universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Long-ranged electrostatic descriptors for machine-learning force fields“

verfasst von / submitted by

Carolin Faller, BSc

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

Master of Science (MSc)

Wien, 2023 / Vienna, 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Georg Kresse
Mitbetreut von / Co-Supervisor:	

Acknowledgements

First and foremost, I would like to thank my supervisor Georg Kresse and my co-supervisor Carla Verdi for their guidance and constant feedback. Special thanks go to Merzuk Kaltak, with whom I collaborated on the development of new radial basis functions, and to Bernhard Schmiedmayer, who always helped me when I encountered technical problems and let me use his computer for my tests. In addition, I would like to thank Andreas Singraber and Ferenc Karsai for their input and setting the framework for the code.

I would also like to thank SFB TACO and all the scientists working in it for the opportunity to participate and for all the fruitful discussions during meetings and retreats.

Abstract

Machine learned force fields have the potential to accelerate first-principles calculations for the prediction of material structures and properties such as the system energy and forces acting on individual atoms. A kernel-based method is used in this project.
So far, only local structural features are considered when mapping to descriptors. Thus, only the interaction between atoms within a certain cutoff radius is included. Even by systematically increasing the cutoff radius, it is very hard to fit the data accurately and in practice, it is virtually impossible to capture long-ranged (electrostatic) interactions. In this work, we perform the calculation of the descriptors in reciprocal space and always use all atoms to indirectly perform a summation over all periodic images and thus be able to capture long-range interactions.
We adopt Slater-type orbitals as radial basis functions, that are used when mapping onto the descriptors, because the corresponding descriptors better fit the slowly decaying nature of long-range interactions.
When combining these short- and long-ranged descriptors we are able to improve the quality of the fit compared to the short-ranged method for materials where both types of interactions are present.

Kurzfassung

Maschinengelernte Kraftfelder haben das Potenzial, first-principle Berechnungen für die Vorhersage von Materialstrukturen und -eigenschaften, wie z. B. der Systemenergie und der Kräfte, die auf einzelne Atome wirken, zu beschleunigen. In diesem Projekt wird eine kernelbasierte Methode verwendet.
Bislang werden bei der Berechnung der Deskriptoren nur lokale Strukturmerkmale verwendet. So wird nur die Wechselwirkung zwischen Atomen innerhalb eines bestimmten Cutoff-Radius berücksichtigt. Auch wenn man den Cutoff-Radius systematisch vergrößert, ist es sehr schwierig, die Daten genau zu fitten und in der Praxis ist es so unmöglich, langreichweitige (elektrostatische) Wechselwirkungen zu erfassen. In dieser Arbeit führen wir die Berechnung der Deskriptoren im reziproken Raum durch und verwenden immer alle Atome, um indirekt eine Summation über alle periodischen Bilder durchzuführen und so langreichweitige Wechselwirkungen zu erfassen.
Wir verwenden Slater-Orbitale als radiale Basisfunktionen, da die entsprechenden Deskriptoren das langsam abfallende Verhalten der langreichweitigen Wechselwirkungen besser widerspiegeln.
Durch die Kombination von diesen lang- und kurzreichweitigen Deskriptoren können wir die Qualität der Fits im Vergleich zur Methode mit kurzer Reichweite für Materialien, bei denen beide Arten von Wechselwirkungen vorhanden sind, verbessern.

Contents

List of Tables

List of Figures

1. Introduction

About this Project

The main goal is to develop a set of long-ranged descriptors for the `Python` version of the Machine Learning (ML) code, that is used in combination with the Vienna Ab *initio* Simulation Package (`VASP`)[1].
This program is called `polipy4vasp` and it is used to test new things easily, as `Python` allows for a quick and compact implementation of many matrix operations and numerical algorithms, especially when using libraries like `NumPy`[2] and `SciPy`[3]. The goal of `polipy4vasp` is to machine learn the polarization and incorporate long-ranged descriptors. This thesis shows how these long-ranged descriptors and different radial basis functions for them are build up and implemented, and then the model is tested for different materials. The `polipy4vasp` coed is developed as part of subproject P03 of the SFB TACO[4].

Machine Learning

ML is a fast developing field of research that has become progressively more important within the last decade. Its goal is to build methods and algorithms that use data to improve the performance of a given task [1]. In ML, models and algorithms are constructed using training data, and subsequently, are capable of generating predictions for novel data without the need to explicitly program the prediction mechanisms [2].
ML methods are used in various fields and applications [3, 4] like image recognition or auto-translation of text, but also in less popularly known applications such as material sciences for learning force fields.
In general in computational materials science Machine learned force fields (MLFF) have the potential to accelerate first principle (FP) calculations, and those MLFF are a powerful method to predict inter-atomic potentials [5–21]. In this thesis, they are used for the prediction of material properties, such as the system energy and forces acting on individual atoms.
For MLFF, kernel based methods and Neural Networks (NN) are the two ML methods that are mainly used. In recent years NN became more prominent. In this project, we adopt a kernel based method because it requires fewer training data and allows for the choice of a kernel function. In the literature, there are many different choices of such kernels like Gaussian or polynomial kernels [18, 22–24].

[1] `https://vasp.at/`
[2] `https://numpy.org/`
[3] `https://scipy.org/`
[4] `https://sfb-taco.at/`

Motivation for the Long-Ranged Descriptors

Many methods for calculating MLFF are local. This means that they calculate a descriptor based on the local properties around one atom within a certain cutoff radius [13, 14, 18, 20, 25] independent of whether kernel methods or neural networks are used. Thus, only the interaction between atoms within this cutoff radius is included. Even by systematically increasing the cutoff radius, it is virtually impossible to capture long-ranged (electrostatic) interactions due to the increasing computational cost.
Nonetheless, electrostatic interactions play an important role in:

- Ionic systems [26]
- Electrode surfaces [27]
- Macroscopically polarized interfaces [28]
- Nano science in general [29].

This is why, long-ranged interactions need to be included in the calculation of MLFF, and in this work we focus on ionic systems.
So far, there are different methods to incorporate electrostatic interactions in models in literature, where the energies and sometimes forces of a bulk material or of single molecules are predicted. Some of these are:

- Calculating an Ewald-like energy and including it into the existing model [6, 30]
- Machine learn partial charges and atomic multipoles [7, 31–35]
- Using a non-local representation of the system and remapping it onto locally defined descriptors [36].

We propose an alternative approach that fits well into the already existing `VASP` code and into `polipy4vasp`. This has the advantage that the whole kernel based ML scheme as described in [20] remains unchanged. The only difference is that the descriptors and radial basis functions are calculated in reciprocal space without a cutoff radius. This new set of descriptors uses all atoms to indirectly perform a summation over all periodic images, and is thus able to capture long-range interactions.

Radial Basis Functions

The mapping from a local environment onto descriptors is ubiquitous in MLFF but depends on the choice of the angular and radial basis functions [6, 37].
`VASP` uses spherical Bessel functions as radial basis functions [18, 38]. They work well in combination with a cutoff radius, a cutoff function and spherical harmonics as angular basis functions. In reciprocal space, it would be better to find a set of basis functions that satisfies the following properties:

- Does not depend on a cutoff radius and a cutoff function

- Can be normalized analytically on the interval $[0, \infty)$
- Is able to capture the slowly decaying behavior of long range interactions.

To fulfill those properties, in this work Slater type orbitals (STOs) are introduced as radial basis functions.

Before we discuss more details on the calculation of the descriptors and on the radial basis functions in chapter 2, we give a short overview on the ML scheme.

ML Scheme of the Kernel Method

Now we introduce the ML scheme used in this thesis. The general workflow is shown in Fig. 1.1.

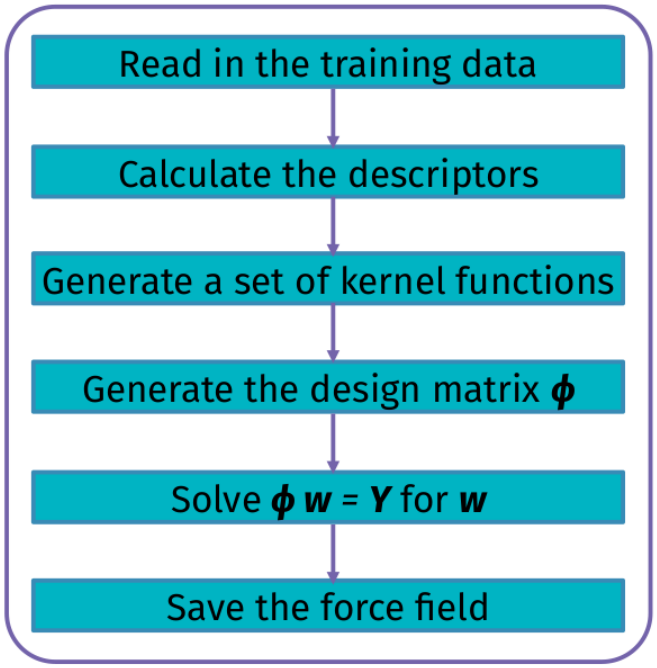

Figure 1.1.: ML scheme of the kernel method used during training. The vector $\mathbf{Y}$ contains energies and forces of the training structures.

As a first step, all necessary data is read in. These are the atomic positions, information about the unit cell and for training, the energies and forces of all configurations which are stored in a vector $\mathbf{Y}$.
Then, structural features based on the atomic density are mapped to a set of descriptors.

This is the important part this thesis focuses on, where either the short- or long-ranged descriptors are calculated using the respective basis functions.
Now, a set of descriptors of representative configurations, called kernel basis functions or Local Reference Configurations (LRC), is selected and the similarity to all descriptors is measured by applying the kernel. The results form the design matrix $\mathbf{\Phi}$.
In the next step, the linear system of equations $\mathbf{\Phi w} = \mathbf{Y}$ is solved, and a set of weights $\mathbf{w}$ is obtained. These weights can be used to predict the energy and force vector $\mathbf{Y}$ of the unseen data by calculating a new matrix $\mathbf{\Phi}$ (similar to the design matrix) for the new data and multiplying it with the learned weights.

2. Theory of Machine Learned Force Fields

In this chapter Machine learned force fields (MLFF) are explained. The goal is to predict properties describing a material, in this case the total potential energy and forces acting on the atoms, up to the accuracy that can be obtained with first principle (FP) calculations. This can be done by describing the environment around each atom with a so called expansion coefficient. These coefficients for all the atoms are then compared to a set of kernel-functions, which we also call Local Reference Configurations (LRC), to predict the properties of interest from the known values of the reference structures.

2.1. The Kernel-Based Method

This section gives an overview about kernel-based methods [39–41] and particularly describes the one, that is used for this project. It is a ML method originally described in [18, 42]. After calculating the descriptors which resemble the environment around all central atoms (for details on how the descriptors are calculated, see the following sections), a kernel is computed. We use a polynomial kernel of order ζ [20]

$$K(\mathbf{X}_i, \mathbf{X}_b) = \left(\hat{\mathbf{X}}_i \cdot \hat{\mathbf{X}}_b\right)^{\zeta} \tag{2.1}$$

that measures the similarity between two normalized descriptors $\hat{\mathbf{X}}_i$ and $\hat{\mathbf{X}}_b$ where the hat indicates normalization. The descriptor $\hat{\mathbf{X}}_i$ around a current central atom i is compared to a reference descriptor $\hat{\mathbf{X}}_b$ of the basis set that is indexed with b.
In general, any kernel can be used such as Gaussian kernels or linear kernels, and normalizing the descriptors is not necessary.
These function evaluations and their negative derivatives with respect to the coordinates r_k^α of each atom k, to account for the forces, are then combined to form a so-called design

matrix

$$\mathbf{\Phi}^I = \begin{bmatrix} \sum_i \frac{1}{N_A^{s=1}} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & \sum_i \frac{1}{N_A^{s=1}} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_1^1} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_1^1} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_1^2} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_1^2} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_1^3} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_1^3} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_2^1} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_2^1} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_2^2} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_2^2} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_2^3} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_2^3} K(\mathbf{X}_i^{s=1}, \mathbf{X}_{b=2}^I) & \cdots \\ \vdots & \vdots & \ddots \\ \sum_i \frac{1}{N_A^{s=2}} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & \sum_i \frac{1}{N_A^{s=2}} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_1^1} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_1^1} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_1^2} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_1^2} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_1^3} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_1^3} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_2^1} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_2^1} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_2^2} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_2^2} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ -\sum_i \nabla_{r_2^3} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=1}^I) & -\sum_i \nabla_{r_2^3} K(\mathbf{X}_i^{s=2}, \mathbf{X}_{b=2}^I) & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}. \tag{2.2}$$

The sum over index i is only taken over atoms of the same atom type I. The index s stands for the training structures/configurations of the data set, b enumerates all kernel basis functions and N_A is the number of atoms in structure s. The operator $\nabla_{r_k^\alpha} K(\mathbf{X}_i, \mathbf{X}_b)$ is the derivative of the kernel with respect to the atomic position of atom k in direction α of the descriptor $\mathbf{X}_i$. The superscript I indicates that kernel is only evaluated for kernel basis functions with the same atom type as atom i. This means that the design matrix holds information about the similarity of the training structures and the LRC.

To determine the fitting parameters w_b during training, the design matrix is used to get an even larger design matrix

$$\mathbf{\Phi} = \left[\mathbf{\Phi}^1, \mathbf{\Phi}^2, \ldots, \mathbf{\Phi}^{N_{\text{type}}}\right] \tag{2.3}$$

taking into account all design matrices for each atom type. This matrix is then used to solve the linear system of equations

$$\mathbf{\Phi w} \stackrel{!}{=} \mathbf{Y}. \tag{2.4}$$

Here $\mathbf{w}$ is the vector containing all fitting parameters w_b, and $\mathbf{Y}$ is a super vector with all the energies and forces obtained from FP calculations.
With the creation of the design matrix and using a kernel, the very high dimensional fitting problem can be reduced to a linear problem that can be solved easily. The reason for this is the so called kernel trick [43]. This transforms the data into a very high

dimensional space where it is linearly separable or distributed such that the relation between the data points can be obtained by a linear regression. Here this is equivalent to solving the linear system of equations given in Eq. (2.4).
In practice, the entries of the vector $\mathbf{w}$ are least squares solution of Eq. (2.4).

For predicting the energies and forces of some unseen data, a new matrix $\mathbf{\Phi}$ is calculated for each structure. This matrix consists only of the first blocks of Eq. (2.2)

$$\mathbf{\Phi}^I = \begin{bmatrix} \sum_i \frac{1}{N_A} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & \sum_i \frac{1}{N_A} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \\ -\sum_i \nabla_{r^1_1} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & -\sum_i \nabla_{r^1_1} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \\ -\sum_i \nabla_{r^2_1} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & -\sum_i \nabla_{r^2_1} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \\ -\sum_i \nabla_{r^3_1} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & -\sum_i \nabla_{r^3_1} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \\ -\sum_i \nabla_{r^1_2} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & -\sum_i \nabla_{r^1_2} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \\ -\sum_i \nabla_{r^2_2} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & -\sum_i \nabla_{r^2_2} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \\ -\sum_i \nabla_{r^3_2} K(\mathbf{X}_i, \mathbf{X}^I_{b=1}) & -\sum_i \nabla_{r^3_2} K(\mathbf{X}_i, \mathbf{X}^I_{b=2}) & \cdots \end{bmatrix} \tag{2.5}$$

since we only predict energies and forces for one unseen structure at the time. These blocks are then combine to the final matrix $\mathbf{\Phi}$ according to Eq. (2.3). This matrix is then multiplied by the weight vector $\mathbf{w}$ to obtain a super vector $\mathbf{Y}$ with the predicted values.

2.2. Descriptors in Real Space

In this section, it is explained how the short-ranged descriptors $\mathbf{X}$ used in `VASP` and `polipy4vasp` are usually calculated.
The descriptors used here are similar to the Smooth Overlap of Atomic Position (SOAP) [44] and Gaussian Approximation Potential (GAP) [6] descriptors.
We suppose that in a system with N_A atoms, the potential energy U can be written as the sum

$$U = \sum_{i=1}^{N_A} U_i \tag{2.6}$$

of local energy contributions U_i. In practice, these local energies are never calculated during the FP calculations. They are determined by the local environment of atom i. This environment is described by a functional of the atom density $F[\rho_i(\mathbf{r})]$

$$\rho_i(\mathbf{r}) = \sum_{j=1}^{N_A} \tilde{\rho}_{ij}(\mathbf{r}) \tag{2.7}$$

where $\mathbf{r}$ is a position in real space and

$$\tilde{\rho}_{ij}(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_{ij}) \tag{2.8}$$

is the likelihood to find atom j at position $\mathbf{r}$ relative to atom i [20] with the distance vector $\mathbf{r}_{ij}$ between atoms i and j.

In Eq.(2.8), the expression is usually multiplied by a cutoff function $f_{\text{cut}}(r)$ that truncates the atom density smoothly at a cutoff radius R_{cut} and removes all information outside of the cutoff sphere describing only local contributions to the atom density. In this way, the important local features around the central atom i, that strongly influence the local potential energy U_i, can be captured. In the present work, the cutoff function

$$f_{\text{cut}}(r) = \begin{cases} \frac{1}{2}\left(\cos\left(\pi \frac{r}{R_{\text{cut}}}\right) + 1\right), & \text{if } r \leq R_{\text{cut}} \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

of Behler and Parinello [5] is used, where r is the distance between atoms.
As in SOAP, the point particle represented by the Dirac-delta function is approximated by a noramized Gaussian

$$\delta(r) \approx \frac{1}{\sqrt{2\pi\sigma_{\text{atom}}^2}} \exp\left(-\frac{r^2}{2\sigma_{\text{atom}}^2}\right) \tag{2.10}$$

with the broadening parameter σ_{atom} of the atom. When combining all of this, the expression for the likelihood $\tilde{\rho}_{ij}(\mathbf{r})$ becomes

$$\tilde{\rho}_{ij}(\mathbf{r}) = \frac{1}{\sigma_{\text{atom}}\sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{r}^2\|}{2\sigma_{\text{atom}}^2}\right) f_{\text{cut}}(r_{ij}). \tag{2.11}$$

Since this expression is not rotationally invariant it is not directly used. Instead rotationally and translationally invariant descriptors depending on the likelihood are used as intermediate functions. Here we distinguish between so called two-body (radial) and three-body (angular) descriptors. These are visualized in Fig. 2.1 taken from [45].

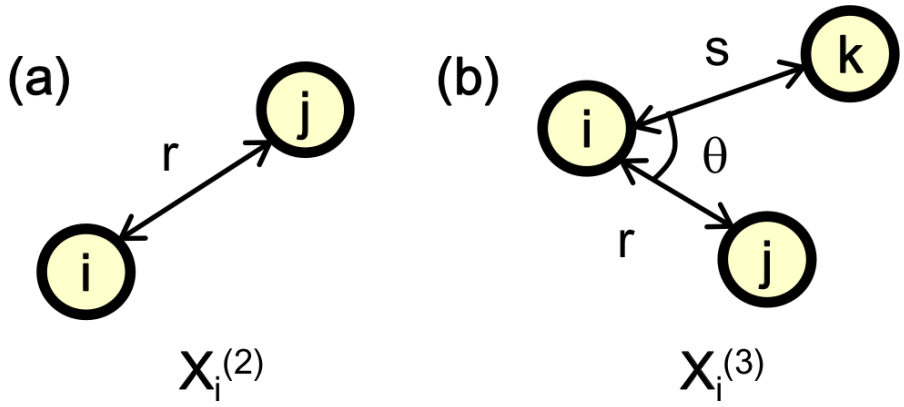

Figure 2.1.: Visualization of two- (a) and three-body (b) descriptors.

The final descriptor of the local environment around atom i is given by

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{X}_i^{(2)} \\ \mathbf{X}_i^{(3)} \end{pmatrix} \tag{2.12}$$

where $\mathbf{X}_i^{(2)}$ is the two- and $\mathbf{X}_i^{(3)}$ is the three-body descriptor. That the use of two and three-body descriptors yields very good results when predicting energies and forces has been shown in [22].

