Machine Learning for Many-body Potentials with Moment Tensors

Tobias Rudolf Olbrich

Born 1st June 1992 in Saarburg

June 3, 2018

Master's Thesis Mathematics

Advisor: Prof. Dr. Michael Griebel

Second Advisor: Prof. Dr. Marc Alexander Schweitzer

Institute for Numerical Simulation

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Contents

1 Introduction

Computational material science has become an integral part of our modern society and a notable consumer of modern supercomputing power. Its applications include the development of new and improved superconductors, catalysts, and bioengineering[20]. In the words of Peter Galison [27]:

> Without the computer based simulation, the material culture of late-twentieth-century microphysics is not merely inconvenienced – it does not exist.

A variety of subfields of digital simulation have contributed towards creating the components of our present-day life, including quantum physics, quantum chemistry, condensed matter physics, computational statistical mechanics, computational materials science, continuum modelling, circuit layout, and more [2].

As computing analytical and exact solutions is not possible for many interesting problems, computer simulations often employ some kind of discretization in order to compute approximate solutions. There are two main approaches that can be used for this. One is to approximate some area of interest by some discrete mesh. This is useful for problems that are well described by *continuum mechanics*. Many problems are easier to describe by considering a system of discrete particles instead. They can be analyzed by methods for meshless discretization, such as particle methods. The latter is the class of problems considered in this thesis; we give a short overview over particle methods based on [29]. The fundamental mathematical model for particle methods is Newton's second law

$$\mathbf{F} = m\mathbf{a}, \tag{1.1}$$

a system of ordinary differential equations of second order. This system describes the relationship between the forces acting on a particle and their resulting acceleration. The goal is to numerically approximate the solution of this system in an efficient manner. Note that the particles that constitute the system can range from very small to very big in both size and mass. Particle methods can be employed both to simulate the development of a system of atoms, where each particle represents a single nucleus, and galaxies, as system of stars. Problems particles methods are applied to include:

- **Biochemistry:** The dynamics of macromolecules, especially proteins, is one of the most prominent applications of particle methods.
- **Astrophysics:** Particle methods are used to test the soundness of theoretical models. Depending on the size of the simulated system, hundreds or thousands of stars represented by mass points act as particles in the simulation. The forces in the system result from the gravitational potential.

- **Solid state physics:** This is the application most relevant to this thesis[1]. Particle methods are used to analyze the properties of existing materials and for the development of new materials. Properties that can be studied include the behavior of the material under change of temperature or pressure, structural failure under shear stress, the impact of defects, and elastic properties.

To approximate the Newtonian ODE in eq. (1.1), some discretization over time is chosen. If we can compute the forces acting on each particle, we can then predict the evolution of the system using classical mechanics[2]. Computing the forces (as the derivative as the system's energy) is thus the core challenge in particle methods.

Particle methods for systems of atoms In principle, all matter is described by quantum mechanics and the Schrödinger wave equation. However, directly computing analytical solutions of this high-dimensional partial differential equation is impossible for all but very simple ensembles of particles. Thus, methods for approximating its solution must be applied, with particle methods being a natural approach due to the discrete nature of the problem. The two classical ways to compute the (approximate) forces acting on the particles are quantum methods and empirical potentials.

Systems of small size, i.e. hundreds or thousands of atoms, can be studied from first principles by directly approximating the Schrödinger equation. This can be achieved through the Born-Oppenheimer approximation, leading to computational quantum mechanical methods like Kohn-Sham *density functional theory* (DFT) [39]. We will present a short overview over the quantum mechanical aspects in Section 2. These *ab initio* methods can yield results of high precision in all situations where they are computationally feasible. However, the need for repeated evaluation of the potential energy function for large systems and its derivatives for purposes like molecular dynamics quickly exceeds the limits of today's computational capabilities. For example, DFT, which yields a good combination of accuracy and runtime efficiency, scales with $\mathcal{O}(N^3)$ in the number of atoms. Other *ab initio* methods like the *Møller-Plesset perturbation theory* (MP) [47] ($\mathcal{O}(N^4)$ to $\mathcal{O}(N^7)$ depending on the mode used) and *coupled clusters with singles and doubles* (CCSD) [33] ($\mathcal{O}(N^6)$) exhibit even worse scaling behavior. Methods with linear scaling exist, for example *df-MP2* [62] or the *density matrix minimization technique* [43, 15]. However, implementations of such approaches tend to include expensive constant costs. This limits the practical use of *ab*

[1] This is not to be understood as a restriction; the machine learning models we present are also used to analyze organic compounds.

