Reinhardt et al. ([29]) demonstrated that bond order parameters can be utilized to force the growth of ice nuclei in supercooled liquids. Nucleation is due to the formation of an interface between the two phases a process with a free energy barrier, making homogeneous nucleation events rare.

For the mono-atomic mW-water model, simulations have been able to reproduce homogeneous nucleation ([25]). However, the mW-water model is overestimating the dynamics drastically, hence it is not the best model for studying nucleation of ice. On the other hand, water models with multiple atoms provide good estimates on the dynamics of water but struggle with ice nucleation. A problem is how a local structure measure can be utilized to drive a simulation.

Examples of quantities distinguishing the local structure are order parameters such as the Steinhardt bond order parameter, as it is was shown in the previous section. Reinhardt et. al ([29]) utilized the probability density for all pairs $d_3(i,j)$, defined in equation 82, to distinguish between liquid and solid states of water. The order $l = 3$ is known to be the best tracer for tetrahedral structures. Figure 4a depicts the probability density of $d_3(i,j)$ for the TIP4P-2005 water model. A distinction between the liquid phase, cubic ice and hexagonal ice is made. A clear peak in the probability density for cubic ice is observed around the value $d_3(i,j) = -1.0$, whereas hexagonal ice shows a peak at $d_3(i,j) = -1.0$ as well as for $d_3(i,j) = -0.1$. The liquid phase does not show any extraordinary features in the probability density.

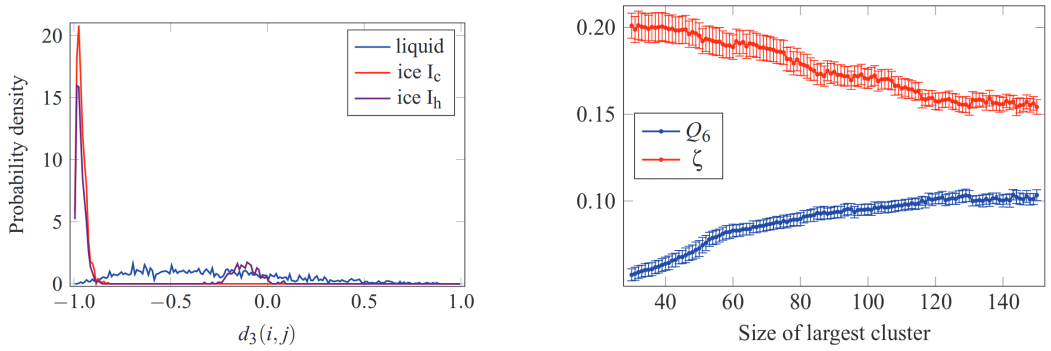
In Reinhardt et al. ([29]) a molecule is considered ice-like if $n_{connections} \geq 3$ and else liquid-like (see equation 83).

$$d_3(i,j) = q_{3i} \cdot q^*_{3j} \tag{82}$$

$$n_{connections}(i) = \sum_{j=1}^{N_n(i)} \Gamma(d_3(i,j)) \tag{83}$$

$$\Gamma(x) = \begin{cases} 1 \text{ if } [(x < -0.825) \ or \ (-0.23 < x < 0.01)], \\ 0 \text{ else} \end{cases} \tag{84}$$

Due to this pronounced peaks for the solid phases of water in the probability density of $d_3(i,j)$, it is possible to use this value to classify molecules as solid or liquid. A possible function to classify the results of figure 4a, is equation 84.



(a) Probability density of $d_3(i,j)$

(b) Averaged Steinhardt bond order parameter $\overline{q}_6$ as function of cluster size

Figure 4: (a): The probability density of the $d_3(i,j)$ correlation function shows distinct features depending on the phase of water. Pronounced peaks are observed for the ice phases, whereas the liquid phase has no comparable features. (b): The averaged Steinhardt bond order parameter $\overline{q}_6$ (in figure denoted as Q_6) shows a clear correlation with the ice cluster size of the TIP4P-2005 water simulation. ζ is a parametrization, here not further considered. Figures taken from: [32].

Monte Carlo simulations in an isobaric-isothermal ensemble were conducted in combination with umbrella sampling. The idea is to use the order parameter as a parametrization of a bias function. This means that the bias b is a function of the order parameter $b = b(\overline{q}_6)$. The bias modifies the acceptance probability in such a way, that certain nucleus sized are preferred. After the simulation the bias can then

be analytically removed from the results.
A clear correlation between the cluster size and the $\overline{q}_6$ Steinhardt bond order parameter, seen in figure 4b, is observed.
The Steinhardt bond order parameter provides a powerful tool for simulations concerning homogeneous nucleation of ice. The first task that can be accomplished by the order parameter is the tracking of the nucleation. A clear correlation of Steinhardt bond order parameter of order $l = 6$ and the cluster size is found. The second task is driving the simulation by providing a parameter for the umbrella sampling .

4.5 Shortcomings of SBOP

Steinhardt bond order parameters are a well established method for characterizing the local structures in glasses and crystals as well as descriptors for nucleation processes. Nevertheless, Steinhardt bond order parameters suffer from some crucial weaknesses. Mickel et al. ([22]) are discussing two major flaws in the definition of the order parameter as explained in detail below.
For a perfect crystal structure a specific value for the Steinhardt bond order parameter is obtained. For the identification of the crystal system this knowledge has been applied. Nevertheless, Kapfer et al. ([16]) showed that non-crystalline local structures can exist with similar order-parameter values. However, this argument speaking against Steinhardt bond order parameters is weakened by statistics in larger configurations.

4.5.1 Ambiguity of the Neighborhood Definition

There are several possible ways to define the nearest neighbors of an atom in molecular dynamics.

- Nearest neighbors via cutoff radius r_c: Every particle which ones center is closer than $a\sigma$ to the center of particle A is considered as nearest neighbor of particle A. σ is here the particles diameter. Values of $a = 1.2$ and $a = 1.4$ are used commonly.

- Nearest neighbors via Delaunay graph: The Delaunlay triangulation is applied, connecting the centers of the particles in such a manner that the circumcircle of the formed triangles does not include other centers.

- Nearest neighbors via fixed number of neighbors: To every particle a fixed number of the closest particles number is assigned and considered as nearest neighbors.

Figure 5 shows the implications of the definition of the nearest neighbors on a sample configuration. A general problem is that small changes in the particle's position can change the values of the order parameter drastically.

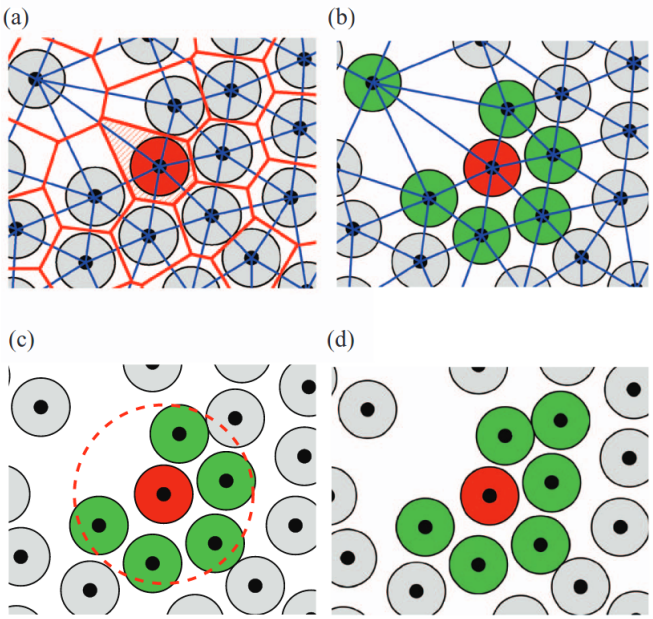


Figure 5: Different definitions for the determination of the nearest neighbors result in distinct designations of the nearest neighbors for the same configuration. (a) Voronoi diagram, (b) nearest neighbors via Delaunay graph, (c) via cutoff radius, (d) via fixed number of neighbors. Figure taken from [22].

The nearest neighbor definition has impact on the Steinhardt bond order parameter, which is depicted in figure 6. For a supercooled liquid four different nearest neighbors definitions are applied as a function of the packing fraction ϕ for the calculation of the averaged order parameter $\overline{q}_6$. It can be seen, that the obtained results $\overline{q}_6$ span over a large range of values where the highest values are achieved when utilizing the cutoff radius definition and the lowest when utilizing the Delaunay definition. The deviations make the calculation of Steinhardt bond order parameters for different definitions difficult to compare. A further observation is that $\overline{q}_6$ is not

constant for changes in the packing fraction ϕ.

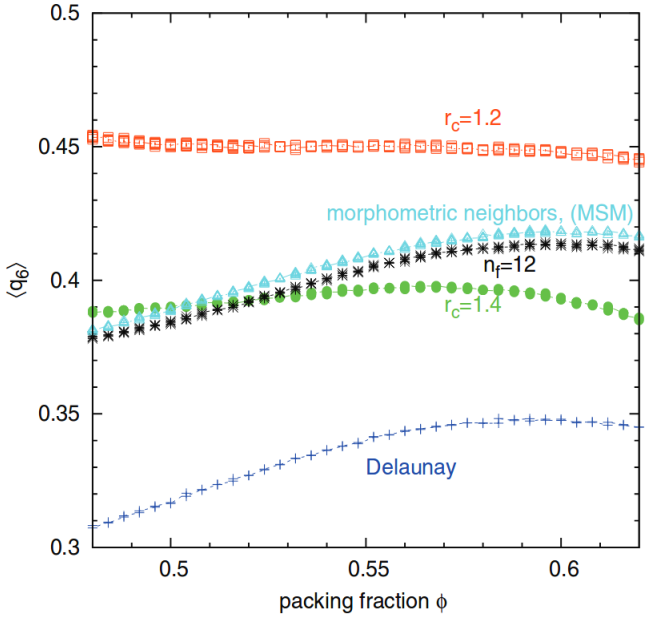


Figure 6: Depicted are the values for the averaged Steinhardt bond order parameter of order $l = 6$ $\overline{q}_6$ as function of the packing fraction ϕ for different nearest neighbors definitions. The red squares represent the cutoff radius definition for $r_c = 1.2\sigma$, the black stars represent the fixed number of neighbors definition with $n = 12$. The green dots show the cutoff radius definition for $r_c = 1.4\sigma$, the blue crosses represent the Delaunay definition. Figure taken from [22].

4.5.2 Minkowski Structure Metrics

Mickel et al. ([22]) introduced a modified order parameter in order to remove the problem of discontinuities in the neighbor assignment. Infinitesimal small changes of the configuration can lead to drastically deviating results for $\overline{q}_l$. To address this problem a weight in respect to the relative area of the Voronoi cell facet $A(f)/A$ is added to the definition of the Steinhardt bond order parameter, seen in equation 85.

For crystal structures where the Voronoi faces all have equal areas - SC,FCC, HCP - the value of q_l' is equal to the value of q_l. However, this is not the case for BCC since the facet area for the second shell neighbors is smaller than the facet area of neighbors in the first shell.

$$q_l'(a) = \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} \left| \sum_f \frac{A(f)}{A} Y_{lm}(\theta_f, \phi_f \right|^2 \right)^{1/2} \tag{85}$$

Most works only consider orders $l = 4$, $l = 6$ of the Steinhardt bond order parameters. However, Mickel et al. ([22]) are emphasizing that q_3' and q_5' can be used for the identification of centro-symmetric structures since the values of the order-parameters vanish for inversion-invariant configurations.
The neighbor definition always has to be considered when comparing q_ls of different studies. In order to eliminate discontinuities in the q_l-values, a new parameter the Minkowsky structure metric q_l', is introduced, where the contribution of a neighbor is weighted according to its Voronoi-facet area.

5 Machine learning

There are many attempts to elegantly define machine learning. I find the following subsumption into one single sentence quite precise:

> "Machine learning is based on algorithms that can learn from data without relying on rules-based programming." [1]

The term "machine learning" was the first time used by the pioneer of artificial intelligence Arthur Samuel and today describes a very broad field of computational sciences, which is focused on algorithms capable of improving its performance from previous inputs ("learning from experience"). Machine learning approaches are commonly utilizes for following tasks:

- Regression: A typical task in machine learning is finding an analytical model to describe the relations between quantities. The idea is to find unknown parameters represented by the vector $\mathbf{p}$ to describe the relation between the independent variables $\mathbf{x}$ and the dependent variable y, which is mathematically expressed by equation 86.

  $$y = f(\mathbf{x}, \mathbf{p}) \tag{86}$$

  A constraint is that $\mathbf{p}$ is chosen is such a way that a certain error function is minimized.

[1] `https://mck.co/2BGeAVh`, last time accessed on 28-07-2018.

- Classification: A dataset has to be classified into classes which are defined by the applicant.
- Clustering: A similar task to the classification. The main difference is that no predefined classes exist and the algorithm autonomously creates classes which are called clusters.
- Dimensionality Reduction: It is sometimes useful to reduce the dimension of the problem in order to simplify future processing. Lets consider a data set with multiple variables which can be best illustrated by thinking of a three dimensional data cube. Furthermore, the data is tightly connected to a plane within the cube. In this case a projection of the data on the plane eliminates one dimension and reduces the complexity of the problem.

A distinction of two machine learning types is useful. The first learning type is the so-called supervised learning. Here, the data used for the training includes the solution of the problem. In the case of optical character recognition (OCR), this means that in addition to the image itself also the character (solution or label) is provided. Both, the regression task and the classification task require supervised learning. Figure 7 shows the results of a classification as well as regression task.
The second learning type is unsupervised learning which is characterized by unlabeled training data. The clustering task, as well as the dimensionality reduction task are both unsupervised learning methods.
In this work supervised learning via an artificial neural network is conducted.

