DIPLOMARBEIT

Van-der-Waals interactions in neural-network force fields

Ausgeführt am Institut für
Materialchemie
der Technischen Universität Wien

unter der Anleitung von Univ.Prof. Georg Kent Hellerup Madsen und Dr. Jesús Carrete Montaña

durch

Nina Bučková

18.8.2023 [redacted]

Datum Unterschrift (Studentin)

Erklärung zur Verfassung der Arbeit

Bc Nina Bučková

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. August 2023

[redacted]

Nina Bučková

Acknowledgements

First of all, I would like to thank my supervisors, Prof. Georg Madsen and Dr. Jesús Montaña, for giving me the chance to work on this thrilling topic and for their excellent supervision. I'm very grateful to all members of Prof. Madsen's group, especially Ralf Wanzenböck, Nico Unglert, Sebastian Bichelmaier, Lijun Pan, Peter Kovacs, and Johannes Schörghuber, for offering me a helping hand whenever I've experienced problems with software or the computational cluster, and also for taking the time to discuss my work.

I am very thankful to Prof. Madsen and Infineon company for granting me the scholarship that enabled me to participate in the 36th Workshop on Novel Materials and Superconductors with a poster presentation and to spend two weeks in Ås at the Norwegian University of Life Sciences. During this time, I had the opportunity to discuss my work with Dr. Kristian Berland and his group and work on the topic under his supervision. I am very grateful to him and his group members—Rasmus Andre Trånas, Elin Dypvik Sødahl, Seyedmojtaba Seyedraoufi, Prutha Nagaraja, Øven Andreas Grimenes, and Dario Alejandro Leon Valido—for the many inspiring discussions and the countless enjoyable moments outside of the university.

Many thanks also go to my extraordinary parents for being an endless source of inspiration, to my amusing sister Linda, my boyfriend Ayurzana Purevdorj and his parents, and my dear friends Mija & Leon, Alex de Cruz, Miriam Jeníčková, Samo Gonos, Marek Mrva, and Šima Müllerová.

Abstract

In recent years, new methods based on machine-learning techniques and neural-networks have been developed to calculate the thermodynamic and spectroscopic properties of various systems with high precision and low computational costs. However, most current neural-network force fields have a significant limitation: they only take into account local, short-range interactions between atoms and molecules, while largely ignoring the long-range interactions that may strongly influence the properties of materials. This limitation may have significant consequences for predicting the physical properties of materials that are primarily determined by long-range interactions. This master thesis discusses the extent to which long-range interactions are taken into account in the current neural-network force field NeuralIL, and how this limitation affects the accuracy of predicted density using molecular dynamics simulations for water as a case study. Moreover a new implementation of the DFT-D3 method for calculating dispersion interactions using the high-performance machine-learning framework JAX will be presented. Finally, the suitability of vdW-DF exchange-correlation functionals to correct for those effects in condensed-phase water systems will be analysed and compared to DFT calculations with RPBE functional with and without the DFT-D3 dispersion correction.

Contents

CHAPTER 1

Introduction

Van der Waals interactions (vdW) are non-bonded interactions that are pervasive in nature, playing a critical role in the cohesion and structural properties of materials. These interactions, governed by the quantum-mechanical behaviour of constantly fluctuating electron clouds, exhibit nonadditive and long-range characteristics. At the nanoscale, vdW interactions can extend up to distances of 10 nm or more [1]. In the field of theoretical chemistry, various methods exist to calculate these interactions. Accurate calculations can be performed using coupled-cluster methods such as CCSD(T), while approximate approaches within the framework of density functional theory (DFT) include specialised van der Waals density functionals [2] and DFT-D methods [3].

Until recently, theoretical chemistry predominantly relied on physics-based methods, such as DFT. However, a groundbreaking development in theoretical chemistry and materials science has emerged, introducing a novel methodology that leverages the power of big data science. This innovative approach integrates machine-learning tools, particularly neural-networks, into the field. The concept showcases the remarkable ability of large datasets to generate highly accurate models, even with relatively simple underlying algorithms such as linear regression. Neural-networks possess significant potential for predicting the physical properties of diverse chemical compounds, offering a substantial advantage over older, well-established techniques.

Neural-networks have emerged as a promising tool particularly in the context of molecular dynamics (MD) simulations. MD simulations are widely used to investigate the physical properties of compounds. These simulations involve solving Newton's equations of motion at each time step, enabling the system to evolve over time. To perform MD simulations, it is crucial to calculate the forces of each configuration generated during the simulation. Traditionally, two primary methods have been employed to calculate the potential energy: ab initio methods, such as DFT, which accurately compute energy using quantum mechanical principles, and classical force fields, which approximate the potential energy surface (PES) using functional forms based on analytical results and a limited set of parameters. While DFT calculations are known for their precision, they can be computationally expensive. In contrast, classical force fields offer faster computations but at the cost of reduced accuracy. The challenge lies in the large number of time steps involved in an average MD simulation, often numbering in the millions. With each time step, a new configurations is generated, necessitating the calculation of potential energies and forces. Slow methods can significantly extend the simulation duration, potentially taking years to complete. On the other hand, imprecise potential energy and force calculations can lead to inaccuracies in the resulting physical properties.

Neural-network force fields (NNFFs) offer a solution to address these challenges. Rather than directly calculating potential energies using computationally expensive methods like DFT for millions of configurations during a simulation, a NNFF can be trained using a set of several thousand structures. This dataset, comprised of high-quality data, enables a NNFF to learn and approximate the potential energy surface (PES) with great accuracy. As a result, the trained NNFF can efficiently calculate the potential energy of any configuration within the same system, offering both high accuracy and significantly reduced computational time compared to traditional methods like DFT. Until today many NNFF have been developed and successfully used. For example the RuNNer NNFF [4] developed by Prof. Behler and his coworkers was successfully used for example to predict the relative stability of the hexagonal and cubic ice phases [5] or to calculate reaction probabilities of nitrogen molecules on a Ru(001) surface with height accuracy [6].

However, to achieve accurate predictions, it is crucial for the NNFF to effectively model the interatomic interactions within the material of interest. This task becomes particularly challenging when it comes to adequately representing vdW interactions. Modelling vdW interactions with sufficient accuracy poses a considerable challenge, as they are inherently non-local and long-range in nature. The ability of the NNFF to capture and reproduce these interactions plays a crucial role in accurately predicting the physical properties of the system under study.

Many NNFFs employed today utilise descriptors based on the assumption of chemical locality, where an extensive property of a configuration is decomposed into contributions from local atomic environments represented by descriptors. This approximation was applied successfully to predict potential energies, forces and properties of ionic liquids [7], water [5], [8] and many more compounds. This approach allows training the NNFF on small fragments and using it to make predictions for larger systems composed of these fragments [9]. However, when studying systems with dominant long-range and nonadditive interactions like vdW interactions, the locality approximation often fails to adequately describe their physical properties [10]. It has been shown, for instance, that omitting vdW interactions from the training set of a NNFF used in molecular dynamics simulations for bulk water leads to inaccurate predictions of water properties, such as density [8]. This discrepancy arises because descriptors, typically with a radius of 4 to 6 Å, can only capture interactions between atoms within this radius. However, vdW interactions, such as London dispersion interactions, persist between atoms at distances of 50 Å and even greater. These interactions can contribute significantly to the total potential energy and possibly cannot be adequately captured by the local descriptors commonly used.

Considering the limitations imposed by the locality approximation, I conducted an analysis to assess the feasibility of modelling dispersion interactions and transferability in water systems using NeuralIL, a neural-network force field developed by Prof. Madsen's group [7]. My analysis involved two datasets: one comprising bulk water configurations and the other consisting of water clusters. The datasets and the findings of my analysis are presented in Section 3.1.

In Section 3.2, I further investigated the impact of the locality approximation on the accuracy of the predicted density of bulk water at a temperature 250 K and a pressure of 1 bar. To accomplish this, we employed two NeuralIL models trained on DFT forces with and without the dispersion correction, computed using the DFT-D3 method, which are analysed in detail in Section 3.1, and performed an MD simulation in a NPT ensemble and using JAX-MD simulation package [11]. The resulting water densities were compared with the density values given in a study carried out using another NNFF that was trained on data with and without DFT-D3 correction [8]. The results of this investigation, including their implications, are thoroughly discussed.

Lastly, in Section 3.4, I explore the potential of van-der-Waals density functionals to create a more accurate water dataset, surpassing the limitations of previous research methodologies. This investigation sheds light on the effectiveness of these density functionals in capturing the intricate properties of bulk water.

During the course of this master's thesis, a program for calculating the DFT-D3 correction was developed. This program, which uses the Python library JAX, is based on the Torch-dftd program [12], [3]. In Section 3.3, the program is thoroughly described, including a comparison with the original Torch-dftd version and the persistent challenges encountered during its development.

By addressing these key aspects, I aim to enhance the general understanding of the modelling of dispersion interactions, improve the accuracy of predicted water properties, and advance the development of more reliable and comprehensive approaches in theoretical chemistry and molecular dynamics simulations.

CHAPTER 2

Theoretical background

2.1 Van-der-Waals interactions in theoretical chemistry

The exact solution of the Schrödinger equation contains the vdW contribution to the energy since it is contained in the exact formulation of the electron interaction energy. However, the explicit solution can not be obtained easily especially in case of systems containing many atoms. In theoretical chemistry the golden standard for modelling of vdW interactions are the wave-function methods such as coupled-cluster [3]. However, the most popular methods in condensed matter science nowadays are based on DFT because the calculations can be carried out faster.

According to the Born-Oppenheimer approximation, the Hamilton operator, which determines the total energy, can be split into a nuclear Hamiltonian and another Hamiltonian that contains the contribution of electrons to the total potential energy [13]. The latter is given by :

$$E[\Psi] = E_{\text{kin}}[\Psi] + E_{\text{ext}}[\Psi] + E_{\text{ee}}[\Psi] \tag{2.1}$$

where E_{kin} is the kinetic energy of the electrons, E_{ext} is an external energy due to the interaction of the electrons with the external (nuclear) field and E_{ee} stands for the electron-electron interaction energy. The electron-electron interaction energy can be split into two parts, the first of which reflects the uncorrelated probability density and is called Hartree energy [13]. The second one is given by the correlated pair-probability density Π_{xc} and reflects the correlated movement of electrons.

$$E_{\text{ee}}[\Psi] = E_{\text{H}}[n] + U[\Pi_{\text{xc}}] \tag{2.2}$$

Van-der-Waals interactions consist of three interactions : Keesom, Debye and London dispersion interactions. The Keesom interactions result from the interactions between molecules with permanent dipole moments and it can be either attractive or repulsive, depending on the relative orientation of the molecules [14]. Debye interactions originate from the polarisation of nonpolar molecules or atoms by an electric field produced by a molecule having a permanent dipole moment and therefore they are attractive. On the other hand, the dispersion interaction is created because the movements of electrons are correlated, which results in a transient dipole moment on the atoms in a molecule or solid that can induce dipole moments on other distant unpolar molecules or atoms [14]. The interaction energies from Keesom and Debye interactions result from the change of the ground state electron density and therefore they are fully contained in the Hartree energy [14]. However, the dispersion interaction is caused by a dynamic effect linked to the correlated motion of electrons, thereby is contained only in $U[\Pi_{\mathrm{xc}}]$. The exaggerated repulsion incorporated in $E_{\mathrm{H}}[n]$ is compensated partly by the dispersion interaction, which is also attractive.

Unfortunately calculating Π_{xc}, and thereby $U[\Pi_{\mathrm{xc}}]$, accurately is computationally intractable for even intermediate size systems. Within DFT, this term is part of the Kohn-Sham energy, which is a functional of the ground-state electron density n, and is specifically included in the exchange-correlation energy $E_{\mathrm{xc}}[n]$:

$$E_{\mathrm{KS}}[n] = T_{\mathrm{s}} + E_{\mathrm{ext}}[n] + E_{\mathrm{H}}[n] + E_{\mathrm{xc}}[n] \tag{2.3}$$

The exchange-correlation energy can be approximated using various exchange-correlation functionals. However, common functionals of the type LDA, GGA or more advanced semilocal meta-GGA functionals fail to describe the long-range electron correlation energy and therefore fail to describe the long-range part of dispersion interaction which is highly nonlocal [15].

Yet the non-relativistic exchange and correlation energy of a system can be formulated exactly on a top of a mean-field method framework like DFT using the adiabatic-connection fluctuation-dissipation theorem [15]. The central quantity in this theorem is the non-local, time-dependent density response function $\chi(\mathbf{r},\, \mathbf{r}',t,t')$, which describes the response of an electron density at position $\mathbf{r}$ and time t to a perturbation at position $\mathbf{r}'$ at time t' [15]. It is related to the nonlocal polarizability $\boldsymbol{\alpha}$ by:

$$\chi(\mathbf{r},\mathbf{r}',t,t') = \nabla_{\mathbf{r}} \cdot \nabla_{\mathbf{r}'} \cdot \boldsymbol{\alpha}(\mathbf{r},\mathbf{r}',t,t') \tag{2.4}$$

This fluctuation-dissipation theorem states that the correlation energy can be calculated for the stationary state using the time-independent, frequency-dependent dynamic polarizability $\boldsymbol{\alpha}(\mathbf{r},\mathbf{r}',\omega)$ using this equation:

$$E_{\mathrm{corr}} = \frac{1}{2\pi}\int_0^\infty d\omega \int_0^1 d\lambda \iint d\mathbf{r}d\mathbf{r}' Tr\{[\alpha_\lambda(\mathbf{r},\mathbf{r}',i\omega) - \alpha_{\lambda=0}(\mathbf{r},\mathbf{r}',i\omega)]\mathbf{T}(\mathbf{r},\mathbf{r}')\} \tag{2.5}$$

where $\mathbf{r}$, $\mathbf{r}'$ are the spatial coordinates, ω is the frequency, λ is a coupling strength that connects the Kohn-Sham system of noninteracting electrons ($\lambda = 0$) to the real system of fully interacting electrons ($\lambda = 1$), $\boldsymbol{\alpha}_{\lambda=0}$ is the polarizability response of a Kohn-Sham system, $\boldsymbol{\alpha}_\lambda$ is the polarizability of a system with interacting electrons and $\mathbf{T}(\mathbf{r}, \mathbf{r}')$ stands for a dipole potential related to the Coulomb potential $v(\mathbf{r},\mathbf{r}')$ via [15]:

$$\mathbf{T}(\mathbf{r}, \mathbf{r}') = -\nabla_{\mathbf{r}} \nabla_{\mathbf{r}'} v(\mathbf{r}, \mathbf{r}') \tag{2.6}$$

All of the currently existing methods for calculating the dispersion interactions within the DFT framework are based on calculating the long range part of the total correlation energy given by Eq.(2.5) by mainly introducing various approximations to the two central quantities : the nonlocal polarizability $\boldsymbol{\alpha}(\mathbf{r}, \mathbf{r}', \omega)$ and the effective interaction potential $v(\mathbf{r}, \mathbf{r}')$. Depending on the kind of approximations they can be classified into several categories. Two of them that are relevant for this work are the fragment-based pairwise vdW methods, to which the family of DFT-D methods belongs, and nonlocal two-point density functional approximations such as the van-der-Waals density functionals (vdW-DF) [15].

2.1.1 DFT-D3

DFT-D3 [3] is currently a widely adopted method for calculating dispersion energy in conjunction with DFT. It belongs to the family of DFT-D methods. The total energy is calculated as a sum of the Kohn-Sham energy and the dispersion energy, calculated using the two-body and three-body energies according to the following equations:

$$E_{\text{total}} = E_{\text{KS}} + E_{\text{disp}} \tag{2.7}$$

$$E_{\text{disp}} = -(E_{\text{2-body}} + E_{\text{3-body}}) \tag{2.8}$$

$$E_{\text{2-body}} = \sum_{AB} \sum_{n=6,8} s_n \frac{C_n^{AB}}{r_{AB}^n} f_{d,n}(r_{AB}) \tag{2.9}$$

where the first sum runs over atomic pairs AB and the second one over different orders of the averaged (isotropic) dispersion coefficient C_n^{AB} ($n = 6{,}8$) for an atom pair AB, r_{AB} is their inter-nuclear distance, s_n is a global scaling factor. For $n > 6$ the value of s_n depends on the density functional that has been used in the calculation of E_{KS} while for $n = 6$ s_n is unity [3]. In order to ensure that the dispersion energy is well behaved for small inter-atomic distances (especially to avoid singularities) a damping function called damping function "zero" of the following form is applied:

$$f_{d,n}(r_{AB}) = \frac{1}{1 + 6(r_{AB}/(s_{r,n} R_0^{AB}))^{-\alpha_n}} \tag{2.10}$$

where $s_{r,n}$ is the order-dependent scaling factor of the cutoff radii R_0^{AB} and has to be adjusted for each density functional. Another parameter that is adjusted manually is the steepness parameter α_n, in such a way that the dispersion correction amounts to less than 1% of the maximal absolute value of the dispersion energy for typical covalent bond distances [3].

Contrary to the older versions of DFT-D method (such as D2 or D1) in DFT-D3 the dispersion coefficients are based on time-dependent (TD) DFT calculations. The reference dispersion coefficients of atoms A and B are calculated using polarizability values for some reference molecules of those atoms, which were chosen to be hydrides of those atoms, and subtracting the contribution of the hydrogens:

$$C_{6,\mathrm{ref}}^{AB} = \frac{3}{\pi}\int_0^{\infty} d\omega \frac{1}{m}[\alpha^{A_m H_n}(i\omega) - \frac{n}{2}\alpha^{H_2}(i\omega)] \times \frac{1}{k}[\alpha^{B_k H_l}(i\omega) - \frac{l}{2}\alpha^{H_2}(i\omega)] \tag{2.11}$$

where $\alpha^{H_2}(\mathrm{i}\omega)$ is the polarizability of the hydrogen molecule, m, n, k, l are the stochiometric factors and $\alpha^{A_m H_n}(\mathrm{i}\omega)$ is the polarizability of the reference molecule $\mathrm{A}_m\mathrm{H}_n$. The use of polarizabilities of hydrides stems from the fact that, within a compound, the polarizability of an atom tends to be lower compared to its free atom state. Since this method is supposed to be used mainly in dense matter and almost every element in the periodic table forms at least one stable hydride, they are appropriate to mimic this behaviour of the polarizability [3]. The use of reference molecules could seem to be a disadvantage at first sight because it leads to ambiguous coefficients. However, it opens the door to the inclusion of some geometrical information about the molecule in form of the coordination number CN.

Although the dynamical polarizability $\boldsymbol{\alpha}$ has a nonlocal character there are some local effects that influence it (and thus the dispersion coefficient). In fact, the contribution of an atom to the total dispersion coefficient of a molecule depends on its chemical environment. One of the most important local electronic effects is hybridization. It is induced by covalent bond formation between atoms which drastically changes the electronic structure. Fractionally occupied atomic orbitals mostly become doubly occupied, energetically lower-lying molecular orbitals. Because this increases the electronic excitation energies the resulting atomic polarizabilities and thus the C_6 coefficients are often much smaller in molecules than in free atoms [3]. Grimme et al. showed that the effect of the hybridization on the dispersion coefficient can be calculated surprisingly well using some information about the local geometry incorporated in the fractional coordination number [3], which is calculated for an atom A using the counting function:

$$\mathrm{CN}^A = \sum_{B\neq A}^{N_{at}} \frac{1}{1 + e^{-k_1(R_{A,\mathrm{cov}}+R_{B,\mathrm{cov}})/r_{AB}^{-1}}} \tag{2.12}$$

where $R_{A,\mathrm{cov}}$ and $R_{B,\mathrm{cov}}$ stand for a scaled covalent and single-bond radius of atom A and B, k_1 and k_2 are empirically derived scaling factors. The procedure for computing the dispersion coefficients is as follows. The polarizability of each element in the molecule is computed using Eq.(2.11) for at least one reference hydride molecule. If an element can be found in several hydrides with different chemical environments, all of them will be taken as a reference. A typical example would be a carbon atom, which can occur for example in isolation or as a part of C-H, ethene, ethyne or ethane. In those compounds carbon has different chemical environments (the number of bound hydrogens changes) which result in different coordination numbers of C (from 0 to 4). Next, the coordination number of the element of interest in each of the reference compounds is calculated. The dispersion coefficient for a pair of atoms A and B in their environment $C_{6,\text{ref}}^{AB}(\text{CN}^A, \text{CN}^B)$ will be calculated using the Eq.(2.11). These serve as points in an interpolation procedure using which any C_6 coefficient with any combination of CNs can be calculated [3]. The relationship between the actual C_6^{AB} coefficients and the reference coefficients is as follows:

$$C_6^{AB}(\text{CN}^A, \text{CN}^B) = \frac{Z}{W} \tag{2.13}$$

$$Z = \sum_i^{N_A} \sum_j^{N_B} C_{6,\text{ref}}^{AB}(\text{CN}_i^A, \text{CN}_j^B) L_{ij} \tag{2.14}$$

$$W = \sum_i^{N_A} \sum_j^{N_B} L_{ij} \tag{2.15}$$

$$L_{ij} = e^{-k_3[(\text{CN}^A - \text{CN}_i^A)^2 + (\text{CN}^B - \text{CN}_j^B)^2]} \tag{2.16}$$

In those equations, N_A and N_B stand for the number of reference molecules for each atomic element A and B, CN^A and CN^B are coordination numbers of the atoms A and B in the system of interest (the actual molecules or solid) and CN_i^A, CN_j^B are coordination numbers for those atoms in the reference molecules i and j.