Two- and Three-Body Expansion Coefficients

The two body descriptor encodes simply the radial distribution function

$$\rho_i^{(2)}(\mathbf{r}) = \frac{1}{4\pi} \int \rho_i \left(r\hat{\mathbf{r}}\right) \mathrm{d}\hat{\mathbf{r}} \tag{2.13}$$

of the neighbouring atoms around central atom i, which is rotationally and translationally invariant.
The three-body descriptor describes the probability of finding one neighboring atom j of the central atom i at a certain distance r and a third atom k at distance s with the angle Θ between the connecting vectors $\mathbf{r}$ of atom i and j and $\mathbf{s}$ respectively. It is defined as

$$\rho_i^{(3)}(r,s,\theta) = \int\int \delta(\hat{\mathbf{r}}\hat{\mathbf{s}} - \cos(\theta))\rho_i \left(r\hat{\mathbf{r}}\right) \rho_i^* \left(s\hat{\mathbf{s}}\right) \mathrm{d}\hat{\mathbf{r}}\mathrm{d}\hat{\mathbf{s}} \tag{2.14}$$

and needs to be calculated in addition to the two-body descriptor. Using only the radial information is often insufficient because it does not describe the potential energy surface well enough.

Eqs. (2.13) and (2.14) are continuous expressions, and cannot be easily used in practice. This is why, the atomic probability density $\rho_i(\mathbf{r})$ is expanded in terms of radial and angular basis functions. For the angular basis functions spherical harmonics Y_{lm} are used, and for the radial basis functions normalized spherical Bessel functions χ_{nl} [18] are used here (see section 2.4.1 for more details on the radial basis functions). This yields

$$\rho_i(\mathbf{r}) = \sum_{l=0}^{L_{\max}} \sum_{m=-l}^{l} \sum_{n=1}^{N_{\max}} c_{nlm}^i \chi_{nl}(r) Y_{lm}(\hat{\mathbf{r}}) \tag{2.15}$$

for the probability density where c_{nlm}^i are so-called expansion coefficients, and the indices n, l and m stand for the main, angular and magnetic quantum number and the numbers of radial and angular basis functions. When those expansion coefficients are known, the probability density can be expressed as a linear combination of the product of the radial and angular basis functions, and the two and three body descriptors can be rewritten as

$$\rho_i^{(2)}(r) = \frac{1}{\sqrt{4\pi}} \sum_{n=1}^{N_{\max}} c_n^i \chi_{nl}(r) \tag{2.16}$$

and

$$\rho_i^{(3)}(r,s,\theta) = \sum_{l=0}^{L_{\max}} \sum_{n=1}^{N_{\max}} \sum_{n'=1}^{N_{\max}} \sqrt{\frac{2l+1}{2}} p_{nn'l}^i \chi_{nl}(r) \chi_{nl}(s) P_l(\cos(\theta)) \tag{2.17}$$

with P_l being Legendre polynomials of order l.

So far, all the equations are only given for one atom type, but they can be expanded to different species by introducing another index J for the atom type and calculating the probability distribution for each type separately.

The expansion coefficients are defined as

$$c_{nlm}^{iJ} = \sum_{\substack{j \neq i \\ j \in J}} \tilde{c}_{nlm}^{ij} \tag{2.18}$$

with index i for the central atom, J for the atom species of atom j. The expansion coefficients $\tilde{c}_{nlm}^{ij}$ describe $\tilde{\rho_{ij}}(\mathbf{r})$ as an expansion in radial and angular basis functions. These coefficients

$$\tilde{c}_{nlm}^{ij} = h_{nl}(r_{ij}) Y_{lm}(\hat{\mathbf{r}}_{ij}) \tag{2.19}$$

are the product of Spherical Harmonics $Y_{lm}(\hat{\mathbf{r}}_{ij})$ and the factor

$$h_{nl}(r) = \frac{4\pi}{\left(2\pi\sigma_{\text{atom}}^2\right)^{\frac{3}{2}}} f_{\text{cut}}(r) \int_0^\infty Q_{nl}(r') \exp\left(-\frac{r'^2 + r^2}{2\sigma_{\text{atom}}^2}\right) \iota_l\left(\frac{rr'}{\sigma_{\text{atom}}^2}\right) r'^2 \mathrm{d}r'. \tag{2.20}$$

This has been taken from [18] and contains the radial basis functions Q_{nl} and the modified spherical Bessel functions of the first kind ι_l coming from the plane wave expansion. The integration over r' is done numerically and instead of integrating over the whole space, the upper bound of integration is set to R_{cut} because we again assume that the contributions of atoms outside of the cutoff sphere are negligible.

From those expansion coefficients one can now calculate the two-

$$c_n^{iJ} = c_{n00}^{iJ} \tag{2.21}$$

and three-body expansion coefficients

$$p_{nn'l}^{iJJ'} = \sqrt{\frac{8\pi^2}{2l+1}} \sum_{m=-l}^{l} c_{nlm}^{iJ} c_{n'lm}^{iJ'} \tag{2.22}$$

and form the vectors

$$\mathbf{X}_i^{(2)} = \begin{pmatrix} c_1^{i1} \\ c_2^{i1} \\ \vdots \\ c_1^{i2} \\ c_2^{i2} \\ \vdots \end{pmatrix} \tag{2.23}$$

$$\mathbf{X}_i^{(3)} = \begin{pmatrix} p_{110}^{i11} \\ p_{111}^{i11} \\ \vdots \\ p_{120}^{i11} \\ p_{121}^{i11} \\ \vdots \\ p_{110}^{i12} \\ \vdots \\ p_{110}^{i22} \\ \vdots \end{pmatrix} \tag{2.24}$$

to obtain the two- and three-body descriptors. These are then again combined to the final descriptor according to Eq. (2.12).

In order to calculate the descriptor for the forces, the derivative with respect to the coordinates r_k^α of the respective atoms of the expansion coefficients (Eqs. (2.19) and (2.20)) has to be taken and analogously combined to obtain the derivatives of the descriptors.

2.3. Descriptors in Reciprocal Space

As we have already seen, the descriptors explained in section 2.2 only describe the local environment around a central atom within a finite cutoff sphere. For example, this can be seen in Eq. (2.20) where it suffices to take the integral from 0 to R_{cut} as the cutoff function removes all contributions outside of the cutoff sphere anyway. As long as there are no long-range interactions between the atoms of the system, this approach yields good results and saves computation time and resources. When long-range interactions are present, those descriptors are not able to describe them with a short cutoff radius. To account for long-range interactions, R_{cut} would need to be very large, such that a lot of atoms and even multiple periodic images of the same atom would lie in the cutoff sphere. With this the computation gets very expensive, since each atom has a large number of neighbor atoms and the calculation of the derivatives becomes very costly. In practice, it is impossible to capture long-ranged (electrostatic) interactions with the descriptors

presented above.
The idea behind this thesis is to calculate the descriptors in reciprocal space, considering all the atoms in one cell and therefore implicitly doing a summation over all the periodic images and incorporate long-ranged interactions. This approach does not require a cutoff radius, so the computational cost is limited by the number of atoms in the cell.

2.3.1. Descriptors without Finite Cutoff Radii

Here, the descriptors in reciprocal space are calculated to resemble the ones from real space, but avoid a cutoff and take into account potential long-ranged interactions.
In order to obtain the new descriptors, we first have to do a Fourier Transform (FT)

$$\mathcal{F}(f(\mathbf{x})) = \int_{-\infty}^{\infty} \exp(-i\mathbf{G}\mathbf{x}) f(\mathbf{x}) \mathrm{d}^3x \tag{2.25}$$

of the unbroadened atom density (2.7) to obtain the corresponding density in reciprocal space.

$$\rho(\mathbf{G}) = \sum_{j=1}^{N_A} \exp\left(i\mathbf{G}\mathbf{r}_j\right). \tag{2.26}$$

Here the vectors $\mathbf{G}$ are reciprocal lattice vectors.
In order to calculate this density the FT of both the unbroadened density and the Gaussian broadening (2.10)

$$\mathcal{F}\left(\exp\left(-\frac{\mathbf{r}^2}{2\sigma_{\text{atom}}^2}\right)\right) = \exp\left(-\frac{|\mathbf{G}|^2\sigma_{\text{atom}}^2}{2}\right) \tag{2.27}$$

are needed. The full calculations can be found in sections A.1.1 and A.1.2.
To obtain the expansion coefficients in reciprocal space, we start with the following equation similar to Eq. (2.15)

$$\rho_i^J(\mathbf{r}) = \sum_{nlm} c_{nlm}^{iJ} f_{nlm}(\mathbf{r}), \tag{2.28}$$

where the density in real space is represented as a linear combination of the basis functions in real space $f_{nlm}(\mathbf{r}) = Q_{nl}(|\mathbf{r}|)Y_{lm}(\hat{\mathbf{r}})$. These basis functions are the product of the radial $Q_{nl}(|\mathbf{r}|)$ and angular $Y_{lm}(\hat{\mathbf{r}})$ basis functions.
To obtain the expansion coefficients in reciprocal space, we need the FT of the radial basis functions and the broadened density. The FT of the density and the broadening have already been shown in Eqs. (2.26) and (2.27). As the FT of the basis functions depends on the choice of the radial basis functions, the exact form of the expression in reciprocal space $f_{nlm}(\mathbf{G})$ will be given in section 2.4. This leads us to the following relation in reciprocal space

$$\rho_i^J(\mathbf{G}) = \sum_{nlm} c_{nlm}^{iJ} f_{nlm}(\mathbf{G}). \tag{2.29}$$

From this, the expansion coefficients can be derived by multiplying both sides with the complex conjugate of $f_{nlm}(\mathbf{G})$ and integrating over $\mathbb{R}^3$. Since the basis functions are orthonormal to each other we arrive at the expansion coefficients

$$c_{nlm}^{jJ} = \sum_{\mathbf{G}} f_{nlm}(\mathbf{G}) \exp(-i\mathbf{G}\mathbf{r}_j)\rho_J(\mathbf{G}). \tag{2.30}$$

In practice, the expansion coefficients are stored into an array containing the product of the radial and angular basis functions for all values of the quantum numbers n, l and m. In contrast to the method presented in section 2.2, here the expansion coefficients are calculated directly without the intermediate calculation of the $\tilde{c}_{nlm}^{ij}$.

Again the derivative with respect to the atom positions r_k^α

$$\frac{\mathrm{d}c_{nlm}^{jJ}}{\mathrm{d}r_k^\alpha} = \left(i \sum_{\mathbf{G}} f_{nlm}(\mathbf{G})\mathbf{G} \exp(-i\mathbf{G}\mathbf{r}_j) \left(\delta_{Jk} \exp(i\mathbf{G}\mathbf{r}_k) \exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2} \right) - \delta_{jk}\rho_J(\mathbf{G}) \right) \right)_\alpha \tag{2.31}$$

is needed for the first derivative and force predictions. Here δ_{Jk} means

$$\delta_{Jk} = \begin{cases} 1, & \text{if atom } k \text{ is of type } J \\ 0, & \text{otherwise.} \end{cases} \tag{2.32}$$

The extended calculation of this derivation can be found in appendix A.2.1.

2.3.2. Implementation

Algorithm 1 shows how the descriptors and the derivatives are calculated efficiently in `Python`.

In Alg. 1, the input parameters are the settings, containing all parameters that can be set by the user such as $N_{\max}$, p or σ_{atom}, the lattice vectors in real space and the positions of the atoms.
Then the function `prep` is defined which calculates the reciprocal grid points $\mathbf{G}$, the normalized version $\hat{\mathbf{G}}$, the atomic density ρ as function of the reciprocal grid points and the basis functions f_{nlm}. This function is needed because the calculation of the returned variables requires some intermediate results that should not be saved longer then necessary, and leaving the function removes them from memory.
In line 12, the exponential in Eq. (2.30) $\exp(-i\mathbf{G}\mathbf{r})$ is calculated for all positions and with it the expansion coefficient c is calculated according to Eq. (2.30). For this, `np.einsum` [46] is used, and the input arrays to multiply and sum over are split up into the basis functions depending on n, l, m, and $\mathbf{G}$, and $\rho \cdot \exp$ depending on $\mathbf{G}$ and the number of atom types to save memory and not store an intermediate result depending on all indices. The same will be done later when calculating the derivatives. There the efficient memory handling becomes even more important because the arrays are much larger.

Algorithm 1: Calculation of the expansion coefficients and their derivatives in reciprocal space. The input parameters are user-settable hyperparameters stored in the class `settings`, information about the `lattice` vectors and atomic `positions`, a list of atom `types` and the maximal number of atom species `maxtypes`.

```
settings, lattice, positions, types, maxtypes
function prep(settings, lattice, positions)
    [mx, my, mz] = settings.ReciprocalPoints
    freq1, freq2, freq3 = get_rec_lattice_points(mx, my, mz)
    G, normG = get_wave_vectors(lattice, freq1, freq2, freq3)
    normG = np.where(abs(normG) < 1e-10, 1e-10, normG)
    normalized_G = np.where(normG == 0, G, G / normG)
    fnlm = calculate_basis_functions(settings, normalized_G, normG)
    rho = calculate_atom_density(G, normG, positions, maxtypes, types,
     settings.SigmaAtom)
    return fnlm, rho, G, normG
fnlm, rho, G, normG = prep(settings, lattice, positions)
exp = np.exp(-1j * G @ positions.T)
c = np.einsum('Mln,MJi->nlJi', fnlm, rho * exp, optimize = True)
rho_term[np.eye(len(positions), dtype= bool)] = rho
k_term = np.exp(1j * G @ positions.T) * np.exp(-normG**2 *
 settings.SigmaAtom**2 / 2)
// initialize dc as zeros
exp = G * exp
for J in range(maxtypes) do
    mask_k = types == J
    type_term = np.where(mask_k, k_term, np.zeros((len(positions),
     rho_term.shape[2])))
    dc_buf = type_term - rho_term
    dc_buf = dc_buf * exp
    helper_dc = np.real(1j * np.einsum('Mln,Mija->nlija', fnlm,
     dc_buf, optimize = True))
    dc = dc + helper_dc
    self_dc[J] = helper_dc[mask_diag]
end
// remove self interaction part from dc
_, v = get_rlattice_and_v(lattice)
c_self = np.sum(flmn * np.exp(- normG**2 * settings.SigmaAtom**2 /
 2), axis = 0)[0]
c[:, 0, :, :]  -= c_self
return np.real(c) / v, dc / v, self_dc / v
```

Next, the derivatives need to be calculated. First, the array called `rho_term` $= \delta_{jk}\rho_J(\mathbf{G})$ and then `k_term` $= \exp(i\mathbf{G}\mathbf{r}_k)\exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right)$ are calculated. They are then used in the for-loop over all atom types in line 19 to calculate `type_term` $= \delta_{Jk}\exp(i\mathbf{G}\mathbf{r}_k)\exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right)$. These two arrays `type_term` and `rho_term` are then subtracted to get the expression in the inner brackets of Eq. (2.31). The result is stored as the variable `dc_buf` and is then multiplied by `exp` that has previously been multiplied by $\mathbf{G}$ in line 16.
This variable `dc_buf` is used to calculate the derivative of the expansion coefficients `dc` for all types. Again, `np.einsum` [46] is used and the input arrays have been prepared like this to optimize the memory requirements. An example of what happens if this optimization is not done and the arrays are simply multiplied with each other and then summed over $\mathbf{G}$ is explained in section A.2.1.
The part where i is equal to j is removed from dc because this part is atom type-dependent and stored in the additional array called `self_dc`.
In the end the self interaction `c_self`

$$\texttt{c\_self} = \sum_{\mathbf{G}} f_{nlm}(\mathbf{G}) \cdot \exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right) \tag{2.33}$$

is subtracted from the expansion coefficients. This self interaction is calculated as a simplified version of Eq. (2.30) where the density does not depend on all atoms but just on the current central atom j. The only case where the self interaction is non-zero is for $l = 0$ because the radial basis functions are equal to one at $r = 0$. For higher l the radial basis functions are zero at $r = 0$.
Then the expansion coefficients and their derivatives are divided by the volume of the reciprocal unit cell and returned to calculate the descriptors.

2.4. Radial Basis Functions

In this section, different choices of radial basis functions are introduced. Starting out from the definition of the atomic density

$$\rho_i^J(\mathbf{r}) = \sum_{l=1}^{L_{\max}} \sum_{m=-l}^{l} \sum_{n=1}^{N_{\max}} c_{nlm}^{iJ} Q_{nl}(r) Y_{lm}(\hat{\mathbf{r}}), \tag{2.34}$$

the general representation $Q_{nl}(r)$ for the radial basis functions is introduced. The functions $Q_{nl}(r)$ vary for different values of n and can be also different for different l.

For the calculations in reciprocal space the FT of the basis function is needed. The basis functions

$$f_{nlm}(\mathbf{r}) = a_{nlm} Q_{nl}(r) Y_{lm}(\hat{\mathbf{r}}) \tag{2.35}$$

are products of spherical harmonics Y_{lm}, radial basis functions Q_{nl} and a normalization factor a_{nml} for the basis functions. For the calculation of the FT of this expression some

properties are needed. The first one is the plane wave expansion

$$\exp(i\mathbf{kr}) = 4\pi \sum_l \sum_m i^l j_l(|\mathbf{kr}|) Y_{lm}(\hat{\mathbf{k}}) Y^*_{lm}(\hat{\mathbf{r}}), \tag{2.36}$$

where an exponential function is factorized into the spherical harmonics and spherical Bessel functions $j_l(x)$. This is again a factorization into a radial (spherical Bessel functions) and an angular (spherical harmonics) part.
The second property is the orthogonality of the spherical harmonics

$$\int_0^\pi \int_0^{2\pi} Y_{lm}(\hat{\mathbf{r}}) Y_{l'm'}(\hat{\mathbf{r}}) \sin(\phi) \mathrm{d}\phi \mathrm{d}\theta = \delta_{ll'}\delta_{mm'}. \tag{2.37}$$

With this the FT of Eq. (2.35) can be calculated as follows

$$\begin{aligned} f_{nlm}(\mathbf{G}) &= a \int_{\mathbb{R}^3} \exp(-i\mathbf{Gr}) Y_{lm}(\hat{\mathbf{r}}) Q_{nl}(r) \mathrm{d}\mathbf{r} \\ &= a \int_{\mathbb{R}^3} 4\pi \sum_{l'} \sum_{m'} i^l j_l(|\mathbf{G}|r) Y_{l'm'}(-\hat{\mathbf{G}}) Y^*_{l'm'}(\hat{\mathbf{r}}) Y_{lm}(\hat{\mathbf{r}}) Q_{nl}(r) \mathrm{d}\mathbf{r} \\ &= 4\pi a \sum_{l'} \sum_{m'} i^l Y_{l'm'}(-\hat{\mathbf{G}}) \int_0^\infty j_l(|\mathbf{G}|r) Q_{nl}(r) r^2 \mathrm{d}r \\ &\qquad \int_0^\pi \int_0^{2\pi} Y^*_{l'm'}(\hat{\mathbf{r}}) Y_{l'm'}(\hat{\mathbf{r}}) \sin(\phi) \mathrm{d}\phi \mathrm{d}\theta \\ &= 4\pi a i^l Y_{lm}(-\hat{\mathbf{G}}) \int_0^\infty j_l(|\mathbf{G}|r) Q_{nl}(r) r^2 \mathrm{d}r. \end{aligned} \tag{2.38}$$

From the first to the second line of Eq. (2.38) the plane wave expansion given in Eq. (2.36) for $\exp(-i\mathbf{Gr})$ was used. Then from step two to three, the Cartesian coordinate system was switched to spherical coordinates. In the last step, the orthogonality of the spherical harmonics given in in Eq. (2.37) was used to perform the integrals over the angular part and the sums are resolved, since only the cases $l = l'$ and $m = m'$ provide a non-zero contribution.
Whether the remaining integral is solved numerically or analytically depends on the choice of the radial basis functions and what approximations are needed to solve this integral.