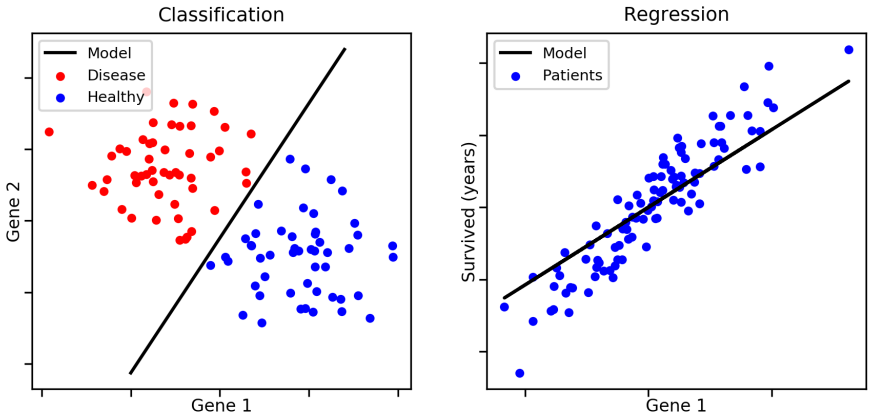


Figure 7: Two typical task of machine learning are the classification in previously defined classes and the fitting of the best model parameters, called a regression. Both task are part of the so-called supervised machine learning method, which requires labeled data. Figure taken from `https://bit.ly/2MYXmaR`, last time accessed on 28-07-2018.

5.1 Neural Network

Artificial neural networks (or only neural networks, NN) are machine learning based computational systems, the design of which is inspired by biological neurons ([34] chapter 3).

NNs have a huge range of applications like optical character recognition (OCR) ([28]), pattern recognition ([20]), speech recognition ([12]), general classification tasks and in the case of this work the calculation of energies and forces for a molecular dynamics simulation ([6]).

The comparison of a biological neuron and an artificial neuron, the so-called perceptron, can be seen in figure 8. Every biological neuron consist of a nucleus and two types of appendices, the dendrites and the axons. Both appendices transmit electrical signal in one direction. The dendrites receive impulses from other neurons and conduct them towards the neuron's nucleus, whereas the axons transmit impulses from the nucleus towards other neurons. Typically, neurons emit signals only when a certain threshold of the input signals is exceeded.

The idea of artificial neural networks is similar. From different artificial neurons individually weighted signals are transmitted towards another artificial neuron, which

only emits a signal itself, when the sum of the input signals exceeds a threshold.

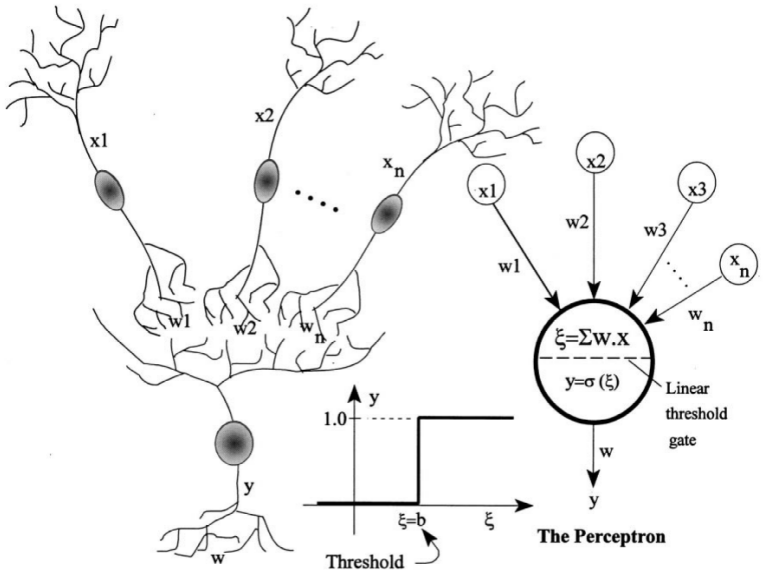


Figure 8: Artificial neural networks are inspired by biological neural networks. In both cases the information flow is unidirectional and thresholds have to be exceeded in order to further pass a signal. Figure taken from `https://bit.ly/2zlkKgt`, last time accessed on 28-07-2018.

Neuronal networks consist of many neurons which are connected, for instance the human brain is estimated to contain $86 \pm 8 \cdot 10^9$ neurons (see [3]). Depending on the task artificial neural networks typically are consisting of hundreds to thousands of neurons, which is significantly smaller and demonstrating the complexity of biological systems.

6 Representing Potential Energy Surfaces by Neural Network Potentials

This section is based on a detailed review article by Behler [6]. Molecular dynamics simulations provide an outstanding tool for studying the properties of molecular systems. For systems of smaller size electronic structure methods, like density-functional theory (DFT), for the calculation of energies and forces can be utilized. Nevertheless, the computational effort for ab-initio methods restricts the simulation

time and the system size. Typical ab-initio molecular dynamics simulations have simulation times in the range of tenths of picoseconds ([30]). To address this limitations the ab-initio methods are replaced by efficient proximate potentials. This allows to determine the potential energy surface (PES) several orders of magnitude faster than ab-initio methods. Unfortunately, the atomistic models fail to reproduce the PES correctly in all facets and many models specializing only on particular problems exist.

Behler divides potentials into two categories: "physical" and "mathematical" potentials. "physical" potentials are including as much information of the system as possible in order to approximate the real behavior. Often a relatively small number of parameters is enough to sufficiently describe the system. An example of a "physical" potential is the Lennard-Jones potential, which depends on of only two free parameter. On the other hand "mathematical" potentials make use of a large number of free parameters, which ones values are determined by "machine learning" algorithms.

The "machine learning" method for the determination of the potential energy surface reviewed here is based on artificial neural networks (NNs), which are receiving a lot of attention recently, although they were already introduced several decades ago ([21]). Originally, neural networks had the disadvantage that not all functions could be reproduced. This problem was solved by the introduction of a hidden layer and the replacement of step functions with arbitrary activation functions ([6]).

6.1 Feed-Forward Neural Networks

The most commonly used type of neural network is the so-called feed-forward NN. A NN network consists of interconnected nodes which are arranged in layers. Due to their similarity to biological neurons, the nodes are also called artificial neurons. The feed-forward NN is characterized by its unidirectional information flow. Figure 9 depicts the principle structure of a feed-forward NN, in this case with two hidden layers. The goal of the NN is to calculate the energy E from the input vector $\mathbf{G}$ (also called input layer). In the case of molecular dynamics $\mathbf{G}$, holds information of the particle configuration and E represent the energy of the configuration. A larger number of hidden layers increases the complexity of the functions that the NN can reproduce. The arrows represent the flow of information and to each arrow a weight

is assigned.

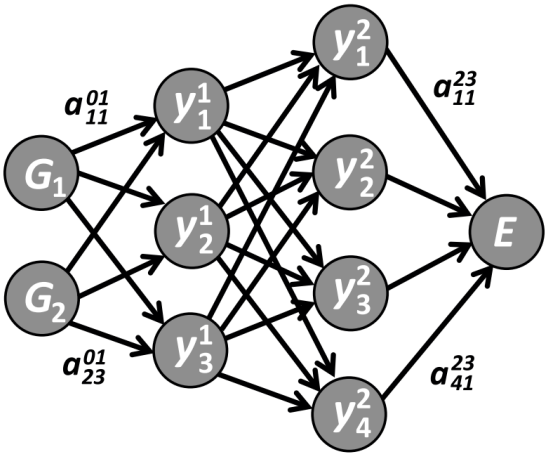


Figure 9: The schematic structure of a feed-forward NN. Via two hidden layers the input vector $\mathbf{G}$ is transformed to a corresponding energy E. The arrows represent the information flow. Figure taken from [6].

The value of the i-th neuron in the j-th layer is y_i^j and is calculated via linear a combination of values from the previous layer, as seen in equation 87.

$$y_i^j = f_i^j\left(b_i^j + \sum_{k=1}^{N_{j-1}} a_{k,i}^{j-1,j} y_k^{j-1}\right) \tag{87}$$

Here, b_i^j is a shift of the input value, the so-called bias, $a_{k,i}^{j-1,j}$ represent the weight of the k-th node from the previous layer $(j-1)$ on neuron i in layer j. Finally, an activation function on the linear combination is applied. Following forms of activation functions are used commonly:

- Linear activation function
- Sigmoid activation function
- Hyperbolic tangent activation function
- Gaussian activation function

The explicit function for the calculation of the energies (E) for the NN in figure 9, is given by equation 88. Here, G_i represents the symmetry functions. Commonly, the last activation function is linear to ensure that a big range of values can be accessed.

$$E = f_l^3\left(b_l^3 + \sum_{k=1}^{4} a_{kl}^{23} f_k^2\left(b_k^2 + \sum_{j=1}^{3} a_{jk}^{12} f_j^1\left(b_j^1 + \sum_{i=1}^{2} a_{ij}^{01} G_i\right)\right)\right) \tag{88}$$

NNs tend to have a huge number of free parameters depending on the number of hidden layers and nodes. Equation 89 gives the total number of parameters N_W, where M_{HL} is the number of hidden layer and N_k the number of neurons in layer k.

$$N_W = \sum_{k=1}^{M_{HL}+1} (N_{k-1}N_k + N_k) \tag{89}$$

In many cases several thousand free parameters have to be determined, which requires a big dataset for the training procedure.

6.1.1 Neural Network Potential

An neural network potential (NNP) is an atomic potential fulfilling following conditions:

- The multidimensional potential energy function is calculated via a NN with help of the atomic coordinates and charges. This means that the system is fully characterized by the positions and charges of the particles, which is basically the same input required for an ab-initio method.
- Typically, the NNP is generated by utilizing ab-initio method values as references. Nevertheless, this is not necessary and also in this work a physical potential is used.
- No physical approximations for the NNP are used. The values from the reference calculations are strictly defining what a successful NNP is.

A NNP aims to reproduce the PES from a reference model (electronic structure) as good as possible.

6.1.2 High-dimensional Neural Network Potential

There are several problems arising when NNP are applied. The input layer nodes contain the symmetry functions which store the information about the local atomic environment. Each atom of the same element has the same set of symmetry functions with distinct values depending on the atoms vicinity. An increase of the particle number in the configurations leads to an increase of the input layer nodes. Subsequently also the hidden layer has to be adapted for the new configuration to cover the new degreed of freedom. Furthermore, a conventional NNP is only applicable to the system size, which has been used for the calculation of the weights. This makes sense, when considering that the number of input nodes is the number of atoms in the system.

A further difficulty of NNPs is that several invariances of the energy E concerning the position of the atoms have to be fulfilled. Rotations and translations of the configuration as well as a permutation of equivalent particles should result in an equivalent PES. This has been the biggest challenge for NNP since the common Cartesian coordinates do not provide these invariances. Every atom is linked to an input layer neuron with a specific weight. It is easily understandable why in exchange of two particles for a NNP with Cartesian coordinates leads to different energies.

The general idea of a high dimensional NNP is that for every single atom the contribution to the total energy is calculated in a separate neural network as seen in equation 90. The single atom's energy is a function of the atom's local environment within a certain cutoff radius R_c which is typically in the range of $6 - 10\mathring{A}$.

$$E_s = \sum_{i=1}^{N_{atom}} E_i \tag{90}$$

The input value is calculated via several symmetry functions describing the local environment of the atom. This corresponds to a transformation of the Cartesian coordinates R_i into a symmetry function vector G_i. The energies of the individual atom E_i have no real physical meaning, only the sum of all atomic energies corresponds to the physical potential energy of the configuration. This method of splitting the total energy E_s into contributions of single atoms solves the problem atom number variations in the configurations.

Figure 10 represent the schematic structure of a high dimensional NNP for a system of three atoms. For every singe atom from the Cartesian coordinates $\mathbf{R}_i$ the symmetry functions vector $\mathbf{G}_i$ are calculated. The symmetry function vector is then the input layer for an atomic NN, which calculates the contribution of the atoms energy E_i to the total potential energy. In this example not only short range interactions are considerer, but also electrostatic contributions are calculated via a separate high dimensional NN.
Since the summation order of the single atomic contributions E_i is arbitrary, it is guaranteed that permutations of atoms result in the same total energy E.

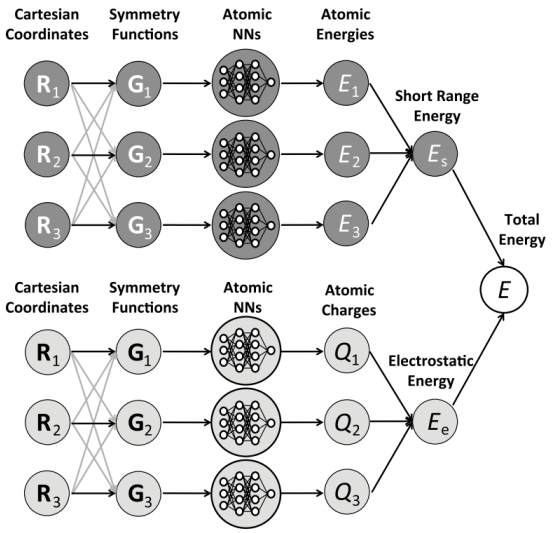


Figure 10: This figure represents the schematic structure of a high dimensional NNP. The system consists of three atoms and electrostatics is considered by including an additional contribution calculated in a separate NN. Figure taken from [6].

6.2 Symmetry Functions

It is of great importance to be able to characterize the local atomic environment and providing suitable input for the NN. In general, Cartesian coordinates do not provide the invariances desired for the input quantity. For this reason symmetry functions are used to describe the local atomic environment. Following criteria have to be fulfilled by by symmetry functions:

- Symmetry functions and their derivatives need to be continuous, which is a general requirement on a PES in order to be able to calculate forces in MD simulations.

- Symmetry function need to be invariant towards rotation and translation of the configuration. For example Cartesian coordinates change their values when a rotation is applied to the system which results in different outputs values.

- The symmetry functions have to be invariant to permutations between same species atoms. Again, this is not fulfilled by Cartesian coordinates. When two atoms are interchanged their Cartesian coordinates change and this results in a distinct output.

- Every atom of the same species must have the same number of symmetry functions.