[2] It is worth considering why we can treat systems of atoms with classical mechanics, rather than quantum mechanics. This is justified by the Born-Oppenheimer approximation; see Section 2.2.

initio methods for molecular dynamics to small systems and relatively small time-frames in the order of picoseconds.

For *mesoscale* problems of tens of thousands of particles and timescales in the order of nanoseconds, which is the scale this thesis will focus on, the predominant approach involves interaction potentials. Usually acting on some relatively small local site, they approximate the system's energy function through some explicit, physically-motivated function which is fitted to the problem though choice of some set of—also empirical—parameters. Empirical potentials have a very long history, going back to 1924 when Sir John Edward Lennard-Jones proposed a function describing the interaction between a pair of neutral atoms or molecules [42]. Today, an enormous amount of such potentials is available and in active use; important examples include the very popular *embedded atom method* [21] and *bond order potentials* like the Tersoff potential [59], the Brenner potential [14] or ReaxFF [60]. Software packages like LAMMPS [51] or ATK-ForceField [55] provide large collections of ready-to-use potentials. Two examples, the simple *Lennard-Jones potential* mentioned above and the relatively complex *modified embedded atom method*, are presented in Section 3.2.1 in greater detail. Their core benefit is the fact that their analytical definition allows for highly efficient implementations which scale at $\mathcal{O}(N)$ with a sufficiently small constant; they can therefore be used to inspect the properties of systems of all relevant sizes and in all relevant timescales. On the other hand, the accuracy and transferability of such potentials is generally limited, both by the fact that the functions they are composed from are approximations and the fact that there is no definite way to find the best set of parameters.

In more recent times, parameterless non-empirical potentials based on machine learning have been considered. These are often universal approximators like neural networks which are "trained on" or fitted to some data set of systems with known properties, which can for example be created with the *ab-initio* methods mentioned above. This removes the need to chose physically motivated functions and parameters as for empirical potentials, replacing them with an unbiased purely mathematical fitting procedure. Two fundamental papers in this endeavor are using a Gaussian kernels and using neural networks [4, 10]. We will investigate their methods in more detail in Section 3.2.2. Machine learning potentials have been applied for example by learning on-the-fly [44] or active learning [52]. Thanks to their general nature, machine learning potentials can theoretically—i.e. given enough training data—approximate the exact potential function to arbitrary precision. They are thus systematically improvable. In contrast, their very general form usually requires training on fairly big data sets before yielding good results, thus demanding a large, but reusable, computational effort upfront. Furthermore, while they are orders of magnitudes faster than methods like DFT and exhibit the desired linear scaling, they usually cannot compete with classical potentials in terms of computational performance.

Computational material science problems of even larger scales will not be discussed in this thesis. Beyond the point of our mesoscale setting, one quickly finds oneself in macroscopic territory where individual atoms are no longer of interest and which is governed by macroscopic physical equations. Such problems usually fall into the realm of continuum mechanics.

Outline In Section 2, we give an overview of the quantum mechanical fundamentals that govern the evolution of systems of atoms, and present the Born-Oppenheimer approximation, which allows us to separate the electrons and the nuclei in our computations. We also investigate the tight-binding model, a further approximation to the complex quantum model.

In Section 3, the basics of interatomic potentials is introduced. Usually acting on some local neighborhood of a given atom, they form the basis for empirical and machine learning potentials used for mesoscale systems. Two examples of each will be discussed in detail.

Section 4 will exhibit a new approach to machine learning potentials introduced in [56]. Polynomials will be used to describe atomic neighborhoods, in a manner that guarantees that chemically equivalent environments will receive the same description. How to construct such polynomials will be explained in Section 4.1. In Sections 4.2 and 4.3, it will be proven that these polynomials can indeed approximate the tight-binding model. Some additional work is required to to make the evaluation of the polynomials sufficiently efficient; this is covered in Section 4.4.

In Section 5, the polynomials are used to construct a complete machine learning potential whose performance is examined with numerical experiments. Lastly, Section 6 summarizes the results of this thesis.

2 Many-body quantum mechanics

In this section, we give a short introduction to the quantum mechanics needed to describe many-body systems of electrons and nuclei. We discuss the *Born-Oppenheimer approximation* which allows us to handle the nuclei and the electrons separately and the tight-binding model, which forms the foundation of interatomic potentials acting on atomic neighborhoods or "sites". Sections 2.1 and 2.2 closely follow the description in [34].