In practice, $\mathbf{G}$ is an array containing all the coordinates of the reciprocal lattice points.

2.4.1. Spherical Bessel Functions

One possible choice of radial basis functions are spherical Bessel functions. There are two reasons that suggest this choice. The first one being the property of the plane wave expansion, that can be seen in Eq. (2.36). Here, a plane wave can be divided into two parts, where the spherical Bessel functions represent the radial part and the spherical harmonics the angular one. As already shown above, this choice leads to very nice properties when integrating out to calculate the basis functions. This is also the reason

why the spherical Bessel functions are chosen as basis functions in real space.
The second reason is that the spherical Bessel functions are complete. This means that every function can be written as a linear combination of spherical Bessel functions, without error, if the basis of spherical Bessel functions is sufficiently large. In practice, only a small number of radial basis functions $N_{\max}$ is needed to describe the radial part of the descriptors with a reasonably small error.
In addition, spherical Bessel functions are orthogonal to each other if we use a certain cutoff radius. This is shown, for example, in Eq. 20 of [47] and always is a useful property for a basis.

In Real Space

In real space spherical Bessel functions are the only implemented radial basis.
Here, not only the simple spherical Bessel functions $j_l(r)$ are used, but a normalized version that is also adapted to the cutoff radius. These functions are written as

$$Q_{nl}(r) = \begin{cases} j_l(q_{nl}r) & \text{if } r \leq R_{\text{cut}} \\ 0 & \text{else} \end{cases} \tag{2.39}$$

with the parameters

$$q_{nl} = \frac{\alpha_{nl}}{R_{\text{cut}}}, \tag{2.40}$$

where α_{nl} the n-th root of j_l and q_{nl} are chosen such that j_l is zero at R_{cut}. In practice, this root-finding is done numerically using the Brent algorithm [48].
This choice of parameters q_{nl} ensures that all radial basis functions and their derivatives are zero at the cutoff radius and they do not contribute beyond that distance.
The hat in Eq. (2.39) states that the functions are normalized, which means that a normalizing factor

$$a_{nl} = \left(\int_0^{R_{\text{cut}}} j_l(q_{nl}r) j_l(q_{nl}r) r^2 \mathrm{d}r \right)^{-\frac{1}{2}} \tag{2.41}$$

for each value of n and l is introduced. This integral can be either calculated numerically or analytically (see Eq.(2.45)).

These radial basis functions are then used for the calculation of the factor h_{nl} according to Eq. (2.20).
To avoid the repeated calculation of this factor for every pairwise nearest neighbour distance r_{ij}, the calculation is done once for a certain set of interpolation points and the values for all the intermediate points are obtained via cubic spline interpolation.

In Reciprocal Space

The remaining integral of Eq. (2.38) can be calculated analytically when using spherical Bessel functions as radial basis functions in reciprocal space. For this we will need the

following relation

$$\int_0^R j_l(kr)j_l(qr)r^2\mathrm{d}r = \frac{R^2}{k^2-q^2}\left(j_l(kR)qj_l'(qR) - kj_l'(kR)j_l(qR)\right) \tag{2.42}$$

where j_l' is the derivative of j_l.
Now we can use Eq. (2.42) and make the approximation that we only evaluate the final integral in Eq. (2.38) from 0 up to a certain cutoff radius R_{cut} and use the same factors q_{nl} as in real space (see Eq. (2.40)) to set the spherical Bessel functions to zero at R_{cut}. This leads to

$$\begin{aligned} f_{nlm}(\mathbf{G}) &= 4\pi a i^l Y_{lm}(-\hat{\mathbf{G}}) \int_0^{R_{\mathrm{cut}}} j_l(|\mathbf{G}|r)j_l(q_{nl}r)r^2\mathrm{d}r \\ &= 4\pi a i^l Y_{lm}(-\hat{\mathbf{G}}) \frac{R_{\mathrm{cut}}^2}{|\mathbf{G}|^2 - q_{nl}^2} j_l(|\mathbf{G}|R_{\mathrm{cut}})q_{nl}j_l'(q_{nl}R_{\mathrm{cut}}). \end{aligned} \tag{2.43}$$

For the case that $|\mathbf{G}|^2$ is close or equal to q_{nl}^2 we use L'Hôpital's rule [49] to derive the following relation

$$f_{nlm}(\mathbf{G}) = 4\pi a i^l Y_{lm}(-\hat{\mathbf{G}}) \frac{R_{\mathrm{cut}}^3}{2} j_l'(q_{nl}R_{\mathrm{cut}}). \tag{2.44}$$

This approximation of the integral is also used to determine the normalization factor

$$a_{nl} = \sqrt{\frac{2}{R_{\mathrm{cut}}^3 j_l'(q_{nl}R_{\mathrm{cut}})}}. \tag{2.45}$$

Here one has to keep in mind that during the construction of the basis functions again a cutoff radius is used and, therefore, this method is again finite ranged. Of course, the cutoff radius can be chosen larger than in real space, since only the normalization of the Spherical Bessel functions depends on it and not the number of nearest neighbour atoms. Nevertheless, there still is a cutoff radius that we initially wanted to avoid.

2.4.2. Arbitrary Radial Basis Functions

Here the final result from Eq. (2.38) is taken and the integral over r is solved numerically such that an arbitrary function Q_{nl} can be taken as radial basis function. In the program, it is important to allow for differently shaped arrays of scaling factors like q_{nl}, since those factors can be l-dependent, but do not have to.

2.4.3. Introducing a Cutoff Function

Starting out from the arbitrary radial basis functions one can introduce a cutoff function to make the descriptors in reciprocal space short ranged, as the ones in real space are. This can be helpful when examining the influence of long-ranged interactions, or compare the different types of descriptors.

To avoid making the atomic density dependent on each atom and keeping it as a function of only reciprocal grid points, no nearest neighbor calculations are introduced. The FT of the basis functions

$$f_{nlm}(\mathbf{G}) = 4\pi a i^l Y_{lm}(-\hat{\mathbf{G}}) \int_0^{R_{\text{cut}}} j_l(|\mathbf{G}|r) f_{\text{cut}}(r) Q_{nl}(r) r^2 \mathrm{d}r \tag{2.46}$$

is calculated by only taking the integral over the radius of the cutoff sphere and letting the basis functions go to zero at the upper bound of the integral by multiplying them with a cutoff function f_{cut}. This cuts off all interactions between atoms further away than a given cutoff radius R_{cut}. It is also the reason why it suffices to evaluate the integral up to R_{cut} and not infinity.
Here, the same cutoff function as in real space, which is given in Eq. (2.9), is used but the method of applying it is different. In real space it is applied after integrating over the smoothed density (compare Eq. (2.20)). In reciprocal space, however, it is applied to the radial basis functions before performing the FT. So here the cutoff function is still applied in real space and therefore cuts off the long-ranged interactions and not the short-ranged ones, as a cutoff-function in reciprocal space would do.

In theory, this way of introducing a cutoff function can be used for any type of radial basis function Q_{nl}. In practice, this would make the analytic solutions of the integral in Eq. (2.46) more complicated and calculating the integral numerically is computationally costly and/or inaccurate.

2.4.4. Slater-Like Functions

We want to describe long-ranged interactions, and this is easily possible using modified radial basis functions as described in this section.
We know that long-ranged interactions decay with $1/r$, and it might therefore be useful to have basis functions that are designed to fit this behavior with very few basis functions.

The first idea was to use exponential functions as basis functions, because of the following relation

$$\frac{1}{r} = \int_0^\infty \exp(-\lambda r) \mathrm{d}\lambda, \tag{2.47}$$

where $1/r$ can be represented as integral over all possible positive exponents λ. If this integral is now approximated by a finite sum, we obtain

$$\frac{1}{r} \approx \sum_{n=1}^{N_{\max}} w_n \exp(-\lambda_n r) \tag{2.48}$$

as an approximation for $1/r$. The weights and abscissas $\{w_n, \lambda_n\}_{n=1}^{N_{\max}}$ are chosen such that the sum in Eq. (2.48) approximates $1/r$ for any $r \in [1, \infty)$ as good as possible.

In practice, the basis described above is not complete and, therefore, is not able to represent the electrostatic energies. The reason for this is that the values of λ_n increase to fast as they are designed to match $1/r$ and no other functions.
The completeness of the basis set depends on the way $\{\lambda_n\}_{n=1}^{N_{\max}\to\infty}$ behaves. Employing the Weierstrass Theorem of completeness allows to deduct that $\lambda_n \sim 1/n$ yields a complete basis [50].

Motivation Behind Slater Type Orbitals

In this section, we motivate why Slater type orbitals (STOs) are a reasonable choice of radial basis functions. For this, we have to look at the electrostatics, as we want to describe long-ranged electrostatic interactions. The derivation below is taken from [51].
In an ionic material the charge density ρ induces an electrostatic potential Φ that is a solution to the Poisson equation

$$\nabla\Phi = -4\pi\rho. \tag{2.49}$$

It is of the form

$$\Phi(\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\mathrm{d}^3r' \tag{2.50}$$

where the potential at $\mathbf{r}$ is introduced by a point charge at $\mathbf{r}'$.
In the following we will need the expression of the Nabla operator

$$\mathbf{r}\cdot\nabla = r\partial_r \tag{2.51}$$

in spherical coordinates.
Now we want to express the right hand side of Eq. (2.50) in a completely factorized form in $\mathbf{r}$ and $\mathbf{r}'$, as it is then easier to calculate the integral. This can be done via Laplace's formula where $1/|\mathbf{r}-\mathbf{r}'|$ is developed into a Taylor series in $\mathbf{r}'$ around $\mathbf{r}'=0$. This is done in Eq. (2.52):

$$\begin{aligned}
\frac{1}{|\mathbf{r}-\mathbf{r}'|} &= \sum_{n=0}^{\infty}\frac{1}{n!}\left(\mathbf{r}'\cdot\nabla'\right)^n \frac{1}{|\mathbf{r}-\mathbf{r}'|}\bigg|_{\mathbf{r}'=0} \\
&= \sum_{n=0}^{\infty}\frac{1}{n!}\left(-\mathbf{r}'\cdot\nabla'\right)^n \frac{1}{r} \\
&= \frac{1}{r}\cdot\frac{1}{\sqrt{1+\frac{r'^2}{r^2}-2\frac{r'}{r}\cos(\gamma)}} \\
&= \sum_{l=0}^{\infty} P_l\left(\cos(\gamma)\right)\frac{r_<^l}{r_>^{l+1}}.
\end{aligned} \tag{2.52}$$

In the first line of the Taylor expansion in Eq. (2.52), the operator ∇' is the Nabla operator and the prime indicates that it only acts on $\mathbf{r}'$. Then we make use Eq. (2.51) and rewrite the equation in spherical coordinates and obtain the functions creating the Legendre polynomials P_l depending on the cosine of the angel γ between the vectors $\mathbf{r}$

and **r'**.
The expressions $r_<$ and $r_>$ are defined as

$$r_< = \max\{r, r'\}, \qquad r_> = \min\{r, r'\}. \tag{2.53}$$

Now we apply the addition theorem

$$P_l\left(\cos(\gamma)\right) = \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_{lm}(\hat{\mathbf{r}}) Y_{lm}^*(\hat{\mathbf{r}}') \tag{2.54}$$

for the Legendre polynomials and when we plug in the result for $1/|\mathbf{r} - \mathbf{r}'|$ into Eq. (2.50) we get the multipole expansion

$$\Phi(\mathbf{r}) = \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_{lm}(\hat{\mathbf{r}}) \int_0^{\infty} r'^2 \frac{r_<^l}{r_>^{l+1}} \int_0^{\pi} \int_0^{2\pi} \rho(\mathbf{r}') Y_{lm}^*(\hat{\mathbf{r}}') \mathrm{d}\phi' \mathrm{d}\theta'. \tag{2.55}$$

In the multipole expansion $l = 0$ describes the monopoles and their induced field that decays with $1/r$. The monopole term is the first order term of the Taylor series and therefore the dominating one. Higher terms like the dipole term for $l = 1$ and the quadrupole term for $l = 2$ are only the second and third order term of the series expansion and decay with $1/r^2$ and $1/r^3$, respectively. The decay of the higher order term is thus much faster than for the monopoles, so in practice it often suffices to use only those to describe electrostatic interactions. If the source terms in a material are monopoles, the description by monopoles is exact. Because of this, in the following we only give all equations for this simplified case.

The central idea is to represent the $1/r$ decays with radial basis functions. Here, the radial part of the STOs

$$Q_n(r) = \exp(-\zeta_n r) \tag{2.56}$$

is a quite intuitive choice because of the properties shown in Eqs. (2.47) and (2.48). But this basis is neither orthonormal nor complete.

It has to be noted that we use these basis functions to describe the charge density and not the electrostatic potential itself. So during the calculation of the descriptors we can keep all the routines and equations the same as described in 2.3.1. This approach is justified because the electrostatic potential is a function of the density (see Eq. (2.55)). More specifically, for monopole source terms the potential $\Phi(\mathbf{r})$ is a unique function of the radial average density around the central point **r**. Therefore, we must also be able to represent the same potential unambiguously with the density decomposed into basis functions.
Here, we could also argue that in reciprocal space the difference between calculating the charge density or the electrostatic potential is only a metric. This metric is the so called Coulomb kernel

$$C_J(|\mathbf{G}|) = \frac{4\pi Z_J}{G^2 V} \tag{2.57}$$

where Z_J is the nuclear charge number of an atom of type J, V is the volume of the reciprocal cell and G^2 is the squared length of the reciprocal grid point vectors $\mathbf{G}$.
Then the electrostatic potential

$$\Phi_J(\mathbf{G}) = -C_J(|\mathbf{G}|)\rho_J(\mathbf{G}) \tag{2.58}$$

is given as the negative product of the Coulomb kernel given in Eq. (2.57) and the atomic density in reciprocal space shown in Eq. (2.26).

Completeness and Normalization

In the previous section, we have seen that STOs are a good choice for the radial basis functions, but in general they do not form an orthonormal basis (ONB).
Here, we introduce the mathematical properties that are needed to construct an orthonormal and complete set of radial basis functions.
Because we want to find a set of radial basis functions, the space we want to represent is $L^2(\mathbb{R}_+)$, that is, all square integrable functions on the interval $\mathbb{R}_+ = [0, \infty)$.
To obtain a complete basis set of $\mathbb{R}_+$, the radial basis functions $Q_n(r)$ must fulfill the completeness relation [52]

$$\sum_{n,n'=1}^{\infty} Q_n(r)[S^{-1}]_{nn'}Q_{n'}(r') = \frac{\delta(r-r')}{r^2}. \tag{2.59}$$

for $r \in \mathbb{R}_+$.
Fulfilling this relation ensures that all functions on $\mathbb{R}_+$ can be represented as a linear combination of the basis functions $Q_n(r)$.
For an ONB $S_{nn'} = \delta_{nn'}$ holds and otherwise the overlap matrix $S_{nn'}$ is defined as

$$S_{nn'} = \int_0^{\infty} Q_n(r)Q_{n'}(r)r^2\mathrm{d}r. \tag{2.60}$$

To measure whether a set of functions forms a complete basis, the completeness profile $Y(\zeta)$ is used. It is obtained from the completeness relation given in Eq. (2.59) by multiplying with sample functions $g_\zeta(r)g_\zeta(r')$ and truncating the sum at $N_{\max}$

$$Y(\zeta) = \sum_{n,n'=1}^{N_{\max}} \langle g_\zeta|Q_n\rangle \, [S^{-1}]_{nn'} \, \langle Q_{n'}|g_\zeta\rangle \, . \tag{2.61}$$

The sample functions have to be normalized such that the right hand side of Eq. (2.61) is equal to 1. This yields

$$Y(\zeta) = 1, \qquad N_{\max} \to \infty, \quad \forall\zeta. \tag{2.62}$$

It has to be noted that all these relations are given in real space, but they also hold true in reciprocal space, since the FT is a unitary linear transformation on the L^2 space [53] and, therefore, preserves orthogonality and normalization.

Orthonormalized STOs

We try to use a set of orthonormalized STOs as radial basis functions, because they are not dependent on a cutoff radius as the spherical Bessel functions are and they are able to describe the $1/r$ behavior of the electrostatic interactions, as we have seen above in section 2.4.4.
Starting out from Eq. (2.56) we use those functions $\tilde{Q}_n(r)$ as a radial basis. The tilde states that they are neither orthogonal nor normalized.
To obtain the ONB

$$Q_n(r) = \sum_{n'=1}^{N_{\max}} S_{nn'}^{-1/2} \tilde{Q}_n(r) \tag{2.63}$$

an overlap matrix S is used. In this case according to Eq. (2.60), it is given as

$$\begin{aligned} S_{nn'} &= \int_0^\infty \tilde{Q}_n(r)\tilde{Q}_{n'}(r) r^2 \mathrm{d}r \\ &= \frac{1}{(\zeta_{nl} + \zeta_{n'l})^3} \cdot 2. \end{aligned} \tag{2.64}$$

In order to calculate $S^{-1/2}$ we use the property that S is symmetric and positive definite and perform a Cholesky decomposition [54] and rewrite it as $S = LL^T$ with the lower triangular matrix L. Then it holds $S^{-1/2} = (L^{-1})^T$ and multiplying this matrix with the non-orthogonal and non-normalized set of basis functions corresponds to applying the modified Gram-Schmidt process [55, 56] to the basis functions to obtain an ONB.
This basis can now be plugged into Eq. (2.38) to obtain

$$f_{n00}(\mathbf{G}) = 4\pi i^l Y_{00}(-\hat{\mathbf{G}}) \sum_{n'} S_{nn'}^{-1/2} \int_0^\infty j_0(|\mathbf{G}|r)\tilde{Q}_{n'}(r) r^2 \mathrm{d}r, \tag{2.65}$$

where we do not need the additional normalizing prefactor a as the normalization is already performed by applying S. The integral in Eq. (2.65) can be calculated analytically with `mathematica` and is given by

$$\int_0^\infty j_0(|\mathbf{G}|r)\tilde{Q}_{n'}(r) r^2 \mathrm{d}r = f \cdot \sqrt{\frac{1}{\zeta_n^3}}. \tag{2.66}$$

Here, $f = 4$ is a prefactor resulting from the FT.
In the approach, that will be called "Slater1" from now on, the exponents ζ_n are calculated such that

$$\begin{aligned} &\min_{w_n, \zeta_n > 0} \quad \max_{\zeta_n \in [\zeta_{\min}, \zeta_{\max}]} |1 - Y(\zeta)| \\ \approx &\min_{w_n, \zeta_n > 0} \quad \max_{\zeta_n \in [\zeta_{\min}, \zeta_{\max}]} \left| 1 - \sum_{n=1}^{N_{\max}} w_n \left| \left\langle e^{-\zeta r} \middle| e^{-\zeta_n r} \right\rangle \right|^2 \right| \\ = &\min_{w_n, \zeta_n > 0} \quad \max_{\zeta_n \in [\zeta_{\min}, \zeta_{\max}]} \left| 1 - \sum_{n=1}^{N_{\max}} w_n \int_0^\infty r^2 e^{-\zeta r} e^{-\zeta_n r} \mathrm{d}r \right| \end{aligned} \tag{2.67}$$

the maximal error of approaching the completeness profile is minimized. Here we want to approach 1 in order to fulfill the completeness relation given in Eq. (2.59) and have a completeness profile, that according to Eq. (2.61) is as close to 1 as possible for the given number of radial basis functions $N_{\max}$.
We assume that exponential functions or STOs for the case $l = 0$ are a good basis to describe the long-range electrostatic interactions. The exact exponents ζ_n are not known. Therefore, we have chosen $g_\zeta(r) = e^{-\zeta r}$ as sample function to represent all $\zeta_{\min} < \zeta < \zeta_{\max}$ as accurately as possible with our basis $Q_n(r) \sim e^{-\zeta_n r}$. The overlap matrix S also depends on ζ_n (see Eq. (2.64)). For this reason, it is impossible to solve the minimization problem of Eq. (2.67). Therefore, as an approximation, we assume that the basis is already orthonormal and do not consider the overlap when inserting the completeness profile of Eq. (2.61) into Eq. (2.67).
This approach yields a very compact set if $\zeta_{\min}$ and $\zeta_{\max}$ are chosen optimal, in the sense that the smallest possible error in Eq. (2.67) is achieved. Compact or concentrated in this case means that the peaks of the STOs are very thin and concentrated around the origin. Further from the origin, the functions rapidly decay to zero instead of approaching it over a longer interval.