The first step towards symmetry functions is to define a cutoff radius in order to only include information of the local atomic environment. A possible choice for a cutoff function is seen in equation 91. This represents a monotonous decreasing function until the cutoff radius R_c is reached where the function attains the value zero.

$$f_{c,1}(R_{ij}) = \begin{cases} 0.5\left[\cos\left(\frac{\pi R_{ij}}{R_c}\right) + 1\right] \text{ if } R_{ij} \leq R_c \\ 0 \text{ if } R_{ij} > R_c \end{cases} \tag{91}$$

Furthermore, a cutoff function of the form as seen in equation 92 can be applied. This cutoff function has the advantage that it avoids discontinuities at the calculation of forces. The choice of the value of R_c itself is important since it defines the region around an atom which is considered in the characterization of the local environment via symmetry function. Every atom outside the cutoff radius is neglected. Hence, it is of great importance to ensure that all energetically relevant particles are included within the cutoff radius. Generally, cutoff radii in the range between $6 - 10\mathring{A}$ suffice in fulfilling this condition.

$$f_{c,2}(R_{ij}) = \begin{cases} \tanh^3\left[1 - \frac{R_{ij}}{R_c}\right] \text{ if } R_{ij} \leq R_c \\ 0 \text{ if } R_{ij} > R_c \end{cases} \tag{92}$$

A distinction between two types of symmetry functions can be made: Radial symmetry functions and angular symmetry functions. Radial symmetry functions have the purpose to describe the radial distribution of atoms, whereas angular symmetry functions describe the angular distribution. A very basic radial symmetry function is the sum of all cutoff functions of the atom's neighbors G_i^1 seen in equation 93. It is important to note that for each chemical element in the simulation a separate symmetry function is calculated.

$$G_i^1 = \sum_{j=1}^{N_{atom}} f_c(R_{ij}) \tag{93}$$

An improved version for a radial symmetry function is given by equation 94. Due to the Gaussian factor, neighbor atoms farther away have a smaller contribution to the symmetry function, which is a good property when considering, that the interactions are weaker for increasing distances between the atoms. The radial symmetry function G_i^3 defined via equation 95 includes a damped cosine term with the parameter κ which is adjusting the period.

$$G_i^2 = \sum_{j=1}^{N_{atom}} e^{-\nu(R_{ij}-R_s)^2} f_c(R_{ij}) \tag{94}$$

$$G_i^3 = \sum_{j=1}^{N_{atom}} \cos(\kappa R_{ij}) f_c(R_{ij}) \tag{95}$$

The parameter ν controls how fast the function is decaying, whereas R_s sets the radius for which the symmetry function is specifically sensitive to. Hence, a combination of several radial symmetry functions with different values for ν and R_s can be utilized to describe the radial distribution of atoms efficiently. The radial symmetry functions G_i^1, G_i^2 and G_i^3 are shown in figure 11, where the cutoff radii R_c for G_i^1 and G_i^2 are varied, η is varied for G_i^2 and κ is varied for G_i^3.

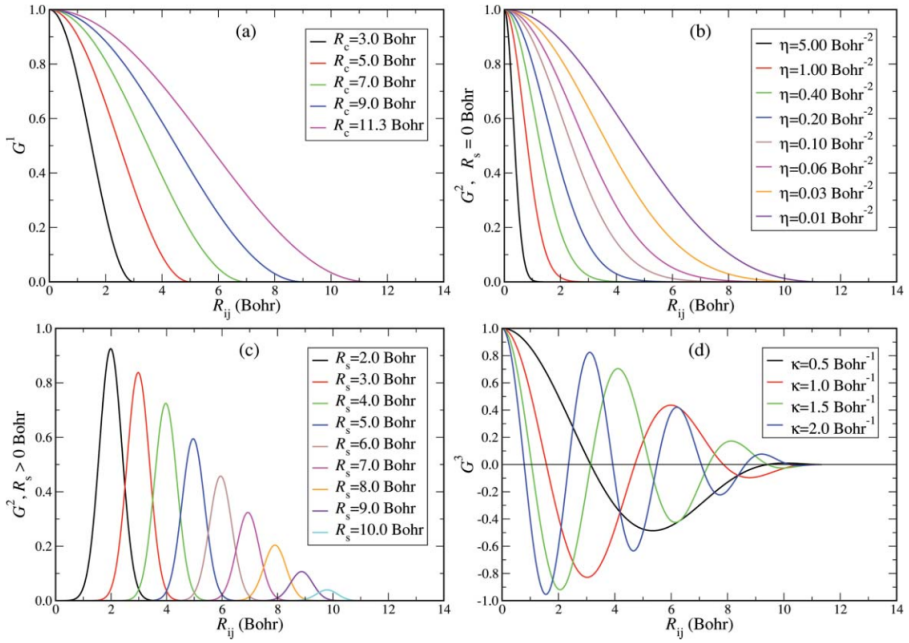


Figure 11: The radial symmetry functions G_i^1, G_i^2 and G_i^3 for different parameters are depicted in this figure. Figure taken from [5].

This radial symmetry function cannot reflect the angular distribution of atoms within the cutoff radius. For this reason additional angular symmetry functions depending on the angle $\theta_{ijk} = \dfrac{\mathbf{R}_{ij}\mathbf{R}_{ik}}{|\mathbf{R}_{ij}||\mathbf{R}_{ik}|}$ are introduced. Here, $\cos\theta_{ijk}$ describes the angle between atom j and k in respect to atom i. Two possible angular symmetry functions are G_i^4 and G_i^5 as seen in equations 96 and 97 respectively.

$$G_i^4 = 2^{1-\zeta} \sum_{j\neq i} \sum_{k\neq i,j} \left[(1+\lambda\cos\theta_{ijk})^\zeta e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \right] \tag{96}$$

$$G_i^5 = 2^{1-\zeta} \sum_{j\neq i} \sum_{k\neq i,j} \left[(1+\lambda\cos\theta_{ijk})^\zeta e^{-\eta(R_{ij}^2+R_{ik}^2)} f_c(R_{ij}) f_c(R_{ik}) \right] \tag{97}$$

These symmetry functions take into account all possible triples within the cutoff radius. The parameter ζ defines the angular resolution of the function. Again, η defines the specific radius for which the symmetry function is sensitive while λ defines where the maximum of the cosine term is found. G_i^4 and G_i^5 have basically the same form, except for the term including R_{ik}. This mean that the distance of

the atom j has no effect on the results. Figure 12 shows the angular symmetry function G_i^4 as function of θ_{ijk}.

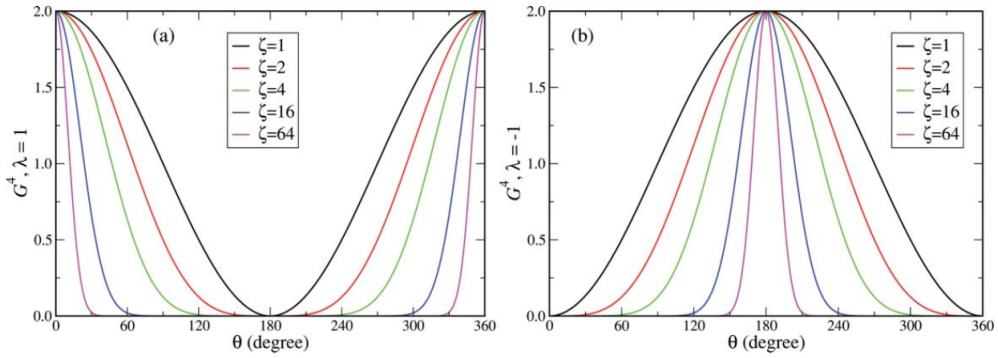


Figure 12: The angular symmetry function G_i^4 for different values of λ. The chart on the left-hand side shows G_i^4 for $\lambda = 1$, whereas the right chart shows G_i^4 for $\lambda = -1$. The angular part of G_i^4 is identical to the angular part of G_i^5. Figure taken from [5].

6.3 Energy Gradients

There is a clear relation between the input values $\mathbf{G}$ of the NNP and the obtained energy E. Therefore it is possible to analytically express the forces:

$$F_\alpha = -\frac{\partial E_s}{\partial \alpha} - \frac{\partial E_{elec}}{\partial \alpha} \tag{98}$$

Here, α is the spatial component of interest, E_s is the short range component of the energy and E_{elec} is the electrostatic energy. Of special interest is the short component which is given by equation 99.

$$F_{\alpha,s} = -\frac{\partial E_s}{\partial \alpha} = -\sum_{j=1}^{N_{atom}} \frac{\partial E_j}{\partial \alpha} = -\sum_{j=1}^{N_{atom}} \sum_{\mu=1}^{N_{sym,j}} \frac{\partial E_j}{\partial G_{j\mu}} \frac{\partial G_{j\mu}}{\partial \alpha} \tag{99}$$

Here, N_{atom} represent the number of atoms in the configuration, N_{sym} is the number of symmetry functions and $G_{j\mu}$ is the μ-th symmetry function of the j-th atom.

6.4 Training of the NNP

In the development of a NNP a process which was not considered yet is the calculation of the weights. This procedure is called training. The goal is to find suitable weights in order to precisely predict the energies and forces of the molecular configuration. These parameters are obtained by a so-called learning process.
An iterative process compares the energies and forces obtained by the NNP with the reference model energies and forces. A quantity describing how good the NNP achieves the reference energies is the root mean squared error as seen in equation 100.

$$RMSE = \sqrt{\frac{1}{N_{struct}} \sum_{i=1}^{N_{struct}} (E_{i,ref} - E_{i,NN})^2} \tag{100}$$

An analog quantity describes the deviation of the forces. There are many optimization algorithms which can be utilized in order to minimize RMSE. The most important ones are discussed in a separate section.
The set of the configurations used for the training procedure is usually split into two subsets. The first subset is the training set itself which ones configurations are actively used to optimize the weights. In order to be able to identify overfitting the second set, called test set, is utilized. The ratio of training set configurations to test set configurations is usually around $10 : 1$.
A typical learning process is shown in figure 13. The right chart depicts the fit of the NN together with the physical function. The discrete data points are uniformly distributed across the range of x. Two data points represent the test set which is not used in the training procedure. Clearly visible are deviations at exactly this "missing" descriptions of the system at this points.
The right-hand side figure shows the so-called learning curve of the NN which is the evolution of the RMSE of the test set as well as the evolution of the training set. Epochs are quantifying the training length. Initially both, the RMSE of the training set as well as the RMSE of the test set, are decreasing. At approximately 20 epochs the RMSE of the test set starts to increase while the RMSE of the training set is still decreasing. This is typical sign of overfitting. At this point the general properties of the system have already been learned by the NN and further learning with the training set leads only to improvement of the weights of the training set,

while impairing the description of the test-set.

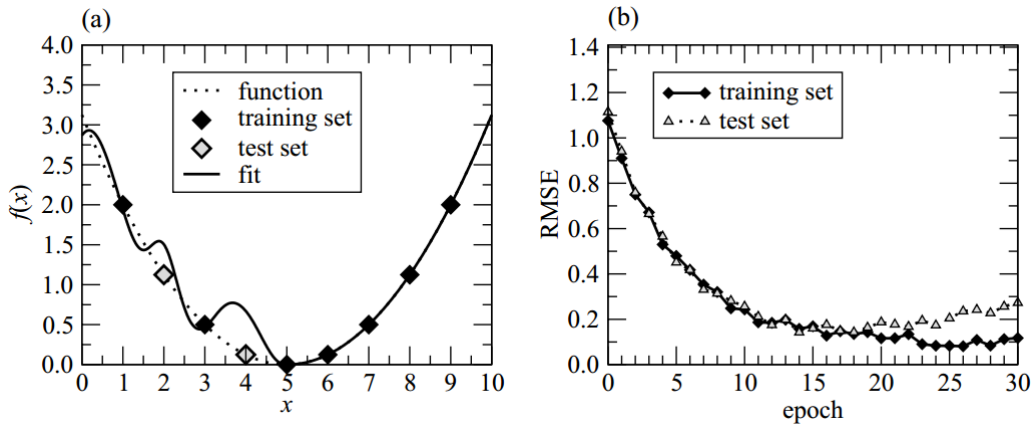


Figure 13: The left chart shows the regression of a NN for a given training set (black points) and a test set (white points). The test points are missing in the training. The NN should be able to reproduce the overall dependence $f(x)$. Overtraining the NN leads to the loss of the overall trend. The right chart shows a learning curve where the black points represent the RMSE of the training set and the white points represent the test set. Here, the overfitting is visible in the increasing RMSE of the test set. Figure taken from [6].

For a good NNP both, the test set as well as the training set are expected to have minimized RMSE. To ensure this it is crucial to choose the training set such that all relevant configurations are represented. Missing regions in the configuration space cannot be represented by the NNP, especially extrapolations beyond the trained parameters are unreliable.

7 Optimization Algorithms

A crucial process in NNs is the choice of the ideal parameters, i.e. the weights and biases. There are many optimization algorithms available but for neural network applied as driver for molecular dynamics, especially the Kalman filter and the gradient descent method are popular. These two methods are reviewed here.

7.1 Kalman Filter

The Kalman filter ([15]) is a recursive method specifically designed for noise contaminated measurements based on linear dynamical systems. At each time-step an operator is applied on the old state and mixed with noise. The mathematical formulation is portrayed in this section.