The higher-order C_8 coefficients are calculated based on the C_6 coefficients recursively via:

$$C_8^{AB} = 3C_6^{AB}\sqrt{Q^A Q^B} \tag{2.17}$$

with

$$Q^A = s_{42}\sqrt{Z^A}\frac{\langle r^4\rangle^A}{\langle r^2\rangle^A} \tag{2.18}$$

where $\langle r^4 \rangle$ and $\langle r^2 \rangle$ are multipole-type expectation values derived from atomic densities which are averaged geometrically, Z is the nuclear charge of atom A and s_{42} is a scaling factor.

In DFT-D3 one can optionally also calculate the three-body term and add it on top of the two-body one. While the two-body term describes the additive interaction between two atoms in a compound, the three-body term describes the nonadditive interaction between three atoms simultaneously. This kind of interaction is nonadditive because the long range part of the interaction between three ground state atoms does not equal to the sum of interactions taken in pairs. The three-body dispersion energy term is described using the following equation:

$$E_{\text{3-body}} = \sum_{ABC} f_{d,(3)}(\overline{r}_{ABC}) E^{ABC} \tag{2.19}$$

where $f_{d,(3)}$ is the three-body damping function, analogous to the two body one, and E^{ABC} is the three-body interaction energy given by the Axilrod-Teller-Muto equation :

$$E^{ABC} = \frac{C_9^{ABC}(3\cos\theta_a \cos\theta_b \cos\theta_c + 1)}{(r_{AB} r_{BC} r_{CA})^3} \tag{2.20}$$

The θ are internal angles of the triangle formed by r_{AB}, r_{AC}, r_{BC} and C_9 is the triple-dipole constant which is calculated approximately using the geometric mean of the dispersion coefficients:

$$C_9^{ABC} \approx -\sqrt{C_6^{AB} C_6^{AC} C_6^{BC}} \tag{2.21}$$

Towards the conclusion of this section, I aim to illustrate the methodology through a straightforward example involving a water dimer configuration. By progressively increasing the distance between the two water molecules and subsequently calculating the corresponding dispersion interaction energy, I can derive what is referred to as the energy dissociation curve, showcased in Figure 2.1. Given that the literature suggests a negligible contribution from the three-body term to the overall dispersion energy in water [16], the dissociation curves were established solely using the two-body term $E_{\text{2-body}}$, along with the zero-type damping function as defined by Eq.(2.10). Notably, some parameters encapsulated in the method, which remain independent of the compound's nature for which dispersion energy is being determined, are outlined in Table 2.1. The parameters $s_{r,6}$ and s_8 are dependent on the selected exchange-correlation functional utilised for the DFT calculation. The other parameters listed in the table maintain a consistent value, irrespective of the functional employed. Moving onto parameters associated with the compound's chemistry and its geometry, the dispersion coefficients C_6^{AB} and C_8^{AB} fall under this category. These coefficients contribute to the total dispersion energy (as energies E_6 and E_8), as visualised in Figure 2.1.

Table 2.1: Some parameters of DFT-D3 which do not depend on the chemical nature or geometry of the compound or material the dispersion energy of which should be calculated. The values of $s_{r,6}$ and s_8 are determined by the choice of the exchange-correlation density functional, which is in this case the Perdew-Burke-Ernzerhof density functional (PBE) [17].

Parameters	Values
s_6	1.0
$s_{r,6}$	1.217
s_8	0.722
$s_{r,8}$	1.0
α_6	14.0
α_8	16.0

The dissociation curve serves as a valuable tool to discern the evolution of dispersion interaction and its constituent contributions E_6 and E_8 in correlation with the expanding molecular distance. As illustrated in Figure 2.1, the dispersion energy reaches its minimum at an intermolecular distance of around 3.0 Å, coinciding with the peak of E_6 and E_8 contributions. Furthermore, as the distance between the molecules contracts, the dispersion energy experiences a sharp decline, attributed to the influence of the damping function, which determines in which interatomic distance region the dispersion energy is in absolute value reduced or vanishes completely. Conversely, a continuous increase in the distance between water molecules prompts the dispersion interaction to decrease slowly, predominantly due to the quickly diminishing E_8 contribution. At more extended distances, the dispersion energy is dominated entirely by the E_6 energy contribution, as visually evident in Figure 2.1. As distance expands, the dispersion energy approaches a small, nonzero constant value in case of the water dimer, which is especially evident in the logarithmic representation of the dissociation curve displayed in Figure 2.2. By increasing the intermolecular distance, the intermolecular dispersion contributions decrease to zero, however, the intramolecular dispersion contributions stay unchanged and nonzero. Because of this, the dispersion energy converges for the water dimer to the dispersion energy value that corresponds to the double value of the dispersion energy of an isolated water molecule surrounded by vacuum.

As the distance between the molecules increases, the dispersion energy undergoes notable changes. One contributing factor is the variance in the dispersion coefficients, C_6^{AB} and C_8^{AB}, influenced by the calculated coordination number CN of atoms within the molecules, which in turn hinges on the distance between the atoms. Using this dependency, the authors of the method try to account for the hybridisation effect mentioned in this Section. This phenomenon is effectively demonstrated by examining the CN of both oxygen atoms and one hydrogen atom from each water molecule within the dimer, as showcased in

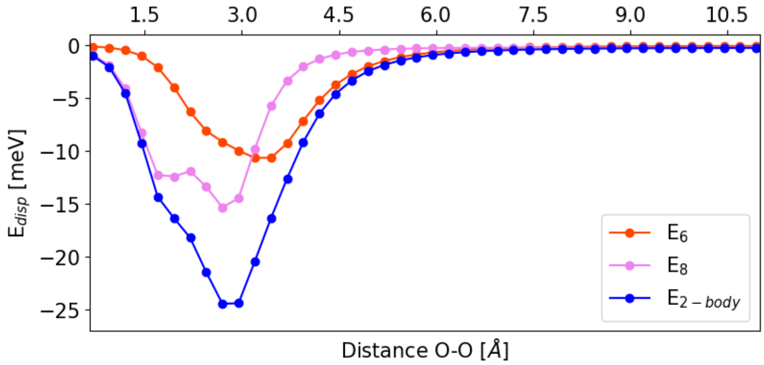

Figure 2.1: Dispersion energy as a function of the distance between the oxygen atoms in two water molecules. The dispersion energies were calculated using the DFT-D3 method with Perdew–Burke-Ernzerhof (PBE) exchange-correlation functional [17], zero damping function and no three-body energy correction.

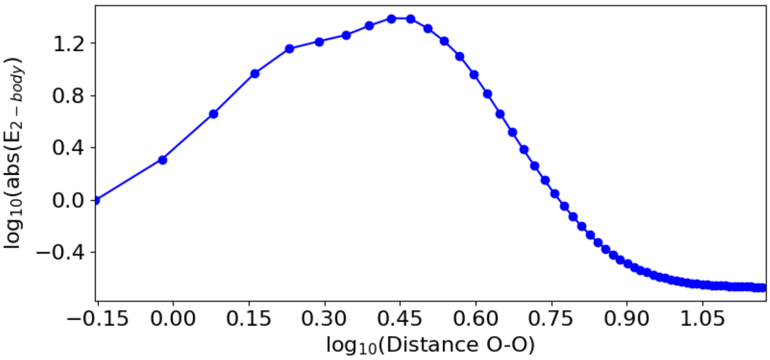

Figure 2.2: Logarithm of the dispersion energy as a function of the logarithm of distance between the oxygen atoms in two water molecules. The dispersion energies were calculated using the DFT-D3 method with PBE exchange-correlation functional, zero damping function and no three-body energy correction.

Figure 2.3. Notably, at the smallest distance of 0.7 Å, the CNs register as relatively high. In the case of oxygen atoms, CNs stand at 3 for O2 and 5 for O1, while hydrogen atoms H2 and H4 possess CNs of 2 and 3 respectively. This behaviour can be rationalised by examining the molecular arrangement at such a close proximity. At this distance, the two molecules are so closely located to each other that the interatomic spacing falls beneath the length of the covalent bond in a single water molecule's O-H linkage, which measures 0.97 Å. Consequently, the DFT-D3 method's algorithm considers O1 to be coordinated with five atoms, O2 with three, and H2 and H4 with three and two other atoms, respectively. With the increase in intermolecular distance, the molecules gradually separate, prompting a decrease in CNs. This shift in CNs leads to a simultaneous elevation in C_6^{AB} and C_8^{AB}. This evolution is exemplified in the interaction between the

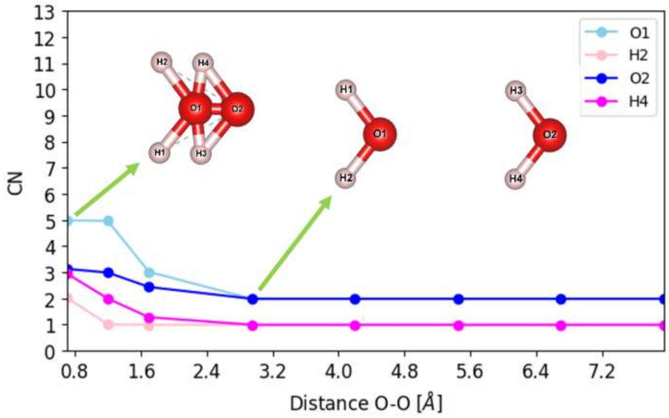

Figure 2.3: Coordination number CN of oxygen atoms O1, O2 and hydrogens H2 and H4 as a function of the interatomic distance between the oxygen atoms. The structures of the water dimer are given for the interatomic distances of 0.7 and 2.95 Å.

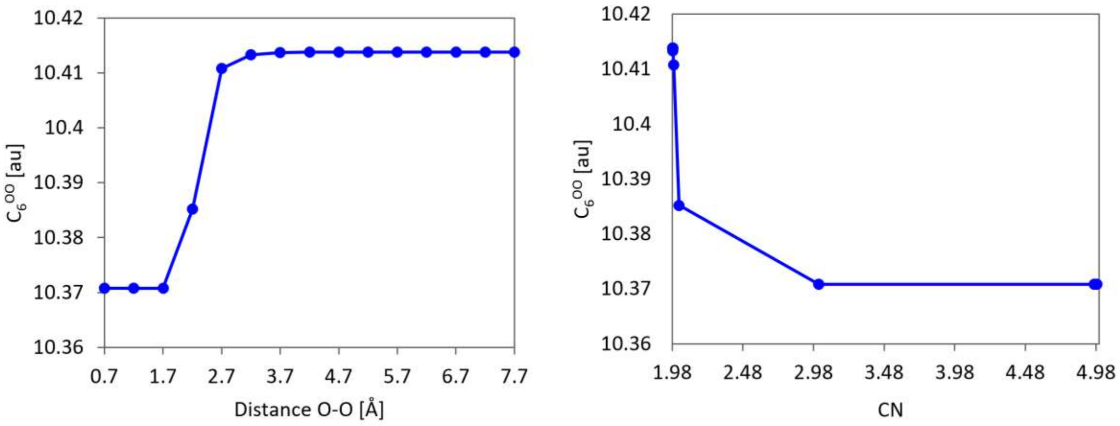

Figure 2.4: The dispersion coefficient of the interaction between the two oxygen atoms (C_6^{OO}) as a function of the interatomic distance between the oxygens and CN of O1.

two oxygen atoms, O1 and O2, as depicted in the right panel of Figure 2.4. As O1's CN diminishes, C_6^{AB} increases. In the same time, C_6^{AB} experiences an uptick as the distance between O1 and O2 expands (as shown in the left panel of Figure 2.4) while CNs of O1 and O2 decrease (Figure 2.3). This trend holds true for any combination of intermolecular atomic pairs within the system. The CN decreases and C_6^{AB}, C_8^{AB} increase until the distance between the molecules gets large. Then their values reach a constant value. It can be seen in the Figures 2.4 and 2.3 that their values do not change anymore already beginning from approximately 3.0 Å.

2.1.2 van-der-Waals density functionals

Dispersion interactions can be included directly within DFT using non-local density functionals. Among those there are also functionals from the vdW-DF family [2]. The vdW-DF method splits the exchange-correlation energy into semilocal and nonlocal contributions:

$$E_{xc}^{\text{vdW-DF}}[n] = E_{xc}^{sl}[n] + E_c^{nl}[n] \tag{2.22}$$

where $E_{xc}^{sl}[n]$ is the semilocal contribution of the exchange-correlation energy and $E_c^{nl}[n]$ is the non-local correlation energy. Within the vdW-DF framework, the semi-local term is built on the GGA such as the revised PBE functional (revPBE) to include the semi-local exchange-correlation energy [15]. The non-local correlation term is characterised by its two point dependency on the electron density via this equation:

$$E_c^{nl}[n] = \frac{1}{2}\int\int n(\mathbf{r})\phi(\mathbf{r},\mathbf{r}')n(\mathbf{r}')\, d\mathbf{r}d\mathbf{r}' \tag{2.23}$$

where ϕ is the so called non-local kernel which is an integral itself and which is constructed to capture the long-range part of the correlation energy. Since the release of the very first density functional of this kind, vdW-DF1 [2], which was universally applicable to all sorts of geometries of various compounds, several versions have been developed, such as vdW-DF2 [18], optPBE-vdW [19], optB88-vdW [20] or vdW-DF-cx [21].

2.2 Molecular dynamics

2.2.1 General concepts

Molecular dynamics is a powerful method that enables the prediction of measurable quantities of various materials in a realistic manner. By linking the microscopic world, which includes electronic structure and inter-atomic interactions, with the physical properties of a material, one can calculate these properties by averaging over multiple configurations that are frozen in time at finite temperature. This approach is analogous to the way physical properties are measured during experimental measurements. Because the ergodic hypothesis postulates, that the ensemble and time averages are equivalent, one can, equivalently to the latter strategy, observe a time evolution of a single state over a sufficiently long time and calculate the physical property of interest by computing the temporal averages of the properties [22]. During the simulation a collection of systems with the same kind of microscopic interactions and macroscopic properties (such as volume, total energy ...) is created, which is called an ensemble [22]. This strategy is the key concept of molecular dynamics.

In molecular dynamics the time evolution is performed in discrete time steps. At every time step Newtons equations of motion are integrated:

$$\dot{\mathbf{p}} = \mathbf{f} \tag{2.24}$$

with

$$\dot{\mathbf{x}} = \frac{\mathbf{p}}{M} \tag{2.25}$$

where $\dot{\mathbf{p}}$ is momentum of a particle, $\mathbf{f}$ is force acting on a particle, $\mathbf{x}$ are the atomic positions and M the atomic or molecular mass of a particle in a simulation box. For this purpose an integrator is used that makes them solvable on a computer using numerical calculations. In this work a JAX implementation of molecular dynamics called JAX-MD [11] with the Nosé-Hoover chain for an isobaric-isothermal ensemble [23] was used. We chose JAX-MD because it is based on the same high-performance python library JAX as the NNFF NeuralIL. The advantages of using JAX framework will be explained in detail in the Section 2.3.1.

2.2.2 Isobaric ensemble

When calculating properties of materials such as density, free energy of formation, redox potentials, or solubilities under constant temperature and pressure, the most suitable ensemble is the NPT [24]. The NPT ensemble operates with a constant number of particles, pressure and temperature. In this work, the temperature and pressure were held on average constant using the Nosé-Hover chains [23] as a barostat and thermostat. To maintain a fixed external pressure, the system's volume must be allowed to fluctuate. This concept can be visualised as an isobaric system coupled with an isotropic piston. The piston compresses or expands the system uniformly in response to instantaneous internal pressure fluctuations, ensuring that the average internal pressure matches the externally applied pressure. In this ensemble, both isotropic and anisotropic volume fluctuations are possible. [24] For the case of isotropic volume change, the internal pressure is defined as the average of the instantaneous pressure, which represents the change in energy with volume. It is given by [25]:

$$P_{int} = \frac{1}{3V}\left[\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i} + \sum_{i=1}^{N}\mathbf{R}_i\mathbf{f}_i - 3V\frac{\partial U}{\partial V}\right] \tag{2.26}$$

where V is the volume of the cell, U stands for potential energy, $\mathbf{p}_{\mathrm{i}}$, R_{i} and m_{i} are the momenta, position and mass of the ith particle and finally $\mathbf{f}_{\mathrm{i}}$ are the forces on the atom i.

In a NPT ensemble, the system is also coupled to an external thermal reservoir that exchanges heat with the system in order to maintain the temperature constant on average.

2.3 Machine learning in theoretical chemistry

2.3.1 NeuralIL

In this work, a NNFF called NeuralIL [7], which was developed in the research group of Prof. Madsen was used. Since the technology related to neural networks evolves very fast, in this work I used two different versions of NeuralIL: the older version NeuralIL 0.5 and the newest NeuralIL+VeLO. Their differences are explained later in this chapter.

The neural network itself consists of several layers of neurons that communicate between each other as depicted in Figure 2.5. The bottommost layer is called the input layer. The other layers are the output layer (the topmost one) and the layers in between called hidden layers. Each neuron in the hidden layer receives the output of each neuron in the previous layer as input I_i and generates an output according to:

$$O = f(b + \sum_{i=1}^{N} \alpha_i I_j) \tag{2.27}$$

where α_i is a set of weights, b the overall bias and f is an activation function, which has been chosen to be Swish-1 [26] in NeuralIL. The set of weights and biases of each neuron altogether represent the parameters of the neural network that have to be optimised during the training.

Before a neural network can be trained, a dataset used as an input for the neural network and the corresponding reference outputs have to be created. The typical input to NeuralIL are atomic positions, forces and/or potential energies, cell parameters and atomic types. In order for the neural network to effectively handle the atomic positions, which typically come in the form of Cartesian coordinates, a transformation into a set of descriptors is necessary. Descriptors possess a notable advantage as they encode the local environment of each atom without relying on the selection of an absolute origin or orientation within the coordinate system. This property grants them translation and rotational invariance, unlike Cartesian coordinates [7].