The exact values for ζ_n are given in section A.3.

STOs with Scaled Exponents

As discussed in the section above, there are different methods for constructing the exponents ζ_n of the STOs. For the Slater1 method a optimization of $\zeta_{\min}$ and $\zeta_{\max}$ needs to be done. Also, the calculation of the ζ_n for increasing main quantum numbers in practice can be only done iteratively. This means that the calculation of the exponents can not be automated and is computationally costly.
So we are looking for an easier way to calculate a set of exponents ζ_n for an arbitrary large number of radial basis functions.
The straight forward way of doing this is starting out with most concentrated radial basis function for four radial basis functions from Slater1. These values are the values in the last column of Tab. A.1. Again, in practice, we only need the monopole term with one angular quantum number.
Now we assume $\zeta_1 = 10.604$ and calculate the values for higher n adopting the following prescription

$$\zeta_i = \frac{\zeta_1}{s^i}. \tag{2.68}$$

Here s is a scaling factor, that is determined empirically for different numbers of radial basis functions.

Fig. 2.2 shows the radial basis functions for only one angular basis function and a scaling constant of $s = 3$ (see section 5.1 for the choice of this constant). We can see that the first function with the highest exponent is the most concentrated one and decays rapidly to zero. Whereas the functions with lower exponents decay slower.

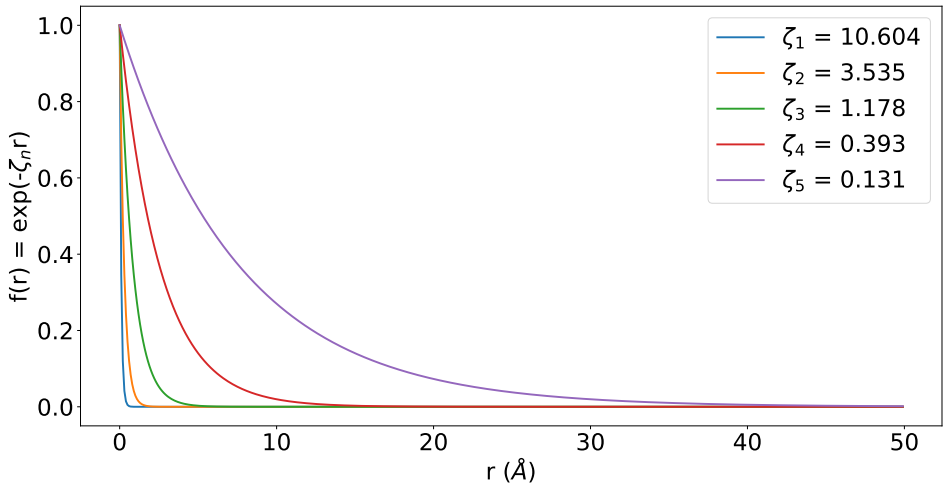

Figure 2.2.: Visualization of the first five radial basis functions $f(r) = \exp(-\zeta_n r)$ for Slater2 with a scaling constant of $s = 3$ between subsequent exponents. The first basis function with the highest exponent ζ_1 decays the fastest to zero and can therefore model more short-ranged interactions. The last one with ζ_5 decays slowly to zero and can model very long-ranged interactions up to more than 40 Å.

Here, we cannot choose the most concentrated exponent arbitrarily large, since we must still be able to represent the resulting basis function smoothly in reciprocal space with a reasonable number of Fourier grid points. This means that we need to be able to capture this very steep decay to zero with our Fourier grid.
Fig. 2.2 also indicates how adding another radial basis function would look like. Since we always start from the most concentrated function, for the reasons described above, the labels in the plot would change. An additional basis function has an even smaller scaling exponent and, therefore, decays even slower.
If we now suppose that the previous basis set was already almost complete and good enough to describe the electrostatic energies, then adding another basis function in theory should not change the results for the predicted energies and forces. This is because the regression scheme we apply in order to learn the weights in Eq. (2.4), should be able to learn that this additional basis function adds no further information and, therefore, set the corresponding weights to zero. We will later see that in practice this is not the case.

The exponents ζ_{nl} resulting from Eq. (2.68) are scaled logarithmically and are used for the radial basis, which is then orthogonalized with the overlap matrix S from Eq. (2.64) as described in the previous section.

This method will be called "Slater2" from here on.

2.5. Combining Short- and Long-ranged Descriptors

Now that we can calculate short-ranged descriptors in real space and long-ranged descriptors with different radial basis functions in reciprocal space, we need to combine the two to get one descriptor that can potentially represent both long- and short-ranged interactions.
When we apply the kernel to measure the similarity between a kernel basis function and the current descriptor, we want the result to show high similarity if and only if both the long-ranged and the short-ranged part of the descriptor are similar. Therefore we need to find a way of combining the descriptors that ensures this property. Hence, we set the final descriptor to

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{X}_i^{(2)} \\ \mathbf{X}_i^{(3)} \\ \mathbf{X}_i'^{(2)} \\ \mathbf{X}_i'^{(3)} \end{pmatrix} \tag{2.69}$$

where we simply append the long-ranged descriptor $\mathbf{X}'$ to the short-ranged one $\mathbf{X}$. Another advantage of this method is that we can choose different numbers of basis functions for the different descriptors. This would not be possible if the final descriptor would be, for example, the sum of the single descriptors.

In order to let the user choose whether he wants to combine the different descriptors or keep only one version, the input parameter `lr`, standing for long-ranged, is introduced in the `Python` code. This parameter can be set to three possible values 0, 1 and 2. A value of 0 means that no long-ranged descriptors are used and only the calculations in real space are performed. Setting `lr` $= 1$ corresponds to taking the long-ranged descriptors only, and the radial basis functions can be chosen by setting the input parameter `rad` accordingly. Using `lr` $= 2$ means that the descriptors are combined, as described above. Then the radial basis functions in reciprocal space can be set by using `rad`, and different values for $N_{\max}$ and $L_{\max}$ in real and reciprocal space can be used.

3. Comparison of the Expansion Coefficients

3.1. Cutoff Function

The figures 3.1-3.3 show the behavior of the expansion coefficients c as a function of the atom distance between two atoms and the influence of a cutoff function. They were created using the parameters given in Tab. 3.1.

Table 3.1.: Hyperparameters used to create figures 3.1-3.3. Here R_{cut} is the cutoff radius, N_{max} is the number of radial basis functions, and L_{max} is the angular quantum number determining the number of angular basis functions. The broadening parameter of the atoms is σ_{atom}, p is the number of reciprocal grid points in each direction, and the lattice constant is the side length of the cubic cell in which two trial atoms were placed.

Parameter	Value
R_{cut}	4 Å
N_{max}	3
L_{max}	0
σ_{atom}	0.4 Å
p	25
lattice constant	16 Å

Two atoms were placed in the cell at the same starting position $x = [0, 1, 1]$ and the second atom was moved away from the first atom in x-direction. Therefore, only the behavior of the expansion coefficients in this direction is plotted. For reasons of simplicity, we only inspect the $\mathrm{l} = 0$ case.
Figure 3.1 shows how the two curves behave when neither the reference calculations in real space nor the calculations in reciprocal space are done with a cutoff function. To switch off the cutoff function in real space the output of this cutoff function was always set to one and then Eqs. (2.18), (2.19) and (2.20) were used for the calculation of the expansion coefficients. In reciprocal space, the spherical Bessel functions were used as a radial basis to make the methods comparable. As the standard approach in reciprocal

space does not include a cutoff function, the expansion coefficients were calculated using Eq. (2.30).

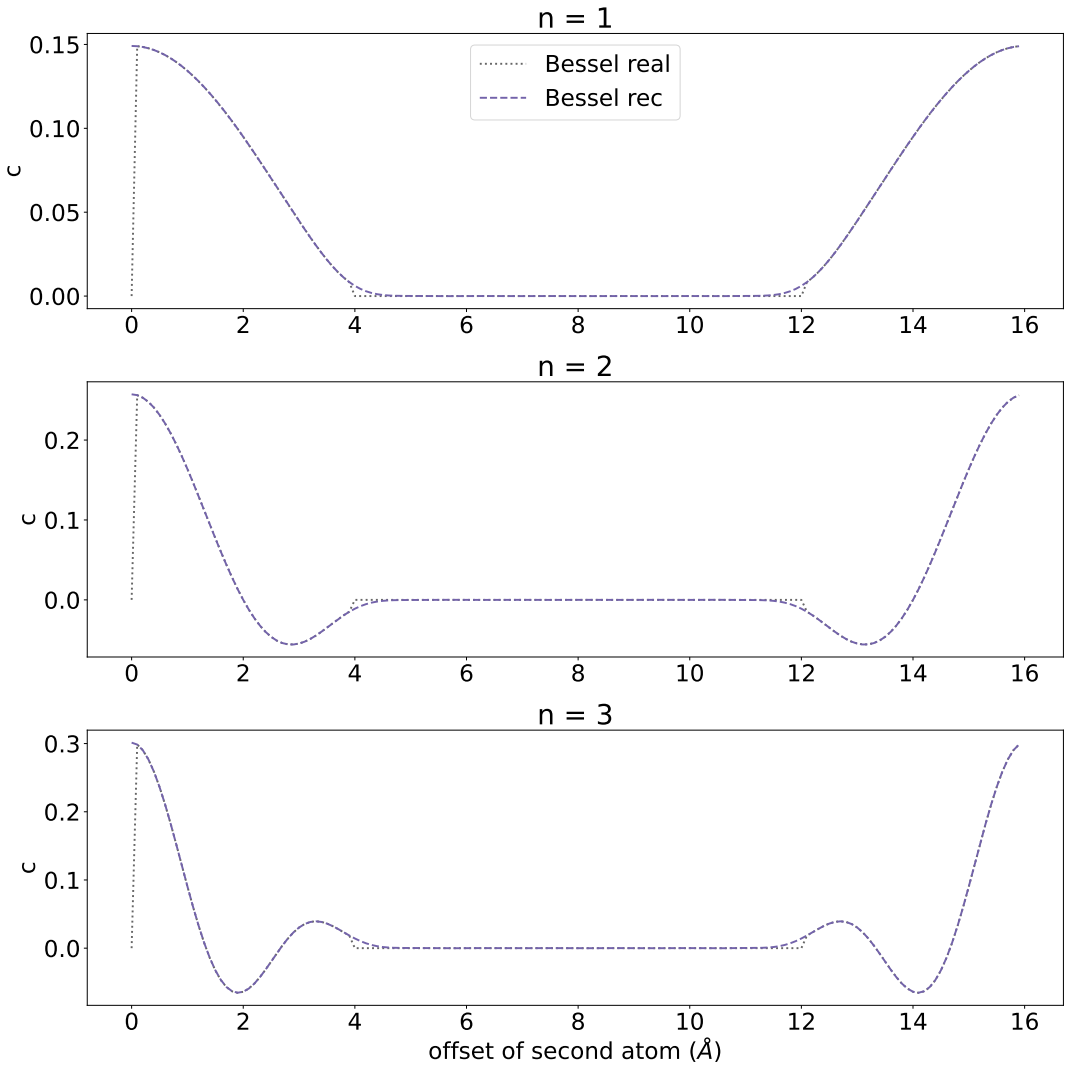

Figure 3.1.: Behavior of expansion coefficients c for without cutoff functions in both real and reciprocal space when spherical Bessel functions are used as radial basis. Two atoms were placed in a cubic cell with a side length of 16 Å, and one was moved away from the other. For the real space method the expansion coefficients abruptly jumps to zero at the cutoff radius $R_{\text{cut}} = 4$ Å.

The expansion coefficients behave as expected. The reference value in real space decreases (with some oscillations depending on n) for increasing distance of the atoms. Then it

abruptly drops to zero at the cutoff radius. This discontinuity occurs because of how the descriptors in real space are designed.
The reciprocal expansion coefficient is not zero at the cutoff radius because of the Gaussian broadening of the atoms. With this broadening, the atoms are of finite size not just points. If the position is outside of the cutoff sphere but still close enough to the central atom such that the exponential tail of the broadened atom is still seen within the sphere, the atom still contributes to the expansion coefficient. In reciprocal space we observe this behavior, but with the real space method all interaction is presently erroneously assumed to be zero as soon as the second atom moves further away than four Å.

The calculation of the descriptors in reciprocal space includes all atoms and, therefore, we see the contributions even though the atom is placed outside of the "cutoff sphere". If we look closely at this region, we see that the curve for the reciprocal expansion coefficient decays exponentially to zero outside of the cutoff sphere, as expected for the exponential broadening.
This discontinuity is smoothed out when introducing the cutoff function, as can be seen in Fig. 3.2, where the expansion coefficients in reciprocal space stay the same and in real space the cutoff function is used.

The cutoff function does not only smooth the behavior of the expansion coefficients in real space, but also sets them and their derivative to zero at the cutoff radius. Thus, in practice, the design error discussed above does not matter, as the expansion coefficients in real space are normally always calculated with the cutoff function, to give less weight to the contribution of distant atoms.

The expansion coefficient in reciprocal space is already smooth without a cutoff function, since it includes the contribution of Gaussian broadened atoms outside of the cutoff sphere. In contrast to the real space method, there are still oscillations shown for higher values of n because they are not suppressed by a cutoff function. If we want to suppress them and obtain a smooth convergence to the value of zero for distances larger than the cutoff radius and weight shorter ranged interactions more strongly than longer ones, we introduce a cutoff function as described in section 2.4.3, especially using Eq. (2.46) for calculating the basis functions. Again, spherical Bessel functions are used, to make the expansion coefficients comparable.

The result for both types of expansion coefficients with cutoff functions is shown in Fig. 3.3. Here the expansion coefficients are now zero for larger atom distances for both methods and the two curves are very similar.
Nevertheless, one has to keep in mind that the place of where the cutoff function is applied in the different methods is different, as discussed in section 2.4.3.

In practice, one would not use any cutoff function in reciprocal space, because it contradicts the main goal of the descriptors in reciprocal space. This goal is to avoid the

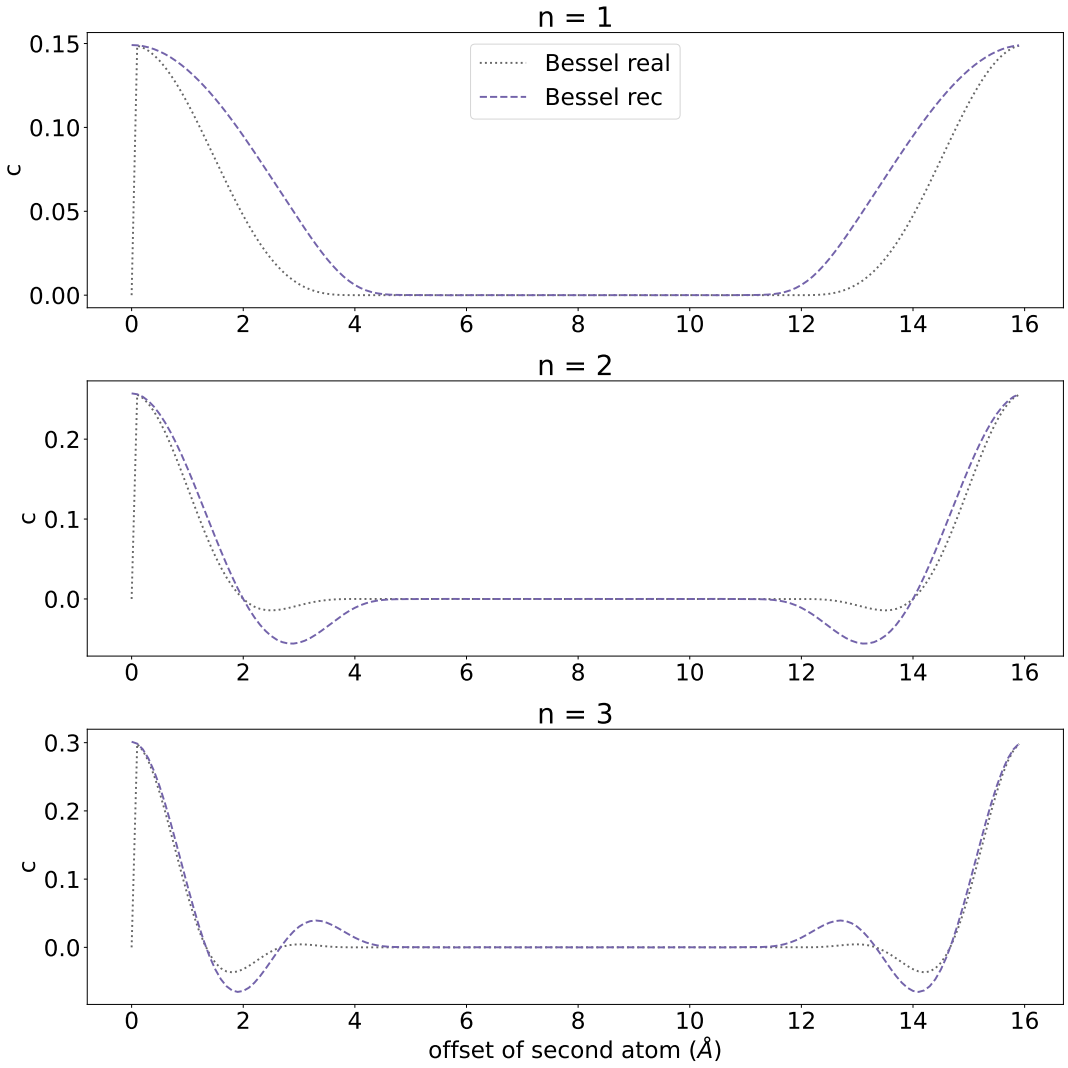

Figure 3.2.: Behavior of expansion coefficients c with a cutoff function in real space and no cutoff function in reciprocal space when spherical Bessel functions are used as radial basis. Two atoms were placed in a cubic cell with a side length of 16 Å, and one was moved away from the other. The real space expansion coefficients and their derivatives become zero at and beyond the cutoff radius $R_{\text{cut}} = 4$ Å.

cutoff radius and allow for long-ranged interactions[1]. Applying a cutoff function is more complicated in reciprocal space, as the integral in Eq. (2.46) is now more complicated

[1]Here one has to keep in mind that this is not possible, as the FT to reciprocal space for spherical Bessel functions as radial basis functions is done one a finite interval.