2.1 Basic principles

2.1.1 Notations and postulates

Quantum mechanics are built upon *wave-functions* acted on by linear Hermitian operators. The wave-functions are L^2 functions of system parameters and time, and describe the entire system. The Hermitian operators acting on the wave-functions correspond to physical observables like position, momentum and energy.

In this thesis, the system parameters are the positions of the electrons and nuclei which constitute the system. Using Latin indices for electronic variables and Greek indices for nuclear variables, the system parameters are written as $\{\{r_i\}, \{r_\alpha\}\}$ and the wave-function is denoted by $\psi(\{r_i\}, \{r_\alpha\}, t)$. Spin is ignored in this thesis as it is not relevant to our primary use case.

We use the *bra-ket* notation due to Dirac [23], with *bras* $\langle\psi|$ and *kets* $|\psi\rangle$ representing state-vectors from some complex Hilbert space, and the corresponding *bra-ket* inner product $\langle\psi|\phi\rangle$. This notation respects that different wave-functions ψ can represent the same state-vector $|\psi\rangle$. In particular, two non-zero state-vectors that differ only by multiplication with a constant describe the same state. We thus assume our state-vectors are normalized, i.e.

$$\langle\psi|\psi\rangle = \|\psi\|_{L^2} = 1.$$

For some observable O, we write $\hat{O}$ for its corresponding operator. It acts on a state vector $|\psi\rangle$ to produce a different state vector $|\phi\rangle$ (which may not be normalized):

$$\hat{O}\,|\psi\rangle = |\phi\rangle$$

Each operator has some set of normalized eigenstates $\{|\chi_n\rangle\}_{n\in\mathbb{N}}$ with corresponding eigenvalues λ_n, i.e.

$$\hat{O}\,|\chi_n\rangle = \lambda_n\,|\chi_n\rangle\,.$$

All eigenvalues of Hermitian operators are real.

For some system in state $|\psi\rangle$, the postulates of quantum mechanics state that

- the outcome of a measurement of some observable is always an eigenvalue λ_n of its corresponding operator
- immediately following a measurement, the state-vector collapses to a state from the span of the eigenstates corresponding to λ_n
- the probability of measuring λ_n is given by

$$\mathbb{P}(\lambda_n) = \sum |\langle \chi_i | \psi \rangle|^2,$$

where the $\langle \chi_i|$ form a normed basis of the eigenspace of λ_n.

2.1.2 Stationary states and the Schrödinger equation

Between measurements, the state-vector evolves in time according to the time-dependent Schrödinger equation:

$$\hat{H} |\psi\rangle = i \frac{\partial}{\partial t} |\psi\rangle . \tag{2.1}$$

In the non-relativistic case, the *Hamiltonian* $\hat{H}$ for our systems of nuclei and electrons is given by

$$\begin{aligned} \hat{H} = -\frac{1}{2} \sum_i \Delta_i - \frac{1}{2} \sum_\alpha \frac{1}{m_\alpha} \Delta_\alpha - \sum_i \sum_\alpha \frac{Z_\alpha}{|r_i - r_\alpha|} \\ + \frac{1}{2} \sum_i \sum_{j \neq i} \frac{1}{|r_i - r_j|} + \frac{1}{2} \sum_\alpha \sum_{\beta \neq \alpha} \frac{Z_\alpha Z_\beta}{|r_\alpha - r_\beta|}, \end{aligned} \tag{2.2}$$

where m_α is the mass of a nucleus and Z_α is its atomic number. The first two terms represent the kinetic energies of the electrons and nuclei, respectively; the remaining three correspond to Coulomb interaction energies between the particles. Relativistic effects are of no interest in our setting because they are relevant only for very heavy atoms, which are not considered in this work.