7.1.1 Initialization

The initial value of the error covariance matrix $\mathbf{P}(t)$ can be manipulated with the error parameter ϵ and the initial value of the process noise covariance matrix $\mathbf{Q}(t)$ is determined by the parameter $q(0)$.

$$\mathbf{P}(0) = \epsilon^{-1}\mathbb{1}, \quad \epsilon = 0.001 \ ... \ 0.01 \tag{101}$$

$$\mathbf{Q}(0) = q(0)\mathbb{1}, \quad q(0) = 0.0 \ ... \ 0.1, \quad q_{min} \approx 10^{-6} \tag{102}$$

7.1.2 Iteration

The Kalman filter is a iterative process minimizing the quantity $y(t)$, which is in this case the error of the energy and force estimation of the NNP:

$$y(t) = E_{pot}^{DFT} - E_{pot}^{NN}(t), \quad and \quad F_{i,\alpha}^{DFT} - F_{i,\alpha}^{NN}(t) \tag{103}$$

The derivative matrix $\mathbf{H}(t)$ is formed by the derivatives of the error $y(t)$ in respect to the weights $\mathbf{w}(t)$:

$$\mathbf{H}(t) = \left(\frac{\partial}{\partial w_1} \ ... \ \frac{\partial}{\partial w_n}\right)^T y(t) \tag{104}$$

Additionally the scaling factor $A(t)$ is calculated and is a function of the learning rate η $\epsilon(0.001...1.0)$, the derivative matrix and the error covariance matrix:

$$A(t) = \left[\frac{1}{\eta} + \mathbf{H}(t)^T\mathbf{P}(t)\mathbf{H}(t)\right]^{-1} \tag{105}$$

The Kalman gain matrix $\mathbf{K}(t)$ can be expressed via a matrix multiplication of the error covariance matrix and the derivative matrix multiplied with the scaling factor:

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}(t)A(t) \tag{106}$$

The values of the weights and the error covariance matrix are recalculated iteratively:

$$w(t+1) = w(t) + \mathbf{K}(t)y(t) \tag{107}$$

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \mathbf{K}(t)\mathbf{H}(t)^T\mathbf{P}(t) + \mathbf{Q}(t) \tag{108}$$

$$\mathbf{Q}(t) = q(t)\mathbb{1}, \quad q(t) = max(q(0)e^{-t\tau}, q_{min}) \tag{109}$$

7.2 Fading Memory Kalman Filter

A further Kalman filter is the so-called fading memory Kalman filter, also called extended Kalman filter ([36]). The idea of the fading memory Kalman filter is to give more weight to recent results and to give less weight to results the older they are. The forgetting factor λ is the parameter controlling how fast previous iteration results are neglected for the current calculation.

7.2.1 Initialization

For the fading memory Kalman filter the error covariance matrix is initialized as unity matrix and the initial value of the forgetting factor, as well as λ_0 have to be set.

$$\mathbf{P}(0) = \mathbb{1} \tag{110}$$

$$\lambda(0) = 0.95 \ ... \ 0.99, \quad \lambda_0 = 0.99 \ ... \ 0.9995 \tag{111}$$

7.2.2 Iteration

The goal of the iterative process is to minimize $y(t)$ which is, as for the Kalman filter, the error of the energy and force estimation of the NNP:

$$y(t) = E_{pot}^{DFT} - E_{pot}^{NN}(t), \quad and \quad F_{i,\alpha}^{DFT} - F_{i,\alpha}^{NN}(t) \tag{112}$$

The derivative matrix in respect to the weights is formed:

$$\mathbf{H}(t) = \left(\frac{\partial}{\partial w_1} \ ... \ \frac{\partial}{\partial w_n}\right)^T y(t) \tag{113}$$

The scaling factor is in the case of the fading memory Kalman filter a function of the forgetting factor λ instead of the learning rate η:

$$A(t) = \left[\lambda + \mathbf{H}(t)^T \mathbf{P} \mathbf{H}(t)\right]^{-1} \tag{114}$$

The Kalman gain matrix $\mathbf{K}(t)$ is calculated in the same way as in the case of the Kalman filter

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}(t)A(t) \tag{115}$$

The iterative calculation of the weights, the error covariance matrix and the forgetting factor is performed by following equations:

$$w(t+1) = w(t) + \mathbf{K}(t)y(t) \tag{116}$$

$$\mathbf{P}(t+1) = \lambda(t)^{-1}\big(\mathbf{P}(t) - \mathbf{H}(t)^T \mathbf{P}(t)\big) \tag{117}$$

$$\lambda(t+1) = \lambda_0 \lambda(t) + 1 - \lambda_0 \tag{118}$$

7.3 Gradient Descent

A commonly applied iterative optimization algorithm is the gradient descent method. The general idea is to find minima of a function by utilizing multidimensional gradients of the function.

Lets consider a function $f(\mathbf{x})$ with multiple variables defined by the parameter vector $\mathbf{x}$. The goal is to find the parameters minimizing the function $f(\mathbf{x})$:

$$\min_{x \epsilon \mathbb{R}^N} f(x) \tag{119}$$

From a starting point $\mathbf{x}_0$ the subsequent point is determined by the direction of the steepest descent which can be expressed via the gradient of the function. An important parameter is the positive step size α.

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n^* \nabla f(\mathbf{x}_n) \tag{120}$$

The step size α_n^* is chosen in such a way that it minimizes $f(\mathbf{x}_n - \alpha_n^* \nabla f(\mathbf{x}_n))$. It is possible to maintain a constant step size α. Nevertheless, it is advantageous to

calculate the ideal step size for every step. Several methods for calculating varying step sizes can be found in [37]. A method proposed by Barzilai-Borwein ([4]) determines the step size by following equation:

$$\alpha_n = \frac{(\mathbf{x}_n - \mathbf{x}_{n-1})^T[\nabla f(\mathbf{x}_n) - \nabla f(\mathbf{x}_{n-1})]}{|\nabla f(\mathbf{x}_n) - \nabla f(\mathbf{x}_{n-1})|^2} \tag{121}$$

Lets assume fixed step sizes. There are several approaches of the gradient descent methods, which will be discussed now.

7.4 Batch Gradient Descent

In machine learning the goal is the minimization of a cost function normally the mean squared error (MSE) or the root mean square error (RMSE). The definition of the MSE can be seen in equation 122. Here, m is the number of training instances, θ is the parameter vector, $y_m^{(i)}$ is the predicted value of the $i-th$ instance and $y^{(i)}$ is the value of the $i-th$ instance.

$$MSE(\theta) = \frac{1}{m}\sum_{i=0}^{m}\left(y_m^{(i)} - y^{(i)}\right)^2 \tag{122}$$

The MSE is the function which should be minimized. Hence, the gradient of this quantity in respect to its parametrization $\nabla_\theta MSE(\theta)$ has to be formed.

$$\theta_{n+1} = \theta_n - \alpha\nabla_\theta MSE(\theta) \tag{123}$$

7.4.1 Stochastic Gradient Descent

In order to calculate the gradient in the batch gradient descent method the whole test set is used. This is computationally expensive. A solution is to use only one instance for the calculation of the gradient which is chosen randomly. This approach is called the stochastic gradient descent. A typical behavior of this method is that the minimum is not reached directly as it is the case for the batch gradient descent and that even at the minimum the parameters are fluctuating.
An upside of using this method is that local minima do not have the devastating effect on the determined parameters as in the case of batch gradient descent since the stochasticity of the methods helps to escape the local minima. The final fluctuations around the minimum can be removed by using gradually decreasing step sizes.

7.4.2 Mini-batch Gradient Descent

The third gradient descent method discussed is the mini-batch gradient descent. For each step several randomly chosen instances are used for the calculation of the next parameter set. This has the advantage that an averaging of some instances reduces the impact of outliers in the data. Furthermore, in comparison with the stochastic gradient descent the fluctuation in the parameter space are reduced once reached the minimum of the cost function.
Typical paths of the three discussed typed of gradient descents are depicted in figure 14 for a two dimensional parameter space.

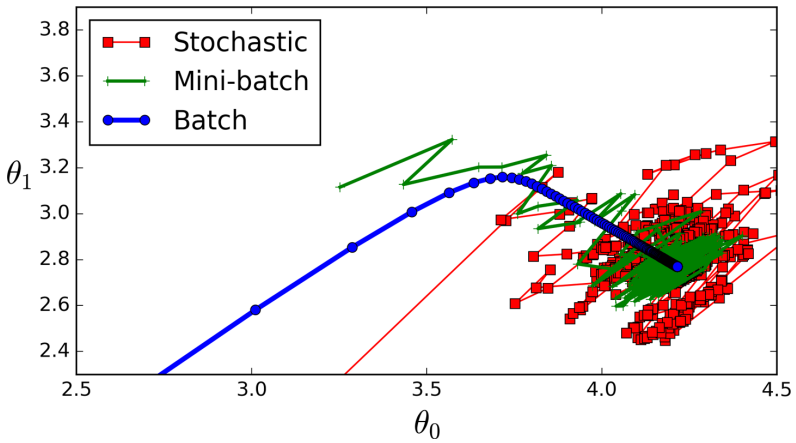


Figure 14: The batch gradient descent (blue lines) always choses the direct path towards the cost functions minimum. The stochastic gradient descent (red) maintains a large fluctuation in the parameter space even when the minimum is reached. The green lines represent the path of a mini-batch gradient descent method with reduced fluctuations. Figure source: [10] p.120.

8 Water Models

8.1 mW-Water

One of the greatest challenges in molecular dynamics is the creation of models which should reproduce important quantities of the molecular configuration without computational expensive ab-initio calculations. Of special interest is water. The goal of the models is representing crucial molecular properties of water. Most of the water

models follow the paradigm of using long-range interactions to create the tetrahedral structure. Into this category fall the popular water models SPC ([8]), SPCE ([7]), TIP3P and TIP4P ([14]). All of them utilize a long-range electrostatic potential and consist of multiple force centers. A question arising is if a coarse-grained model, this means a model where water is represented by a single particle, without long-range interactions, can still reproduce crucial properties of water.

Existing coarse-grained water models using only short range interactions and spherically symmetric potentials are not capable of reproducing the angular distribution function, while being good at representing the radial distribution function. An idea of Molinero et al. ([24]) is considering elements that form tetrahedral structures which are silicon and germanium. For these elements low-density amorphous solids with tetrahedral coordinations are found. A further similar property of silicon and germanium in comparison to water is the density anomaly. All of this indicated that long-range interactions are not the key ingredient in models to create tetrahedral structures.

The potential to describe water is similar to the Stillinger-Weber ([33]) (SW) silicon potential. Equation 124 shows the so-called mW mono-atomic water model potential introduced by Molinero et al. ([24]). It is a function of the distances of "water" pairs (r_{ij}, r_{ik}) and the angles between three particles (θ_{ijk}).

$$E = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j\neq i} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) \tag{124}$$

$$\phi_2(r) = A\varepsilon \left[B\left(\frac{\sigma}{r}\right)^p - \left(\frac{\sigma}{r}\right)^q \right] exp\left(\frac{\sigma}{r - a\sigma}\right) \tag{125}$$

$$\phi_3(r, s, \theta) = \lambda\varepsilon[\cos\theta - \cos\theta_0]^2 exp\left(\frac{\gamma\sigma}{r - a\sigma}\right) exp\left(\frac{\gamma\sigma}{s - a\sigma}\right) \tag{126}$$

Here, r and s are the distances to two neighboring atoms. To ensure that tetrahedral structures are formed the parameter $\theta_0 = 109.47°$ is set. The parameter determining the strength of the tetrahedral term is λ whereas ε determines the magnitude of the two-body interaction.

The mW-model is able to reproduce the density of water in the temperature range from 250K-350K within an error of 1%. Figure 15 shows the temperature-density diagram for different water models. The depicted atomistic models are capable to qualitatively correspond with the experimental data. However, mono-atomic,

isotropic models do not show this behavior. Only when adding angle-dependency to the mono-atomic potential, results according to the experimental data can be achieved. For mW-water the temperature of maximum density is 250K which deviates from the experimental value of 277K. Nevertheless, mW-water outperforms all other models except for TIP5P (285K) and TIP4P (253K).
The best agreement with the experimental data is found for the melting temperature of hexagonal ice ($T_m = 274.6$K, $T_{m,exp} = 273.15$K), the density of the liquid at T_m ($\rho_{liquid} = 1.001\text{gcm}^{-3}$, $\rho_{liquid,exp} = 0.999\text{gcm}^{-3}$), the density of the liquid at $T = 298$K ($\rho_{liquid} = 0.997\text{gcm}^{-3}$, $\rho_{liquid,exp} = 0.997\text{gcm}^{-3}$), the enthalpy of fusion at T_m ($\Delta H_m = 1.26\text{kcalmol}^{-1}$, $\Delta H_{m,exp} = 1.436\text{kcalmol}^{-1}$).

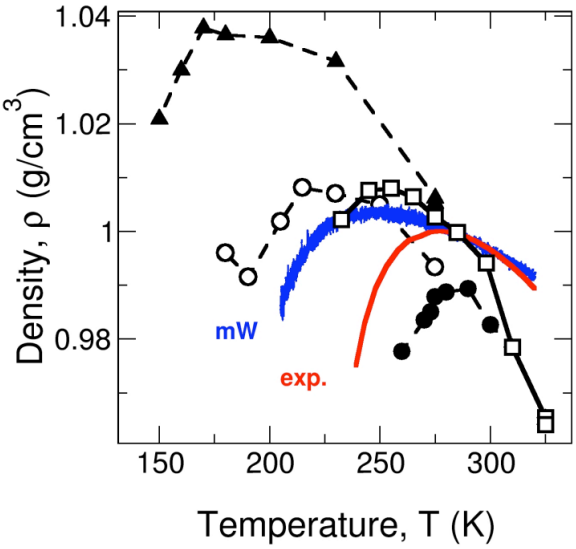


Figure 15: Various atomistic models of water show qualitatively the same results in the temperature-density diagram. TIP5P (black circles), TIP4P (white squares), TIP3P (black triangles), SPC (white circles), experimental data (red line), mW-water (blue line). In the case of mono-atomic models only non-isotropic models reproduce this behavior. Figure taken from [24].

Since mW-water is a mono-atomic model it lacks of degrees of freedom in comparison to atomistic models. Molinero and Moore ([24]) claim that this causes the heat capacity at $25°C$ to be less than half of the experimental value. Furthermore,

the model is not capable of reproducing the diffusion coefficient. At 298K $D = 6.5{\cdot}10^{-5}\text{cm}^2/\text{s}$ are achieved, whereas the experimental value is $D_{exp} = 2.3{\cdot}10^{-5}\text{cm}^2/\text{s}$. The diffusion coefficient is drastically overestimated. A possible explanation favored by the authors of the original paper is that the missing hydrogen atoms, which normally slow down the water molecule due to reorientation processes, are causing the overestimation of the diffusion coefficient.
By the introduction of the tetrahedral mono-atomic mW-water model it is possible to overcome the shortcomings of isotropic water models, which are not capable of simultaneously reproducing the energetic and structural properties of water. Advantages in comparison to the atomistic models are the relatively low computational costs, with the mW-water model being 100-times faster than the computationally cheapest atomistic model.