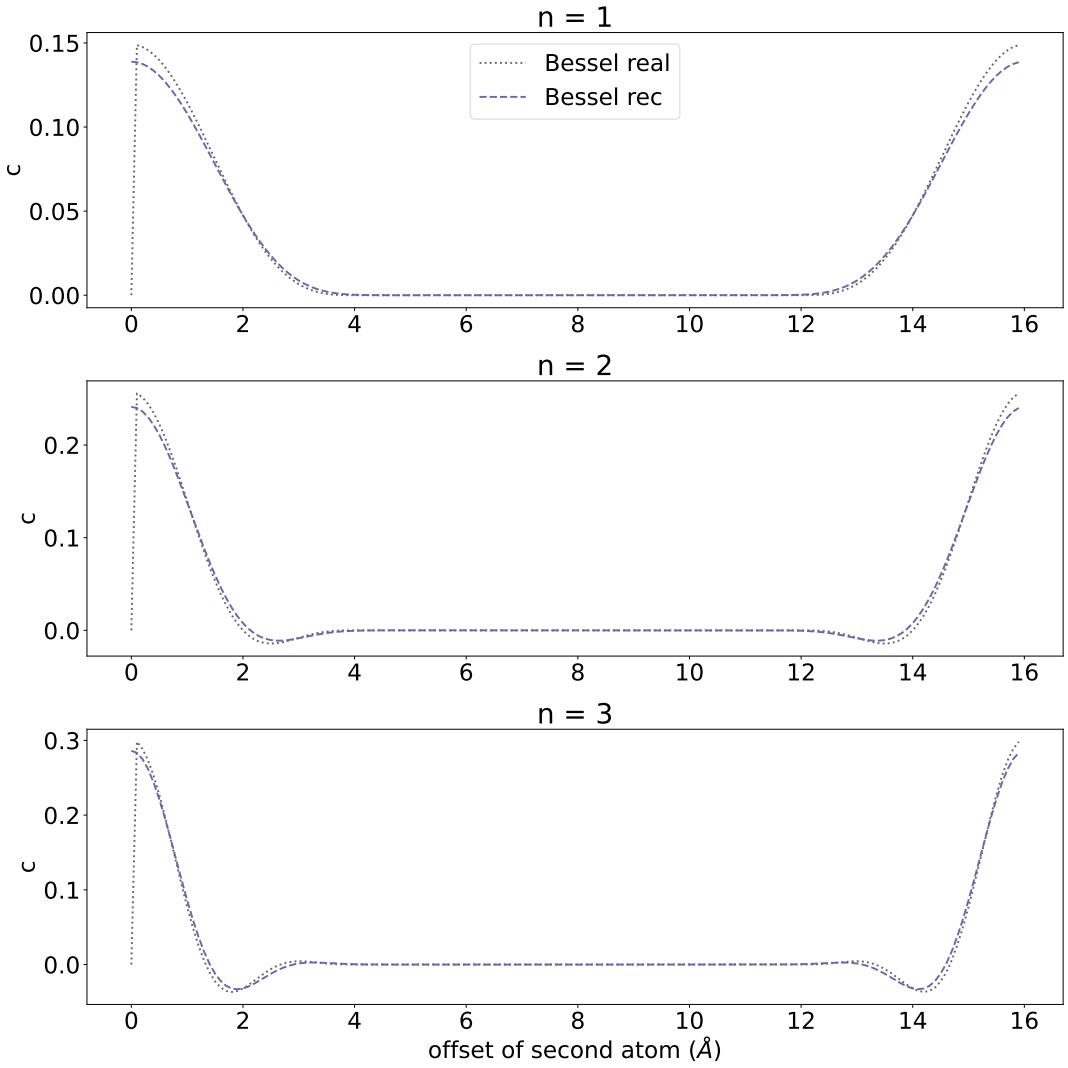

Figure 3.3.: Behavior of expansion coefficients c with cutoff functions both in real and reciprocal space when spherical Bessel functions are used as radial basis. Two atoms were placed in a cubic cell with a side length of 16 Å, and one was moved away from the other. Both types of expansion coefficients and their derivatives become zero at and beyond the cutoff radius $R_{\text{cut}} = 4$ Å.

and cannot be calculated analytically, forcing the use of numerical integration, which is not as precise and also more costly.
Not only is numerical integration costly, but the entire approach of using a cutoff function in reciprocal space is unnecessarily inefficient. When we want to have only short-ranged interactions and a cutoff radius, the reference approach in real space is more efficient and

less expensive. This method does not take into account all atoms but only the nearest neighbours. Therefore, descriptors and their derivatives are not calculated for atoms outside the cutoff sphere, but are made obsolete by the cutoff function.

Here, we see that the introduction of the cutoff function in reciprocal space makes the expansion coefficients more similar to the short-ranged real space method. At first glance, this might seem counter intuitive since a cutoff function in reciprocal space ideally would cutoff all interactions of atoms further away than the cutoff radius. In reciprocal space, this would mean cutting off the short ranged interactions and making the results differ from the short-ranged real space method.
That this is not the case is due to the fact that the cutoff function is introduced when transforming to reciprocal space (see Eq. (2.46)) and not afterwards. So the cutoff is applied before performing the integral of the FT so that the cutoff functions is effectively still applied in real space and the result is transformed to reciprocal space. Because of this, the results become more similar to the real space method and do not behave as one would intuitively expect.

3.2. Expansion Coefficients for Different Radial Basis Functions

Already in the previous section 3.1, we have looked at the behavior of the expansion coefficients, when we have two atoms in the unit cell and one atom moves away from the other. Now, we want to know how the Slater-like radial basis function influence that behavior and what we can learn from this. The parameters given in Tab. 3.2 were used for the following tests.
Looking at Fig. 3.2 we see four different curves. The grey one shows the expansion coefficients calculated with spherical Bessel functions in real space. The purple curve shows the ones with Bessel functions in reciprocal space and the light turquoise line shows the result for Slater1 and the dark turquoise for Slater2.
One has to note that we again only look at the $l = 0$ expansion coefficients since all higher ones are constant in this simple test cases.
The first thing we notice again is that both methods using spherical Bessel functions are short-ranged, as they go to zero at 7 Å for the real space method or directly afterwards for the reciprocal one. This is best seen in the first subplot for the first radial basis function and $n = 1$. The other curves show that the descriptors with Slater-like basis functions are long-ranged. We need a unit cell of 65 Å to see that at very far distances even these expansion coefficients converge towards zero. This behavior is especially pronounced for $n = 1$.

If we look closely at the Slater methods, we see that the expansion coefficients converge the slowest for $n = 1$ but very fast for higher n.
In general, the Slater curves behave differently from all the Bessel ones, as there are oscillations for small distances in the first functions which are not present for the Bessel

Table 3.2.: Hyperparameters used to create figure 3.4. Here R_{cut} is the cutoff radius, N_{max} is the number of radial basis functions, and L_{max} is the angular quantum number determining the number of angular basis functions. The broadening parameter of the atoms is σ_{atom}, p is the number of reciprocal grid points in each direction, and the lattice constant is the side length of the cubic cell in which two trial atoms were placed.

Parameter	Value
R_{cut}	7 Å
N_{max}	3
L_{max}	0
σ_{atom}	0.4 Å
p	100
lattice constant	65 Å

functions. For higher radial basis functions the Slater methods do not show these oscillations whereas the Bessel functions do. For which curve the oscillations stop and how large the peak for very small distances becomes, depends on the Slater method and therefore on the exponents ζ. This is due to the order of normalization of the basis functions and whether the normalization starts from $n = 4$ (Slater1 and Slater2) or from the basis function for $n = 1$ (Bessel real and Bessel rec).
Currently, our best hypothesis is that this behavior for small distances shows that we cannot properly describe short-range interactions with this set of radial basis functions: The oscillations for small distances make the descriptor very sensitive to small changes in atomic positions, since the slope and hence the derivative of the functions lead to larger changes in the expansion coefficients. This would mean that the descriptors for similar configurations can be very different and therefore the algorithm cannot properly distinguish between very similar configurations.

For small changes in atomic positions further away from the central atom the descriptor barley changes, as the slope of the expansion coefficients is very flat. This means that here the algorithm will be perfectly capable to determine the similarity between descriptors and therefore configurations.
In a structure with multiple atoms we have both, neighbors at shorter or longer distances so the similarity measure will always be a compromise between the long-range part, that can be described very well with Slater functions, and the short ranged part, that is problematic.

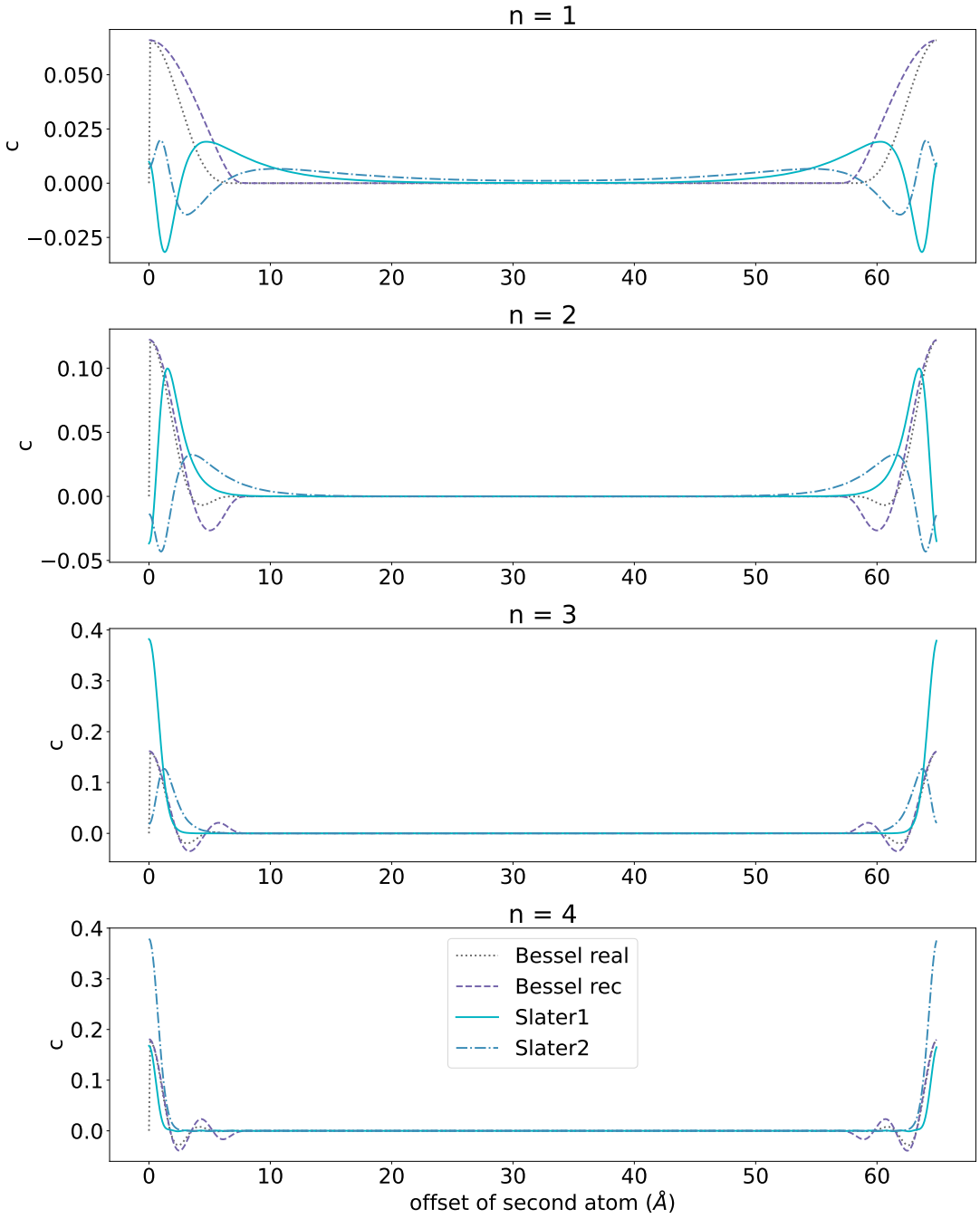

Figure 3.4.: Behavior of expansion coefficients c for different radial basis functions (Bessel functions in real and reciprocal space and STOs Slater1 and Slater2). Two atoms were placed in a cubic cell with a side length of 65 Å, and one was moved away from the other. The methods with Bessel functions decay to zero around the cutoff radius $R_{\text{cut}} = 7$ Å. The Slater methods are long-ranged and decay slower, at least for the first two basis functions ($n = 1$ and $n = 2$).

4. Test Materials

This chapter introduces different materials that were used to train and test the ML model.

4.1. Overview

The data for the material that should be trained or predicted is read in from a `VASP` file. This can be either a so called ML_AB(N) file or a POSCAR file. The ML_AB file contains the results of previous ab initio calculations [57] and not only the information about all the different configurations of the material, but also the corresponding energies, forces and the LRC. They are read in and used for training the model. When an existing MLFF is used for predicting energies and forces only the information describing the configurations, such as lattice vectors, atom positions and atom types are read in from POSCAR files. They usually contain the information needed to start an Molecular Dynamics (MD) run for this material [58].
Tab. 4.1 gives a brief overview on the most important properties of the different materials and the ML_AB files that were used as input.

Table 4.1.: Overview over all test materials including the atom types, the number of atoms per type N_{atom}, and the number of configurations/structures N_{conf}.

Material	Atom type	N_{atom}	N_{conf}
Random Gas	H	64	200
Zirconia (ZrO_2)	Zr	32	592
	O	64	
Sodium Chloride (NaCl)	Na	64	1014
	Cl	64	

Tab. 4.1 shows the name of the materials in the first column and the different atom species of those materials in the second column. The third column states the number of atoms of each type present in the primitive cell.
The fourth column contains the number of configurations that were used for training and testing this material.

4.2. Random Gas

There is no material to test the long-ranged electrostatic descriptors, because most realistic materials are well described by local methods or at least include a strong short-ranged interactions. To deal with this issue, a material, where it is clear that only long-ranged electrostatic interactions are important, is needed. A simple test case for this is a random gas, as it was also used by Griasfi and Ceriotti in [36].
Here, we want the atoms to be randomly distributed in the primitive cell, ensuring that they do not get to close to each other, to still have mostly electrostatic interactions, and avoiding to large differences in pairwise energies. To avoid large energy differences, it is necessary that the atoms maintain a minimum distance of about 2.5 Å. Still, we want to place 64 atoms in a cubic cell with side lengths between 12 and 20 Å. With random placement of atoms, this is virtually impossible. So in a first step, the atoms were placed on a grid and then we performed a Metropolis Monte Carlo (MC) calculation for hard spheres [59,60] to obtain a set of independent configurations with random atomic positions and a minimum distance/hard sphere radius of 2.5 Å.

Then the atoms were randomly split into two groups of 32 atoms and stored in a list sorted according to these two groups. This list was then used to create the POSCAR file. In the POTCAR file the types of the atom groups were specified. Both were set to be hydrogen atoms with positive charge for the first half of the atoms and negative charge for the second group. For this test, POSCAR files for a set of 200 training and validation structures were created.
These files were then used to calculate the Ewald energies and forces using `VASP`, and this information was stored and used as training data by putting it into a file that resembles the ML_AB file and contains all the training data for the different structures. This file can then be used to perform the usual training and prediction using the `Python` code. [1]

Since the source terms in this material are single atoms and therefore only monopoles, the use of a single angular basis function is sufficient.

4.3. Zirconia

This material with many configurations serves as a realistic test case, as there might be long-ranged electrostatic interactions present in Zirconia. Whether this is the case will be elaborated in section 6.2,
During the creation of this data set the energies were weighted more strongly. Because of this, it makes sense that for our results, the error in the energies might be lower in comparison to the force error than this would be the case for other materials where the energy was weighted not as strongly.

[1] To avoid complications when reading in the ML_AB file with the `polipy4vasp` code, the first atom group was labeled as hydrogen atoms and the second one as helium, but this is just to distinguish the differently charged atoms as two types.

4.4. Sodium Chloride

This is another large test set, as it can also be used to train a NN, that requires more training data than the kernel methods. We use it here because it is a realistic set of a material where long-ranged electrostatic interactions are potentially present due to the large difference in electronegativity between Na and Cl.

5. Choice of Scaling Constant and Hyperparameters for Slater2

In this chapter, we examine Slater2 and aim to find the optimal scaling factor for the exponents as well as the best number of radial basis functions and the optimal hyperparameters that we need when we apply the model to the random gas.
As already stated in section 4.2, it is sufficient to only use one angular basis function for this material.

5.1. $N_{\mathbf{max}}$ and Scaling Factor

The first thing that needs to be determined when working with the Slater2 method, is the optimal scaling factor as shown in Eq. (2.68) and the number of radial basis functions. In order to test this, a Gaussian broadening of the atoms of 0.6 Å, 32 reciprocal grid points in each direction and 250 kernel basis functions were used.
Fig. 5.1 shows the root-mean-square percentage error (RMSPE) of the energies and forces as a function of the resulting minimal exponent of the STOs ζ_{min}. The RMSPE is the root-mean-square error (RMSE) of the predicted properties divided by the variance of the exact results. Thus, we have a unitless measure that does not depend on the magnitude of the exact values.

For a fixed number of radial basis functions varying the scaling constant s and always starting with $\zeta_{\text{max}} = 10.604$ obviously yields different minimal exponents (see Eq. (2.68)). These are used as x-axis in Fig. 5.1.
Independent of the number of radial basis functions, the plot shows a region of minimal exponents that yield the smallest errors. But clearly 5 radial basis functions yield the best results and increasing the number of radial basis functions for the same scaling factor does not improve the error further. This is surprising since, as discussed in section 2.4.4, adding additional basis functions with smaller exponents should not change the results. Why this happens nonetheless is discussed in the following paragraphs.
In choosing the optimal scaling factor, we focus on the graph of the forces because the energies are prone to overfitting, since we have little training data for them. So this test shows that a scaling factor of 3 with 5 radial basis functions[1] is the best choice.

That the error increases for larger ζ_{min} than the best choice is obvious. If ζ_{min} is too large, we can not model long distance contributions to the density as the STOs decay

[1] This corresponds to a minimal exponent $\zeta_{\text{min}} = 0.131$.

too quickly. Therefore, the scaling constant needs to be increased to have slowly decaying functions in our basis set.

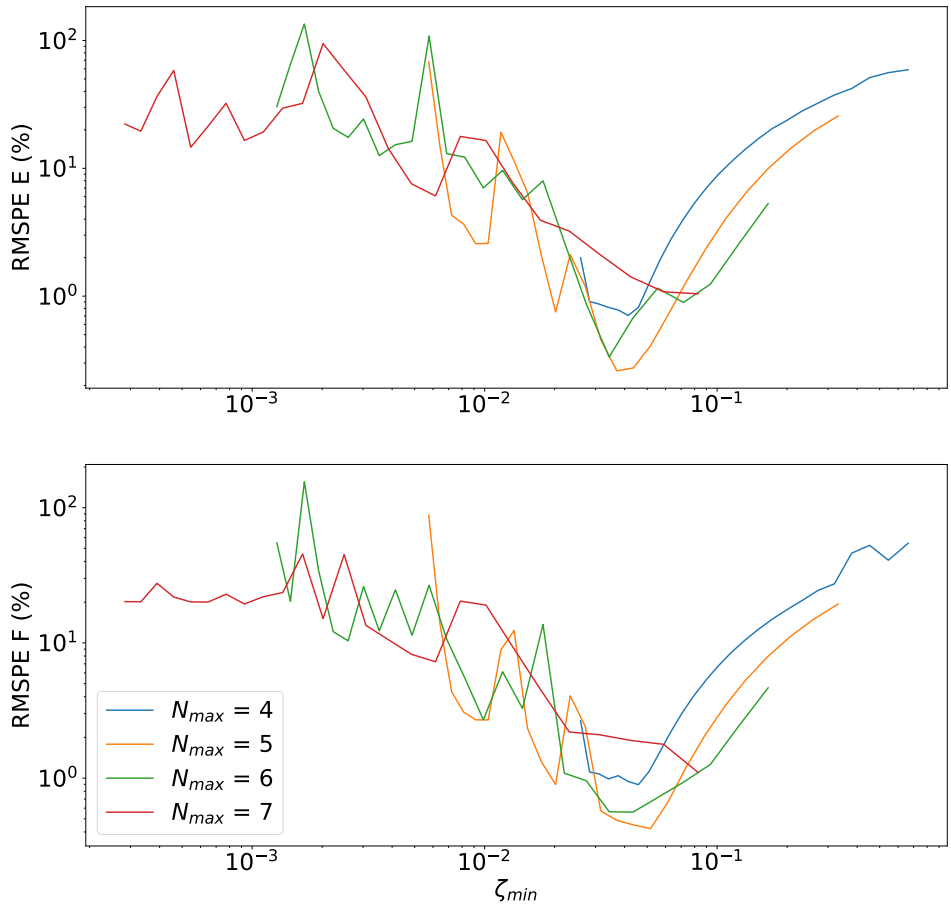

Figure 5.1.: RMSPE as a function of different minimal exponents ζ_{min} for different numbers of radial basis functions N_{max} for the random gas with maximal exponent $\zeta_{\text{max}} = 10.604$. The smallest error is achieved for $N_{\text{max}} = 5$ and a scaling factor of $s = 3$ between two subsequent exponents.