The eigenvalue equation for the Hamiltonian yields the time-independent Schrödinger equation

$$\hat{H} |\psi\rangle = E |\psi\rangle , \tag{2.3}$$

where the eigenvalue E denotes the total energy of the system. Here, the time-dependence of the state takes the simple form

$$\psi(\{r_i\}, \{r_\alpha\}, t) = \tilde{\psi}(\{r_i\}, \{r_\alpha\}) \Theta(t), \tag{2.4}$$

which yields the following system:

$$\hat{H} \tilde{\psi}(\{r_i\}, \{r_\alpha\} = E \tilde{\psi}(\{r_i\}, \{r_\alpha\}), \tag{2.5}$$

$$i \frac{\mathrm{d}}{\mathrm{d}t} \Theta(t) = E \Theta(t). \tag{2.6}$$

Solving the ordinary differential equation (2.6) gives us the form of the eigenfunctions of the Hamiltonian as

$$\psi(\{r_i\},\{r_\alpha\},t) = \tilde{\psi}(\{r_i\},\{r_\alpha\})\exp(-iEt).$$

The eigenstates of the Hamiltonian are also known as *stationary states*, because the expectation values of time-independent operators for these states are constant in time:

$$\begin{aligned}\langle\psi|\hat{O}|\psi\rangle &= \int \psi^*(\{r_i\},\{r_\alpha\})\hat{O}\psi(\{r_i\},\{r_\alpha\})\prod_j \mathrm{d}r_j \prod_\beta \mathrm{d}r_\beta \\ &= \int \tilde{\psi}^*(\{r_i\},\{r_\alpha\})e^{iEt}\hat{O}\tilde{\psi}(\{r_i\},\{r_\alpha\})e^{-iEt}\prod_j \mathrm{d}r_j \prod_\beta \mathrm{d}r_\beta \\ &= \left\langle\tilde{\psi}\middle|\hat{O}\middle|\tilde{\psi}\right\rangle\end{aligned}$$

As the energy of the system is our core interest, we will only deal with the time-independent state $|\tilde{\psi}\rangle$ from here and ignore the exponential time dependency.

2.2 Born-Oppenheimer approximation

The Born-Oppenheimer approximation [12] is a strategy to separate the motion of the electrons and the nuclei in the Schrödinger equation. As electrons and protons carry the same charge (with different signs), the forces acting on them and thus the changes in their momentum are of the same order of magnitude. It thus makes sense to assume that their actual momenta are of the same order of magnitude, too. Due to the huge difference in mass between nuclei and electrons, we can conclude that the nuclei have much smaller velocity than the electrons. As we are interested in the movement of nuclei in this work, the electrons can be assumed to instantaneously relax to their ground state when looking at the time-scale relevant for nuclei movement. This lets us consider the nuclei to be stationary when solving the time-independent Schrödinger equation (2.3) resulting from the Hamiltonian given in eq. (2.2) for the ground state, which can then be used to obtain the forces on the nuclei and thus update their positions using e.g. a Verlet time step for the Newton laws of motion applied to the nuclei [28].

To this end, we decompose the eigenstate of the Hamiltonian into an electronic part and a nuclear part as

$$\tilde{\psi}(\{r_i\},\{r_\alpha\}) = \psi_{\{r_\alpha\}}(\{r_i\})\phi(\{r_\alpha\}),$$

where $\psi_{\{r_\alpha\}}(\{r_i\})$ is a wave function in $\{r_i\}$ satisfying the time-independent

Schrödinger equation for electrons in a static array of nuclei:

$$\underbrace{\left[-\frac{1}{2}\sum_i \Delta_i - \sum_i \sum_\alpha \frac{Z_\alpha}{|r_i - r_\alpha|} + \frac{1}{2}\sum_i \sum_{j\neq i} \frac{1}{|r_i - r_j|}\right]}_{=:\hat{H}^{(e)}} \psi_{\{r_\alpha\}}(\{r_i\}) = \mathcal{E}^{(e)}_{\{r_\alpha\}} \psi_{\{r_\alpha\}}(\{r_i\})$$

Applying the whole Hamiltonian to $\tilde{\psi}(\{r_i\}, \{r_\alpha\})$:

$$\begin{aligned}
&\hat{H}\tilde{\psi}(\{r_i\}, \{r_\alpha\}) \\
&= \left[-\frac{1}{2}\sum_\beta \frac{1}{m_\beta}\Delta_\beta + \hat{H}^{(e)} + \frac{1}{2}\sum_\beta \sum_{\gamma\neq\beta} \frac{Z_\beta Z_\gamma}{|r_\beta - r_\gamma|}\right] \tilde{\psi}(\{r_i\}, \{r_\alpha\}) \\
&= \psi_{\{r_\alpha\}}(\{r_i\}) \left[-\frac{1}{2}\sum_\beta \frac{1}{m_\beta}\Delta_\beta + \mathcal{E}^{(e)}_{\{r_\alpha\}} + \frac{1}{2}\sum_\beta \sum_{\gamma\neq\beta} \frac{Z_\beta Z_\gamma}{|r_\beta - r_\gamma|}\right] \phi(\{r_\alpha\}) \\
&\quad -\frac{1}{2}\sum_\beta \frac{1}{m_\beta}\left[2\boldsymbol{\nabla}_\beta \phi(\{r_\alpha\}) \cdot \boldsymbol{\nabla}_\beta \psi_{\{r_\alpha\}}(\{r_i\}) + \phi(\{r_\alpha\})\Delta_\beta \psi_{\{r_\alpha\}}(\{r_i\})\right]
\end{aligned} \tag{2.7}$$