In NeuralIL, the spherical Bessel descriptors are implemented, which are calculated by projecting the density $\rho_{ij}(\mathrm{r})$ of each chemical element J around each atom i within a sphere with a predefined cutoff radius r_{cut} on a set of orthonormal basis functions B_{nlm}:

$$\rho_{ij}(\mathbf{r}) = \sum_{\substack{j \in J \\ R_{ij} < r_{cut} \\ j \neq i}} \delta(\mathbf{r} - \mathbf{R}_{ij}) \tag{2.28}$$

$$B_{nlm}(\mathbf{r}) = g_{n-l,l}(r) Y_l^m(\hat{\mathbf{r}}) \tag{2.29}$$

In Eq. 2.29 n, l and m are index values, analogous to the quantum numbers of atomic orbitals, that determine the number (n_{max}) of radial basis functions and the shape of the

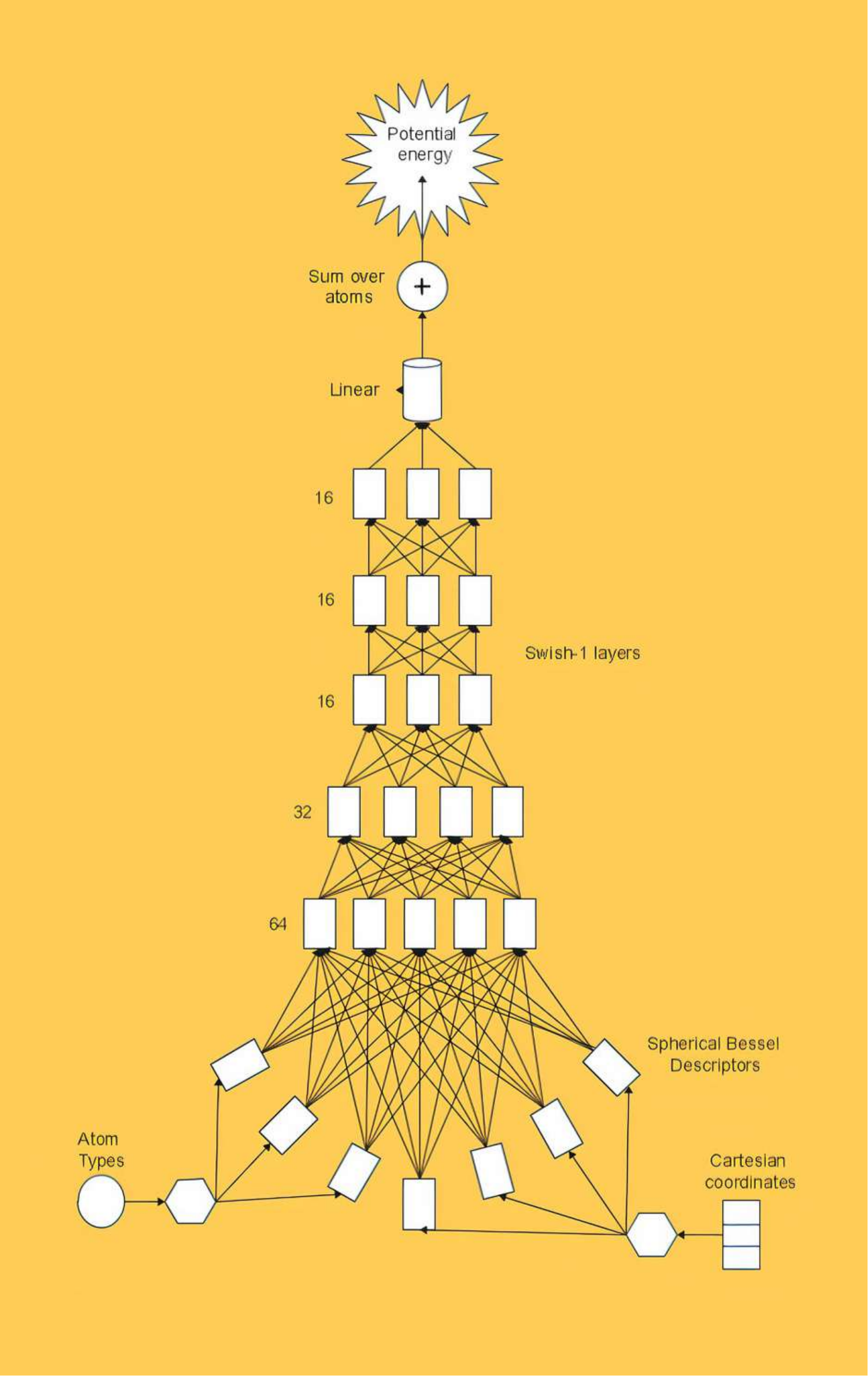

Figure 2.5: Scheme of neural network NeuralIL. Adapted from [7] with permission of the authors.

basis functions $g_{n-l,l}(r)$ (l index) and spherical harmonics Y_l^m (l and m indices). Their values lie between $0 \leq n \leq n_{max}$, $0 \leq l \leq n$ and -$l \leq m \leq l$.

Increasing the value of n_{max} improves the resolution within the sphere, but it also leads to longer computation times [7].

The training phase in NeuralIL starts by randomly splitting a dataset into a training set and a validation set. While the training set is used for the parameter optimisation of the neural network, the validation set is used to estimate how good the neural network is at the current stage of training. The training set is divided randomly into several small subsets called mini-batches. Those are, one after another, used as input for the hidden layers.

The neural network computes the total potential energy of a configuration by utilising a set of Bessel descriptors, denoted as p_α. These descriptors are constructed using the atomic positions, cell information, and types. Additionally, the network incorporates embedding coefficients w_β. The total potential energy is determined by summing the contributions of local atomic environments described by the descriptors, as expressed by the following equation:

$$E_{\text{pot}}(\mathbf{p}_\alpha, \mathbf{w}_\beta) = \sum_{i=1}^{n_{atoms}} \Omega(\mathbf{p_i}, \mathbf{w_i}) = \sum_{i=1}^{n_{atoms}} e_i \tag{2.30}$$

where Ω is a very flexible function implemented by the neural network [7]. Atomic forces are calculated as a gradient of the potential energy with respect to the atomic coordinates r_i:

$$\mathbf{f}_i = -\frac{\partial E_{\text{pot}}}{\partial \mathbf{r}_i} \tag{2.31}$$

During the training we want to find the parameters of the function Ω so that the output of NeuralIL (which can be for example the potential energy or atomic forces) approximates the inputs from the validation set as much as possible. This can be evaluated using the so called loss function which was in case of NeuralIL chosen to be of the log-cosh-type:

$$\alpha_{\text{log-cosh}} = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \beta \log \left[\cosh \left(\frac{y_i - \bar{y}_i}{\beta} \right) \right] \tag{2.32}$$

where N_{batch} is the number of mini-batches, β is a scale parameter (in versions NeuralIL 0.5 and NeuralIL+VeLO it was chosen to be 0.1 eV/Å and 1.0 eV/Å respectively), y_i stands for the reference value of either the potential energy of atomic forces, $\bar{y}_i$ is the calculated energy or force by NeuralIL. The loss function itself is a smooth approximation to the mean average error (MAE) [7].

During the training the value of the loss function is minimised with respect to the parameters of the neural network. In the older version NeuralIL 0.5, the adaptive moment estimation (ADAM) algorithm [27] was used, which makes use of running averages and second momenta. In combination with ADAM, NeuralIL 0.5 implements a dynamic learning rate strategy during each single training epoch. This strategy involves three distinct phases with varying training rates. In the initial 45% of the training the training rate is linearly increased to a predetermined value. Subsequently, for the following 45% of the training, the rate is decreased back to the starting value. Finally, as the loss function approaches its minimum, the rate is further reduced to reach the final value. This adaptive learning rate approach substantially enhances the efficiency of the learning process, resulting in significant improvements in learning time [7]. However, in the newest version NeuralIL+VeLO ADAM and dynamic learning rate were replaced by the fully nonlinear learned optimiser VeLO [28] which can reduce the training time even more drastically, as will be also demonstrated in the Section 3.1.3.

When the training with all of the mini-batches is finished, we say that an epoch has been completed.

The code of both versions of NeuralIL is built upon the high-performance machine learning framework called JAX (and related libraries Flax and Optax) [29], specifically designed for Python. JAX brings several advantages to the table. Firstly, it enables automatic differentiation of Python and NumPy functions, supporting both reverse and forward differentiations. This capability allows for swift calculations of Hessian and Jacobian matrices of descriptors with respect to the Cartesian coordinates of the atomic positions [7]. Another notable advantage of JAX is its incorporation of an accelerated linear algebra compiler (XLA). This feature not only facilitates code execution on CPUs but also extends compatibility to GPUs and TPUs without necessitating substantial modifications to the existing code. Furthermore, JAX enables the just-in-time compilation of any suitably coded Python function. This optimisation technique significantly enhances the execution speed of functions, leading to substantial performance improvements [29].

CHAPTER 3

Results and discussion

3.1 On the search for the best neural network model

3.1.1 The dataset

Since water is a very important compound with physical properties that are influenced strongly by the long-range interactions, the dataset that was used for training of the neural-network models consisted of 1593 condensed-phase structures of 64 water molecules. The structures were created by Dr. Cheng et al. [5] and successfully used to train their own NNFF to predict physical properties of water such as density. In the Dr. Cheng's dataset the last 593 structures were created using path integral molecular dynamics at 300 K and the first 1000 structures were chosen in such a way that they cover a large configurational space [5]. The corresponding energies and forces were calculated using DFT with the revPBE0 exchange-correlation functional. The dispersion correction was calculated using DFT-D3 with zero damping function and no three-body term. One of those structures is given in the Figure 3.1.

However, due to the focus of my research group on metal-water and metal-liquid interfaces, we decided to recalculate the potential energies and forces of the structures using a different density functional, namely the revised Perdew-Burke-Ernzerhof functional RPBE [30]. This choice was based on the successful utilisation of RPBE with DFT-D3 dispersion correction with a zero damping function and no three body term for simulating the physical properties of water/platinum electrode interfaces [31].

The energies and forces were calculated using the GPAW package [32], [33]. The wave functions were expanded in plane waves using the projector-augmented-wave formalism with an energy cutoff of 340 eV. Due to the rather large size of the cell (12×12×12 Å on average), the reciprocal cell was sampled using 1 k point only. Finally the energies and forces were corrected using the DFT-D3 method with zero-type damping function

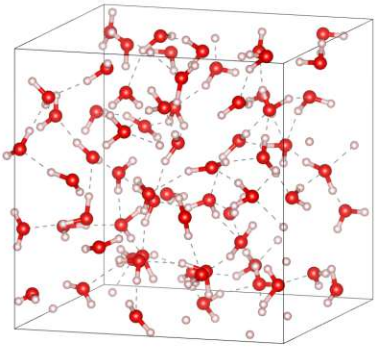

Figure 3.1: One of 1593 bulk water structures from the datased created by Dr. Cheng et al. [5]

without the three-body correction and taking a maximum interaction distance within atomic pairs of 50 Å, as suggested in the literature [31].

The forces and energies calculated using RPBE and DFT-D3 were analysed separately through histograms, as illustrated in Figures 3.2, 3.3, 3.4 and 3.5. It is evident that both force sets exhibit a symmetric bell-shaped distribution centered around 0 eV/Å (with mean values of 0.0001 eV/Å for RPBE and -5.0^{-19} eV/Å for DFT-D3 forces), which is desirable. Notably, the dispersion forces display a narrower distribution compared to the RPBE forces. The variance of the dispersion forces is 0.0009 eV/Å, with a standard deviation of 0.031 Å, whereas the RPBE forces have a variance of 5.67 eV/Å and a standard deviation of 2.38 eV/Å. Further analysis reveals that approximately 35% of the dispersion forces are smaller than 10^{-3} eV/Å. Since the dispersion forces are so small in this case, their addition to the RPBE forces barely changes the distribution of the RPBE forces, as can be seen in the Figure 3.3. The resulting variance is 5.64 eV/Å and standard deviation 2.38 eV/Å which are very similar to the variance and standard deviation of the RPBE forces. The Figures 3.4 and 3.5 show, that also the energies calculated using DFT-D3 are very small compared to the RPBE energies, since the DFT-D3 energies correspond on average only to around 10.7 eV while the RPBE energies are around 617 eV. The DFT-D3 energies are so small that, when added to the RPBE potential energies, they would constitute only between 1.15 - 2.6 % of the total potential energy of the bulk water configurations.

3.1.2 Model optimisation

Before proceeding with the training, a crucial decision had to be made regarding the choice of models. Our initial selection was the same NeuralIL model, that was employed by Prof. Montaña et al. as a force field for ionic liquids [7] which corresponds now to NeuralIL 0.5. This model, trained on atomic forces only, proved successful in accurately capturing

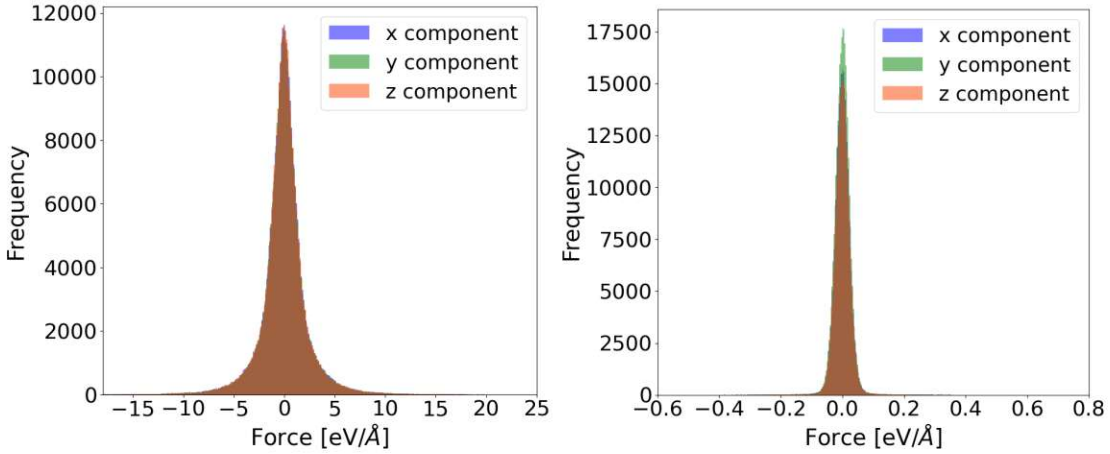

Figure 3.2: Distribution of x, y and z components of atomic forces calculated using RPBE functional (left) and using only DFT-D3 (right)

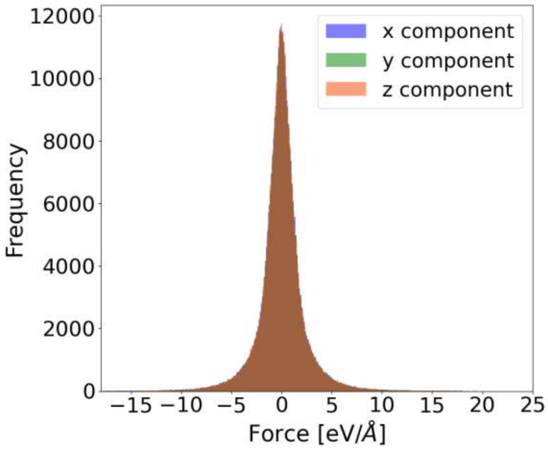

Figure 3.3: Distribution of x, y and z components of atomic forces consisting of forces calculated using RPBE functional and DFT-D3 dispersion correction.

the intricate long-range interactions among the constituents of ionic liquids, despite the absence of an explicit treatment of long-range interactions in this model. Motivated by these encouraging results, I opted to employ the NeuralIL model on the water dataset to evaluate its performance in that context. Although water differs from ionic liquids, it also exhibits significant vdW interactions between its molecules. Therefore, we sought to determine whether the NeuralIL model could effectively capture and represent the behaviour of water, even without explicitly accounting for long-range interactions. This NeuralIL model consists of 6 layers: one with 64 neurons, one with 32 neurons and three layers with 16 neurons. The hyperparametrs were : size of embedding (set to two), learning rates of 10^{-3}, 10^{-2} and 10^{-5}, r_{cut} (set to 3.5 Å) and n_{max} (set to 4 Å).

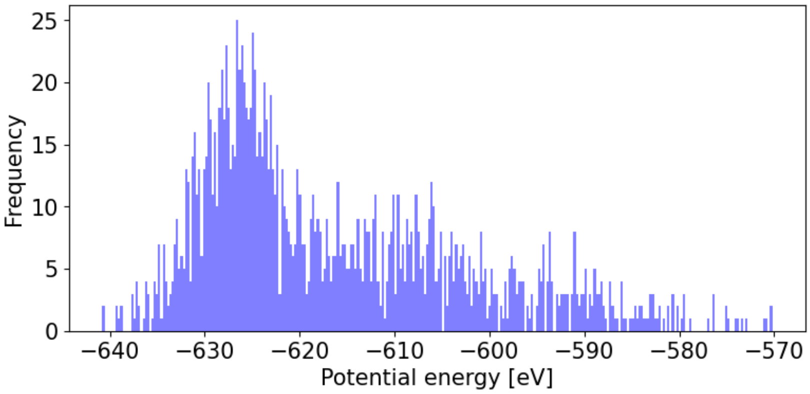

Figure 3.4: Potential energy distribution calculated using RPBE only for all configurations in the bulk water dataset.

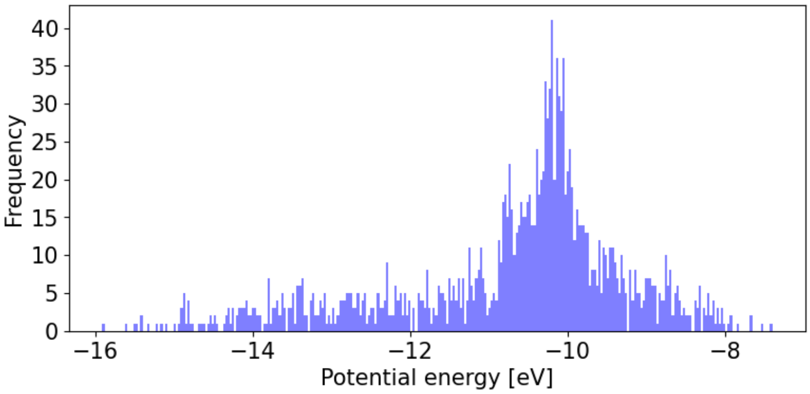

Figure 3.5: Distribution of dispersion energy calculated using DFT-D3 for all configurations in the bulk water dataset.

The second model was derived from the initial one by increasing the r_{cut} parameter and searching for an optimal value of n_{max}. As discussed in the Background chapter, the selection of the cutoff radius is a critical parameter for systems involving vdW interactions. It determines the range at which atoms within a compound interact with each other. Consequently, we sought to assess the performance of the NeuralIL model when trained on forces and its ability to predict forces that include vdW contributions, as the cutoff radius was increased.

In water systems, due to the long-range nature of vdW interactions, a typical cutoff radius used in the literature is 6 Å [5], [34]. However, employing such a large cutoff radius in the NeuralIL model would necessitate choosing a correspondingly large n_{max} parameter to achieve a relatively low root mean square error value. This would lead to a significantly longer training time, potentially spanning several days. In order to avoid excessive training time, I decided to make a compromise between a smaller cutoff

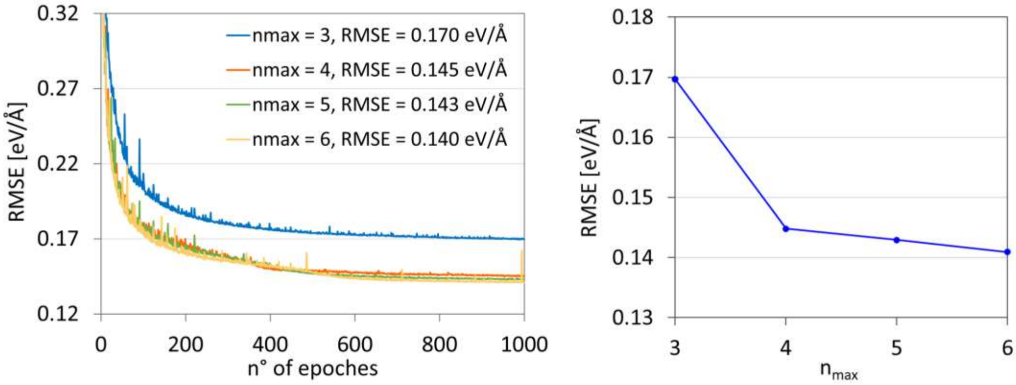

Figure 3.6: Left: the convergence of RMSE for the models with r_{cut} = 4.0 Å and n_{max} = 3, 4, 5 and 6. Right: the evolution of the converged, minimal value of RMSE for the models with n_{max} = 3, 4, 5 and 6.

radius of 3.5 Å and the larger 6 Å typically used, and selected a cutoff radius of 4 Å. Subsequently, I aimed to determine the optimal value of n_{max} for this specific cutoff radius.

In order to find it, I ran the training of NeuralIL for the same cutoff radius of 4 Å and various n_{max} values (3,4,5 and 6). At the beginning of the training the configurations in the dataset were split randomly into a training set (containing 80 % of the configurations) and a validation set (containing 20 %). During a repeated random splitting of the dataset, however, different configurations will land in the training set. Usually some configurations in a training set help the model to learn faster and better than some others. Because of this, the value of the RMSE can be in some cases smaller than in the others. Because of this dependency of the RMSE on the distribution of the data, it was important to run all of the training of the model with the same data distribution. The training was carried out for 1000 epochs and 8 mini batches per epoch with 32 configurations per batch.

During the training the RMSE was evaluated and recorded. The evolution of RMSEs for each n_{max} is depicted in the Figure 3.6 left. It can be seen that the worst performing model had the value of n_{max} of three, because after the training finished the reached RMSE value is large compared to the other models. On the other hand, the performance of the three other models is very similar - they converge similarly quick to a similar minimal RMSE. In order to see how exactly the minimal reached RMSE changed with n_{max}, it is plotted in the Figure 3.6 right. It can be see again that the model with an n_{max} of three was performing rather badly. Beginning with $n_{max} = 4$, the minimal RMSE changes very little with increasing n_{max} value. However, the training time increases drastically. Because $n_{max} = 5$ had reached a reasonably small RMSE relatively quickly compared to n_{max} of 4 and 6 and it was not computationally too expensive, the decision was made to take the model with this value of n_{max}.