The question remains why the error increases again if one chooses a larger scaling constant and thus a smaller ζ_{min} while the number of basis functions stays constant. In this case, the basis functions are too spread out and no longer complete enough. This means that there are decays of a certain slope that cannot be modeled with this basis set, as all

decays of the basis functions are too far off. In theory this problem should be resolved by choosing a smaller scaling constant and more basis functions. But Fig. 5.1 shows that this is not the case even for more than 5 radial basis functions, as the RMSPE increases again for $\zeta_{\text{min}} < 0.131$.

Another problem is that using more than 5 basis functions and a scaling factor of $s = 3$ also leads to an increase in the error.
This indicates that there is not only an optimal minimal exponent, but at the same time an optimal number of radial basis functions and thus an optimal scaling constant to reach this minimal exponent. Adding more basis functions in between the most and least concentrated one (staying at the same ζ_{min} in the plot but going to higher N_{max}) as well as adding another basis function, that may not provide any further information, with the same scaling factor (lowering ζ_{min} and simultaneously increasing N_{max}) worsens the results.

In order to explain this behavior, we look at the leverage score [20, 61] of the different descriptors for the random gas to have a measure of their relative importance. The higher the leverage score, the more important is the corresponding descriptor and its contribution to the energies and forces.
The leverage score is calculated as follows: First the square matrix

$$\mathbf{A} = \mathbf{X}^{(2)}\mathbf{X}^{(2)T} \tag{5.1}$$

with the radial descriptor matrix $\mathbf{X}^{(2)} = \left[\mathbf{X}_1^{(2)}, \mathbf{X}_2^{(2)}, ...\right]$ containing all radial descriptors $\mathbf{X}_i^{(2)}$ is defined.
Then the eigenvalues λ_ξ and eigenvectors $\mathbf{u}_\xi$ of this symmetric and positive definite matrix $\mathbf{A}$ are calculated. The correlation of the eigenvalues can then be measured with a leverage score:

$$w_j = \frac{1}{K}\sum_k u_{jk}. \tag{5.2}$$

The index k runs over all eigenvectors from whose eigenvalues the correlation is to be measured. The number of these eigenvalues is K.

In Tab. 5.1 we look at the leverage scores when using a scaling factor of 3 and 5 or 7 radial basis functions. In order to make things comparable and tell which descriptors are the important ones, we only use either the first 10 features (first 5 descriptors of positively (+) and negatively (-) charged atoms) or the remaining ones. When we us the first 10, we look at how much the sixth and seventh descriptors would contribute to the result. Looking at the first 10 descriptors corresponds to taking $k \in [1, ..., 10]$ in Eq. (5.2). When calculating the leverage score for the last 4 descriptors, we see how strongly they are correlated to all other descriptors. This corresponds to setting $k \in [11, ..., 14]$ in Eq. (5.2).
In theory, the sixth and seventh descriptors should not change the result, if we assume that 5 basis functions are sufficient. This would imply that their leverage score is zero or

at least close to zero when calculating it from the first ten eigenvalues. This should be the case, since the first ten eigenvalues correspond to the first ten descriptors built with the first five basis functions. There are 10 descriptors instead of 5 because we have two atom types.
When we calculate the leverage scores for the last 4 eigenvalues, we expect the results to be high for these descriptors and low for all others. This would show that the descriptors resulting from the additional basis functions only are correlated to themselves and that they are not a linear combination of already existing functions and therefore add new information to the basis.

As the random gas has two different atom types, there are $2N_{\max}$ many descriptors. Tab. 5.1[2] clearly shows in the third column that the leverage scores of the sixth and seventh descriptors ($n = 6$ and $n = 7$ for both atom types) are not always close to zero, but significantly smaller than the scores of the first ten descriptors.
The fourth column of Tab. 5.1 shows that when using only four descriptors and asking what are the four descriptors that are correlated to the sixth and seventh descriptors, the last four lines show by far the highest scores, as expected. This means, when introducing a sixth and seventh basis function to the model, these basis functions add additional information and are not just a linear combination of already existing ones.
The second column of the table shows that when using only five radial basis functions and calculating the corresponding leverage scores with respect to all ten features, all descriptors are equally important and contribute to the model.
One thing we notice is that for 7 radial basis functions, the third, fourth and fifth descriptors are the most important ones. This makes sense because the corresponding basis functions are the long-ranged ones. Whereas the first and second basis functions have higher exponents and decay faster.

These results tell us that the ML algorithm is not capable to distinguish which descriptors are important and contribute to the energies and forces and which not. This is due to the fact that adding new, independent basis functions that provide new but unnecessary information affects the similarity measure of the kernel method. If the expansion coefficients originating from these additional basis functions are similar, the underlying configurations might still not be. This can happen because at large distances the slope of the basis functions is flat and the expansion coefficient has similar values, even if the distances are very different. Therefore, applying the kernel gives a higher similarity measure than it would with only five basis functions. Then this unnecessary information influences the design matrix and therefore the weights of the fit and ultimately the energies and forces.

This means that the user needs to run some tests and fix the number of radial basis functions, such that all necessary information about the pair distribution function is encoded in the radial basis functions, but no additional meaningless information.

[2] It has to be noted that this table only shows the leverage scores for the atom types interacting with the same atom type. The mixed terms are not shown.

Table 5.1.: Leverage scores for different numbers of radial basis functions $N_{\max}$ calculated for different descriptors (see line 2 and text for details) of the random gas and central atoms with positive (+) and negative (-) charge. For $N_{\max} = 7$ the last four descriptors only do not contribute to the energies and forces and only correlate to themselves.

	$N_{\max} = 5$	$N_{\max} = 7$	$N_{\max} = 7$
leverage score calculated for descriptors	all	first 10	last 4
$n = 1$ (+)	0.1	0.072671	0.067301
$n = 1$ (-)	0.1	0.072360	0.073991
$n = 2$ (+)	0.1	0.066593	0.081027
$n = 2$ (-)	0.1	0.063454	0.097253
$n = 3$ (+)	0.1	0.089656	0.024424
$n = 3$ (-)	0.1	0.085885	0.046994
$n = 4$ (+)	0.1	0.099173	0.001958
$n = 4$ (-)	0.1	0.098834	0.003988
$n = 5$ (+)	0.1	0.099976	0.000057
$n = 5$ (-)	0.1	0.099951	0.000052
$n = 6$ (+)		0.007406	0.249611
$n = 6$ (-)		0.002473	0.249794
$n = 7$ (+)		0.001174	0.249913
$n = 7$ (-)		0.000513	0.249973

5.2. Other Hyperparameters

The hyperparameters are variables that partly determine how good the fit of the model might be. This includes the number of radial basis functions $N_{\max}$ (main quantum number), the number of angular basis functions determined by setting $L_{\max}$, the angular quantum number, the width of the Gaussian broadening σ_{atom}, the number of kernel basis functions, two weighting parameters for the radial and angular descriptors $[\beta_{\text{rad}}, \beta_{\text{ang}}]$, the power of the polynomial kernel, weighting factors w_{energy} and w_{force} that determine how strong the energies and the forces should influence the fit, the cutoff-radius R_{cut} for short-ranged models and, if the calculations are done in reciprocal space, the number of reciprocal grid points p in each direction.
As discussed above, it is sufficient to only use $l = 0$. The weighting factors for energies

and forces are both set to 1 and we use a polynomial kernel of order 4.

Now that the best scaling constant and number of radial basis functions have been found, we can optimize the different hyperparameters by performing tests over ranges of different parameters and pick those that minimize the error. As we have multiple parameters to optimize, a full optimization by searching in the whole hyperparameter space is very costly, as we would need to calculate the RMSPE for all possible combinations of hyperparameters. For this reason, we start with a most general choice of the hyperparameters (parameters that have been generally proven to work) and optimize them one by one, starting with the most influential parameter.
Initially, we used the same set of hyperparameters as for the tests of the scaling constant, namely $p = 32$ reciprocal grid points in each direction, which is very large so that the chances to introduce an additional error are low, a Gaussian broadening of the atoms of $\sigma_{\text{atom}} = 0.6$ Å and 250 kernel basis functions.

5.2.1. σ_{atom}

As a first parameter, σ_{atom} is optimized by varying it in steps of 0.1 Å. This is the first parameter to optimize, since it can strongly influence the results and the error does not converge to a certain value, but has a strict minimum that needs to be found. The results are shown in Fig. 5.2 and we see that $\sigma_{\text{atom}} = 0.5$ Å gives the smallest errors, and even changing it slightly increases the error noticeably. Therefore, this value will be used for further calculations.

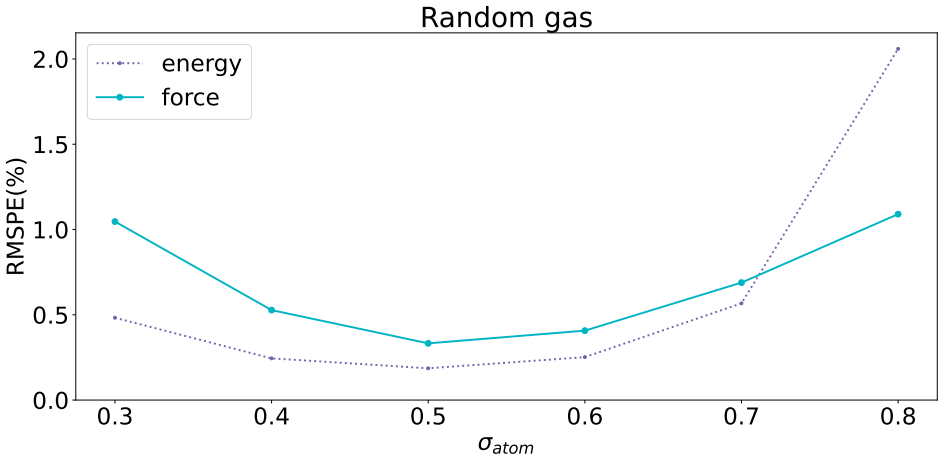

Figure 5.2.: RMSPE in energies and forces as a function of the broadening parameter σ_{atom} for the random gas with Slater2 radial basis functions. A clear minimum in the error at $\sigma_{\text{atom}} = 0.5$ is shown.

5.2.2. Number of Kernel Basis Functions

The next parameter we look at is the number of kernel basis functions, because here we have to choose enough to get the RMSPE to converge to a fixed value. On the other hand, we want to keep the number as small as possible to speed up the calculations. In general, the kernel basis functions are selected randomly.
The results are presented in Fig. 5.3 and it becomes clear, as we again focus mainly on the error of the forces, that 250 already was a good choice, but the error in the forces slightly improves if we chose 300 kernel basis functions. So the greater number is even better and used for the following calculations.

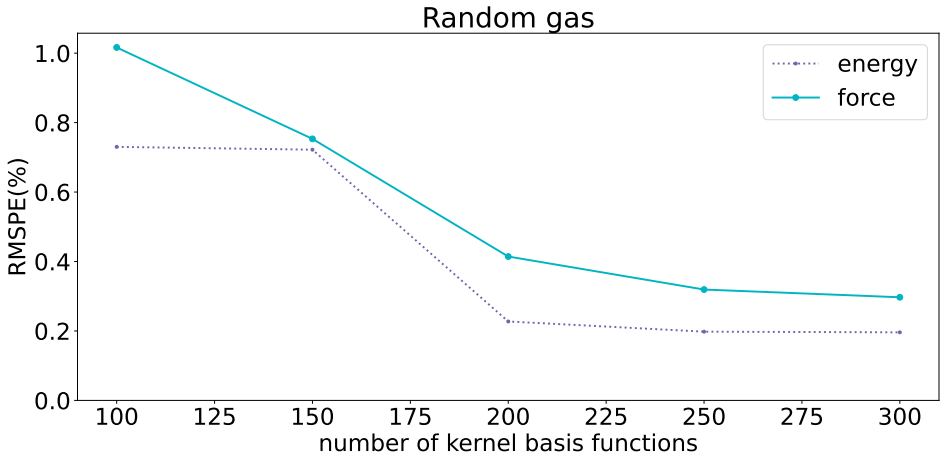

Figure 5.3.: RMSPE in energies and forces as a function of the number of kernel basis functions for the random gas with Slater2 radial basis functions. The error is converged for 300 kernel basis functions.

5.2.3. Number of Reciprocal Grid Points

In a final step, we want to see if we can reduce the number of reciprocal grid points, as 32 is already a very large number, leading to long runtimes and very high memory requirements[3].

In Fig. 5.4 we see that $p = 32$ gives the best results but $p = 28$ is only slightly worse. Therefore, $p = 28$ is used for subsequent calculations, as it is less critical in terms of runtime and memory and yields almost the same precision as larger numbers of reciprocal grid points.
One has to keep in mind that p is the number of grid points for a FT. Here this FT is done analytically, but it could also be done numerically via the fast Fourier Transform

[3]On the machine we ran the tests $p = 32$ is already the maximum number of reciprocal grid points we can use without exceeding the available memory of 500 GB RAM.

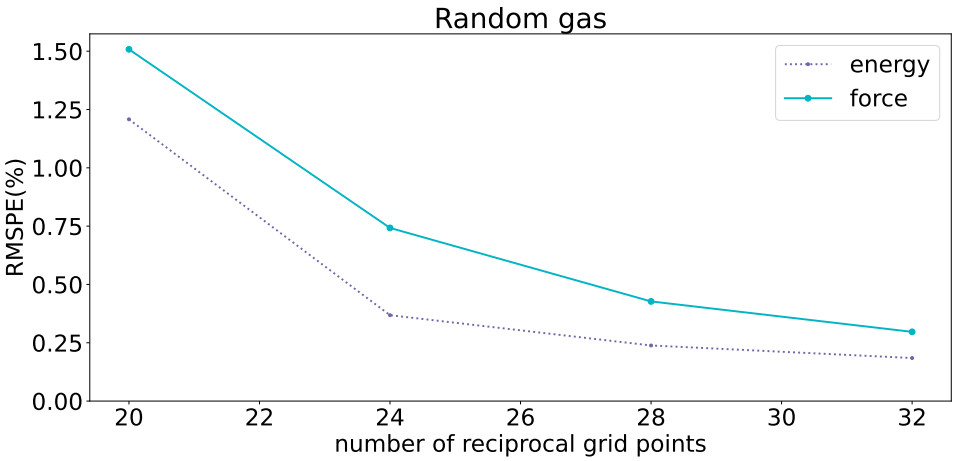

Figure 5.4.: RMSPE in energies and forces as a function of the number of reciprocal grid points in each direction for the random gas with Slater2 radial basis functions. It shows that the more reciprocal grid points we use, the better the results become. Due to memory requirements and runtime, and the fact that the change from 28 to 32 is not very large, the use of 28 points is recommended.

(FFT) algorithm [62]. In this algorithm the FT is evaluated pairwise on points and the array of FFT grid points is set to a length of 2^n in `numpy` because the Cooley–Tukey algorithm [63] is used. Here n is chosen as small as possible and the excess entries are set to zero.
With this in mind, the optimal choice for p would be a number which full fills $p = 2^n$ or at least is an even number.
There exist other algorithms, like Bluestein's algorithm [64] where not only FFT grid points as powers of two are taken, but also allow for powers of small prime numbers. This algorithm is implemented in `scipy` in addition to the Cooley–Tukey algorithm. Thus, if one wants to perform an FFT in `python`, doing it with `scipy` is recommended, as the algorithms allow for more flexibility when choosing the number of reciprocal grid points and still perform the FFT fast and memory efficient.

6. Comparison of Radial Basis Functions for Different Materials

6.1. Random Gas

In tab. 6.1 the optimized hyperparameters for the different radial basis functions, the runtime t and the RMSPE are shown.
The results in the table were produced using only one angular basis function, which is sufficient for the random gas because we only have monopoles. Four or five radial basis functions were used and a cutoff radius of 9 Å for the calculations with spherical Bessel functions as radial basis functions. This cutoff radius is already relatively large compared to the usual 5 or 6 Å. Depending on the cell size the cutoff sphere for some atoms covers already multiple periodic images.
When calculating the descriptors in reciprocal space, only these descriptors were used and not a combination with the short-ranged ones.

Table 6.1.: Optimal hyperparameters and RMSPE in energies (E) and forces (F) for different types of radial basis functions for the random gas. With the angular quantum number $L_{\max} = 0$ only one angular basis function is used and $N_{\max}$ is the number of radial basis functions, R_{cut} is the cutoff radius used in the localized methods, σ_{atom} is the broadening parameter for the atoms, p is the number of reciprocal grid points in each direction, LRC is the number of kernel basis functions, which are selected randomly, and t is the runtime of the training for all 200 training configurations. The Slater2 method yields by far the best results and the short-ranged methods are not able to capture the long-ranged interactions.

	$N_{\max}$	$L_{\max}$	R_{cut} (Å)	σ_{atom} (Å)	p	LRC	RMSPE E (%)	RMSPE F (%)	t (s)
Bessel real	4	0	9	0.6		250	26.7	33.6	36
Bessel rec	4	0	9	0.6	20	250	133.3	18.2	252
Slater1	4	0		0.6	20	250	3.6	2.7	248
Slater2	5	0		0.5	28	300	0.2	0.4	652

When looking at the results in the table, we focus on the results for the forces, as the energies are prone to overfitting. We can see that the short-ranged descriptors have an error of 33.64% which is very sizable. Normally, we expect results in the low single-digit percentage range, as, for example, heated up silicon has a RMSPE of 1.7% in the energies and 3.8% in the forces. Going to reciprocal space and using spherical Bessel functions about halves the errors, but they are still too large.
Only the different Slater-like types of basis functions give results of the expected order, or even one order of magnitude better in case of Slater2. We see that Slater2 works better than Slater1, but requires more reciprocal grid points and more kernel basis functions for the errors to converge. Because of this small number of reciprocal grid points, Slater1 is slightly faster than Slater2.
Interestingly, Slater2 is by far the best approach, yielding errors of less than 1%. This small error can not be achieved by the other methods, even if more reciprocal grid points and radial basis functions are used.

So far we have only looked at the training error, but we still need to consider the prediction error to see whether the model is overfitting.

Table 6.2.: RMSPE in energies (E) and forces (F) and runtime for predicting all 200 configurations of random gas validation data for different types of radial basis functions. Again, Slater2 yields the best results and overall the validation errors are similar to the training errors. Therefore, the model is not overfitting.

	RMSPE E (%)	RMSPE F (%)	runtime (s)
Bessel real	36.9	35.9	16
Bessel rec	196.0	19.1	749
Slater1	5.2	2.7	790
Slater2	0.4	0.4	2219

Comparing Tab. 6.1 and Tab. 6.2, where the errors for the validation set are shown, we see that there is hardly any overfitting and the results for training and validation agree well. Also, when predicting with a trained MLFF, it becomes clear that the approach in real space is much faster than in reciprocal space. This is again because of the greater number of derivatives with respect to the nearest neighbours. Here, Slater2 is by far the slowest approach because of the additional reciprocal grid points and kernel basis functions.
Now as we see that the errors for Slater2 are the smallest, the question remains whether they are small enough and how good the result actually is.