9 Computational Methods and Set-up

The main part of the work is coding on the NNP which is based on the work of Behler ([6], [5]) and currently maintained and improved by Andreas Singraber ([26]). Till now only radial and angular symmetry functions were available. I have implemented the SBOPs as alternative to the angular symmetry functions. The programming language used is C++ and the implementation is tested with the BOOST-test library [2]. The calculation of reference values of the SBOPs for perfect lattices as well as the three dimensional visualization of the SBOPs is performed python scripts.
Due to its simplicity the chosen reference model is the mW-model. Prior to the training, the set of symmetry functions has to be chosen. For all simulations, an identical set of 8 radial symmetry functions is chosen, which ones parameters can be seen in table 1. In this thesis two different methods for the NNP training are performed. For the training with angular symmetry functions, 36 functions are chosen, which are optimized for the application on the mW-water model. The parameters of the angular symmetry function can be seen in table 1. For the SBOPs all orders from $l = 3, ..., 12$ are used for three different cuff-off radii of $4, 5$ and $6\mathring{A}$. This gives in total 30 functions, which is comparable to the number of angular

[2] `https://bit.ly/2LrM26q`, last time accessed on 28-07-2018

Table 1: Parameter list of radial symmetry functions (equation 94) applied for the training

ν	0.001	0.01	0.03	0.06	0.15	0.3	0.6	1.5
r_{shift} (B)	0	0	0	0	4	4	4	4
r_c (B)	12	12	12	12	12	12	12	12

Table 2: Parameter list of angular symmetry functions applied for the training, which are the same for both the G_i^4 (equation 96) and G_i^5 (equation 97)

η	λ	ζ	r_c (B)	η	λ	ζ	r_c (B)
0.06	1	1	12	0.03	-1	3	12
0.06	-1	1	12	0.03	1	9	12
0.06	1	3	12	0.03	-1	9	12
0.06	-1	3	12	0.01	1	1	12
0.06	1	9	12	0.01	-1	1	12
0.06	-1	9	12	0.01	1	3	12
0.03	1	1	12	0.01	-1	3	12
0.03	-1	1	12	0.01	1	9	12
0.03	1	3	12	0.01	-1	9	12

symmetry functions.

The training set consists of approximately 100 configurations each of about 128 atoms. In order for the NNP learning process to capture all relevant information of the reference model, it is important that the training set contains as many different relevant physical cases as possible. This means that the training set should contain liquid configurations as well as perfect crystals. However, also displaced crystal structures and crystals with removed lattice points should be included.
The training set consists of liquid water configurations at a $T = 298$K for pressures of $p = -2000, -1000, 0, 1000, 2000, 3000$atm and icedIh configurations at $T = 274.6$K for $p = -2000, -1000, 0, 1000, 2000, 3000$atm. Furthermore, configurations with random displacements and random particle removals are in the training set. Finally, configurations with randomly changed box vector angles and box vector lengths are included. For each epoch in the training all energies from the training configurations

are used and only 0.3% of all available forces. This ensures that approximately the same number of energies and forces is used. The training process consists of two steps: Before the actual training a scaling routine is applied to the training configurations in order to normalize the symmetry functions to an input between zero and one. This increases the flexibility of the NN. Secondly the training process evaluates the weights and biases of the nodes.
The MD simulations are conducted with LAMMPS [3], a popular MD simulation package. The weight data file, the NN configuration file, as well as the initial configuration are needed to perform the time evolution. The calculations for the RDF and the diffusion coefficient are both based on the MD simulation of a liquid mW configuration at $T = 298$K as well as an ice configuration at $T = 150$K. The simulation time for the RDFs and diffusion coefficients are 10ps and 20ps respectively. In all cases the used integration scheme is the velocity Verlet algorithm, the time step is 0.5fs with a Nosé-Hoover thermostat for a NPT ensemble. The pressure is chosen to be 1atm for all simulations.

[3] `https://lammps.sandia.gov/`, last time accessed on 28-07-2018

10 Calculation and Visualization of SBOPs for Perfect Structures

10.1 Visualization of SBOP

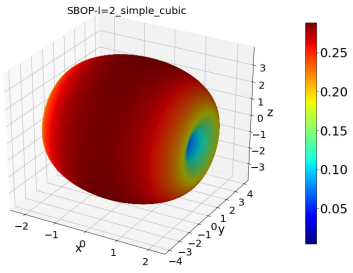
The overall goal of this thesis is to use Steinhardt bond order parameters instead of angular symmetry functions. A set of these functions should provide information on the distribution of the angles between the center atom and neighboring atoms within the cutoff radius.
The visualization of the Steinhardt bond order parameters provides a mean to identify symmetries which could reduce the amount of information provided about the local atomic environment. For a given atomic configuration and order l the values q_l are calculated for all $\phi \in (0, 2\pi)$ and $\theta \in (0, \pi)$. This is done by shifting the position of one atom of the nearest neighbors over all positions on the sphere while keeping all other atoms at fixed positions. The radius in the visualization of the corresponding SBOP value to the angle pair (ϕ, θ) is given with $r(\phi, \theta) = 1 + aq_l(\phi, \theta)$, where a is a fixed scaling factor.

10.1.1 Simple Cubic

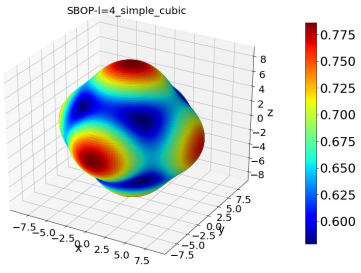
In the case of the simple cubic lattice which is given by the atom configuration as presented in table 3, the results of the visualization for the even orders can be seen in figure 16 and for the odd orders in figure 17.

Table 3: The nearest neighbor positions in the simple cubic lattice

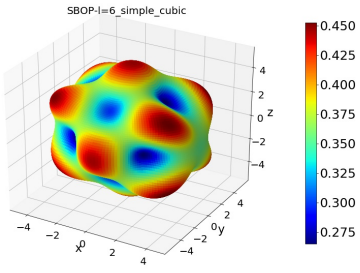
atom	1	2	3	4	5	6
x	1	-1	0	0	0	0
y	0	0	1	-1	0	0
z	0	0	0	0	1	-1



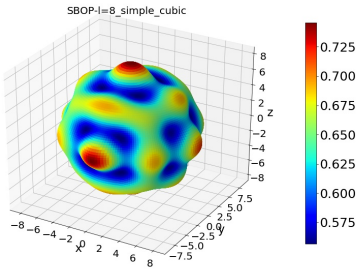
(a) SBOP of order $l = 2$



(b) SBOP of order $l = 4$



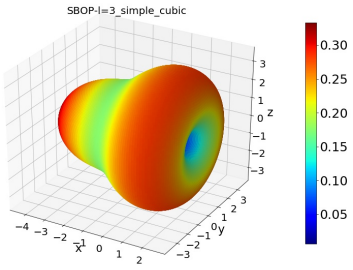
(c) SBOP of order $l = 6$



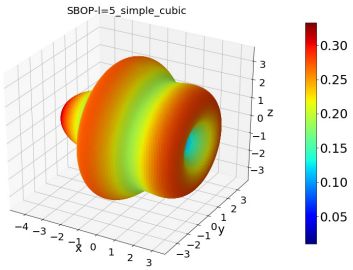
(d) SBOP of order $l = 8$

Figure 16: Visualization of SBOPs of even order for the simple cubic lattice.

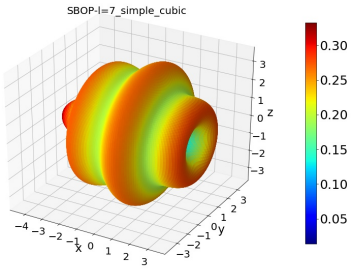
One immediate observation is that all plots for odd orders are showing a rotational symmetry, whereas the even orders do not show such a symmetry, except for the order $l = 2$.



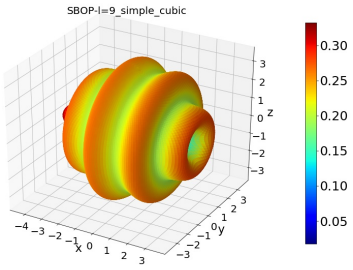
(a) SBOP of order $l = 3$



(b) SBOP of order $l = 5$



(c) SBOP of order $l = 7$



(d) SBOP of order $l = 9$

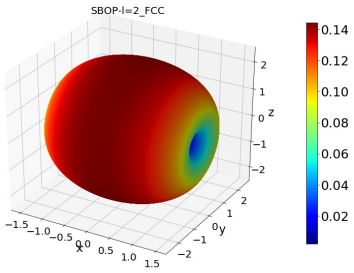
Figure 17: Visualization of SBOPs of odd order for the simple cubic lattice.

10.1.2 FCC

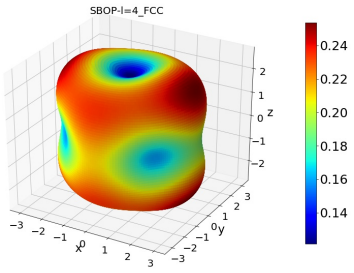
Another visualized cubic lattice type is the FFC lattice given by the atomic configuration in table 4. The results of the SBOP visualization for even orders can be seen in figure 18 and for odd orders in figure 19.

Table 4: The nearest neighbor positions in the FCC lattice

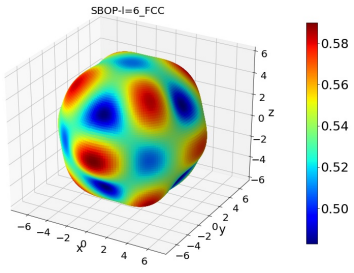
atom	1	2	3	4	5	6	7	8	9	10	11	12
x	1	-1	1	-1	0	0	0	0	1	-1	1	-1
y	1	1	-1	-1	1	-1	1	-1	0	0	0	0
z	0	0	0	0	1	1	-1	-1	1	1	-1	-1



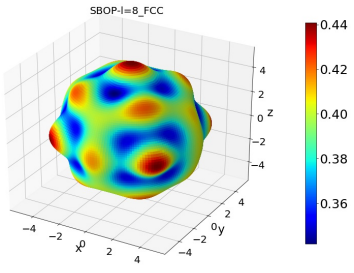
(a) SBOP of order $l = 2$



(b) SBOP of order $l = 4$

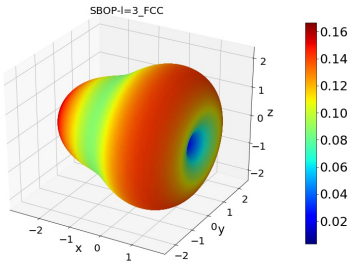


(c) SBOP of order $l = 6$

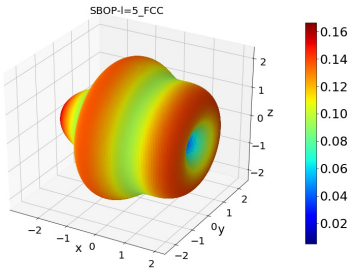


(d) SBOP of order $l = 8$

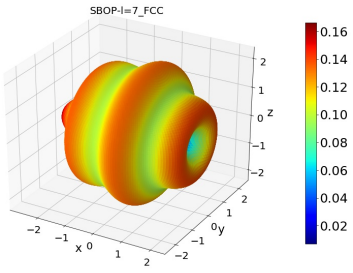
Figure 18: Visualization of SBOPs of even order for the FCC lattice.



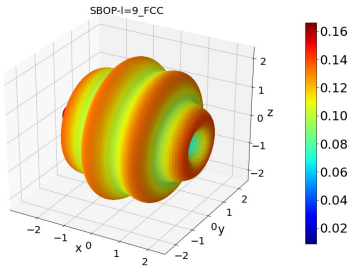
(a) SBOP of order $l = 3$



(b) SBOP of order $l = 5$



(c) SBOP of order $l = 7$



(d) SBOP of order $l = 9$

Figure 19: Visualization of SBOPs of odd order for the FCC lattice.

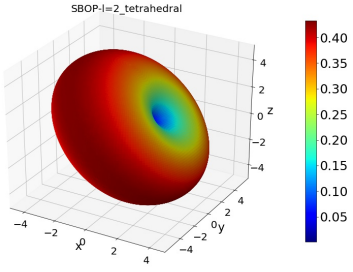
Similarly to the results for the simple cubic lattice it is found that odd orders show rotational symmetries which are not found in the even orders of the SBOPs. An exception is again the SBOP $l = 2$ where a rotational symmetry is found.

10.1.3 Tetrahedral Structure

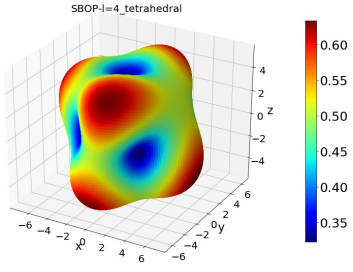
The list of nearest neighbors for the tetrahedral structure is given by table 5. The visualization of even orders of SBOPs is seen in figure 20 while the odd orders are seen in figure 21.