3.1.3 Training and validation

In order to test the influence of the presence of vdW interactions on the training performance of a neural network, the two models, as described in the previous section, were trained according to two different training scenarios:

1. Training solely on forces without considering dispersion contributions
2. Training on forces including the dispersion contributions

Initially, the training was conducted using the previous version of NeuralIL for a duration of one thousand epochs. The evolution of the RMSE during the training process for both models and training scenarios is illustrated in Figures 3.7 and 3.8. Notably, the two models converged to different minimal and stable RMSE values. In the training scenario without considering dispersion interactions, model 1 (Figure 3.7) achieved an RMSE of approximately 0.141 eV/Å. Conversely, model 2 (Figure 3.8 right), under the same training scenario, yielded a slightly higher RMSE value, around 0.144 eV/Å. Furthermore, the two models exhibited differences in the speed of convergence. Model 2 required approximately 1000 epochs to reach a relatively stable value, whereas model 1 had already converged by around 700 epochs. Although model 2 theoretically possessed more information about the atomic environment due to larger cutoff radius of descriptors and a larger n_{max} parameter, it resulted in comparable or slightly larger errors in predicted forces on atoms, along with significantly increased computational time. Consequently, it appears that the larger parameter values in model 2 did not provide any notable improvement at this stage.

Figures 3.7 and 3.8 also present the RMSE evolution for the second training scenario of both models. It is evident that the RMSE values for both models and the second training scenario exhibit a convergence pattern very similar to that observed in the first training scenario. The convergence speed remains practically the same for both training scenarios and models.

In the second training scenario, the achieved RMSE values for model 1 and model 2 are 0.139 eV/Å and 0.145 eV/Å, respectively, demonstrating their close performance. Additionally, the variance of the training set, consisting of RPBE forces only, was 5.72 eV/Å, while the variance for the training set containing RPBE forces with DFT-D3 correction was 5.68 eV/Å. The attained RMSE values for both models and training scenarios are significantly smaller than the variance of the forces in the training set, indicating that the RMSEs are reasonably low, and the models are well trained. Moreover, when compared to the RMSE obtained using the same structures but with energies obtained using a different density functional for training Dr. Cheng's neural network, which was 120 meV/Å, the reached RMSEs appear to be similar to this value.

Once training was completed, the forces on atoms for the validation set were estimated using the neural networks and compared to the original forces obtained from DFT

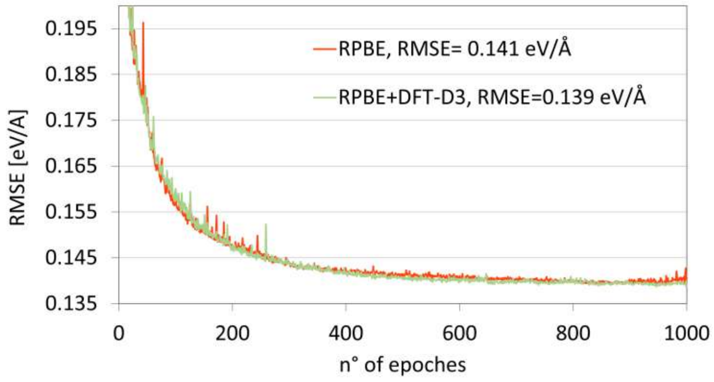

Figure 3.7: The convergence of RMSE of model 1. The case that the training was performed on RPBE forces without the DFT-D3 contribution corresponds to the red line, the training with DFT-D3 corresponds to the green line.

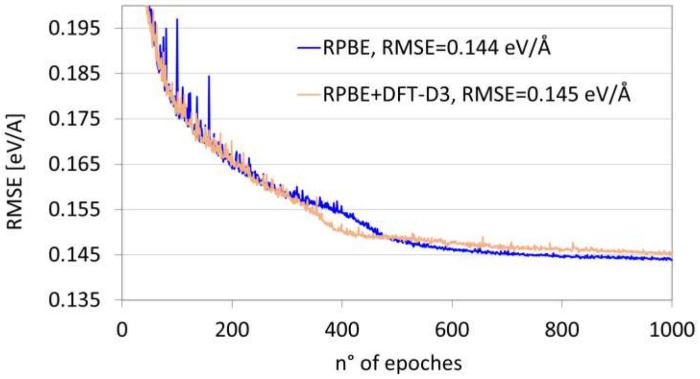

Figure 3.8: The convergence of RMSE of model 2. The case that corresponds to the training on RPBE forces without the DFT-D3 contribution is depicted by the blue line, the training with DFT-D3 corresponds to the pink line.

calculations. Parity plots (Figures 3.9 and 3.10) were utilised to visualise the comparison. Ideally, the reference and predicted forces should align perfectly, indicating that the neural network accurately reproduces the forces with 100% fidelity. In such cases, a straight line would be observed on the parity plot. Examining the parity plots for both models and training scenarios, it becomes evident that the predicted forces closely match the reference forces. The majority of the data points align, indicating a strong agreement between the predicted and reference forces. This observation is consistent across both models and training scenarios. The parity plots for both models confirm that there is minimal difference between the two training scenarios concerning each model. Consequently, it can be concluded that the inclusion of the dispersion contribution of forces in the training set had negligible influence on the convergence speed of RMSE or the minimal error achieved of predicted forces when the training was exclusively performed on forces. Interestingly, from the comparison between model 1 and 2 it emerges,that after a point, when more information about the system is offered to the neural network using descriptors with a larger radius, it does not seemingly help the neural network to learn better. On the contrary, it seems that the learning process becomes harder.

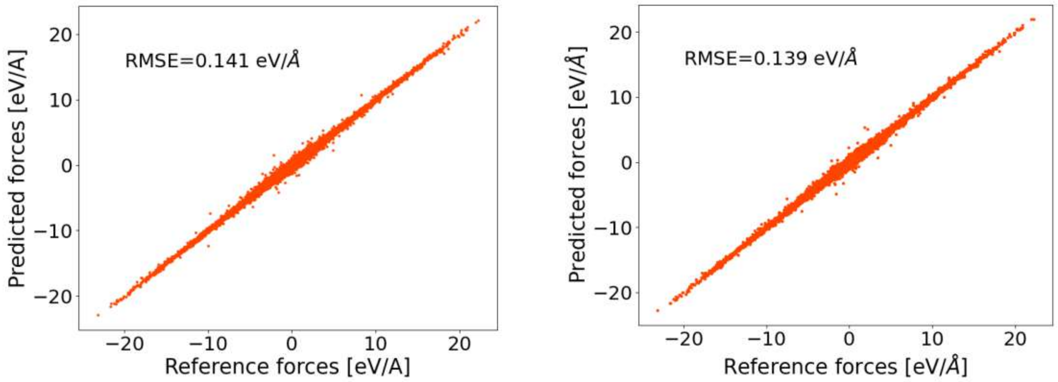

Figure 3.9: Predicted forces using model 1 versus reference forces. Left: model 1 trained without DFT-D3 contribution. Right: model 1 trained with DFT-D3 contribution.

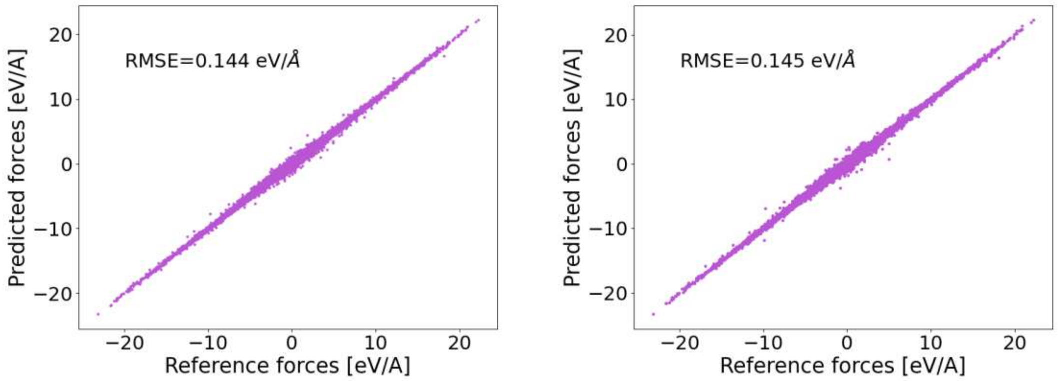

Figure 3.10: Predicted forces using model 2 versus reference forces. Left: model 2 trained without DFT-D3 contribution. Right: model 2 trained with DFT-D3 contribution.

During the course of my master's thesis, a newer version of NeuralIL was developed, taking advantage of the rapidly evolving neural network technology. As discussed in the Theoretical background chapter, this updated version introduced significant changes, primarily in the algorithm employed to minimise the RMSE. The new version incorporates a machine-learned optimiser called VeLO, which has proven to accelerate the training process dramatically, reducing the required number of epochs. Given the advances in the code, our intention was to utilise the latest version of NeuralIL for molecular dynamics simulations. However, before proceeding, I aimed to evaluate the training performance of the two models derived from the newer version. I specifically focused on comparing the performance under the two training scenarios and assessing the improvement in training speed. Consequently, I trained the two models based on NeuralIL+VeLO for 1000 epochs under the conditions of the two training scenarios. The evolution of the RMSEs as a function of the number of epochs is presented in Figure 3.11.

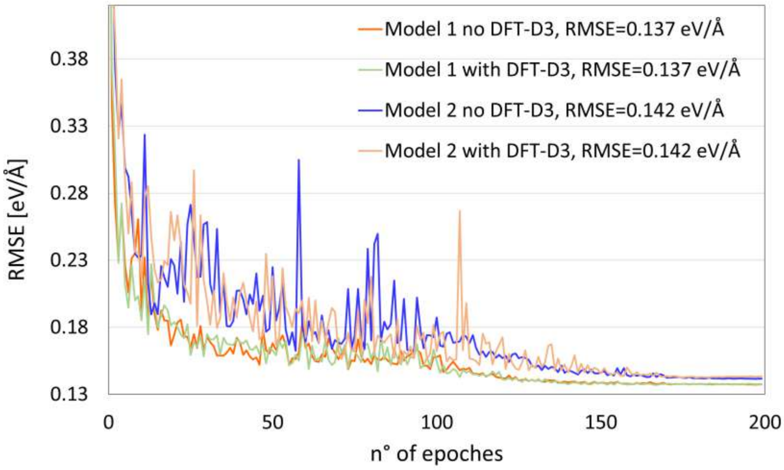

Figure 3.11: Convergence of model 1 and model 2 derived from the new neural network NeuralIL+VeLO. The red line represents the evolution of RMSE for model 1 trained without DFT-D3 contributions, while the green line shows the evolution of RMSE for model 1 trained with DFT-D3 contributions. Additionally, the blue line represents the RMSE for model 2 trained without DFT-D3 contributions, and the pink line represents the RMSE for model 2 trained with DFT-D3 contributions.

Observing the results, it is apparent that the achieved RMSEs for the two neural network models derived from the newer version of NeuralIL were slightly lower compared to the RMSEs obtained using the previous version. Interestingly, similar to the previous observations, the inclusion of vdW forces in the training set did not have a significant impact on the resulting RMSE values. However, the notable improvement lies in the training speed facilitated by the new optimiser. With the utilisation of the VelO optimiser, only approximately 150 and 180 epochs were required for model 1 and model 2 to converge, respectively. This represents a significant speedup of approximately 5.3 times for model 1 and 5.6 times for model 2 compared to the previous version. The introduction of the new optimiser has substantially reduced the training time while maintaining comparable or even improved RMSE values.

3.1.4 Transferability of the neural network model

It has been shown in the previous section that the vdW forces can be computed relatively correctly using both versions of NeuralIL. However it is questionable how transferable this result is to systems also consisting only of water molecules but having different chemical environments. To find this out I created a dataset containing 76 water clusters.

It was made by taking the dataset of 38 water clusters from the Benchmark Energy and Geometry Database (BEGDB) that contained clusters with 2 up to 10 water molecules per cluster [35]. Some examples of structures from the BEGDB are given in the Figure 3.12. The first step on creating the resulting 76 structures consisted in first minimising the 38 structures using DFT and RPBE functional implemented in GPAW. The chosen convergence criterion was that the error in forces needed to be smaller than 0.001 eV/Å and the chosen energy cutoff was 340 eV. Since it was desirable to train NeuralIL on forces again, structures that would have nonzero forces on the atoms were needed. This could be achieved by displacing the atoms in the structures little bit, so that they do not correspond to the relaxed structures anymore. For this purpose the function "rattle", which is part of the python library ASE, was used. The function rattle generates a set of numbers that are normally distributed and are characterised by a certain standard deviation value stdev. The function chooses randomly a set of numbers from those and adds them to the atomic positions. The 76 configurations were created in such a way that the relaxed structures were rattled twice using different values of the stdev parameter, in order to make two sets à 38 structures each that would differ. The values of the stdev parameter used were 0.001 and 0.05. Finally the clusters were put into a simulation box of 13×13×13 Å and the potential energies were calculated using GPAW and RPBE density functional with the DFT-D3 correction using the zero damping function, taking 340 eV for the energy cutoff value again.

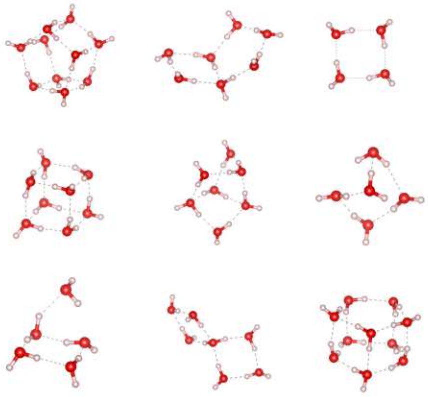

Figure 3.12: An example of some clusters contained in the BEGBT water cluster dataset

Using this dataset, my main objective was to assess the accuracy of predicted forces for water clusters when employing a neural network trained on condensed phase water. The datasets involved multiple water molecules, but there were clear differences between them. The bulk water dataset exclusively comprised water molecules surrounded by other water molecules, while the cluster water dataset featured molecules exposed partly to vacuum and partly to other water molecules. Through this analysis, our aim was to evaluate the neural network's ability to accurately predict forces for structures with novel chemical environments, i.e., those not encountered during the training phase.

The two models, model 1 and model 2, both derived from NeuralIL 0.5, were trained using forces with and without the DFT-D3 contribution, following the methodology described in Section 3.1.3. Subsequently, these trained models were used to predict forces for the entire water cluster dataset. The predicted forces were then compared to the reference forces in parity plots, depicted in Figures 3.13 and 3.14. Surprisingly, both models achieved remarkably low RMSE values. Moreover, there was not a significant difference in RMSE between the two training scenarios. For model 1, the RMSE was 0.078 eV/Å without dispersion interactions and 0.083 eV/Å with dispersion interactions, while model 2 achieved 0.079 eV/Å and 0.101 eV/Å, respectively. Notably, the error in the predicted forces was even smaller than the error obtained for the predicted forces of bulk water configurations. This result was unexpected, as we initially assumed that the neural network would struggle to accurately predict forces for unseen structures. However, it is possible that the neural network, aided by the relatively small cutoff radius used in the models, was able to recognise cluster-like structures within the bulk water, distinct from the surrounding molecules. To confirm this hypothesis, a principal component analysis of descriptors could be used.

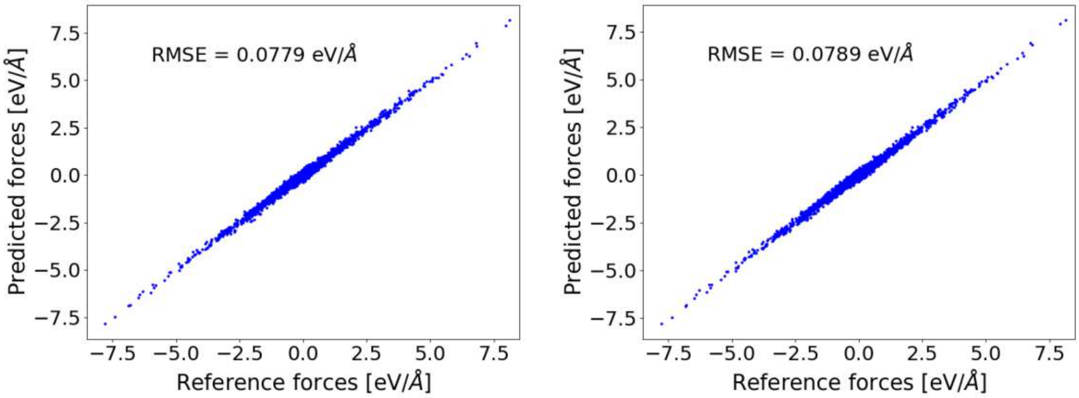

Figure 3.13: Predicted forces for 76 water clusters using model 1 versus the reference forces. Left: model 1 trained without DFT-D3 contribution. Right: model 1 trained on data including DFT-D3 contribution.

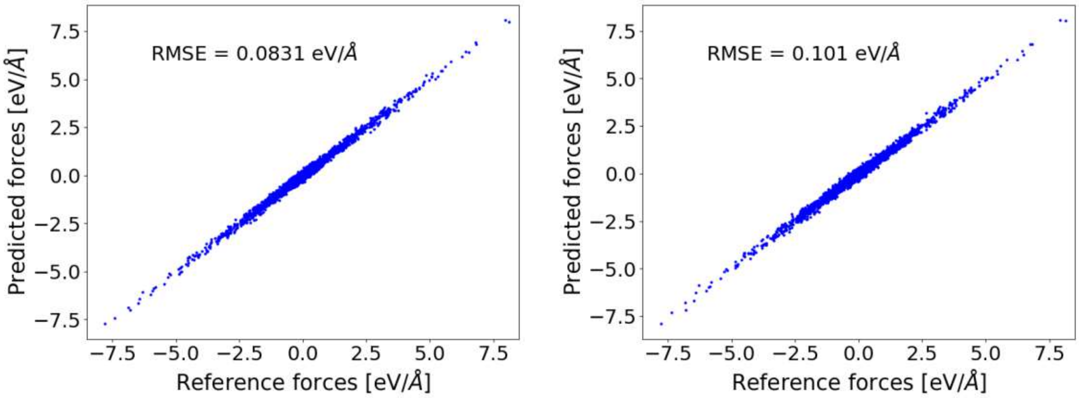

Figure 3.14: Predicted forces for 76 water clusters using model 2 versus the reference forces. Left: model 2 trained without DFT-D3 contribution. Right: model 2 trained on data including DFT-D3 contribution.

3.2 Molecular dynamics

After successfully training the four neural network models, our next objective was to employ these models as force fields in MD simulations. As mentioned in the Introduction section, the locality approximation can pose challenges when attempting to model materials with significant dispersion interactions. Those systems may exhibit physical properties that cannot be accurately calculated using molecular dynamics without accounting for dispersion interactions. This concern is particularly relevant for bulk water, which is a key focus of this work. In previous studies, such as in the work by Prof. Dellago et al. [8], the density of water at various temperatures were calculated using neural network force fields trained on RPBE energies and forces with DFT-D3 correction (zero damping function with no three-body term, given its negligible effect in water [16]), as well as RPBE forces and energies only, similar to our NeuralIL models. The resulting densities were compared with experimental data, and the corresponding density curves are illustrated in Figure 3.15.

The plot clearly demonstrates the impact of including dispersion interactions in the training set data. When dispersion interactions are omitted, the resulting density values deviate significantly from the experimental data, and the trend in the evolution of density with increasing temperature is completely incorrect. However, when dispersion interactions are included in the training, the density values exhibit improved agreement with experimental data, and the overall trend of density evolution is captured more accurately. This highlights the importance of considering dispersion interactions in the training of neural network force fields to achieve accurate predictions of physical properties.

Inspired by this study, my primary focus was to predict water densities at various temperatures within the range depicted in Figure 3.15 and compare them with the corresponding reference values from the study. However, the computationally intensive

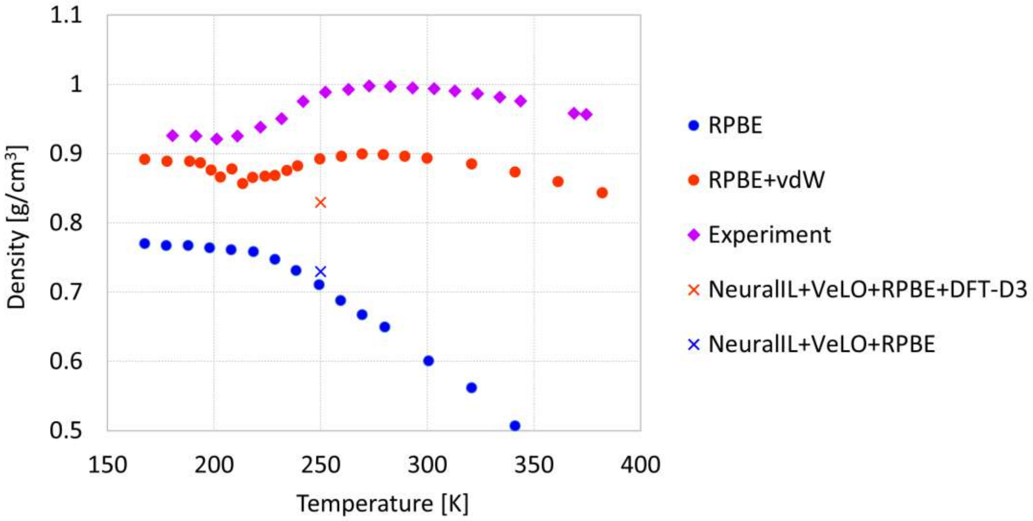

Figure 3.15: Density of water as a function of temperature. The blue circles depict the densities calculated using the RPBE functional without the DFT-D3 correction, the orange circles are densities from simulations with dispersion correction and finally, the experimental values correspond to the violet diamonds. The red cross indicates the density value obtained from from the NPT MD for model 1 + VeLO trained on dataset with DFT-D3 correction, at 250 K and 1 bar and the blue cross indicates the density value obtained from from the NPT MD for model 1 + VeLO trained on RPBE forces only, at the same temperature and pressure. The experimental, RPBE and RPBE+DFT-D3 data were adapted from [8].

nature of MD simulations, involving millions of time steps, imposed considerable time limitations on our research. Throughout the course of this master's thesis, it became apparent that each simulation of the bulk water system required approximately three weeks and half to complete. Consequently, I was only able to conduct MD runs for model 1 trained with and without dispersion correction at a single temperature, which will be the main focus of our analysis in this thesis.