Fig. 6.1 shows that the errors for Slater1 are already relatively small. It shows the

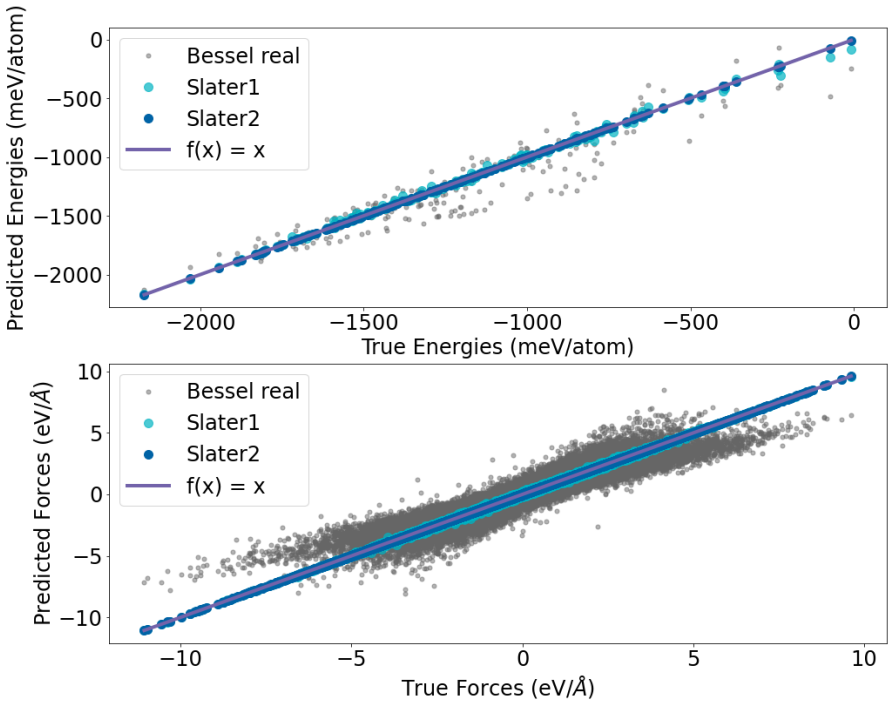

Figure 6.1.: Plot of predicted versus actual energies and forces of the validation data set for the random gas using different types of radial basis functions. The short-ranged method is not able to capture long-ranged interactions and make correct predictions. Slater1 produces good results with very few outliers, Slater2 works even better and no outliers are shown.

predicted energies and forces over the true values of the validation set, and both energies and forces show a nice linear behavior with very few outliers. When looking at the results from Slater2, we see that this linear behavior is even more pronounced and no outlieres are present.
To demonstrate how well the long-ranged descriptors work, also the results obtained with the short-ranged method are plotted in the background. It becomes clear that this approach does not work for the random gas, since the points of the energies are very wide spread around the ideal line and the forces do not even show a linear behavior.

The results for this toy model look really promising. As expected, long-ranged descriptors for Coulomb interaction work really well with the right choice of basis functions. Nonetheless, when comparing the errors for the validation set to the long-distance equivariant (LODE) framework presented in [36], we find that these errors are even lower at a RMSPE of about $2 \cdot 10^{-2}\%$ for the energies. For the forces there exists no value for comparison, since LODE only predicts the energies of a structure and therefore also needs more training configurations. This is probably also the reason why our errors are larger, because we fit

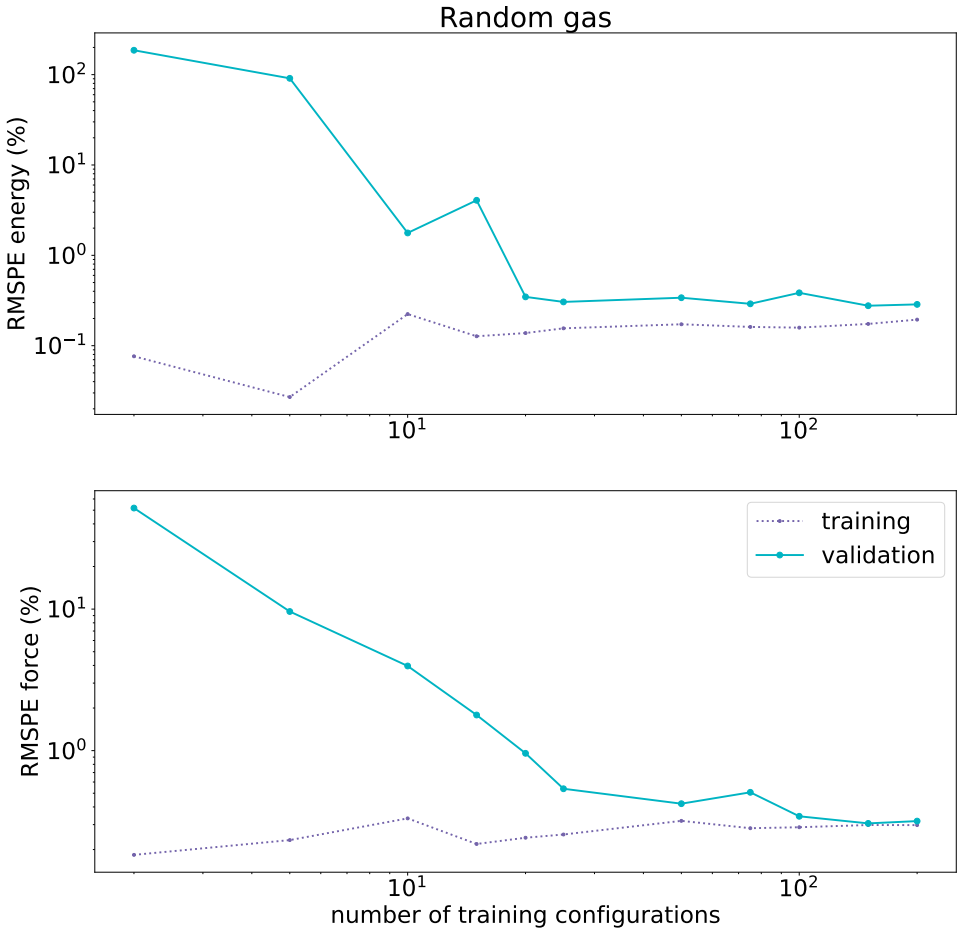

Figure 6.2.: Learning curve of the Slater2 method for the random gas as a function of the number of training structures. For more than 50 training configurations the model hardly improves and the error is almost constant. For fewer training structures the model is clearly overfitting.

both properties with fewer training configurations and focus on the forces. If we want to fit more than one property with a single model, there will always be a compromise on how good the results for the different properties can be. This might imply that by fitting the forces we potentially lose precision in the energies.
Still, our results for the random gas are in the single-digit percentage range, which is what we also generally obtain for other materials such as zirconia using the short-ranged

descriptors. For this see section 6.2.
For this reason, we consider the results for the random gas as a success, especially the ones from Slater2, and continue applying long-ranged descriptors to more complex materials where also short-ranged interactions are present.

As a final test, we want to look at the learning curve for the Slater2 method, as it is the best one, and determine how many training configurations we need and whether we need even more than the current 200 to further improve the results.
To do this, we use different numbers of training configurations, train the model, and then use it to predict energies and forces for 20 random structures of the validation set. Since we choose both the training and validation configurations randomly, we average the resulting errors over multiple runs, to get a stable curve.

The final learning curve is shown in Fig. 6.2. We see that the errors of training and validation agree well and are converged for the 200 configurations we have used so far. This tells us that the tests that we have done so far, are valid and their outcome would not change, if we would use more training structures. On the contrary, we could have used even less configurations, as the errors hardly improve beyond 50 structures.
For less than 50 structures, we see the typical behavior of such a learning curve, where the errors in training are low and the prediction errors are large. That means that in this region we observe overfitting and the model adapts too much to the training structures and is not general enough. This generality is achieved when using more training structures.

6.2. Zirconia

Zirconia is a material that is already described very well using only short-ranged descriptors, as we can see in Tab. 6.3. Nevertheless the question arises, whether there are electrostatic interactions present because of the difference in electronegativity between Zr and O.
Here we know from other tests that in this large data set, no overfitting happens for the calculations in real space and we can only look at the accuracy of the training data. For this, we use the 831 and 1090 kernel basis functions given in the ML_AB file and do not calculate them again. Whenever we use long-ranged descriptors (labeled with Slater), we use them in combination with the short-ranged ones (labeled wit Bessel[1]) as we already know that short-ranged interactions are very pronounced in zirconia and the force field can be trained using only short-ranged descriptors. To be able to compare the results, the short-ranged part is calculated with the same parameters as the reference values in real space.

In Tab. 6.3 the results for training on zirconia are shown for $\sigma_{\text{atom}} = 0.6$. We see that the short-ranged descriptors give the best results. The combined descriptors perform

[1]Bessel functions are only used in real space, because we only want to use long-ranged basis functions in reciprocal space. This means that whenever we write "Bessel", we refere to the basis functions in real space.

Table 6.3.: RMSPE in energies (E) and forces (F) for local and combined descriptors and $\sigma_{\text{atom}} = 0.6$ for zirconia. In the combined approach 6 Bessel functions and 4 STOs are used as radial basis functions ($N_{\max} = 4 + 6$). The angular quantum number $L_{\max}$ encodes the number of angular basis functions. For the short ranged descriptors the cutoff radius $R_{\text{cut}} = 5$ was used, p is the number of reciprocal grid points in each direction ant t is the runtime of the training with all 592 structures. The purely local approach (Bessel) works better than the combined method (Slater1 + Bessel). This means that there are no long-ranged interactions present in zirconia.

	$N_{\max}$	$L_{\max}$	R_{cut} (Å)	p	RMSPE E (%)	RMSPE F (%)	t (s)
Bessel	6	3	5		2.5	7.7	1762
Slater1 + Bessel	4 + 6	0 + 3		24	3.6	9.5	4916

slightly worse than the purely short-ranged ones.
This shows that there are no long-ranged electrostatic interactions present in zirconia, or at least none that we can capture with our set of descriptors. A reason for this might be that the cell used for zirconia is rather small, with a maximum side length of 10.14 Å. It also has a very high atom density with 96 atoms in that cell. Thus, choosing a cutoff radius of 5 Å already includes almost all atoms, and adding the long-ranged descriptors only adds some additional values with no further information to the final descriptor and only worsens the results, as the model is not capable of dealing with unnecessary information, as we have already seen in chapter 5.

Because we already know that there are no long-ranged interactions present in zirconia, we do not need to train the MLFF using the computationally costly Slater2 method.

6.3. Sodium Chloride

Another material where long-ranged electrostatic interactions are expected to be relevant is NaCl. Again, we use the kernel basis functions (2974 for Na and 2401 for Cl) given in the ML_AB file. When we combine long- and short-ranged descriptors, we use the same parameters for the short-ranged part as applied for the short-ranged descriptors. So in Tab. 6.4, if the combined descriptors are used, the columns for $N_{\max}$ and $L_{\max}$ show the values used for the long-ranged descriptors when they are used. The broadening parameter was always set to 0.5 Å.

Tab. 6.4 shows that descriptors in real space and single reciprocal descriptors do not produce good results with small errors. The combined descriptors with Slater1 and Slater2

Table 6.4.: RMSPE in energies (E) and forces (F) for local, long-ranged and combined descriptors for NaCl with the broadening of the atoms of $\sigma_{\text{atom}} = 0.5$. The parameter $N_{\max}$ indicates the number of radial basis functions used per method. The angular quantum number $L_{\max}$ encodes the number of angular basis functions. For the short ranged descriptors the cutoff radius $R_{\text{cut}} = 5$ was used, p is the number of reciprocal grid points in each direction ant t is the runtime of the training with all 1014 structures. The Slater2 method in combination with the local approach (Bessel) gives the small-eat errors and works better than all other method. This means that our combined descriptors successfully describe materials with both short- and long-ranged interactions.

	$N_{\max}$	$L_{\max}$	R_{cut} (Å)	p	RMSPE E (%)	RMSPE F (%)	t (s)
Bessel	8	3	5		14.3	15.7	19715
Slater1	4	0		24	51.2	50.8	12019
Slater2	5	0		24	32.2	27.2	*7622*
Slater1 + Bessel	4 + 8	0 + 3		24	3.4	4.0	31433
Slater2 + Bessel	5 + 8	0 + 3		28	2.7	3.8	*16176*

as radial basis functions give very good results with small errors.

When looking at the results for Slater2, we find that in the purely long-ranged case it again gives the best results even for only 24 reciprocal grid points.
When we combine this method with the short-ranged descriptor, the results are only slightly better than the ones obtained with Slater1 + Bessel. The reason for this could be that Slater2 works especially well for long-ranged interactions, as we have seen for the random gas. But the error in this case is mainly limited by the prediction of the short-ranged interactions, and it makes no difference whether we choose Slater1 or Slater2 for the long-ranged part, as both of them work sufficiently well.

Here we can see that combing both short- and long-ranged descriptors works better than both individual approaches. As expected, neither short-range interactions nor long-range interactions can describe the material sufficiently well. It should be noted, however, that the purely short-ranged method is not as bad as the purely long-ranged one. This means that short-ranged interactions are stronger, but long-ranged electrostatics cannot be neglected. This again points towards the fact that the final error in the combined method is limited by the short-ranged part.
Combining the two descriptors and using Slater1 or Slater2 basis functions gives very good results and shows that for materials where both types of interactions are present

this new approach works well.
Again, one has to note that the combined method has in the present implementations a steep price in terms of long runtimes[2]. Training the system with the combined descriptors takes approximately as long as the runtime of both individual approaches added together. That the runtime for the long-ranged method is lower, is due to the fact that we use only four radial basis functions and only one angular one, whereas in real space eight radial basis functions and four angular ones are used. If we would use the same number in reciprocal space, the runtime would significantly increase.
Since choosing only four radial basis functions in reciprocal space is sufficient, the runtime for the combined descriptors is only 1.64 times that for the method in real space, which is a very small factor compared to the improvement it delivers.

[2]For Slater2, the runtime is written in italics because the calculations were performed on more cores (16 instead of 4) and is therefore lower.

7. Discussion and Outlook

7.1. Conclusion

7.1.1. The Approach

First, we developed a non-local scheme for computing descriptors capable of describing long-range electrostatic interactions in our kernel-based Machine Learning (ML) model to generate a Machine learned force fields (MLFF) for a given material. For this we perform the calculations in reciprocal space and do not use a cutoff radius. So all atoms are used for the calculation of the descriptors and indirectly a summation over all periodic images of the atoms is performed and thus long-ranged interactions can be captured.

In a second step, a new set of radial basis functions was developed. This again does not depend on a cutoff radius and is designed to mimic the slowly decaying $1/r$ behavior of electrostatic interactions. The basis set we used is the radial part of Slater type orbitals (STOs) (2.56) with different sets of exponents.
One method for determining these exponents is to satisfy the completeness profile (2.67) to obtain a complete basis. This approach is called Slater1.
Another method for choosing the exponents is to scale them logarithmically (2.68) and empirically determine the optimal scaling constant. This set of basis functions is labeled Slater2.

With this new set of descriptors and radial basis functions we are able to describe purely long-ranged electrostatic interactions very well. But in order to apply the ML model to realistic materials, where also short-ranged interactions are present, we need to combine the new long-ranged descriptors with the localized ones used previously. This is done by appending both descriptor vectors to form a new super vector that encodes information about both short- and long-ranged interactions.

7.1.2. Summary of the Results

We have shown that for a random gas of point charges, where, by construction, only long-ranged monopole-monopole interactions are present, the Slater2 method with its logarithmically scaled exponents gives the best results. It produces a root-mean-square percentage error (RMSPE) smaller than 1%. Using the Slater1 set of basis functions, we obtain errors in the low single digit percentage range. This is smaller than errors of about 10% we usually expect from our ML model [18, 20, 38], that has been proven to work well for standard short-ranged problems.

It has also become clear that the localized descriptors in real space are not capable of correctly predicting energies and forces in this material, even when using a large cutoff radius.
Our results are not as good as the ones obtained with the long-distance equivariant (LODE) framework in [36]. But in addition to the energies, we can predict forces, which is not done in LODE. This means we are able to fit a toy model where only Coulomb interactions are present with our long-ranged descriptors.
In this toy model we predicted only monopole-monopole interactions. Thus, another test could be to use the model to predict long-ranged electrostatic interactions also for higher orders like dipolar interactions, and using more angular basis functions.

The new descriptors in reciprocal space are only able to predict long-ranged interactions, rather than a combination of long- and short-ranged interactions, as we initially hoped. To achieve a combined description, we would need another set of radial basis functions that resemble spherical Bessel functions or any other set of short ranged radial basis functions in the vicinity of the central atom and then capture the 1/r-decay outside the cutoff sphere.
As long as we do not have such a basis, we need to combine long- and short-ranged descriptors when we want to successfully apply them to realistic materials as already described above.

When using these combined descriptors for the predictions of energies and forces of zirconia, we observe that there are no long-ranged electrostatic interactions present. For this material the localized real-space method gives the best results, and the RMSPE is again in the single digit percentage range. Adding the long-ranged descriptor slightly worsens the results.
For liquid sodium chloride, both the single short- and long-ranged methods give no satisfactory results with large errors. But when we use the combined descriptors we obtain better results with an RMSPE of about 4% in the forces and even smaller in the energies. In this case it does not matter which STOs radial basis functions are used, as the remaining error is mostly influenced by the short-ranged part of the interactions.
For NaCl with the long-ranged descriptors, it is again sufficient to model only the monopole-monopole interactions and use four or five radial basis functions. Compared to the real space part where we need eight radial basis functions and even more angular basis functions, this is a very small number.
This tells us that the combined descriptors work well for materials where strong long-ranged interactions are present.

All in all, there are some things to be aware of when using the long-ranged descriptors, such as choosing the right radial basis functions or when to combine the descriptors with the short-ranged ones, and how to choose the right hyperparameters when a tradeoff between real and reciprocal space is required. If the hyperparameters are well optimized, the new model works well and is able to successfully capture long- and short-ranged

interactions within the same material.

7.1.3. Different STOs

It is quite astonishing that for all our test cases, the logarithmically scaled exponents from Slater2 perform (at least slightly) better than the "optimized" ones from Slater1. Approximating a complete basis set for Slater1 is the theoretically correct and justifiable approach for calculating the exponents of the STOs to obtain a basis that satisfies these conditions as well as possible.
In this light, it is rather unexpected that Slater2 outperforms this approach, as it is a purely empiric method where the scaling and minimal exponent were found with computer experiments.
The next step here is to find a set of exponents for the Slater1 method that mimic the logarithmic scaling and still fulfill the completeness relation (2.59). This should improve the results even further.

7.2. Outlook

Now let us discuss what else can be done to improve the model, what are its current limitations, and how they might be overcome.

7.2.1. Independent Hyperparameters for Short- and Long-Ranged Descriptors

The ML model can run into problems when a hyperparameter, e.g. the broadening parameter σ_{atom}, should be chosen very differently in real and reciprocal space to obtain optimal results. For now we have to make a tradeoff between the different needs and set the parameter to one fixed value.
The obvious solution to this problem is to include two different parameters for σ_{atom} or the respective other hyperparameters. One for the calculations in real and one for reciprocal space, as it has already been done for the number of radial and angular basis functions. Then no such compromises are required. On the other hand, this would allow for more possible combinations of parameters, and the user has to be knowledgeable and possibly do some testing to select the right ones.

This way of decoupling the calculations of long- and short-ranged descriptors can be spun even further, and we could make both methods truly independent from each other. It would mean that all input parameters can be adjusted independently for both methods. This would include the weighting of energies and forces, that throughout all test has been set to 1 for both energies and forces, the order of the polynomial kernel and the number of kernel basis functions to choose. Ultimately, appending both descriptors would still be the best way to form a new descriptor encoding both long- and short-ranged interactions. Then the user has more possibilities to optimize the parameters and therefore the results.

7.2.2. Cutoff Function

As stated above, as long as we do not have a set of radial basis functions that can capture both short- and long-ranged interactions at the same time, we need to combine the two sets of descriptors, because the long-ranged descriptors are not capable of describing short-ranged interactions. Still, this unnecessary and incorrect short-range part is included in the long-range descriptors and should ideally be avoided. This could be done with a cutoff function that is multiplicative in real space.
However, a multiplicative function in real space is a convolution in reciprocal space, and far to expensive to implement.
A possible solution is using the fast Fourier Transform (FFT). The transformation into reciprocal space with a FFT would be fairly simple for energy calculations, but at the same time mathematically complicated for the forces. The corresponding equations have not been worked out yet, but would increase the complexity of the code considerably.

7.3. Summary

In summary, we have shown that long-ranged descriptors can substantially improve predictions for cases where the interactions are strongly ionic and long-range. This is the case for a model of alternating point charges ("random gas"), liquid NaCl, and likely other strongly ionic systems with weak screening.
For other materials, for instance ZrO_2, which one would assume to have at least some ionic contribution to bonding, we found no improvements when including long-ranged descriptors.