Table 5: The nearest neighbor positions in the tetrahedral structure

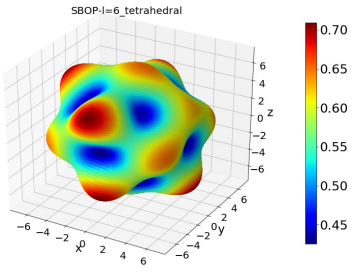
atom	1	2	3	4
x	-1	1	1	-1
y	-1	-1	1	1
z	-1	1	-1	1



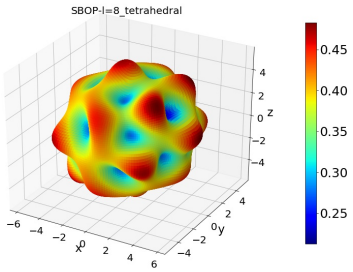
(a) SBOP of order $l = 2$



(b) SBOP of order $l = 4$

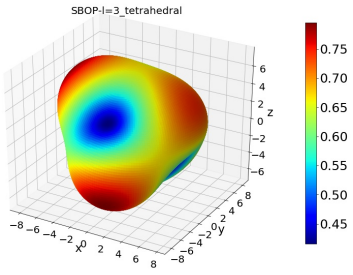


(c) SBOP of order $l = 6$

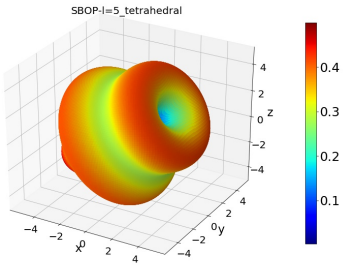


(d) SBOP of order $l = 8$

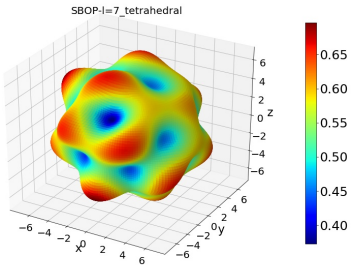
Figure 20: Visualization of SBOPs of even order for the tetrahedral structure.



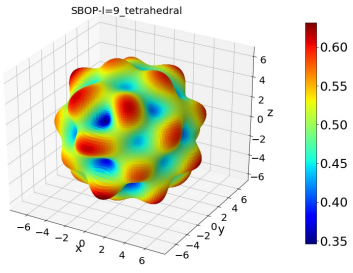
(a) SBOP of order $l = 3$



(b) SBOP of order $l = 5$



(c) SBOP of order $l = 7$



(d) SBOP of order $l = 9$

Figure 21: Visualization of SBOPs of odd order for the tetrahedral structure.

It is found, that in the case of a non-cubic symmetry like the tetrahedral structure, the distinction between even and odd orders in terms of rotational symmetry is no longer valid.

10.2 SBOPs for Crystal Structure

For a perfect lattice the SBOPs yield a fixed value, which is the reason why SBOPs of different order can be utilized to identify the crystal type in MD simulations. The results for the calculations can be seen in table 6. For cubic lattices it is typical that the odd orders of the SBOPs vanish, while the even orders obtain non-zero values, except for the order $l = 2$. When comparing the calculated values of the SBOPs with the visualization, we see that for every order, where the SBOP value vanishes the rotational symmetry in the visualization is found.

Table 6: List of SBOPs for perfect structures including only the nearest neighbors. It can be seen that all odd orders disappear for cubic symmetries, while in the case of a tetrahedral structure this is not true.

q_l	SC	FCC	BCC	tetrahedral
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.7454
4	0.7638	0.1909	0.5092	0.50924
5	0.0	0.0	0.0	0.0
6	0.3536	0.5745	0.6285	0.6285
7	0.0	0.0	0.0	0.6120
8	0.7181	0.4039	0.2128	0.2128
9	0.0	0.0	0.0	0.5179
10	0.4114	0.01286	0.65015	0.6502
11	0.0	0.0	0.0	0.3514
12	0.6955	0.6001	0.4153	0.4153

11 Derivatives for SBOP

In order to be able to analytically calculate the forces acting on the particles, the derivatives of the symmetry function in respect to the centered and neighboring particles coordinates are needed, as it can bee seen in equation 99:
The SBOP takes the has following functional form:

$$G_{101} = q_l(i) = \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l} |\frac{1}{N_b(i)}\sum_{j=1}^{N_b(i)} Y_{lm}(\vec{R_{ij}}) f_c(|\vec{R_{ij}}|)|^2} \tag{127}$$

The derivatives in respect to the coordinates of the centered atom $\vec{\nabla}_i q_l(i)$ itself as well as the derivatives of the SBOP in respect to the neighboring atoms $\vec{\nabla}_k q_l(i)$ are required for the force calculation.

$$\vec{\nabla}_i q_l(i) = C \sum_{m=-l}^{l} (\vec{\nabla}_i q_{lm})^* q_{lm} + q_{lm}^* (\vec{\nabla}_i q_{lm}) \tag{128}$$

$$\vec{\nabla}_k q_l(i) = C \sum_{m=-l}^{l} (\vec{\nabla}_k q_{lm})^* q_{lm} + q_{lm}^* (\vec{\nabla}_k q_{lm}) \tag{129}$$

The constant C is given by

$$C = \frac{2\pi}{(2l+1)} \frac{1}{q_l(i)}. \tag{130}$$

For the derivatives of the SBOP the value q_{lm} as well as its derivatives in respect to coordinates of the centered atom and the neighboring atoms are required:

$$q_{lm} = \frac{1}{N_b(i)} \sum_{j=1}^{N_b(i)} Y_{lm}(\vec{R_{ij}}) f_c(|\vec{R_{ij}}|), \tag{131}$$

$$\vec{\nabla}_i q_{lm} = \frac{1}{N_b(i)} \sum_{j=1}^{N_b(i)} f_c(|\vec{R_{ij}}|) \vec{\nabla}_i Y_{lm}(\vec{R_{ij}}) + Y_{lm}(\vec{R_{ij}}) \frac{\vec{R_{ij}}}{|\vec{R_{ij}}|} \frac{\partial f_c}{\partial |\vec{R_{ij}}|}, \tag{132}$$

$$\vec{\nabla}_k q_{lm} = \frac{1}{N_b(i)} f_c(|\vec{R_{ik}}|) \vec{\nabla}_k Y_{lm}(\vec{R_{ik}}) - Y_{lm}(\vec{R_{ik}}) \frac{\vec{R_{ik}}}{|\vec{R_{ik}}|} \frac{\partial f_c}{\partial |\vec{R_{ik}}|}. \tag{133}$$

12 NNP Training Results with SBOP

12.1 Even Order and Odd Order SBOP

Due to the large discrepancy of the results in the visualization and calculation of the SBOPs, a classification into even and odd orders of SBOPs seems to be reasonable. Here, the training progress for a NNP using radial symmetry functions and odd order SBOPs is analyzed and compared to the training progress for a NNP using radial symmetry functions and even order SBOPs. The goal is to find a minimal set of SBOPs without compromising accuracy in the training.

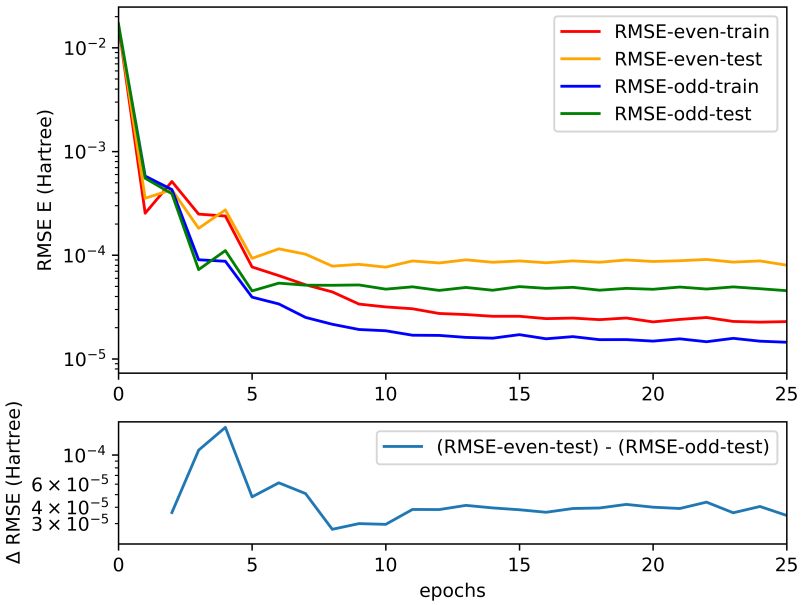


Figure 22: The energy learning curve for even and odd orders of SBOPs shows that in the case of energies odd orders perform better than the even orders. The upper chart shows a comparison of the test and training set performance measured by the RMSE for both NNP setups. The lower chart depicts the difference of the training set energy RMSE between even order SBOPs and odd order SBOPs.

Figure 22 shows the result of the training process for a training set of approximately 90 configurations, with each one consisting of about 128 water molecules. It is found, that in general the odd order SBOPs perform significantly better and the final energy RMSE is half of the error of the even orders. Furthermore, the lower chart shows that the error is consistently about the same value after about 12 epochs, which shows that the deviation is systematic. No overfitting is observed in the training procedure.

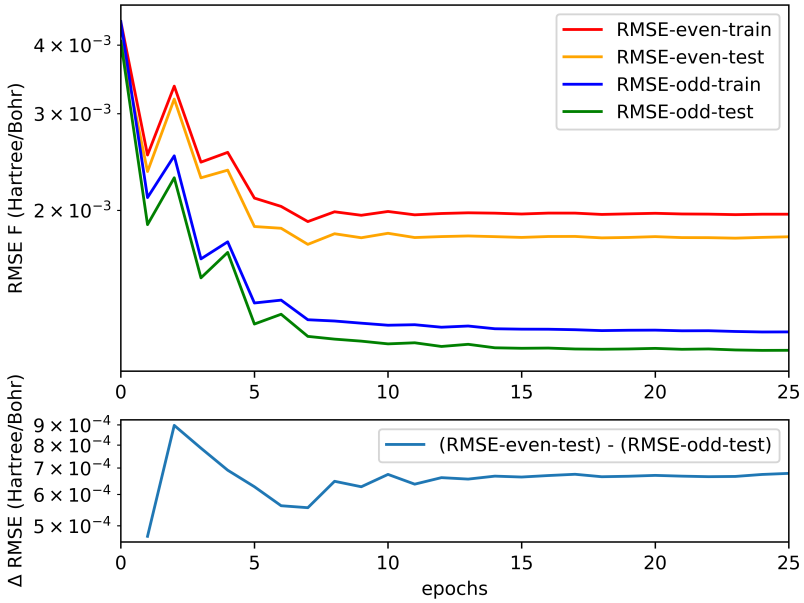


Figure 23: The comparison of learning curves of even and odd orders in term of force calculation show that odd orders outperform even orders of SBOPs in the case of forces. The lower chart shows the difference of the training set force RMSE between angular symmetry functions and odd SBOPs.

A similar situation occurs for the forces which is depicted in figure 23. Again, the odd order SBOPs perform better than the even orders, with the separating factor in the RMSE being ≈ 2. The lower chart shows that the difference of the two approaches is constant after approximately 10 epochs. It can be concluded that in both cases the odd order SBOPs outperform the even order SBOPs.
Nevertheless, when the training is performed with only odd orders of SBOPs the performance is still decreased in comparison to even and odd orders combined. For this reason, all orders are utilized in further trainings.
To summarize, it was found that in general for cubic lattices a rotational symmetry for the odd order SBOPs exists. How can this surprising result be interpreted with the knowledge from the visualization? At first one has to remember that the symmetries are only applicable to systems with a cubic lattice. This is not the general case for atomic configurations. In the training process many different configurations of mW-water are used including solid states, liquid configuration and even displaced lattices. However, water forms tetrahedral structures where

no difference between even and odd orders of SBOPs in terms of symmetries in the visualization can be observed. This makes it currently not possible to give an answer for the significantly better training with odd order SBOPs

12.2 Training Comparison Angular Symmetry functions and SBOP

The crucial part of this thesis is the comparison of performance between the angular symmetry functions and the SBOPs. For this case a NNP is trained with radial and angular symmetry functions and another one is trained with the same radial symmetry functions and SBOPs. Both approaches are trained with the same training and test set.

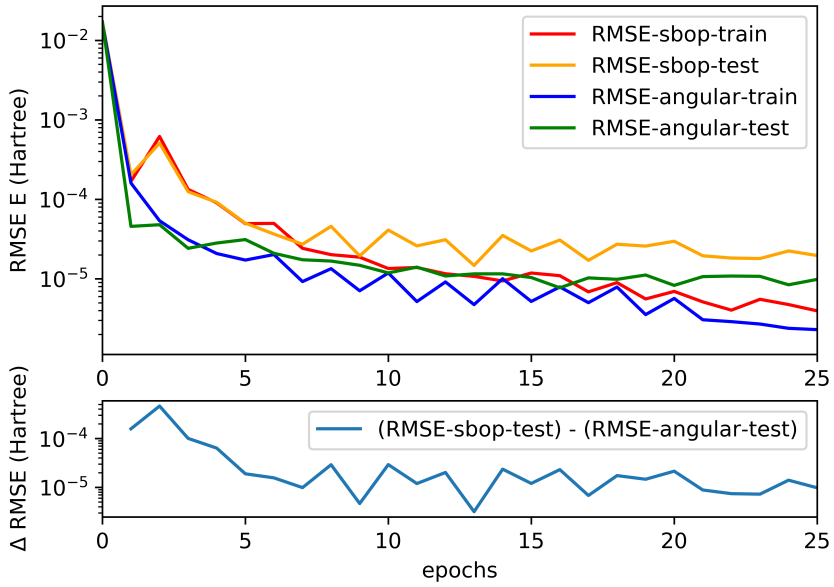


Figure 24: The energy learning curve for the NNP with angular symmetry functions and the NNP with SBOPs show that the angular symmetry functions perform better. The upper chart shows a comparison of the test and training set performance measured by the RMSE for both NNP setups. The lower chart depicts the difference of the training set energy RMSE of the results for SBOPs and the results for angular symmetry functions.