3.2.1 Choosing parameters of the simulations

To accurately predict the density of water at specific temperatures, it was crucial to select an appropriate ensemble for our MD simulations. The choice of ensemble determines the conditions under which the simulation takes place and affects the behavior of the system. Considering that density is influenced by changes in volume with temperature, I found the NPT ensemble to be the most suited for this task. Another argument for the usage of NPT ensemble was, that with a NVT or NVE ensemble one would need to do one whole simulation per volume which would require a much larger amount of time. The NPT ensemble maintains constant temperature, pressure, and the number of particles within the simulation box while the volume of the cell changes during the simulation. Typically at the beginning of this kind of simulation the value of volume changes drastically, since the system is in process of adjustment to the external simulation conditions. This phase is known as equilibration period. After some time the volume begins oscillating around the equilibrium value. Once the volume stabilises over a sufficient duration, the simulation can be terminated, and the density can be calculated based on the average volume value during the equilibrium period. Since we are interested in water density at equilibrium, the calculated volumes during the unstable equilibration period of simulation are not relevant, so they are discarded. The density is given by:

$$\langle \rho \rangle = \frac{NM}{N_A \langle V \rangle} \tag{3.1}$$

where N is the number of water molecules in the simulation box, M stands for the molar mass of the molecules, N_A is the Avogadro number and $\langle V \rangle$ the average volume.

Before a production MD simulation could be conducted, I first needed to determine the paramaters of an NPT MD simulation for a chosen simulation temperature and pressure. These are the equilibration times of thermostat (τ_T) and barostat (τ_P), the time step of the simulation and whether the volume expansion should be isotropic or anisotropic. Given the relative ease of conducting MD simulations at lower temperatures, we selected 250 K as the temperature for our simulations. Additionally, since the density curves presented in Figure 3.15 were computed at ambient pressure, we set the pressure in our simulations to 1 bar to align with these conditions. The search for the optimal parameters was conducted using model 1, which uses VeLO and was trained on forces without DFT-D3 contributions. However, as we will demonstrate in the subsequent chapters, these parameters can also be applied to the simulation using model 1, which incorporates DFT-D3 forces.

So the first decision that I needed to make was, whether to allow isotropic or anisotropic volume changes in the simulation. Considering that anisotropic volume changes are typically recommended for solids and water is a solid at 250 K, we initially chose to implement anisotropic volume changes. In this setup, we first used equilibration times of $\tau_T = 50$ fs for the thermostat and $\tau_P = 500$ fs for the barostat. Next, we needed to decide which time step of the simulation should be used. One of the rules of thumb of MD is, that the time step should be 10 - 100 smaller than the period of oscillation of the fastest normal mode vibration than one is interested in. The frequency of the fastest vibration in solid water at 250 K can be approximated by taking the fastest vibration in liquid water. In liquid water, the fastest degree of freedom corresponds to the asymmetric stretching of O - H bonds which has the frequency of 3615 cm^{-1} in water IR spectra [36]. The rule of thumb is fulfilled by the Nyquist-Shannon sampling theorem [37]. The concept is the following: during molecular dynamics one is trying to sample continuous vibrations using discrete time signals, produced by doing discrete time steps during the simulation. In order to capture the full information about the vibrational movement, the simulation has to be sampled using an appropriate sampling frequency. Its inverse corresponds to the sampling period which corresponds to the time step used in a MD simulation. The Nyquist-Shannon theorem says, that the sufficient sampling period for a range of signal with the highest frequency B is any period that is smaller than 1/2B. This is formulated in the following equation:

$$\frac{1}{2B} > \tau \tag{3.2}$$

Setting B equal to the asymmetric vibrational frequency of water, one comes to the conclusion that the sampling period should be smaller than 4.63 fs. Using the rule of thumb of MD, the time step that should be used than would be approximately 0.5 fs, which fulfils the Nyquist theorem. Because of this, we decided to take this time step value first.

The initial structure for the simulation was selected from the bulk water dataset. However, we encountered an issue known as the "flying-ice-cube" phenomenon, which is an artefact of the molecular dynamics algorithm. During simulations affected by this phenomenon, the system accumulates a non-zero total momentum, causing the energy to be primarily allocated to translational or rotational motion of the molecules, while the other degrees of freedom, such as vibrations, are effectively frozen [38]. This behavior was observed in some of our initial simulations at 250 K. Figure 3.16 depicts the changes in the magnitude and components of the total momentum vector during the simulation. Initially, the total momentum vector had a magnitude of zero. However, after approximately 1 ns, the total momentum began to increase and reached around 55 kg m s^{-1}. The exponential growth in the norm of the total momentum vector is attributed to the exponential increase in the x and z components of the momentum vector. As shown in Figure 3.16, this resulted in all water molecules gaining translational motion along the diagonal in the xz plane of Cartesian coordinates. This behavior is highly unphysical and violates the theorem of equipartition of energy, which dictates that energy must be equally distributed among all degrees of freedom in thermal equilibrium.

In an attempt to address the flying-ice-cube phenomenon, we made adjustments to the parameters of the thermostat. We extended the equilibration times to $\tau_T = 100$ fs and $\tau_P = 1000$ fs, and $\tau_T = 250$ fs combined with $\tau_P = 2500$ fs. Also the time step was modified to 0.3 fs. However, these modifications did not lead to any improvement. To potentially overcome this problem, alternative solutions were considered. One possibility was to utilise a stochastic thermostat instead of deterministic ones like the Nose-Hoover chains since they have been known to mitigate the flying ice cube artefact [38]. Another option was to explore the use of isotropic volume expansion instead of the anisotropic one, as it might provide a more stable simulation environment.

We decided to try out the isotropic volume expansion first, since this required the smallest changes in the simulation code.

Furthermore, an alternative initial configuration was employed to initiate the MD simulation due to the instability observed with the first configuration from the bulk dataset,

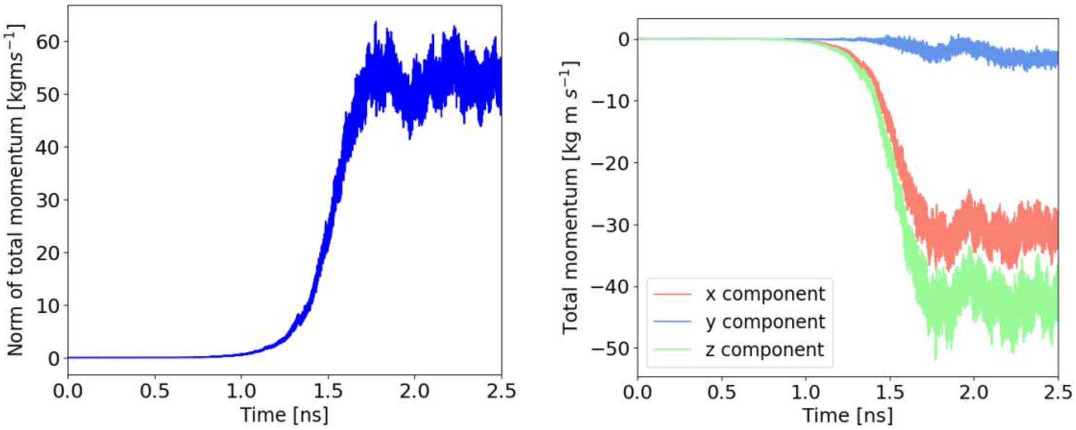

Figure 3.16: The evolution of the norm of the total momentum vector of the simulation box (left) and of the x, y and z components of the momentum vector (right) monitored during a NPT simulation at 250 K, 1 bar and with anisotropic volume expansion.

which resulted in volume explosions. To rectify this issue, the problematic configuration underwent relaxation using the LBFGS optimizer implemented in the Python library ASE [33]. Convergence was achieved when the forces reached a threshold of 0.1 eV/Å. Subsequently, this relaxed structure served as the input configuration for an NVT simulation, where the system was equilibrated at a constant temperature, volume, and number of particles. The NVT simulation was conducted for a duration of 0.4 ns (with a time step of 0.3 fs) at 250 K. The target volume was set to that of the original structure, and the thermostat parameter τ_T was set to 250 fs. This procedure aimed to generate a structure with a more reasonable distribution of molecular velocities. The resulting configuration from the NVT simulation was then utilised as the initial configuration for subsequent NPT simulation runs, employing $\tau_T = 100$ and $\tau_P = 1000$ fs, with the previous time step of 0.5 fs. Through these steps, a stable and physically meaningful molecular dynamics simulation was achieved.

However, the intended simulation time was at least of 15 ns, which was also used by Dellago and coworkers to create the density curves in Figure 3.15. Choosing the time step of 0.5 fs effectively means that during the total duration of the simulation of 15 000 000 fs, 30 million calculations would need to be carried out. It is clear that the smaller the time step, the longer the simulation takes. The dynamics of hydrogen atoms is strongly influenced by quantum-mechanical effects and therefore can not be modelled well using the classical mechanics approach. Therefore, as is common in the literature [39], the mass of hydrogen atoms can be adjusted in case of calculation of an observable, that is independent on mass, in order to reach a more convenient calculation duration. By increasing them one can reduce the value of the highest frequency of the vibrations of those which allows one to increase the time step of the simulation. Because of that, the atomic masses of hydrogens were increased in case of simulations with both neural networks from 1 a.u. to 8 a.u. By doing so, the frequency of the antisymetric vibrations in water could be decreased to approximately 1518.7 cm^{-1}. By setting B in the Nyquist theorem equal to this new frequency value, one founds out that the time step could be increased to around 1.1 fs. Because the simulation can be carried out faster using the time step of 1.0 fs, I decided to use it in case of both simulations, described in detail in the following sections.

3.2.2 Molecular dynamics using NeuralIL+VeLO trained on forces with DFT-D3 correction

Since the system needed a relatively large amount of time to get into the equilibrated state (approximately 6 ns), I decided to run the simulation for 17.6 ns instead of only 15 ns. This way I could gain more data from the equilibrated part of the simulation and thus to get more accurate density prediction. The evolution of the pressure and temperature of water with respect to the average values over the course of the simulation, is shown in the Figures 3.17. It is evident from these plots that the average temperature closely approached the target value, with a deviation of only 0.47 K. However, the average pressure exhibited a larger deviation from the desired value. Instead of the intended 1 bar, the average pressure was measured at 13.54 bar, indicating a slightly larger deviation. In fact, the deviation of the average pressure value correlates with the intensity of the pressure fluctuations throughout the simulation, which correlate with the size of simulation box via the following equation, for the case of an isotropic fluid:

$$\sigma_{\mathrm{p}} = \sqrt{\frac{\rho c^2 k_{\mathrm{B}} T}{V}} \tag{3.3}$$

where c it the speed of sound, ρ is density, T is a temperature, k_B is the Boltzmann constant and V is the volume of the simulation box. It follows from this equation, that the smaller the simulation box, the smaller is also its volume and so the larger is the variance of the pressure σ_p. The consequence of larger fluctuations is, that it becomes more probable that the average value of the pressure will differ from the intended value. This correlation could be clearly observed also in case of this simulation. By looking at the evolution of the volume of the simulation box in the Figure 3.20 left and the evolution of the pressure in the Figure 3.17 right, one can see that the fluctuations of the pressure are the larges between 1 - 6 ns of the simulation, when also the smallest volume was observed. On the other hand, in time intervals of 0 - 1 and 8 - 11 ns the fluctuations of pressure were the smallest while the reached average volume values seem to be the largest of the whole simulation. The effect of the size of the fluctuations on the average pressure calculated every 0.25 ns can be seen in the Figure 3.18. Between 1 - 6 ns, where the fluctuations are the larges, the average pressure value fluctuates the most, while between 8 - 11 ns the average value lies closer to the intended value of 1 bar.

Another possible reason for the large mean pressure fluctuation could be, that stress (being defined in context of isotropic volume expansion as the change of potential energy with respect to the volume) in the Eq.(2.26) is calculated inaccurately during the simulation.

Stress plays a crucial role in calculating the pressure within the simulation box, thus if it is calculated inaccurately, it might have a consequence on the capability of barostat to keep the average pressure close to the intended value. The possible inaccuracy of the calculated stress could originate from the fact that the used NeuralIL model was trained solely on atomic forces, which, contrary to the potential energy, do not incorporate information about stress. By training the neural network models on both forces and energies, the stress contribution to the energy can be captured better, because the predicted energies could be more accurate. Because of this reason, the accuracy of stress calculated using NeuralIL trained solely on atomic forces should be investigated.

Nonetheless, this deviation of the pressure is considered acceptable, as the density evolution shown in Figure 3.20 (right) demonstrates reasonable values throughout the simulation in relation to the reference density value at 250 K from Figure 3.15.

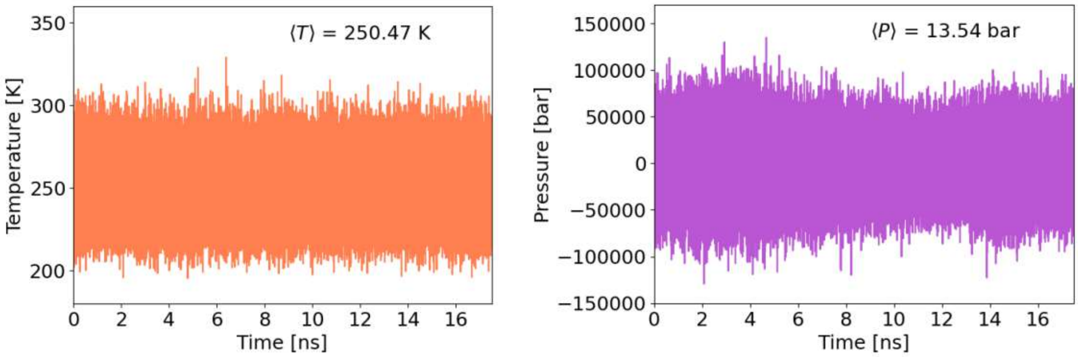

Figure 3.17: The fluctuations and mean values of temperature (left) and pressure (right) during a NPT simulation using using model 1 + VeLO trained on DFT-D3 forces. The simulation was performed at 250 K, 1 bar and using the isotropic barostat.

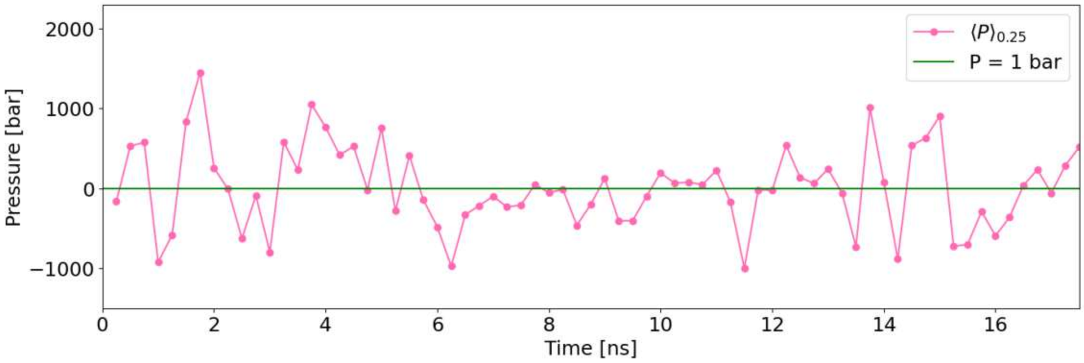

Figure 3.18: The evolution of the average pressure calculated every 0.25 ns (pink) with respect to the intended pressure value of 1 bar (green).

The behavior of the total momentum components of the simulation box was analysed prior to calculating the average equilibrium density to assess the occurrence of the flying-

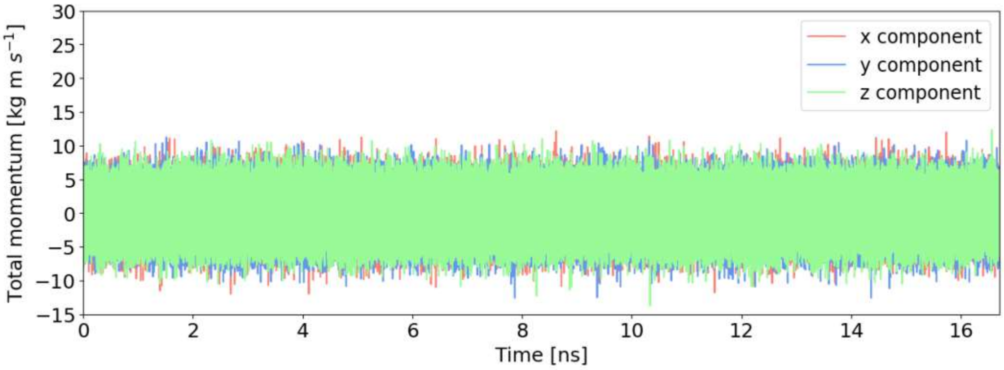

Figure 3.19: The evolution of the x, y and z component of the total momentum vector of the simulation box during NPT MD with model 1 + VeLO, trained on DFT-D3 at 250 K, 1 bar.

ice-cube artefact. Figure 3.19 illustrates the evolution of all three components of the total momentum. It is observed that the values of these components oscillate symmetrically between negative and positive values, resulting in an average total momentum close to zero. This indicates that the flying ice cube phenomenon did not occur in this simulation, which is a positive outcome.

The average equilibrium density of bulk water at 250 K and 1 bar was calculated by determining the period in which there was no significant trend in the volume evolution. The density was calculated with an accuracy of approximately 10^{-3} g/cm^3, which was deemed reasonable considering that the density values can not be obtained with higher accuracy from the reference plot in Figure 3.15. Since a high level of accuracy was not required for the density calculation, the volumes used for the calculation did not need to be perfectly converged. Therefore, the time interval, during which the volume change was constant on two decimal places and thus was chosen to calculate the average density, was 6 - 16.7 ns. By calculating the average volume during this period and using Equation 3.1, the equilibrium density was determined to be 0.83 g/cm^3. Comparing this value, depicted in the Figure 3.15 as a red cross, with the predictions of Dellago's neural network trained on their own dataset using RPBE+DFT-D3, it was found to be approximately 0.06 g/cm^3 lower. Furthermore, the equilibrium density was approximately 0.16 g/cm^3 lower than the experimentally measured value.

However, a closer analysis of the total energy of the system revealed an issue during the simulation. In an NPT ensemble with the Nosé-Hoover chains, the total energy, comprising the potential energy calculated by the neural network, kinetic energy, and energy contributions from the Nosé-Hoover chains, should remain on average constant throughout the simulation. However, as shown in Figure 3.21, this quantity increased continuously throughout the simulation, starting from negative values and progressing to positive ones. Further examination indicated that while the potential energy and kinetic energy remained relatively constant on average, this undesired behaviour occurred. It suggests the possibility of a bug in the simulation program. Our suspicion is that the simulation program uses scaled units for atomic positions to calculate atomic forces, while NeuralIL employs Cartesian coordinates for force calculations. We have informed the authors of JAX-MD about this issue and are currently awaiting their response. As a result, the presented results should not be considered definitive due to this problem.