Efficiency was not a major concern of this work, and before practical applications can be considered, one would need to investigate how the calculation of the descriptors and their derivatives could be accelerated. The results indicate that this would be potentially worthwhile, but only for a fairly limited class of materials.

Bibliography

[1] T. M. Mitchell, *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

[2] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 14413–14423, 2020.

[3] L. Hardesty, "Explained: Neural network." `https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414`, 2017. Accessed: 10.10.2022.

[4] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. M. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. Zhu, "A survey of uncertainty in deep neural networks," *ArXiv*, vol. abs/2107.03342, 2021.

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

[6] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Phys. Rev. Lett.*, vol. 104, p. 136403, 2010.

[7] J. Behler, "Neural network potential-energy surfaces in chemistry: A tool for large-scale simulations," *Physical chemistry chemical physics : PCCP*, vol. 13, pp. 17930–55, 2011.

[8] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical Review Letters*, vol. 108, 2012.

[9] Z. Li, J. R. Kermode, and A. De Vita, "Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces," *Phys. Rev. Lett.*, vol. 114, p. 096405, 2015.

[10] K. Miwa and H. Ohno, "Molecular dynamics study on β-phase vanadium monohydride with machine learning potential," *Phys. Rev. B*, vol. 94, p. 184109, 2016.

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

[12] J. Behler, "First principles neural network potentials for reactive simulations of large molecular and condensed systems," *Angewandte Chemie International Edition*, vol. 56, pp. 12828–12840.

[13] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," *Science Advances*, vol. 3, p. e1603015, 2017.

[14] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, "Machine learning unifies the modeling of materials and molecules," *Science Advances*, vol. 3, p. e1701816, 2017.

[15] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, "Machine learning force fields: Construction, validation, and outlook," *The Journal of Physical Chemistry C*, vol. 121, pp. 511–522, 2017.

[16] T. Huan, R. Batra, J. Chapman, S. Krishnan, L. Chen, and R. Ramprasad, "A universal strategy for the creation of machine learning-based atomistic force fields," *npj Computational Materials*, vol. 3, 2017.

[17] A. Glielmo, C. Zeni, and A. De Vita, "Efficient nonparametric n-body force fields from machine learning," *Phys. Rev. B*, vol. 97, p. 184307, 2018.

[18] R. Jinnouchi, F. Karsai, and G. Kresse, "On-the-fly machine learning force field generation: Application to melting points," *Phys. Rev. B*, vol. 100, p. 014105, 2019.

[19] V. L. Deringer, M. A. Caro, and G. Csányi, "Machine learning interatomic potentials as emerging tools for materials science," *Advanced Materials*, vol. 31, p. 1902765, 2019.

[20] R. Jinnouchi, F. Karsai, C. Verdi, R. Asahi, and G. Kresse, "Descriptors representing two- and three-body atomic distributions and their effects on the accuracy of machine-learned inter-atomic potentials," *The Journal of Chemical Physics*, vol. 152, p. 234102, 2020.

[21] S. Chmiela, V. Vassilev-Galindo, O. T. Unke, A. Kabylda, H. E. Sauceda, A. Tkatchenko, and K.-R. Müller, "Accurate global machine learning force fields for molecules with hundreds of atoms," *ArXiv*, 2022.

[22] A. Glielmo, C. Zeni, and A. De Vita, "Efficient nonparametric n-body force fields from machine learning," *Phys. Rev. B*, vol. 97, p. 184307, 2018.

[23] M. Willatt, F. Musil, and M. Ceriotti, "Feature optimization for atomistic machine learning yields a data-driven construction of the periodic table of the elements," *Physical Chemistry Chemical Physics*, vol. 20, 2018.

[24] A. Bartok and G. Csányi, "Gaussian approximation potentials: a brief tutorial introduction," *International Journal of Quantum Chemistry*, vol. 115, 2015.

[25] L. Zhang, J. Han, H. Wang, R. Car, and W. E, "Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics," *Phys. Rev. Lett.*, vol. 120, p. 143001, 2018.

[26] R. Kjellander, "Focus article: Oscillatory and long-range monotonic exponential decays of electrostatic interactions in ionic liquids and other electrolytes: The significance of dielectric permittivity and renormalized charges," *The Journal of Chemical Physics*, vol. 148, p. 193701, 2018.

[27] R. Jorn, R. Kumar, D. P. Abraham, and G. A. Voth, "Atomistic modeling of the electrode–electrolyte interface in li-ion energy storage systems: Electrolyte structuring," *The Journal of Physical Chemistry C*, vol. 117, pp. 3747–3761, 2013.

[28] Z. Guo, F. Ambrosio, W. Chen, P. Gono, and A. Pasquarello, "Alignment of redox levels at semiconductor–water interfaces," *Chemistry of Materials*, vol. 30, pp. 94–111, 2018.

[29] R. H. French, V. A. Parsegian, R. Podgornik, R. F. Rajter, A. Jagota, J. Luo, D. Asthagiri, M. K. Chaudhury, Y.-m. Chiang, S. Granick, S. Kalinin, M. Kardar, R. Kjellander, D. C. Langreth, J. Lewis, S. Lustig, D. Wesolowski, J. S. Wettlaufer, W.-Y. Ching, M. Finnis, F. Houlihan, O. A. von Lilienfeld, C. J. van Oss, and T. Zemb, "Long range interactions in nanoscale science," *Rev. Mod. Phys.*, vol. 82, pp. 1887–1944, 2010.

[30] Z. Deng, C. Chen, X.-G. Li, and S. P. Ong, "An electrostatic spectral neighbor analysis potential for lithium nitride," *npj Computational Materials*, 2019.

[31] T. Bereau, D. Andrienko, and O. A. von Lilienfeld, "Transferable atomic multipole machine learning models for small organic molecules," *Journal of Chemical Theory and Computation*, vol. 11, pp. 3225–3233, 2015. PMID: 26575759.

[32] K. Yao, J. Herr, D. Toth, R. Mckintyre, and J. Parkhill, "The tensormol-0.1 model chemistry: a neural network augmented with long-range physics," *Chemical Science*, vol. 9, 2017.

[33] T. Bereau, R. A. DiStasio, A. Tkatchenko, and O. A. von Lilienfeld, "Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning," *The Journal of Chemical Physics*, vol. 148, p. 241706, 2018.

[34] P. Bleiziffer, K. Schaller, and S. Riniker, "Machine learning of partial charges derived from high-quality quantum-mechanical calculations," *Journal of Chemical Information and Modeling*, vol. 58, pp. 579–590, 2018. PMID: 29461814.

[35] B. Nebgen, N. Lubbers, J. S. Smith, A. E. Sifain, A. Lokhov, O. Isayev, A. E. Roitberg, K. Barros, and S. Tretiak, "Transferable dynamic molecular charge assignment using deep neural networks," *Journal of Chemical Theory and Computation*, vol. 14, pp. 4687–4698, 2018. PMID: 30064217.

[36] A. Grisafi and M. Ceriotti, "Incorporating long-range physics in atomic-scale machine learning," *The Journal of Chemical Physics*, vol. 151, p. 204105, 2019.

[37] A. S. Christensen, L. A. Bratholm, F. A. Faber, and O. Anatole von Lilienfeld, "Fchl revisited: Faster and more accurate quantum machine learning," *The Journal of Chemical Physics*, vol. 152, p. 044107, 2020.

[38] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, and M. Bokdam, "Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with bayesian inference," *Phys. Rev. Lett.*, vol. 122, p. 225701, 2019.

[39] V. N. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.

[40] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[41] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, pp. 1171 – 1220, 2008.

[42] "The vasp manual." `https://www.vasp.at/wiki/index.php/File:MLFF_rad_and_ang_descriptor.png`. Accessed: 08.07.2022.

[43] K. Koutroumbas and S. Theodoridis, *Pattern Recognition*. Elsevier B.V., 2008.

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

[45] F. Karsai, "File:mlff rad and ang descriptor.png." `https://www.vasp.at/wiki/index.php/File:MLFF_rad_and_ang_descriptor.png`. Accessed: 09.04.2022.

[46] "The numpy documentation." `https://numpy.org/doc/stable/reference/generated/numpy.einsum.html`. Accessed: 30.09.2022.

[47] T. Moloi, "Spherical bessel functions," 2021.

[48] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[49] G. de l'Hôpital, "Analyse des infiniment petits," p. 145–146, 1696.

[50] B. Klahn and W. A. Bingel, "The convergence of the rayleigh-ritz method in quantum chemistry," *Theoretica chimica acta*, vol. 44, pp. 27 – 43, 1977.

[51] J. D. Jackson, *Classical electrodynamics*. Wiley, New York, 1999.

[52] D. P. Chong, "Completeness profiles of one-electron basis sets," *Canadian Journal of Chemistry*, vol. 73, pp. 79–83, 1995.

[53] G. B. Folland, *Harmonic Analysis in Phase Space. (AM-122)*. Princeton University Press, 1989.

[54] D. Dereniowski and M. Kubale, "Cholesky factorization of matrices in parallel and ranking of graphs," in *Parallel Processing and Applied Mathematics* (R. Wyrzykowski, J. Dongarra, M. Paprzycki, and J. Waśniewski, eds.), (Berlin, Heidelberg), pp. 985–992, Springer Berlin Heidelberg, 2004.

[55] J. P. Gram, "Ueber die Entwickelung reeller Functionen in Reihen mittels der Methode der kleinsten Quadrate," *die Reine und Angewandte Mathematik*, vol. 94, pp. 41–73, 1883.

[56] E. Schmidt, "Zur Theorie der linearen und nichtlinearen Integralgleichungen I. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener," *Mathematische Annalen*, vol. 63, pp. 433–476, 1907.

[57] "The vasp manual." `https://www.vasp.at/wiki/index.php/ML_AB`. Accessed: 08.07.2022.

[58] "The vasp manual." `https://www.vasp.at/wiki/index.php/POSCAR`. Accessed: 08.07.2022.

[59] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines,"

[60] J. Gubernatis, "Marshall rosenbluth and the metropolis algorithm," *Physics of Plasmas*, vol. 12, pp. 057303–057303, 2005.

[61] "The vasp manual." `https://www.vasp.at/wiki/index.php/Machine_learning_force_field:_Theory#Sparsification_of_angular_descriptors`. Accessed: 28.11.2022.

[62] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*. Society for Industrial and Applied Mathematics, 1992.

[63] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.

[64] L. Bluestein, "A linear filtering approach to the computation of discrete fourier transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 18, pp. 451–455, 1970.

Acronyms

`VASP` Vienna Ab *initio* Simulation Package. 1, 2, 7, 35, 36

FFT fast Fourier Transform. 45, 46, 58

FP first principle. 1, 5–7

FT Fourier Transform. viii, 12, 15, 16, 19, 22, 23, 30, 32, 45, 46, 67, 68

GAP Gaussian Approximation Potential. 7

LODE long-distance equivariant. 49, 56

LRC Local Reference Configurations. 4–6, 35, 47

MC Monte Carlo. 36

MD Molecular Dynamics. 35

ML Machine Learning. xi, 1–3, 5, 35, 42, 55, 57

MLFF Machine learned force fields. 1, 2, 5, 35, 48, 52, 55

NN Neural Networks. 1, 37

ONB orthonormal basis. 22, 23

RMSE root-mean-square error. 39

RMSPE root-mean-square percentage error. ix, xi, 39–41, 44–49, 52, 53, 55, 56

SOAP Smooth Overlap of Atomic Position. 7, 8

STOs Slater type orbitals. viii, 3, 20–24, 34, 39, 52, 55–57

A. Appendix

A.1. FTs

This first section of the appendix shows the different FTs in more details.
We will need

$$\int_{-\infty}^{\infty} \exp\bigl(-(ax^2+bx)\bigr)\mathrm{d}x = \sqrt{\frac{\pi}{a}}\exp\left(\frac{b^2}{4a}\right) \tag{A.1}$$

for the following integrations.

A.1.1. Atom Density

Starting out from the expression for the atom density given in Eqs. (2.7) and (2.8), we calculate the FT to get the equivalent expression in reciprocal space shown in Eq. (2.26).

$$\begin{aligned}\rho(\mathbf{G}) &= \int_{\mathbb{R}^3} \exp\left(i\mathbf{G}\mathbf{r}\right)\rho(\mathbf{r})\mathrm{d}\mathbf{r} \\ &= \int_{\mathbb{R}^3} \exp\left(i\mathbf{G}\mathbf{r}\right)\sum_{j=1}^{N_A}\delta(\mathbf{r}-\mathbf{r}_j)\mathrm{d}\mathbf{r} \\ &= \sum_{j=1}^{N_A}\exp\left(i\mathbf{G}\mathbf{r}_j\right)\end{aligned} \tag{A.2}$$

In the first step, we simply plug in the expression for the density from Eq. (2.7) into the FT. Then we can move the sum out of the integral and use the property

$$\int_{-\infty}^{\infty} f(t)\delta(t-T)\mathrm{d}t = f(T) \tag{A.3}$$

of the δ-distribution to gain the final result.

A.1.2. Gaussian Broadening

Here the prefactor of Eq. (2.11) is taken into consideration.

$$\begin{aligned}
\mathcal{F}\left(\exp\left(-\frac{\mathbf{r}^2}{2\sigma_{\text{atom}}^2}\right)\right)\frac{1}{\sqrt{2\pi\sigma^2}^3} &= \frac{1}{\sqrt{2\pi\sigma^2}^3}\int_{-\infty}^{\infty}\exp(-i\mathbf{G}\mathbf{r})\exp\left(-\frac{\mathbf{r}^2}{2\sigma^2}\right)\mathrm{d}^3r \\
&= \frac{1}{\sqrt{2\pi\sigma^2}^3}\int_{-\infty}^{\infty}\mathrm{d}r_1\exp\left(-\left(iG_1r_1+\frac{r_1^2}{2\sigma^2}\right)\right) \\
&\qquad \int_{-\infty}^{\infty}\mathrm{d}r_2\exp\left(-\left(iG_2r_2+\frac{r_2^2}{2\sigma^2}\right)\right) \\
&\qquad \int_{-\infty}^{\infty}\mathrm{d}r_3\exp\left(-\left(iG_3r_3+\frac{r_3^2}{2\sigma^2}\right)\right) \\
&= \frac{\sqrt{2\pi\sigma^2}^3}{\sqrt{2\pi\sigma^2}^3}\exp\left(-\frac{G_1^2\sigma^2}{2}\right)\exp\left(-\frac{G_2^2\sigma^2}{2}\right)\exp\left(-\frac{G_3^2\sigma^2}{2}\right) \\
&= \exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right)
\end{aligned} \tag{A.4}$$

The three-dimensional integral is split up into three different integrals, one over each direction in space. These integrals then can be solved using Eq. (A.1). In the final step, the three resulting terms are then re-combined into the vector formulation.
It becomes visible that the FT of a Gaussian is again a Gaussian that we can use in reciprocal space to broaden the density, as we did in real space.

A.2. Derivatives

A.2.1. Derivative of Long-Ranged Descriptors

Starting out from the expansion coefficients

$$\begin{aligned}
c_{lmn}^{jJ} &= \sum_{\mathbf{G}} f_{lmn}(\mathbf{G})\exp(-i\mathbf{G}\mathbf{r}_j)\rho_J(\mathbf{G}) \\
&= \sum_{\mathbf{G}} f_{lmn}(\mathbf{G})\exp(-i\mathbf{G}\mathbf{r}_j)\exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right)\sum_{\mathbf{r}_J}\exp(i\mathbf{G}\mathbf{r}_J),
\end{aligned} \tag{A.5}$$

where both the density and the factor originating form the FT depend on the atom position $\mathbf{r}$, the derivative with respect to one position coordinate of one atom r_k^α can be calculated with the following equations.

$$\frac{\mathrm{d}c_{lmn}^{jJ}}{\mathrm{d}r_k^\alpha} = \frac{\mathrm{d}c_{lmn}^{jJ}}{\mathrm{d}r_k}\frac{\mathrm{d}r_k}{\mathrm{d}r_k^\alpha} \tag{A.6}$$

$$\frac{\mathrm{d}r_k}{\mathrm{d}r_k^\alpha} = \mathbf{e}_\alpha \tag{A.7}$$

$$\begin{aligned}\frac{\mathrm{d}c_{lmn}^{jJ}}{\mathrm{d}r_k} &= \sum_{\mathbf{G}} f_{lmn}(\mathbf{G}) \exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right) \cdot \left(\frac{\mathrm{d}\exp(-i\mathbf{G}\mathbf{r}_j)}{\mathrm{d}r_k} \sum_{\mathbf{r}_J} \exp(i\mathbf{G}\mathbf{r}_J)\right. \\ &\qquad \left. + \frac{\mathrm{d}\sum_{\mathbf{r}_J}\exp(i\mathbf{G}\mathbf{r}_J)}{\mathrm{d}r_k} \exp(-i\mathbf{G}\mathbf{r}_j)\right) \\ &= \sum_{\mathbf{G}} f_{lmn}(\mathbf{G}) \exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right) \cdot \left(- \delta_{jk} i\mathbf{G} \exp(-i\mathbf{G}\mathbf{r}_j) \sum_{\mathbf{r}_J} \exp(i\mathbf{G}\mathbf{r}_J)\right. \\ &\qquad \left. + \delta_{Jk} i\mathbf{G} \exp(i\mathbf{G}\mathbf{r}_k)\exp(-i\mathbf{G}\mathbf{r}_j)\right) \\ &= i\sum_{\mathbf{G}} f_{lmn}(\mathbf{G})\mathbf{G}\exp(-i\mathbf{G}\mathbf{r}_j) \cdot \left(\delta_{Jk}\exp(i\mathbf{G}\mathbf{r}_k)\exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right) - \delta_{jk}\rho_J(\mathbf{G})\right)\end{aligned} \tag{A.8}$$

$$\frac{\mathrm{d}c_{lmn}^{jJ}}{\mathrm{d}r_k^\alpha} = \left\langle \mathbf{e}_\alpha, i\sum_{\mathbf{G}} f_{lmn}(\mathbf{G})\mathbf{G}\exp(-i\mathbf{G}\mathbf{r}_j)\left(\delta_{Jk}\exp(i\mathbf{G}\mathbf{r}_k)\exp\left(-\frac{|\mathbf{G}|^2\sigma^2}{2}\right) - \delta_{jk}\rho_J(\mathbf{G})\right)\right\rangle \tag{A.9}$$

When calculating all those different derivatives in practice, the array to store them becomes quite large. Suppose that we have a cell with 128 atoms of two different types with 64 atom per type and values of $N_{\max} = 8$ and $L_{\max} = 4$, then the intermediate array before the summation over $\mathbf{G}$ would be of size $j \times (j-1) \times 3 \times 2 \times N_{\max} \times (L_{\max}+1)^2 \times p^3$. With a number of $p = 20$ reciprocal grid points in each direction, that would be an array with $128 \cdot 127 \cdot 3 \cdot 2 \cdot 8 \cdot 25 \cdot 8000 = 1.56 \cdot 10^{11}$ entries. These many entries in double precision result in a memory requirement of 1248 GB. This is the reason why this array is never calculated explicitly, but two different terms are calculated. The first depending only on the basis functions ($f_{lmn}(\mathbf{G})$) with the indices for $N_{\max}$ and $L_{\max}$, but not on the different atoms. The second part, on the other hand, includes all other parts of Eq. (A.8) and depends on the atom indices and types, but not on the basis functions. Only in one final summation step over $\mathbf{G}$ both arrays are used together, but they are not directly multiplied, as the `numpy.einsum()` [46] routine is used.

A.3. Values for ζ_n for Slater1

Tab. A.1 shows the values for ζ_n rounded to three decimal digits used for Slater1.

Table A.1.: Exponents ζ_n for Slater1 method for $N_{\max} = 4$ radial basis functions.

	$n=1$	$n=2$	$n=3$	$n=4$
$l=0$	0.236	0.884	2.828	10.604