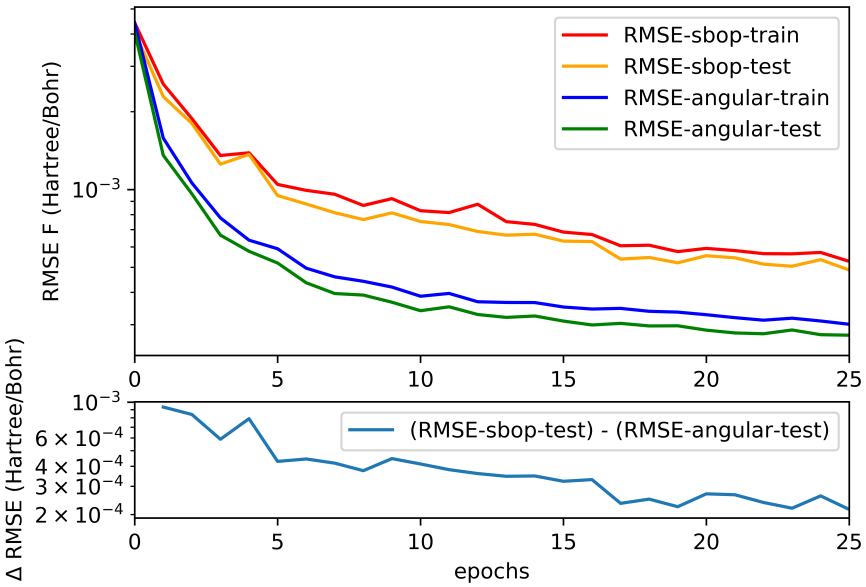
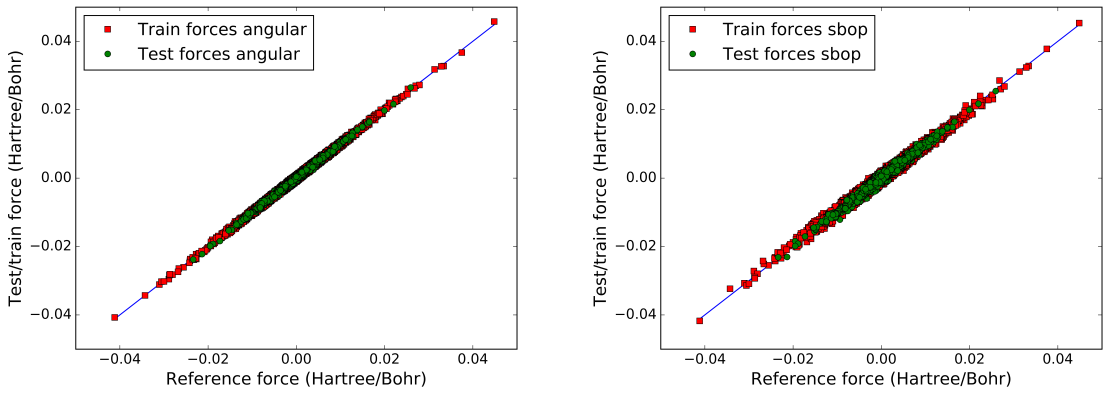


Figure 25: The force learning curve for the NNP with angular symmetry functions and the NNP with SBOPs show that the angular symmetry functions perform significantly better. The upper chart shows a comparison of the test and training set performance measured by the RMSE for both NNP setups. The lower chart depicts the difference of the training set force RMSE of the results for SBOPs and the results for angular symmetry functions.

The comparison of the energy learning curve of the two approaches can be seen in figure 24 and for the force learning curve in figure 25. In both cases it is clearly visible that the angular symmetry functions perform better. For the force RMSE the difference of the test set is decreasing over time but does not change significantly after 20 epochs. This shows that the differences are systematic and cannot be compensated by longer training sessions. Figure 26 shows the correlation of the forces of all molecules in the training and test set after the training process. The chart on the left-hand side shows the force correlation for the NNP with angular symmetry functions, whereas the chart on right-hand side shows the force correlation for the SBOP NNP. It is apparent that the deviations of the SBOP forces are larger than those of the angular symmetry functions. This is consistent with the results of the learning curves.



(a) Angular symmetry funciton

(b) SBOP

Figure 26: Comparison of test and training forces with the reference values after the training.

Another interesting question is, how the accuracy of the training is improved by adding further SBOPs into the the learning process. A comparison between the energy learning curves of a NNP with 18 SBOPs and a NNP with 45 SBOPs is seen in figure 27. The NNP with 45 SBOPs has a smaller RMSE than the NNP with 18 SBOPs. This is not surprising, since more descriptors of the local molecular environment should also provide a better accordance with the reference model. Nevertheless, the magnitude of the improvement is not justifying the increased computational costs of utilizing more SBOPs.
A similar situations occurs for the comparison of the force learning curves depicted in figure 28. The RMSE for the NNP with 45 SBOPs is smaller than the RMSE of the NNP with 18 SBOPs. Nevertheless, the differences are smaller than in the case of the energy training. Generally, it can be concluded that the optimal number of SBOPs for the training process is about 30. The accuracy of the learning process is increased with the number of SBOPs used. However, this is connected with immense computational efforts.

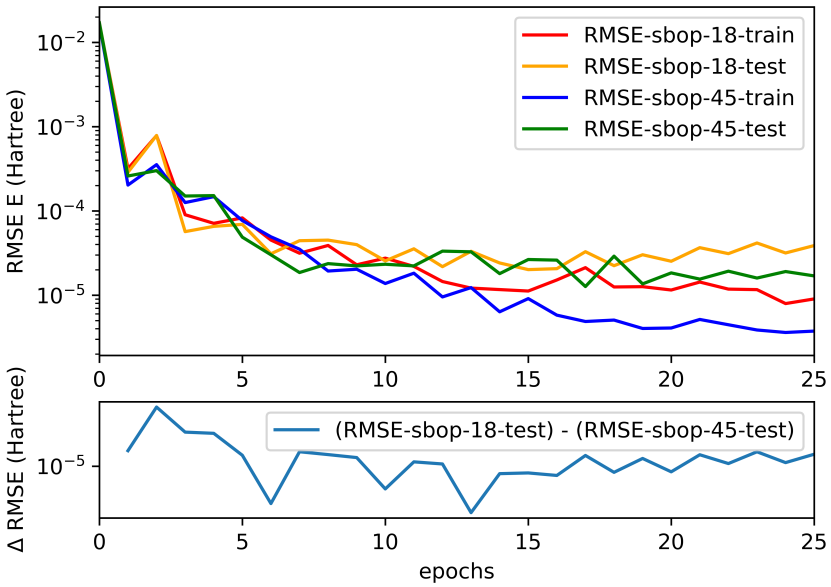


Figure 27: The energy learning curve for a NNP training with 18 and 45 SBOPs respectively: The NNP with 45 SBOPs has a significantly smaller RMSE for the energy training. The difference is consistent over the whole training process as seen in the lower chart, which depicts the differences of the test set's RMSE between the NNP with 18 SBOPs and the NNP with 45 SBOPs.

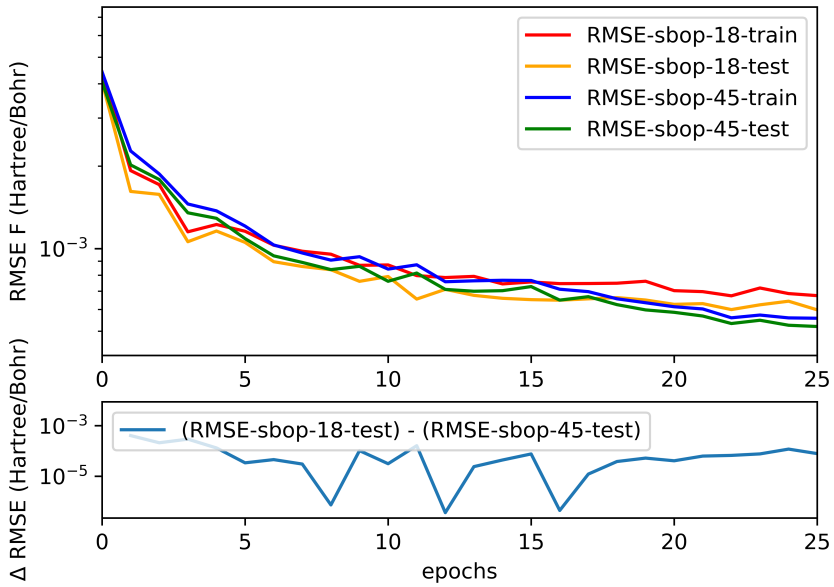


Figure 28: The force learning curve for a NNP training with 18 and 45 SBOPs respectively: The NNP with 45 SBOPs has a smaller RMSE for the force training. Also in this case the difference is consistent over the whole training process as seen in the lower chart, which depicts the differences of the test set's RMSE between the NNP with 18 SBOPs and the NNP with 45 SBOPs.

13 MD Simulations with SBOP NNP

13.1 Radial Distribution Function

In order to test the quality of the results of MD-simulations with the newly implemented SBOPs for the NNP, some basic quantities like the radial distribution function can be utilized. In this case a liquid configuration of mW-water at a temperature of $T = 298$K with 128 molecules is used. For a simulation time of 10ps the radial distribution functions for the mW-model, the NNP with angular symmetry function and the NNP with SBOPs are depicted in figure 29. Both the NNP with angular symmetry functions as well as the NNP with SBOPs are trained with the mW-model as reference model. Ideally the NNPs reproduce the same values as the reference model for the investigated quantities. It is found that the NNP with SBOPs show stronger peaks than the reference model, as well as the NNP with an-

gular symmetry functions. The NNP with SBOPs is not capable of reproducing the RDF correctly, whereas the NNP with angular symmetry functions is doing quite well.

Furthermore, the RDF of a solid phase of mW-water at $T = 150$K is depicted in figure 30. In this case the fit of the SBOP NNP is reproducing the shape of the reference model better than in the case of the liquid configuration. Nevertheless, a radial shift of the RDF is found, which is increasing for bigger values of r. This findings indicate that the NNP with SBOPs performs better in solid configurations than in liquids.

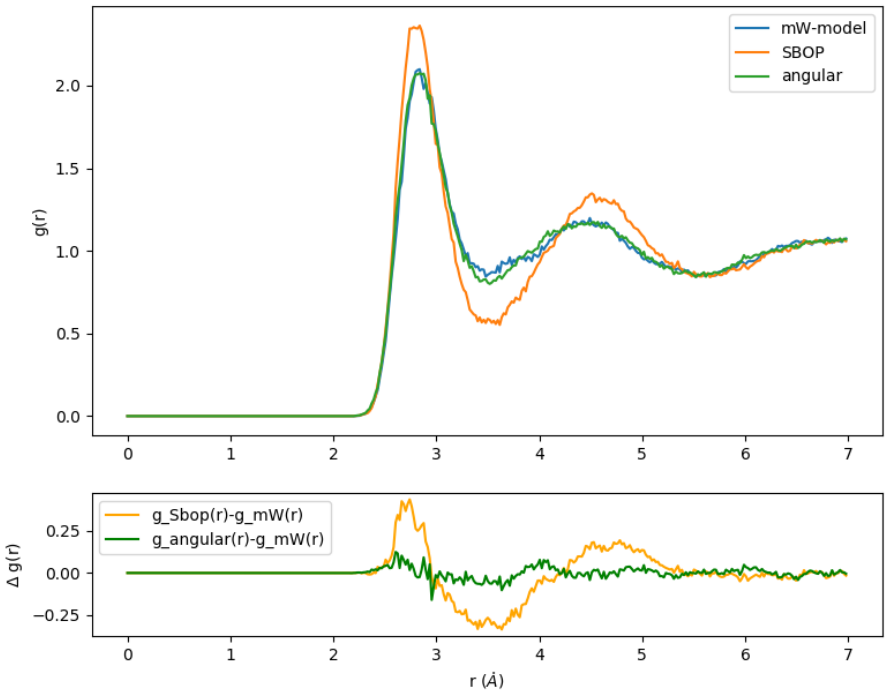


Figure 29: Liquid configuration at $T = 298$K: The upper chart depicts the radial distribution function for three different calculations of mW-water at $T = 298$K. The mW-model of water is represented by the blue line whereas the NNP with angular symmetry functions is represented by the green line. The newly implemented method of NNP with SBOPs is given by the yellow line. The lower chart shows the deviations of the two NNP approaches to the reference mW-model.

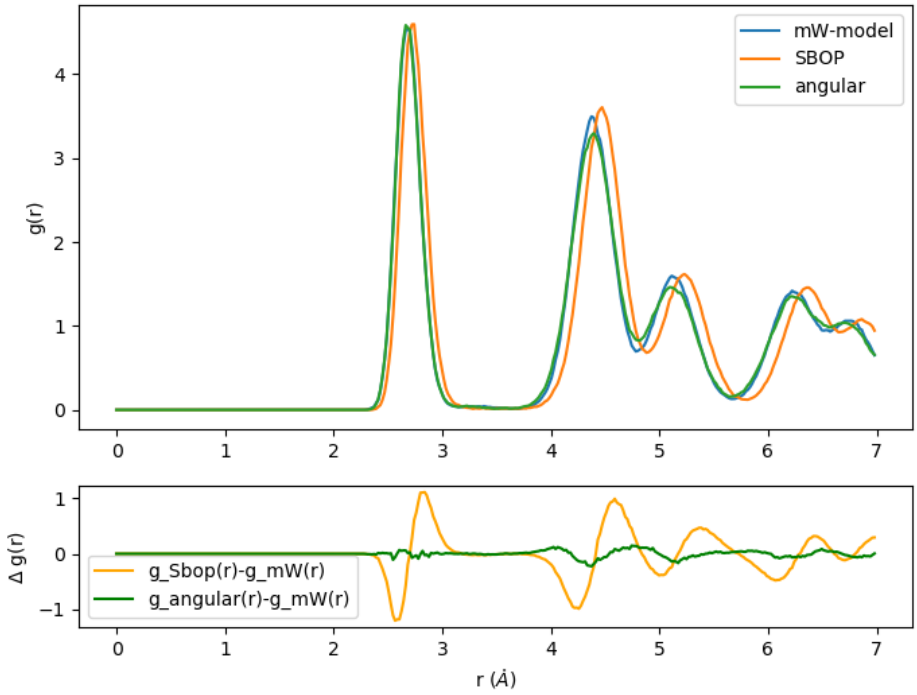


Figure 30: Solid configuration at $T = 150$K: The upper chart depicts the radial distribution function for three different calculations of mW-water at $T = 150$K. The mW-model of water is represented by the blue line whereas the NNP with angular symmetry functions is represented by the green line. The newly implemented method of NNP with SBOPs is given by the yellow line. The lower chart shows the deviations of the two NNP approaches to the reference mW-model.