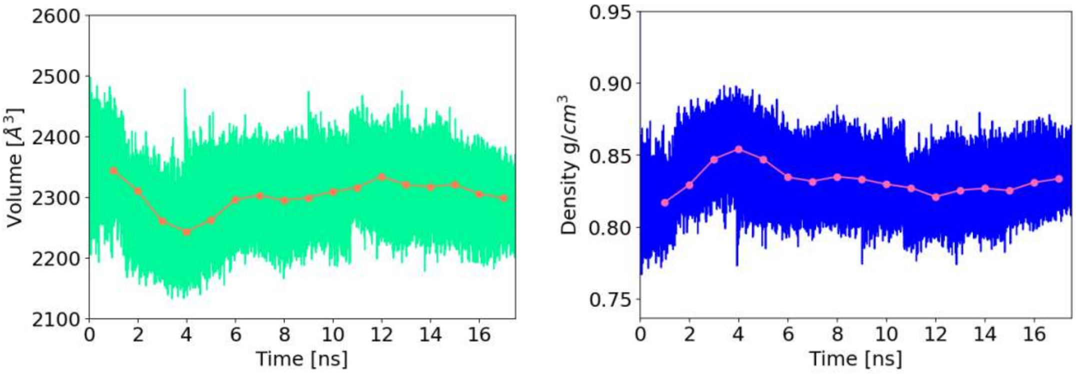

Figure 3.20: The fluctuations of volume and density values during the MD simulation (the blue and green curves) and their mean values calculated each one ns (the orange and pink curves).

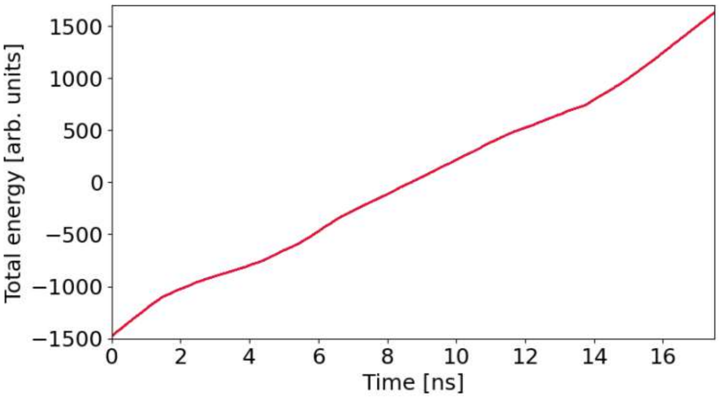

Figure 3.21: The evolution of the total energy of the system over the course of NPT MD at 150 K and 1 bar using model 1 with DFT-D3 forces

3.2.3 Molecular dynamics using NeuralIL+VeLO trained on forces without DFT-D3 correction

Since also in case of this simulation the equilibration period was relatively long, it was carried out for 17.6 ns in order to get statistically more significant result. The evolution of temperature and pressure using model 1 trained on DFT forces only is shown in Figure 3.22 (left and right), along with the average temperature and pressure values achieved during the simulation. As observed, the average temperature closely corresponds to the desired simulation temperature of 250 K, similar to the simulation with model 1 trained on DFT-D3. However, the average pressure deviates by 11.6 bar from the target pressure of 1 bar, as seen in the plot. This deviation could be rationalised in the same way as in the previous section. Nevertheless, this deviation can be considered acceptable since the volume and density of the system (depicted in Figure 3.24) fluctuated around the reference density value at this temperature, which is approximately 0.71 g/cm^3.

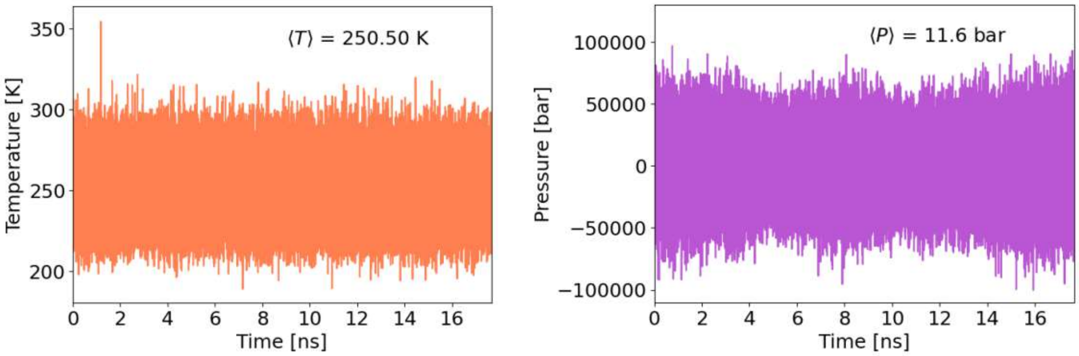

Figure 3.22: The fluctuations and mean values of temperature (left) and pressure (right) during a NPT simulation using using model 1 + VeLo trained on RPBE forces only. The simulation was performed at 250 K, 1 bar and with the isotropic volume expansion.

Similar to NeuralIL trained on DFT-D3, the components of the total momentum were analysed to detect any signs of the flying ice cube artefact. The components of the total momentum are displayed in Figure 3.23. As observed, they oscillated symmetrically around 0 kg m/s, indicating the absence of the flying ice cube artefact. This finding is favourable, as it suggests that the simulation maintained a stable behaviour.

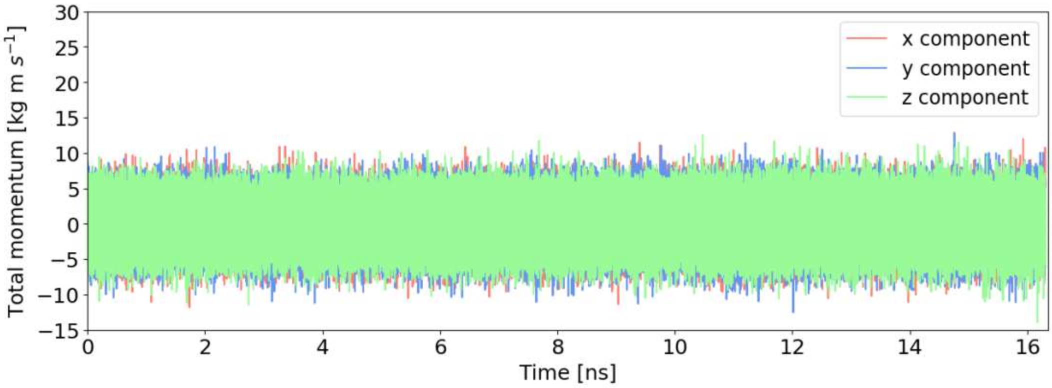

Figure 3.23: The evolution of the x, y and z component of the total momentum vector of the simulation box during the NPT MD simulation at 250 K, 1 bar and model 1 + VeLO trained on RPBE forces only.

After ensuring the absence of the flying-ice-cube artefact, a closer analysis of the volume and density evolution was conducted. The plots in Figure 3.24 illustrate the evolution of volume and density throughout the simulation. It can be observed that the volume did not converge as well as in the simulation with model 1 trained on DFT-D3 in the training set. Notable abrupt and drastic volume changes, reminiscent of a phase transition, were observed during the initial 10 ns of the simulation. However, after the 10 ns mark, the average volume and density appeared to stabilise. The volume and density converged to the desired level of precision, up to two decimal places. As the system demonstrated relative equilibration during the 10-16.7 ns interval, the equilibrium density could be calculated using the volume values from this interval.

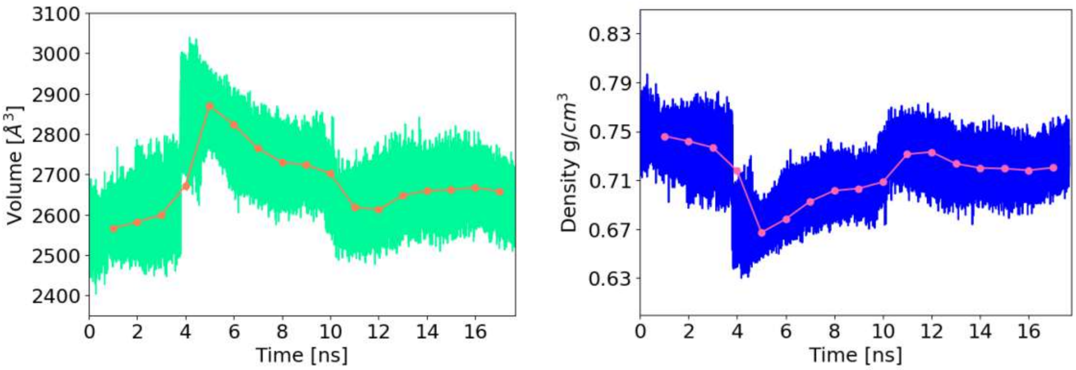

Figure 3.24: The fluctuations of volume and density values during the NPT MD simulation (the blue and green curves) and their mean values calculated each one ns (the orange and pink curves).

The resulting equilibrium density was found to be 0.73 g/cm^3, which is in close proximity to the reference water density value of approximately 0.71 g/cm^3, as depicted in Figure 3.15. The deviation of only around 0.02 g/cm^3 from the reference density is smaller compared to the deviation observed in the case of NeuralIL trained with DFT-D3 in the training set. This indicates that at 250 K, our NeuralIL model trained solely on RPBE forces and the neural network model developed by Dellago's group, which was trained on a different and larger dataset consisting of RPBE data only, exhibit comparable performance. Furthermore, similar to the previous section, the total energy of the system was plotted and examined in this case as well. Unfortunately, as observed with model 1 trained on DFT-D3, the total energy, which should remain constant on average, exhibited an increasing trend. This detected problem with JAX-MD casts the results into doubt.

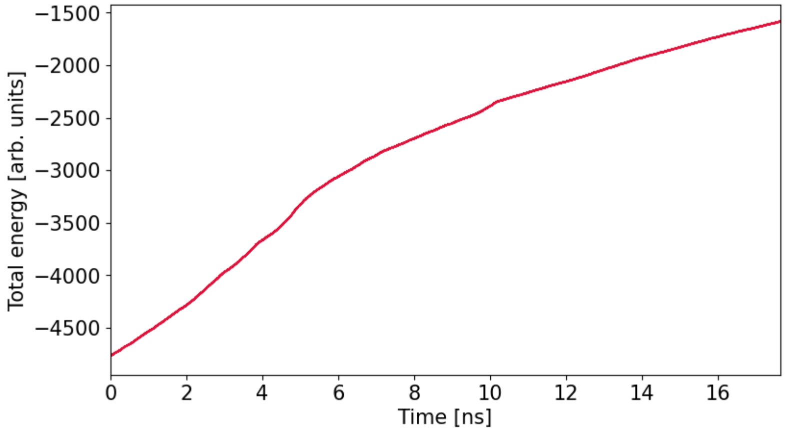

Figure 3.25: The evolution of the total energy of the system over the course of NPT MD using model 1 +VeLO trained purely on RPBE forces.

3.3 DFTD3-JAX

Since a NNFF based on the locality principle, such as NeuralIL, excels in reproducing energies and forces derived from local interactions, we were intrigued to explore the scenario where NeuralIL models are solely trained on DFT forces, with long-range dispersion interactions added subsequently to the predicted forces and energies within the neural network model. Our objective was to assess whether this approach, which treats forces resulting from short and long-range interactions separately, yields more accurate reproduction of the reference forces and more precise density values for water.

To achieve compatibility with NeuralIL, I develeoped a DFT-D3 implementation of using the JAX framework. For this purpose, I used an existing Python program called Torch-dftd [12] as a reference implementation of DFT-D3 method using the PyTorch library [40]. The first step involved modifying the program by replacing all PyTorch functions with their JAX-Numpy equivalents. However, not all PyTorch functions and methods have direct counterparts in JAX-Numpy, so I had to implement some of them from scratch. For example, I needed to create an equivalent function for Tensor.scatter_add(dim, index, src). This function adds values from the src tensor into the self tensor at the indices specified in the index tensor. The index mapping is determined by the indices in src for dimensions other than dim, while the corresponding values in the index tensor are used for dimension = dim. I programmed a similar function called scatter_add_jax(ndarray, ind, src) in JAX, which takes an n-dimensional jax.numpy array as input and performs the same operation. The modified version of DFT-D3, named Dftd3-jax, with the necessary functions implemented in JAX, is stored on the TUW GitLab.

Once Dftd3-jax was implemented, it was necessary to assess its performance and verify the correctness of the results. To accomplish this, a series of test cases were conducted. The tests aimed to evaluate the accuracy and consistency of Dftd3-jax compared to the

Table 3.1: Parameters of DFT-D3 which do not depend on the chemical nature or geometry of the compound or material the dispersion energy of which should be calculated. The values of $s_{r,6}$ and s_8 are determined by the choice of the exchange-correlation density functional, which is in this case RPBE.

Parameters	Values
s_6	1.0
$s_{r,6}$	0.872
s_8	0.514
$s_{r,8}$	1.0
α_6	14.0
α_8	16.0

original Torch-dftd program. The test cases included the following structures: bulk water, a water dimer, and a single water molecule. Each structure served a specific purpose in the evaluation process. The bulk water structure was used to examine the dispersion interactions in a true bulk material and involved applying periodic boundary conditions in all directions. On the other hand, the water cluster structure represented a small assembly of water molecules in a vacuum, where no periodic boundary conditions were necessary. Additionally, the water dimer was utilised to compare dissociation curves obtained using Dftd3-jax and Torch-dftd using the RPBE functional, ensuring their consistency. The test code, written using the Pytest library [41], compared the values calculated by Dftd3-jax with the corresponding values obtained from Torch-dftd in DFT-D3 mode. The test was considered successful if the Dftd3-jax values matched the Torch-dftd values with a precision of at least 10^{-4} eV for energies and 10^{-6} for forces. All of the test cases were sucessfully passed by the Dftd3-jax program.

As evidence of the successful dissociation curve test case, Figure 3.26 illustrates the dispersion energies of two water molecules at varying distances calculated using both Torch-dftd and Dftd3-jax. The compound nonspecific DFT-D3 parameters and their values for this calculations are given in the Table 3.1. The values of $s_{r,6}$ and $s_{r,8}$ are smaller compared to those of the PBE functional, used in Section 2.1.1 which results in larger absolute value of the dispersion energy. The plot clearly shows that the dispersion energy values obtained from both programs are practically identical, further confirming the accuracy and reliability of Dftd3-jax.

However, although the mentioned tests were all passed successfully, it turned out that at this point Dftd3-jax can not be used in a NeuralIL model. There are several reasons for this. First, in order to account in case of periodic boundary conditions for interactions at up to 50 Å, which is a default distance value, the program adds to each side of the cell several copies of the cell so that those distances of up to 50 Å could be modelled. Moreover the indexes of atoms that interact pairwise with each other (the so called neighbour list)

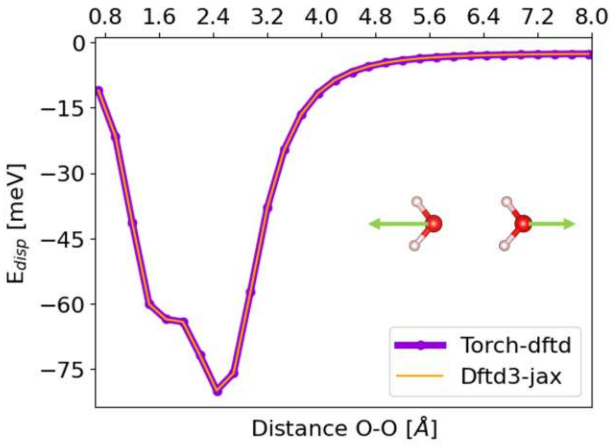

Figure 3.26: Dispersion energy dissotiation curves calculated using Torch-dftd (violet curve) and Dftd3-jax (orange curve) for RPBE functional, zero damping and no tree-body interactions

need to be determined. For this purpose, a Pymatgen's [42] implementation of neighbour list or the primitive_neighbor_list function from ASE python package [33] are used in both Torch-dftd and Dftd3-jax. However, when the Dftd3-jax was used inside a NeuralIL model, it turned out that Pymatgen is incompatible with it and that the ASE function is too slow. So it would be necessary to use some other function, ideally in JAX that would be compatible with NeuralIL and would be quick. In the JAX-MD package there are some functions that could be used for this purpose. However, during an NPT MD simulation the volume of the cell changes and so also the number of pairwise interacting atoms at a distance up to 50 Å changes too. This could lead to different sizes of neighbour list array which would make the calculation slower, since the Dftd3-jax would need to be recompiled at every time step. This is something that needs to be solved too.

Another significant issue with the current implementation of Dftd3-jax is its non-jittable nature, which prevents effective utilisation of the high-performance XLA compiler. Consequently, the program is not performing as fast as it has the potential to be. JAX imposes strict limitations on jitability, particularly for functions that contain conditionals (if-else structures) and accept JAX-incompatible types like strings. Unfortunately, the Dftd3-jax program heavily relies on if-else structures due to the DFT-D3 method's parameter dependence, with certain parameter values being calculated on the fly based on other parameters. To enhance the program's speed, efficiency, and usability within NeuralIL, it would be necessary to reprogram it in a way that makes it jittable. This involves addressing the if-else structures and ensuring compatibility with JAX's requirements for jitability. This effort to reprogram Dftd3-jax to enable jitability will be a crucial objective for future work. By making the program jittable, it can unlock its full potential for accelerated performance and effective integration with NeuralIL.

3.4 New Dataset with vdW-DF

The DFT-D3 method offers numerous benefits, such as being highly efficient and conceptual simplicity [3]. However, it does involve rather drastic approximations. For instance, the non-additive many-body dispersion interaction is approximated by a pairwise interaction between atomic pairs. Additionally, the dispersion coefficient is not directly dependent on the electron density, which renders this method unsuitable for systems with charge transfer, such as in the case of redox reactions [3], [15].

On the other hand, vdW density functionals in DFT are more grounded in physics and incorporate electron density-dependent dispersion energy. Research has demonstrated the successful application of vdW functionals, such as vdW-DF-cx, in accurately calculating the total potential energy, including dispersion energy, for both bulk metal surfaces and charge transfer systems [43]. Unlike DFT-D3, vdW functionals can provide a more robust approach for capturing dispersion interactions on metallic surfaces [43], [44].

Given the research focus of my group on modeling liquid-solid and solid-solid interfaces, I have been exploring vdW-DF as an alternative method to the vdW-D3 approach, which may not be the most suitable for our purposes. Consequently, I am interested in investigating whether a bulk water dataset, where energies and forces are calculated using a vdW-DF functional, would provide superior quality data for training a neural network compared to RPBE corrected with DFT-D3. Since vdW-DF functionals are more rooted in physics, the calculated energies and forces may contain more physically accurate information about the interactions between molecules. This, in turn, could potentially facilitate and expedite the training process of the neural network and improve the neural network's ability to predict dispersion interaction forces despite the used locality approximation, beeing a handicap in this regard.

During my two-week stay in Norway, I had the opportunity to join Dr. Berland's team. Dr. Berland is one of the foremost experts in vdW density functionals and has played a crucial role in the development of several notable functionals, including vdW-DF-cx and vdW-DF3-mc, among others. Currently affiliated with the Norwegian University of Life Sciences (NMBU) in Ås, Dr. Berland provided valuable guidance and expertise throughout the analysis described in the following sections.

3.4.1 Choosing the best energy cutoff for plane waves

The question at hand was which vdW density functional is best suited for the bulk water dataset, specifically in accurately modelling the interactions between water molecules. To address this, we compared the vdW-DF functionals with the exact calculations obtained from the coupled cluster method CCSD(T) using the BEGDB. Two datasets from BEGDB were particularly relevant for our analysis: the S66x8 water dimer dissociation curve dataset [45] and the dataset comprising the potential energies of 38 water clusters [35].

For calculating the potential energies using vdW density functionals in Quantum Espresso [46], which employs the plane wave formalism and utilises pseudopotentials to treat electrons, we opted for the Optimized Norm-Conserving Vanderbilt (ONCV) pseudopotential. This pseudopotential, recommended and provided by the research group of Dr. Berland, was deemed suitable for our purposes. To determine the energy cutoff for the plane waves, we sought an optimal value specifically tailored for water compounds. The desired cutoff should be large enough to ensure that the calculated potential energy per water molecule remains relatively stable within a range of values around the chosen cutoff. At the same time, it should be small enough to avoid unnecessary computational overhead and expedite the calculations.

To determine the ideal cutoff, we selected two water clusters: a water dimer and a cluster comprising five water molecules. Each cluster was placed in a simulation box with dimensions of 10×10×10 Å for the water dimer and 13×13×13 Å for the five-water molecule cluster. For both cases, a 1×1×1 *k*-point grid was employed to cover the first Brillouin zone. Additional parameters used during the calculations can be found in the Table 3.2.

Table 3.2: The parameters used for all the calculations carried out using Quantum Espresso program.

Keyword	Option
occupations	smearing
degauss	0.05
smearing	gaussian
conv_tr	$1e^{-6}$
diagonalization	cg

The potential energy of the water dimer was computed using the vdW-DF1 functional for various energy cutoffs, namely 544.4, 680.5, 816.6, 952.7, 1088.8, 1224.9, 1361, 1497.1, and 1633.2 eV. The change in potential energy as a function of the energy cutoff is depicted in Figure 3.27. It was evident that the potential energy reached a constant value only when using a cutoff of 1088.8 eV. Therefore, it was deemed unnecessary to evaluate the other water cluster at a small cutoff such as 544.4 eV. Consequently, the analysis of the potential energy as a function of the energy cutoff for the second cluster started at the value of 816.6 eV. The evolution of the potential energy for this cluster is illustrated in Figure 3.28.

In the Figure, it is evident that for the second water cluster, the potential energy reaches a constant value at a larger cutoff compared to the dimer. Specifically, the potential energy becomes constant starting from a cutoff of 1224.9 eV. This indicates that a higher cutoff is necessary to accurately capture the potential energy behavior for the larger water cluster.

Next, in order to assess the difference in potential energy per water molecule between the dimer and the other cluster, the calculated potential energies at various cutoff were

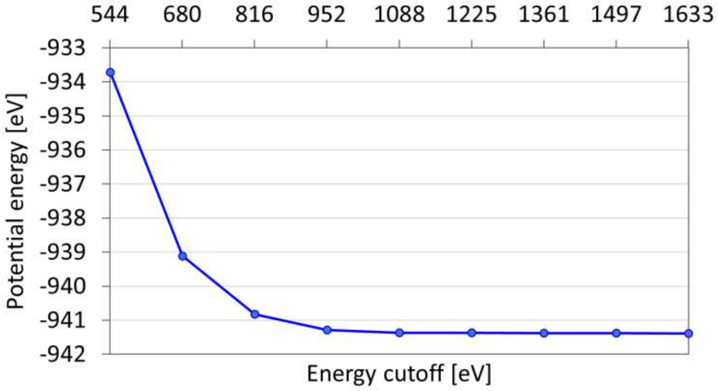

Figure 3.27: Potential energy of a water dimer as a function of the energy cutoff used for plane waves.

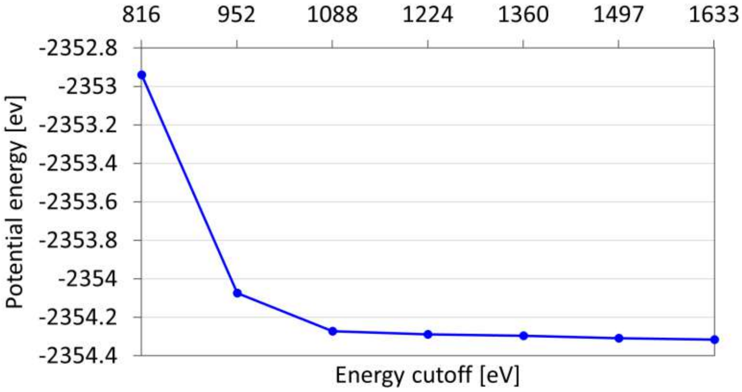

Figure 3.28: Potential energy of a water cluster comprising five water molecules as a function of the energy cutoff used for plane waves.

divided by the respective number of molecules in each cluster. Then, the energy of the dimer was subtracted from the energy of the other cluster using the following equation:

$$E_{diff} = \frac{E_{\text{pot}}^{\text{dimer}}}{2} - \frac{E_{\text{pot}}^{\text{5mer}}}{5} \tag{3.4}$$

Finally, the accuracy of the potential energy calculation around the selected energy cutoff values was determined by calculating the differences in potential energy differences, denoted as Δ. Using lowercase letters a, b, c, and so on to represent the potential energy differences at specific cutoff energy values in the Figure 3.29, the delta values were calculated according to the following scheme:

$$\Delta_1 = a - b \tag{3.5}$$
$$\Delta_2 = b - c \tag{3.6}$$
$$\ldots \tag{3.7}$$

The changes in Δ are illustrated in Figure 3.29 (b). It follows from the figure that the smallest Δ, indicating the highest accuracy, was achieved for the intervals 2, 3, and 6, which correspond to the energy cutoff ranges of 952.7-1088.8, 1088.8-1224.9, and 1497.1-1633.2 eV.

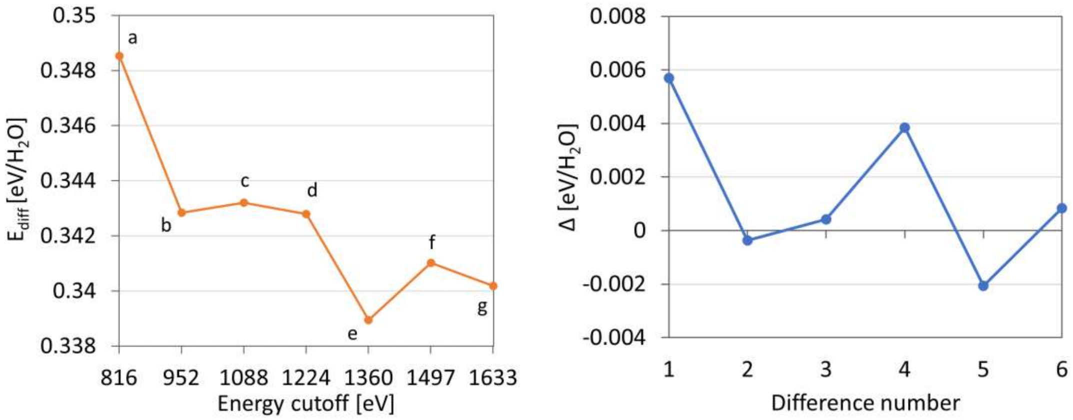

Figure 3.29: The differencies of potential energies per water molecules of water dimer and water pentamer (left) and the differences of those (Δ) (right)

The accuracy criterion for choosing the energy cutoff was set at a precision of at least 10^{-3} eV/Å, considering that approximately one third of the dispersion forces in the bulk water dataset fall within this range as we found out in the Section 3.1.1. The accuracy achieved in the interval of 952.7-1224.9 eV was found to be $-3.6 \cdot 10^{-4}$ and $4.0 \cdot 10^{-4}$ eV, satisfying the criterion. However, as the energy cutoff was further increased, the accuracy initially decreased and then increased again in the range of 1497.1-1633.2 eV, reaching $8.0 \cdot 10^{-4}$ eV. Due to the observed accuracy drop and the indication that accuracy was not yet converged at that point, the energy cutoff radius of 1497.1 eV was chosen as it is more likely to have higher convergence of accuracy.

3.4.2 Water dimer dissociation curves

Once the decision about the cutoff energy was made, I was able to proceed to calculate the dissociation curves of a water dimer. The dissociation curve was calculated using water dimer structures from the BEGDB database, that consisted of two water molecules at different distances between 1.81 Å to 4.02 Å. Those were reached by multiplying the closest intermolecular distance between the molecules by scaling factors of 0.9, 1.0, 1.05,

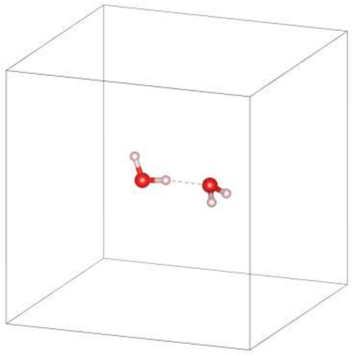

Figure 3.30: The water dimer structure which corresponds to distance scaling of 0.90

1.1, 1.25, 1.5 and 2.0. The water dimers were put into a box of constant size of 10×10×10 Å and used again 1×1×1 k-point. The first dimer configuration from the dissociation curve that corresponds to the scaling factor of 0.9 is depicted in the Figure 3.30.

In this case, the potential energy of the configurations was calculated using several kinds of vdW density functionals. The chosen functionals are vdW-DF1, vdW-DF2, vdW-DF-cx, vdW-DF-opt and vdW-DF-opt2. Since for calculating energies and forces of the bulk water dataset described in previous section were used RPBE and RPBE with the D3 correction, we wanted to analyse them too and see how good they can actually model the interactions between the water molecules compared to vdW-DF and CCSD(T). The calculations were carried out using the same parameters for the Quantum Espresso as in the previous section. The potential energies calculated via various density functionals for water molecules at different distances are depicted in the Figure 3.31.

One can see that various vdW density functionals approximate the CCSD(T) potential energy curve differently good at various distances of the water molecules. At rather small scalings of 0.9-1.15 it turns out that the density functional that approximates CCSD(T) the best is vdW-DF-cx. However in case of larger scalings than 1.15 the vdW-DF-cx, just like all chosen vdW and GGA density functionals, overestimates the potential energy. In this region, all the vdW density functionals performed simillarly well. However, the values obtained using the vdW-DF-opt seemd to be the closes to the CCSD(T) values in this region. Interesstingly, the RPBE functional showed a large tendency to underestimate the potential energy around its minimum. However, the inclusion of the dispersion correction appeared to correct RPBE potential energies, since they deviate less from the CCSD(T) energies.

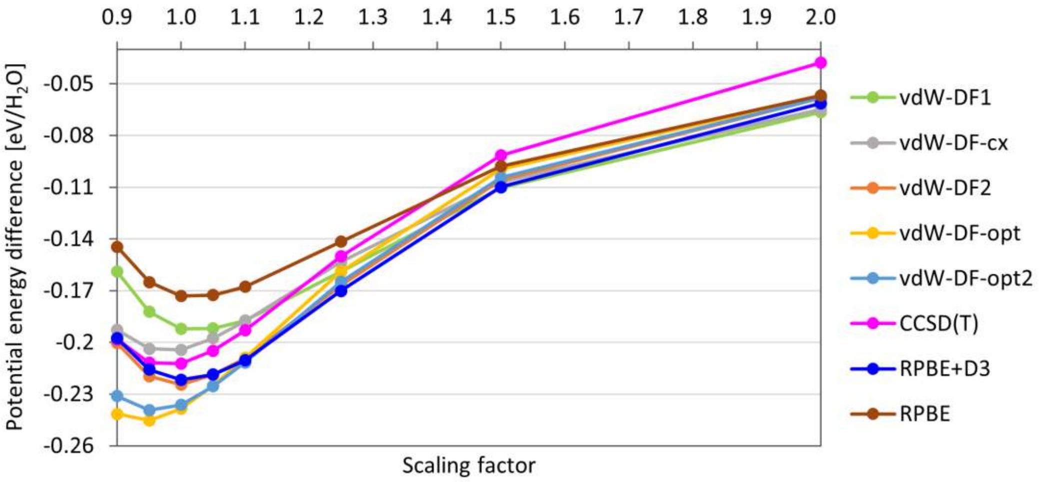

Figure 3.31: Dissociation curves of a water dimer calculated using various density functionals compared to the reference CCSD(T) values

3.4.3 Potential energy of water clusters

In the next step, the 38 water cluster structures from the BEGDB database [35], ranging from clusters consisting of 2 to 10 water molecules, were enclosed within a box measuring 13×13×13 Å. Some examples of the structures contained in the database are depicted in the Figure 3.12. The reciprocal space was once again sampled using a 1×1×1 k-point grid. The calculations were performed using Quantum Espresso, employing the same parameters as in the previous steps. The resulting potential energies, both calculated and reference values, are presented in Figure 3.32 and Figure 3.33. The potential energies are plotted as a function of the cluster names.

From Figures 3.32 and 3.33, it is evident that the density functionals that provide the closest reproduction of the potential energies are once again vdW-DF-cx, RPBE+D3 and, notably, vdW-DF2. It is worth noting that RPBE, on the other hand, tends to underestimate the potential energies, as evidenced by the calculated values being higher than those obtained from CCSD(T). However, a notable observation is that the addition of the DFT-D3 correction helps bring the calculated potential energies closer to the reference CCSD(T) values, just like it could be observed in case of water dimer dissociation curves in the Figure 3.31.

Based on the analysis of the water clusters and water dissociation curves, it was determined that the vdW-DF functional capable of reproducing the CCSD(T) data was vdW-DF-cx. Consequently, it was chosen to calculate the potential energies and forces of the bulk water dataset. However, it is important to note that the analysis of the new dataset will be part of future work and is not included in this master's thesis due to the significant computation time required (of several months).

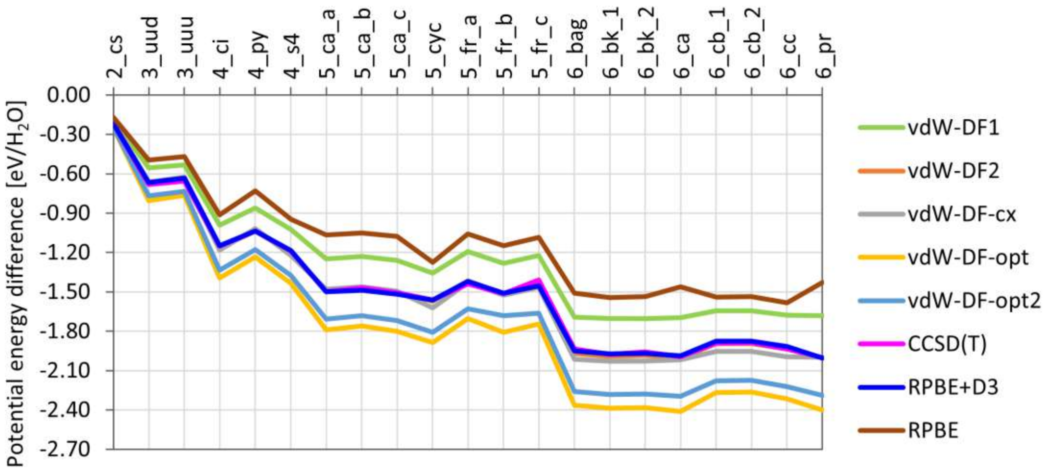

Figure 3.32: Potential energies of water dimer, trimers, tetramers, pentamers and hexamers calculated using various density functionals compared to the reference CCSD(T) values.

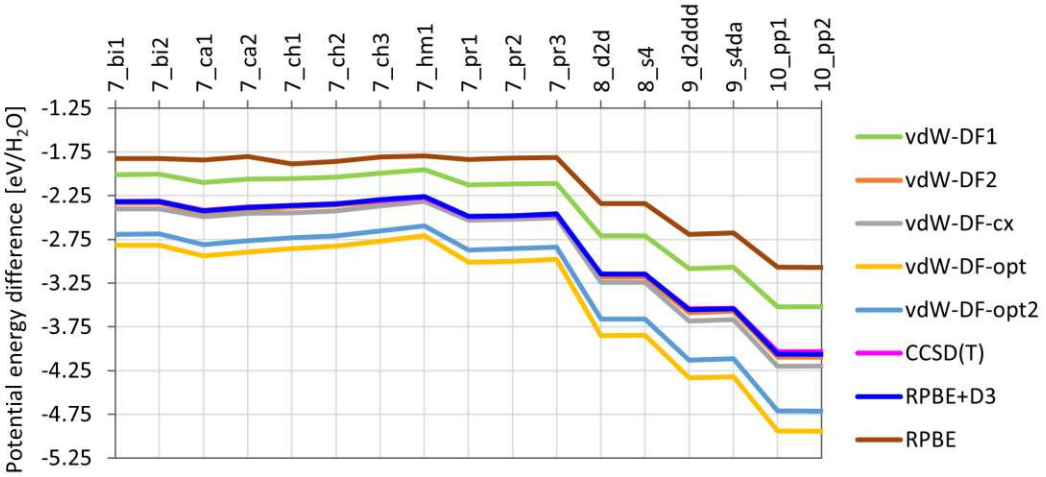

Figure 3.33: Potential energies of water heptamers, octamers, nonamers and decamers calculated using various density functionals compared to the reference CCSD(T) values.

3.4.4 Volume expansion

Additionally, an investigation was conducted to examine how the potential energy, including the dispersion interaction energy, of a water system changes as the volume of the unit cell is expanded. This analysis aimed to understand the sensitivity of the system's potential energy, specifically with regard to increasing intermolecular distances resulting from the expansion of the cell. To begin, we opted to test this phenomenon on a large water cluster. The selected cluster, named 10-PP1 in the BEGDB database's water cluster dataset, was chosen due to its high number of water molecules, making it suitable for our purposes. Since there was no existing Python package available for expanding the cell's volume without enlarging the interatomic distances within the molecules, I developed a custom Python script to accomplish this task.

The central step in my Python program involved extracting individual molecules as separate objects from the entire configuration. To accomplish this, I utilised a function called "decompose" from the Moltools module, generously provided by Dr. Berland and his team. Once the molecules were successfully split into separate objects, the calculation of their respective center of mass and the overall structure's centre of mass followed.

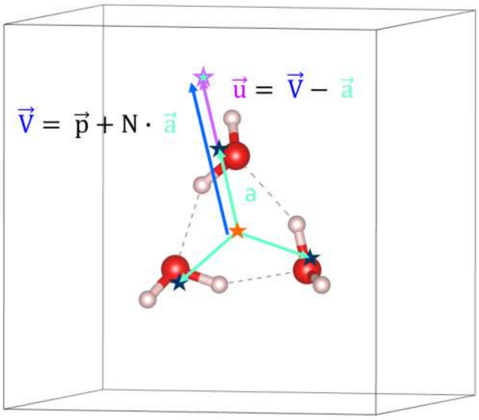

Figure 3.34: Sketch of the concept of the volume expanding program: directional vectors (turquois arrows a) that point away from the center of mass of the whole cluster (orange star) to the centers of masses of the individual molecules (black stars). By multiplying those vectors by a scale factor N (in this case N=0.5, 1.0, ...) and adding them to the initial positions of the molecular centers of masses, a new vector V can be calculated. By subtracting the directional vectors from the new vectors V, the actual shift vector u is obtained, that is added to the positional coordinates of the atoms in molecules, by which the molecules are shifted.

The subsequent task was to determine the direction vector that would govern the positional changes of individual molecules during the volume expansion. This vector was defined as originating from the global center of mass of the structure and pointing towards the centers of mass of the individual molecules. By multiplying these directional vectors appropriately, the positions of the molecules could be translated. This involved adding the product of the directional vector's coordinates to the coordinates of the molecular centers of mass. Consequently, the difference in coordinates between the new and old centers of masses was computed. This difference was then added to the coordinates of the atoms in the molecules, effectively moving the molecules away from the global center of mass. This procedure is also illustrated using a water trimer in Figure 3.34. To ensure that the long-range interactions between structure images were negligible, the

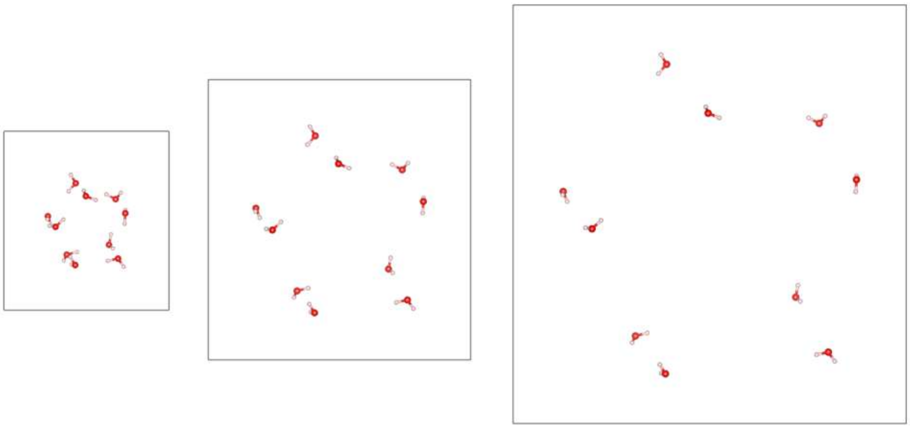

Figure 3.35: The change in the volume of the cell and distances between water molecules in the 10-PP1 cluster for the scale factors of translational vector of 0.5, 3.0, and 4.0.

cell's volume was modified to include a 4 Å vacuum region on each side. This adjustment guaranteed minimal interaction between the structure's images.

For the cluster 10-PP1 a series of seven structures were generated by progressively expanding the cell. These shift values were set at 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5 and 4.0. This expansion resulted in an increase in the distances between water molecules, ranging from around 2.0 Å in the original structure to around 13 Å in the structure with a shift of 4.0. The expansion of the cell containing the 10-PP1 cluster is also illustrated in the Figure 3.35.

Subsequently, the potential energy of each structure was computed using the Quantum Espresso software package, employing the RPBE and vdW-DF-cx density functionals, vdW density functional being the most accurate as shown in the two previous sections, with the same parameters as described earlier. Additionally, for the RPBE functional the dispersion correction was calculated using the DFT-D3 method. To facilitate comparisons, the potential energies were adjusted by subtracting the highest values within the potential energy dataset of the same structure for each density functional. These modified potential energies were then plotted. Figure 3.36 depicts the potential energies for RPBE, vdW-DF-cx, and RPBE+DFT-D3 as a function of scale factor of translational vector. Notably, it can be observed that for all density functionals, the potential energy steadily increased until the scale factor of translational vector reached 3.5, which corresponds to the distance between the molecules of around 12 Å. Beyond this point, there was no further change in the potential energy, which remained constant.