13.2 Diffusion Coefficients

The diffusion coefficient in a molecular dynamics simulation can be calculated via the mean squared displacement (MSD) of the particles as shown in equation 50. Here, the diffusion coefficient for three different models and a liquid configuration at $T = 298$K is calculated by applying a linear fit with minimization of the RMSE (numpy.polyfit[4]) on the mean squared distance. The results can be seen in figure 31, where the mW-water is compared with the NNP trained with two different types

[4] `https://bit.ly/2jd9dqt`, last time accessed on 28-07-2018

Table 7: This table contains results of the calculation of the diffusion coefficient at $T = 298$K. The reference value for mW-water is from the original paper of Molinero and Moore ([24]). The difference between the mW-model and the mW-test value is caused by finite size effects.

model at $T = 298$K	mW	mW-test	angular	SBOP	experimental
$D(10^{-5}\text{cm}^2/\text{s})$	6.5	4.5	3.2	2.2	2.3

of symmetry functions: angular symmetry functions and the newly implemented SBOPs. A comparison to experimental values, as well as a published diffusion coefficient for the mW-model, are compiled in table 7.

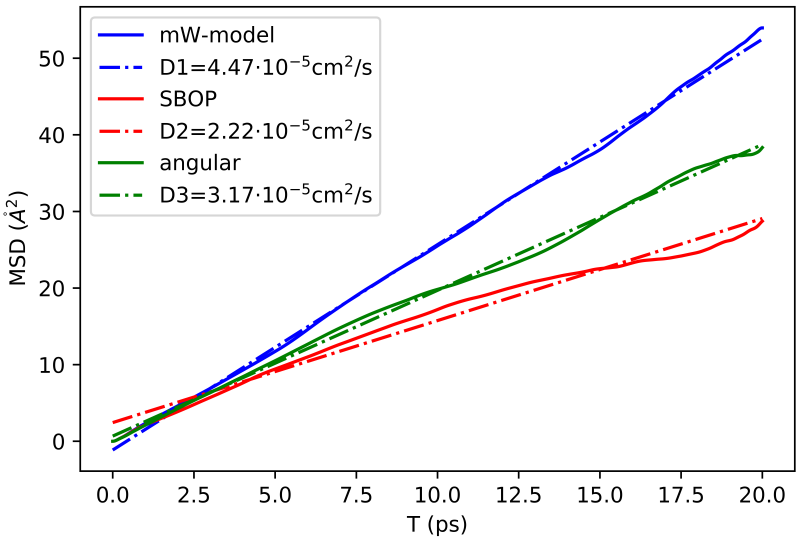


Figure 31: The diffusion coefficient can be calculated via the slope of the time evolution of the mean squared displacement. The blue line is the simulation result for mW-water model while the dotted line represents the linear fit. The green line represents the results of the NN with angular symmetry functions and the red line represents the results of the NN with SBOPs.

It is found that both, the angular symmetry functions, as well as SBOPs underestimate the mobility of the molecules in comparison to the reference value of the pure mW-model. The reference value for the diffusion coefficient is $D = 6.5 \cdot 10^{-5}\text{cm}^2/\text{s}$.

Nevertheless, due to missing finite size correction the used reference value for mW-water is the one found for the same initial configuration with the mW-model, which is $4.5 \cdot 10^{-5}\text{cm}^2/\text{s}$. The NNP with SBOPs fails to reproduce the diffusion coefficient correctly and underestimates it by a factor of approximately 2. It is worth to mention that also the NNP with angular symmetry functions fails to find the correct value and lies somewhere between the reference value and the NNP with SBOPs.

14 Conclusion

The goal of this master's thesis was the implementation of SBOPs which are used to describe the angular distribution of the local molecular environment for a NNP in order to replace the angular symmetry functions. The upsides of SBOPs are, that they provide a simpler and more generic parametrization, making the setup process more straightforward. Each SBOP is characterized only by the order l and and the cutoff radius r_c. The simplification is immediately apparent, when comparing this to the more complex form in equations 96 and 97 with many free parameters.
The three dimensional visualization of the SBOPs revealed, that for perfect cubic lattices the odd order SBOPs display a rotational symmetry. This perfectly coincides with vanishing values of the SBOP for odd orders in cubic systems. An investigation of the learning curves led to the conclusion, that the utilization only of odd orders gives significantly better results in the energy and force RMSE for the test set. Nevertheless, the accuracy suffers in comparison to the usage of both even and odd orders. For this reason both are implemented for further investigations.
The direct comparison of learning curves from a NNP with angular symmetry functions and a NNP with SBOPs showed, that in general the angular symmetry functions yield a smaller error than the SBOPs. This leads to the conclusion that SBOPs are not as capable in describing the general angular distribution of an atom's local environment as the angular symmetry functions are. However, the SBOPs perform extremely well in determining ordered structures. For the learning curve of the SBOPs the RMSE is roughly by a factor 2 larger than the RMSE of the angular symmetry functions. This is the case for both energies and forces.
Molecular dynamics simulations performed with LAMPPS at a fixed temperature of $T = 298\text{K}$ and $T = 150\text{K}$ were conducted in order to test the applicability of the

SBOP NNP as driver of a MD simulation. To make the results comparable, further simulations were performed, one with a NNP utilizing angular symmetry functions and another simulation based on the mW-model as reference. For all three simulations the same initial configuration and conditions are used. The RDF shows that the SBOP NNP tends to produce more pronounced peaks in the RDF of liquids, compared to the reference model, whereas the angular symmetry function is almost identical to the reference model. In the case of solids the results of the SBOP NNP are closer to the reference model. The calculation of the diffusion coefficient leads to similar results. The NNP with SBOPs underestimates the diffusion coefficient by a factor of 2 with the value $D_{SBOP} = 2.2 \cdot 10^{-5}\text{cm}^2/\text{s}$, while the reference simulation yields a value of $D_{mW} = 4.5 \cdot 10^{-5}\text{cm}^2/\text{s}$. The angular symmetry function yields a value of $D_{angular} = 3.2 \cdot 10^{-5}\text{cm}^2/\text{s}$.

Despite the outstanding performance of SBOPs in the determination of local ordered structures, i.e. crystal lattices, they are not the best choice for a general descriptor of the angular distribution of an atom's local environment and are outperformed in this task by angular symmetry functions.

Abstract

In dieser Arbeit wird ein neuer Typ von Symmetriefunktionen als Input für ein Neural Network Potential (NNP) untersucht. Die Symmetriefunktionen werden an Stelle von kartesischen Koordinaten verwendet, da von den Eingangsgrößen zusätzlich Invarianzen erfüllt werden müssen. Es gibt zwei Arten von Symmetriefunktionen für das NNP: radiale Symmetriefunktionen sowie Winkelsymmetriefunktionen. In dieser Arbeit werden die Winkelsymmetriefunktionen durch sogenannte Steinhardt bond order parameter (SBOP) ersetzt, welche eine breite Anwendung in der Physik sowie Chemie finden. Die bis dato verwendeten Winkelsymmetriefunktionen haben den Nachteil, dass deren funktionale Form sehr variantenreich ist und dadurch die Wahl der besten Funktionen und dazugehörigen Parameter erschwert wird. Anders ist es bei den SBOPs: Hier reicht es, ein generisches Set verschiedener Ordnungen zu verwenden, um die lokale Winkelverteilung der Nachbaratome zu reproduzieren. Es zeigt sich, dass für kubische Gitter und ungerade Ordnungen der SBOPs die dreidimensionale Visualisierung durchwegs Rotationssymmetrien aufweist. Dies koinzidiert mit dem Fakt, dass für ungerade Ordnungen die berechneten Werte von idealen kubischen Kristallen null sind. Nichtsdestotrotz wurde gezeigt, dass SBOPs in der Anwendung auf ein NNP schlechtere Ergebnisse erzielen als die bisher eingesetzten Winkelsymmetriefunktionen. Konsistent zeigen die Lernkurven eine größere Abweichung vom Referenzmodell mW-Wasser als die Winkelsymmetriefunktionen. Obwohl SBOPs sehr effizient im Auffinden von kristallinen Strukturen sind, fehlt ihnen das Vermögen zur ganzheitlichen Beschreibung der Winkelverteilung der Nachbaratome. Dies ist auch deutlich an den großen Abweichungen in Ergebnissen aus molekulardynamischen Simulationen sichtbar. Die radiale Verteilungsfunktion zeigt deutliche Abweichungen zum Referenzmodell, sowie Simulationen mit Winkelsymmetriefunktionen. Der berechnete Diffusionskoeffizient wird ebenfalls um ca. eine Faktor 2 unterschätzt.

Acknowledgements

I would like to thank everybody who supported me at the writing of my master's thesis. Special thanks to Christoph Dellago for his supervision, Andreas Singraber and Clemens Moritz for materials and their outstanding scientific expertise, Nils Clees, Matthias Kiss, Jakob Michl for their help and Max Innerbichler for great discussions, as well as to my parents who enabled my education financially and always supported my decisions.

References

[1] H Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4):2384–2393, 1980.

[2] G Arfken. *Mathematical Methods for Physicists*. Academic Press San Diego, 1985.

[3] F Azevedo, L Carvalho, L Grinberg, J Farfel, R Ferretti, R Leite, W Filho, R Lent, and S Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541, 2009.

[4] J Barzilai and J Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[5] J Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, 2011.

[6] J Behler. Representing potential energy surfaces by high-dimensional neural network potentials. *Journal of Physics: Condensed Matter*, 26(18):183001, 2014.

[7] H Berendsen, R Grigera, and T Straatsma. The missing term in effective pair potentials. *Journal of Physical Chemistry*, 91(24):6269–6271, 1987.

[8] H Berendsen, J Postma, W van Gunsteren, and J Hermans. Interaction models for water in relation to protein hydration. In *Intermolecular Forces*, pages 331–342. Springer, 1981.

[9] D Frenkel and B Smit. *Understanding molecular simulation: From algorithms to applications*, volume 1. Elsevier, 2002.

[10] A Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2017.

[11] M Green. Markoff random processes and the statistical mechanics of time-dependent phenomena. ii. irreversible processes in fluids. *The Journal of Chemical Physics*, 22(3):398–413, 1954.

[12] G Hinton, L Deng, D Yu, G Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, T Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[13] J Jones. On the determination of molecular fields.—ii. from the equation of state of a gas. *Proc. R. Soc. Lond. A*, 106(738):463–477, 1924.

[14] W Jorgensen, J Chandrasekhar, J Madura, R W Impey, and M Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983.

[15] R Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[16] S Kapfer, W Mickel, K Mecke, and G Schröder-Turk. Jammed spheres: Minkowski tensors reveal onset of local crystallinity. *Physical Review E*, 85(3):030301, 2012.

[17] R Kubo. Statistical-mechanical theory of irreversible processes. i. general theory and simple applications to magnetic and conduction problems. *Journal of the Physical Society of Japan*, 12(6):570–586, 1957.

[18] T Kühne. Second generation car–parrinello molecular dynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(4):391–406, 2014.

[19] W Lechner and C Dellago. Accurate determination of crystal structures based on averaged local bond order parameters. *The Journal of Chemical Physics*, 129(11):114707, 2008.

[20] C Looney et al. *Pattern recognition using neural networks: theory and algorithms for engineers and scientists*. Oxford University Press New York, 1997.

[21] W McCulloch and W Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[22] W Mickel, S Kapfer, G Schröder-Turk, and K Mecke. Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter. *The Journal of Chemical Physics*, 138(4):044501, 2013.

[23] P Minary, G Martyna, and M Tuckerman. Algorithms and novel applications based on the isokinetic ensemble. i. biophysical and path integral molecular dynamics. *The Journal of Chemical Physics*, 118(6):2510–2526, 2003.

[24] V Molinero and E Moore. Water modeled as an intermediate element between carbon and silicon. *The Journal of Physical Chemistry B*, 113(13):4008–4016, 2008.

[25] E Moore and V Molinero. Ice crystallization in water's "no-man's land". *The Journal of Chemical Physics*, 132(24):244504, 2010.

[26] T Morawietz, A Singraber, C Dellago, and J Behler. How van der waals interactions determine the unique properties of water. *Proceedings of the National Academy of Sciences*, 113(30):8368–8373, 2016.

[27] S Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984.

[28] A Rajavelu, M Musavi, and M Shirvaikar. A neural network approach to character recognition. *Neural networks*, 2(5):387–393, 1989.

[29] A Reinhardt, J Doye, E Noya, and C Vega. Local order parameters for use in driving homogeneous ice nucleation with all-atom models of water. *The Journal of Chemical Physics*, 137(19):194504, 2012.

[30] U Röthlisberger, M Sprik, and J Hutter. Time and length scales in ab initio molecular dynamics. In *Bridging time scales: Molecular simulations for the next decade*, pages 413–442. Springer, 2002.

[31] T Schlick. *Molecular modeling and simulation: an interdisciplinary guide: an interdisciplinary guide*, volume 21. Springer Science & Business Media, 2010.

[32] P Steinhardt, D Nelson, and M Ronchetti. Bond-orientational order in liquids and glasses. *Physical Review B*, 28(2):784, 1983.

[33] F Stillinger and T Weber. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.

[34] M van Gerven and S Bohte. *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

[35] L Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1):98, 1967.

[36] J Witkoskie and D Doren. Neural network models of potential energy surfaces: Prototypical examples. *Journal of Chemical Theory and Computation*, 1(1):14–23, 2005.

[37] Y Yuan. Step-sizes for the gradient method. *AMS IP Studies in Advanced Mathematics*, 42(2):785, 2008.