Upon comparing the two different types of functionals, RPBE and vdW-DF-cx, a slight difference can be observed in the shape of the potential energy curves. However, an interesting observation arises when the RPBE energies are corrected using DFT-D3. In this case, the difference in the shape of the curves between the two functionals becomes almost negligible.

To sum up, it seems that during an isotropic expansion of the volume of a larger, symmetric water cluster, the vdW-DF-cx functional and dispersion corrected RPBE+DFT-D3 tend to behave very similarly. However, it would be also interesting to investigate, if this result could be reproduced in case of any water configuration out of the bulk water dataset. This will be researched more in the future work.

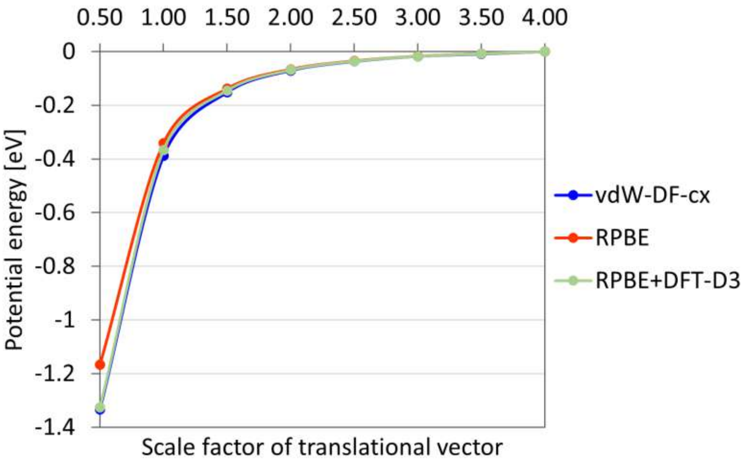

Figure 3.36: Potential energy of the 10-PP1 cluster at various degees of cell expansion (various scale factor of translation vector values) calculated using vdW-DF-cx and RPBE with and without the DFT-D3 correction.

CHAPTER 4

Conclusions

The analysis conducted in Section 3.1.1 revealed that the dispersion interactions in the bulk water dataset were found to be very small. Approximately 35% of the dispersion forces in this dataset were smaller than 10^{-3} eV/Å, indicating their relatively limited contribution compared to the forces calculated using DFT with the RPBE functional. Next, in case of model 1 and model 2, the inclusion of dispersion interactions using the DFT-D3 correction had a very small influence on the resulting RMSE in predicted forces and the quality of the training. In case of model 1, the inclusion of DFT-D3 correction lead to smaller RMSE by 0.002 ev/Å and in case of model 2 to larger RMSE by 0.001 eV/Å compared to the training on forces without DFT-D3. There were also some tiny differences in performance between the two models. Model 2, having larger radius of descriptors, was not able to reach as small RMSE during the training than model 1 having smaller radius of descriptors. Because of larger descriptors, model 2 should be able to capture more information about the interactions as model 1, however it seems that this extra piece of information did not help the model to get trained better. From the comparison of the older version of NeuralIL model with the newest NeuralIL+VeLO results, that using NeuralIL+VeLO it was possible to reach a clear improvement in the speed of the training (speed up of 5.3 to 5.6 reached for model 1 and 2 compared to the older version) and even the differences in the RMSE between the two training scenarios seemingly disappeared. Finally by testing the predictions of the two neural network models trained on bulk water on the water cluster dataset, I could show that they were able to predict the forces for those correctly, even though they were trained on a dataset, where the water molecules had a different local chemical environment. The reached RMSE for the water cluster dataset were very small, for model 1 it was 0.0778 and 0.0789 ev/Å and for model 2 it was 0.0831 and 0.101 ev/Å. So the two NeuralIL models showed a good transferability in case of condensed phase water datasets.

The molecular dynamics simulations described in Section 3.2 highlighted the importance of parameter selection and the time-consuming nature of finding the right parameters for a simulation. From the results presented in Sections 3.2.2 and 3.2.3, it was observed that utilising model 1 trained on forces with the dispersion correction yielded a density value at 250 K and 1 bar that was closer to the experimental value compared to the neural network without DFT-D3 training. This suggests that incorporating information about long-range interactions in the training set can alleviate the shortsightedness of NeuralIL to some extent. However, it should be noted that even with the inclusion of DFT-D3 contributions in the training set for model 1, the interatomic interactions were still not accurately modelled in NeuralIL. As a result of this, the density value deviated by 0.16 g/cm^3 from the experimental value, which was larger than the deviation obtained by another neural network developed by Dellago's group, which was approximately 0.1 g/cm^3. Additionally, the observed lack of invariance in the total energy of the simulation box indicated a potential issue with the simulation settings or the simulation program itself. As a result, the obtained results from these simulations should be interpreted with caution, and further investigation is needed to resolve the underlying error before drawing definitive conclusions.

The JAX version of the Torch-dftd program for calculating the dispersion interactions in condensed matter using the DFT-D3 scheme is capable to calculate correctly the energies and forces in bulk and molecular water and is able to reproduce correctly the dissociation curve of a water dimer. However because of the issues described in the Section 3.3, it is still not possible to use it directly inside a NNFF.

In Section 3.4, it was demonstrated that vdW-DF-cx, vdW-DF2 and RPBE+DFT-D3 functionals perform better in modelling the interactions between water molecules compared to the RPBE. This conclusion was supported by comparing the potential energies of water clusters and water dimer dissociation curves with the exact energies obtained from CCSD(T) calculations. RPBE showed limitations in accurately reproducing the interactions between water molecules, but the inclusion of the dispersion correction DFT-D3 to the RPBE energies led to a significant improvement in the potential energy values, bringing them closer to the CCSD(T) values. The sensitivity of dispersion interactions to changes in volume and intermolecular distances is an interesting finding. This information is particularly relevant in the context of molecular dynamics simulations utilising an NPT ensemble, where the pressure is determined by calculating the stress, which corresponds to the change in potential energy associated with changes in the cell size. Understanding the influence of dispersion interactions on the overall system behaviour can provide valuable insights for accurately simulating and characterising molecular systems.

CHAPTER 5

Outlook

In the future, it would be worthwhile to consider training NeuralIL with a descriptor cutoff value of 6 Å, as commonly done in the literature. This investigation could provide insights into whether using this cutoff value improves the training performance or not, and compare it to what was observed for model 2. Furthermore, the hypothesis proposed in Section 3.1.4, regarding the neural network's ability to recognise cluster-like structures within bulk water with the assistance of the relatively small cutoff radius used in the models, could be explored further. This could involve conducting principal component analysis of the descriptors to gain a deeper understanding of the network's recognition capabilities.

Moving forward, it is crucial to identify and rectify the errors in the simulation program used for MD. Once resolved, the MD simulations should be repeated to obtain accurate and reliable results. Similarly, the same type of simulation should be conducted for model 2 to investigate whether the larger size of descriptors improves the predicted water density. In addition to the existing four models, two additional models should be developed. Those will be similar to model 1 and model 2 but trained solely on local forces. However, during force and energy predictions, the DFT-D3 correction will be calculated using Dftd3-jax program that will be integrated directly into the neural network models. Using these six NeuralIL models, the densities of water should be calculated at various temperatures and compared to the reference density curves [8]. This analysis will provide insights into the performance and accuracy of the different models in reproducing water density.

Furthermore, it is of interest to explore the performance of the models in predicting water densities when trained on both forces and energies simultaneously. Currently, the NeuralIL models were trained solely on atomic forces, which do not incorporate information about stress. As was explained in the Section 3.2.2, in MD simulations, stress plays a crucial role in calculating the pressure within the simulation box. Inaccurately calculated stress might influence negatively the barostat's capability to keep the average pressure close to the intended value. By training the neural network models on both forces and energies, the stress contribution to the energy can be captured, possibly ensuring more accurate predictions of energies. To investigate the impact of training on both forces and energies, MD simulations using the NeuralIL models trained on a combination of forces and energies should be performed. The resulting water densities should be analysed to assess any improvements in the accuracy of the predicted values compared to the models trained solely on forces. Additionally, the error in stress predicted by the models trained only on forces should be evaluated and compared to the models trained on both forces and energies to determine the significance of incorporating energy information in stress calculation.

In order to further enhance the modelling capabilities, a new bulk water dataset will be constructed, incorporating energies and forces calculated using the vdW-DF-cx and vdW-DF2 density functionals. This new dataset will be derived from the structures in the previous bulk water dataset. The neural network models will then be trained using this updated dataset. The training process will be analysed once again, as described in Section 3.1.3, to assess the performance and effectiveness of the models. Additionally, it will be investigated whether the utilisation of this new dataset, which better captures the intermolecular interactions in water, leads to more accurate results during MD simulations. Furthermore, it should be noted that the energies and forces in the original bulk water dataset, calculated using the RPBE functional, were based on a relatively small energy cutoff of the plane waves, resulting in lower precision compared to the vdW-DF-cx bulk water dataset. To address this discrepancy, it is advisable to recalculate the energies and forces using an energy cutoff that provides comparable accuracy to the vdW-DF-cx dataset. This will enable a more meaningful comparison between the RPBE and vdW-DF-cx functional in terms of their performance and predictive capabilities. By employing the more accurate data, a comprehensive assessment of the RPBE and vdW-DF-cx functionals can be conducted, shedding light on their respective strengths and weaknesses in modelling water systems.

In order to gain a deeper understanding of the role of dispersion interactions in the NeuralIL models, it is valuable to analyse how much of the total dispersion interaction can be described within the descriptors of radius 3.5 - 4.0 Å, as utilised in model 1 and model 2. A comparison will be made with the scenario where a selected water molecule interacts with all neighbouring molecules within a distance of 50 Å, as defined in the DFT-D3 method. By assessing the differences in the calculated dispersion interactions between these two approaches, valuable insights can be gained regarding the effectiveness of the local interaction description employed by NeuralIL in capturing long-range dispersion interactions in the bulk water system. This analysis will provide a comprehensive understanding of the interplay between the local and long-range dispersion interactions, elucidating why NeuralIL, despite its focus on local interactions, has exhibited relative success in modelling the long-range dispersion interactions in the bulk water system.

List of Figures

List of Tables

Bibliography

[1] Alberto Ambrosetti et al. "Wavelike charge density fluctuations and van der Waals interactions at the nanoscale". In: *Science* 351.6278 (2016), pp. 1171–1176. DOI: `10.1126/science.aae0509`.

[2] M. Dion et al. "Van der Waals Density Functional for General Geometries". In: *Phys. Rev. Lett.* 92 (24 June 2004), p. 246401. DOI: `10.1103/PhysRevLett.92.246401`.

[3] Stefan Grimme et al. "A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu". In: *The Journal of Chemical Physics* 132.15 (Apr. 2010), p. 154104. ISSN: 0021-9606. DOI: `10.1063/1.3382344`.

[4] Jörg Behler. "First Principles Neural Network Potentials for Reactive Simulations of Large Molecular and Condensed Systems". In: *Angewandte Chemie International Edition* 56.42 (2017), pp. 12828–12840. DOI: `https://doi.org/10.1002/anie.201703114`.

[5] Bingqing Cheng et al. "Ab initio thermodynamics of liquid and solid water". In: *Proceedings of the National Academy of Sciences* 116.4 (2019), pp. 1110–1115. DOI: `10.1073/pnas.1815117116`.

[6] Khosrow Shakouri et al. "Accurate Neural Network Description of Surface Phonons in Reactive Gas–Surface Dynamics: N2 + Ru(0001)". In: *The Journal of Physical Chemistry Letters* 8.10 (2017), pp. 2131–2136. DOI: `10.1021/acs.jpclett.7b00784`.

[7] Hadrián Montes-Campos et al. "A Differentiable Neural-Network Force Field for Ionic Liquids". In: *Journal of Chemical Information and Modeling* 62.1 (2022), pp. 88–101. DOI: `10.1021/acs.jcim.1c01380`.

[8] Tobias Morawietz et al. "How van der Waals interactions determine the unique properties of water". In: *Proceedings of the National Academy of Sciences* 113.30 (2016), pp. 8368–8373. DOI: `10.1073/pnas.1602375113`.

[9] Kevin Maik Jablonka et al. "Big-Data Science in Porous Materials: Materials Genomics and Machine Learning". In: *Chemical Reviews* 120.16 (2020), pp. 8066–8129. DOI: `10.1021/acs.chemrev.0c00004`.

[10] Andrea Grisafi and Michele Ceriotti. “Incorporating long-range physics in atomic-scale machine learning”. In: *The Journal of Chemical Physics* 151.20 (Nov. 2019), p. 204105. ISSN: 0021-9606. DOI: `10.1063/1.5128375`.

[11] Schoenholz et al. “JAX M.D. A Framework for Differentiable Physics”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020. URL: `https://papers.nips.cc/paper/2020/file/83d3d4b6c9579515e1679aca8cbc` `Paper.pdf`.

[12] So Takamoto et al. *PFP: Universal Neural Network Potential for Material Discovery.* 2021. arXiv: `2106.14583 [cond-mat.mtrl-sci]`.

[13] G. K. H. Madsen. “Introduction to Electronic Structure”. In: (2021), pp. 22–23.

[14] János Ángyán. *London dispersion forces in molecules, solids and nano-structures : an introduction to physical models and computational methods / by János Ángyán, John Dobson, Georg Jansen and Tim Gould.* eng. Theoretical and computational chemistry ; 16. Croydon: Royal Society of Chemistry, 2020. ISBN: 9781782620457.

[15] Jan Hermann, Robert A. Jr. DiStasio, and Alexandre Tkatchenko. “First-Principles Models for van der Waals Interactions in Molecules and Materials: Concepts, Theory, and Applications”. In: *Chemical Reviews* 117.6 (2017), pp. 4714–4758. DOI: `10.1021/acs.chemrev.6b00446`.

[16] Romain Jonchiere et al. “Van der Waals effects in ab initio water at ambient and supercritical conditions”. In: *The Journal of Chemical Physics* 135.15 (Oct. 2011), p. 154503. ISSN: 0021-9606. DOI: `10.1063/1.3651474`.

[17] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Phys. Rev. Lett.* 77 (18 1996), pp. 3865–3868.

[18] Kyuho Lee et al. “Higher-accuracy van der Waals density functional”. In: *Phys. Rev. B* 82 (8 Aug. 2010), p. 081101. DOI: `10.1103/PhysRevB.82.081101`.

[19] Jiří Klimeš, David R Bowler, and Angelos Michaelides. “Chemical accuracy for the van der Waals density functional”. In: *Journal of Physics: Condensed Matter* 22.2 (Dec. 2009), p. 022201. DOI: `10.1088/0953-8984/22/2/022201`. URL: `https://dx.doi.org/10.1088/0953-8984/22/2/022201`.

[20] Jiří Klimeš, David R Bowler, and Angelos Michaelides. “Chemical accuracy for the van der Waals density functional”. In: *Journal of Physics: Condensed Matter* 22.2 (Dec. 2009), p. 022201. DOI: `10.1088/0953-8984/22/2/022201`. URL: `https://dx.doi.org/10.1088/0953-8984/22/2/022201`.

[21] Kristian Berland and Per Hyldgaard. “Exchange functional that tests the robustness of the plasmon description of the van der Waals density functional”. In: *Phys. Rev. B* 89 (3 Jan. 2014), p. 035412. DOI: `10.1103/PhysRevB.89.035412`.

[22] Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation.* Oxford: Oxford university press, 2010, pp. 61–63.

[23] Mark E. Tuckerman et al. “A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal isobaric ensemble”. In: *Journal of Physics A Mathematical General* 39.19 (May 2006), pp. 5629–5651. DOI: `10.1088/0305-4470/39/19/S18`.

[24] Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford: Oxford university press, 2010, pp. 214–216.

[25] Glenn J. Martyna, Douglas J. Tobias, and Michael L. Klein. “Constant pressure molecular dynamics algorithms”. In: *The Journal of Chemical Physics* 101.5 (Sept. 1994), pp. 4177–4189. ISSN: 0021-9606. DOI: `10.1063/1.467468`.

[26] P. Ramachandran, B. Zoph, and Q. V. Le. “Searching for Activation Functions”. In: *arXiv:1710.05941 [cs.NE]* (2017). URL: `https://arxiv.org/abs/1710.05941`.

[27] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: `http://arxiv.org/abs/1412.6980`.

[28] L. Metz et al. “VeLO: Training Versatile Learned Optimizers by Scaling Up”. In: *arXiv preprint arXiv:2211.09760* (2022). URL: `velo-code.github.io`.

[29] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: `http://github.com/google/jax`.

[30] B. Hammer, L. B. Hansen, and J. K. Nørskov. “Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals”. In: *Phys. Rev. B* 59 (11 Mar. 1999), pp. 7413–7421. DOI: `10.1103/PhysRevB.59.7413`.

[31] Sung Sakong, Katrin Forster-Tonigold, and Axel Groß. “The structure of water at a Pt(111) electrode and the potential of zero charge studied from first principles”. In: *The Journal of Chemical Physics* 144.19 (May 2016), p. 194701. ISSN: 0021-9606. DOI: `10.1063/1.4948638`.

[32] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen. “Real-space grid implementation of the projector augmented wave method”. In: *Phys. Rev. B* 71.3 (2005), p. 035109. DOI: `10.1103/PhysRevB.71.035109`.

[33] Ask Hjorth Larsen et al. “The atomic simulation environment—a Python library for working with atoms”. In: *Journal of Physics: Condensed Matter* 29.27 (2017), p. 273002. URL: `http://stacks.iop.org/0953-8984/29/i=27/a=273002`.

[34] Shuwen Yue et al. “When do short-range atomistic machine-learning models fall short?” In: *The Journal of Chemical Physics* 154.3 (Jan. 2021), p. 034111. ISSN: 0021-9606. DOI: `10.1063/5.0031215`.

[35] Berhane Temelso, Kaye A. Archer, and George C. Shields. “Benchmark Structures and Binding Energies of Small Water Clusters with Anharmonicity Corrections”. In: *The Journal of Physical Chemistry A* 115.43 (2011), pp. 12034–12046. DOI: `10.1021/jp2069489`.

[36] Kazuo Nakamoto. *Infrared and Raman Spectra of Inorganic and Coordination Compounds. Part A: Theory and Applications in Inorganic Chemistry; Part B: Application in Coordination, Organometallic, and Bioinorganic Chemistry, 5th Edition*. Wiley-Interscience, 1997, p. 170.

[37] Claude E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the Institute of Radio Engineers* 37 (1949), pp. 10–21.

[38] Efrem Braun, Seyed Mohamad Moosavi, and Berend Smit. “Anomalous Effects of Velocity Rescaling Algorithms: The Flying Ice Cube Effect Revisited”. In: *Journal of Chemical Theory and Computation* 14.10 (2018), pp. 5262–5272. DOI: `10.1021/acs.jctc.8b00446`.

[39] Jonathan Lahnsteiner et al. “Room-temperature dynamic correlation between methylammonium molecules in lead-iodine based perovskites: An ab initio molecular dynamics perspective”. In: *Phys. Rev. B* 94 (21 Dec. 2016), p. 214114. DOI: `10.1103/PhysRevB.94.214114`.

[40] Adam Paszke et al. “Automatic differentiation in PyTorch”. In: *NIPS-W*. 2017.

[41] Holger Krekel et al. *pytest 7.4*. 2004. URL: `https://github.com/pytest-dev/pytest`.

[42] Shyue Ping Ong et al. “Python Materials Genomics (pymatgen) : A Robust, Open-Source Python Library for Materials Analysis”. In: *Computational Materials Science* 68 (2013), pp. 314–319.

[43] Kristian Berland et al. “van der Waals density functionals built upon the electron-gas tradition: Facing the challenge of competing interactions”. In: *The Journal of Chemical Physics* 140.18 (Apr. 2014), 18A539. ISSN: 0021-9606. DOI: `10.1063/1.4871731`.

[44] Michele Cutini, Lorenzo Maschio, and Piero Ugliengo. “Exfoliation Energy of Layered Materials by DFT-D: Beware of Dispersion!” In: *Journal of Chemical Theory and Computation* 16.8 (2020), pp. 5244–5252. DOI: `10.1021/acs.jctc.0c00149`.

[45] Jan Řezáč, Kevin E. Riley, and Pavel Hobza. “S66: A Well-balanced Database of Benchmark Interaction Energies Relevant to Biomolecular Structures”. In: *Journal of Chemical Theory and Computation* 7.8 (2011), pp. 2427–2438. DOI: `10.1021/ct2002946`.

[46] Paolo Giannozzi et al. “QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials”. In: *Journal of Physics: Condensed Matter* 21.39 (Sept. 2009), p. 395502. DOI: `10.1088/0953-8984/21/39/395502`.