In the equation above, $\mathcal{E}^{(e)}_{\{r_\alpha\}}$ is called the *adiabatic* contribution of the electrons to the energy. The contributions from the last line in eq. (2.7) are small, as can be demonstrated with pertubation theory: the first order correction from the non-adiabatic contribution has the form:

$$\begin{aligned}
&-\int \Psi^*_{\{r_\alpha\}}(\{r_i\})\phi^*(\{r_\alpha\}) \\
&\qquad \sum_\gamma \frac{1}{m_\gamma}\left[\boldsymbol{\nabla}_\gamma \phi(\{r_\alpha\}) \cdot \boldsymbol{\nabla}_\gamma \psi_{\{r_\alpha\}}(\{r_i\})\right] \prod_j \mathrm{d}r_j \prod_\beta \mathrm{d}r_\beta \\
&= -\sum_\gamma \int \phi^*(\{r_\alpha\}) \nabla_\gamma \phi(\{r_\alpha\}) \\
&\qquad \cdot \underbrace{\left[\int \psi^*_{\{r_\alpha\}}(\{r_i\}) \nabla_\gamma \psi_{\{r_\alpha\}}(\{r_i\}) \prod_j \mathrm{d}r_j\right]}_{=:I} \prod_\beta \mathrm{d}r_\beta \,,
\end{aligned}$$

and the integral I evaluates to

$$I = \frac{1}{2}\nabla_\gamma \int |\psi_{\{r_\alpha\}}(\{r_i\})|^2 \prod_j \mathrm{d}r_j = \frac{1}{2}\nabla_\gamma(1) = 0$$

because the normalization of the electronic wave-function is constant under the movement of nuclei. Thus the first order contribution vanishes. The second term in the last line of eq. (2.7) will be maximal when the electrons are

tightly bound to their nuclei (see Section 2.3), i.e. $\psi_{\{r_\alpha\}}(\{r_i\}) = \psi(\{u_{(i,\alpha)}\})$ with $u_{(i,\alpha)} = r_i - r_\alpha$. With that, we get for the first order correction:

$$
\begin{aligned}
&-\int \psi^*(\{u_{(i,\alpha)}\})\phi^*(\{r_\alpha\}) \sum_\gamma \frac{1}{2m_\gamma} \left[\phi(\{r_\alpha\})\Delta_\gamma \psi(\{u_{(i,\alpha)}\})\right] \prod_j \mathrm{d}r_j \prod_\beta \mathrm{d}r_\beta \\
&= -\sum_\gamma \frac{1}{2m_\gamma} \left[\int |\phi(\{r_\alpha\})|^2 \prod_\beta \mathrm{d}r_\beta\right] \\
&\qquad \left[\int \psi^*(\{u_{(i,\alpha)}\})\Delta_\gamma \psi(\{u_{(i,\alpha)}\}) \prod_{(j,\beta)} \mathrm{d}u_{(j,\beta)}\right] \\
&= -\sum_{(k,\gamma)} \frac{1}{m_\gamma} \int \psi(\{u_{(i,\alpha)}\}) \frac{1}{2} \Delta_{(k,\gamma)} \psi(\{u_{(i,\alpha)}\}) \prod_{(j,1b)} \mathrm{d}u_{(j,\beta)}
\end{aligned}
$$

This is of the same order of magnitude as the kinetic energy multiplied by the ratio between atom and electron masses. As this factor is also small (in the order of 10^{-4} or 10^{-5}), we can also neglect the contributions from this term.

With that, eq. (2.7) gives us a Schrödinger equation of the form

$$
\left[-\frac{1}{2}\sum_\beta \frac{1}{m_\beta}\Delta_\beta + \mathcal{E}^{(e)}_{\{r_\alpha\}} + \frac{1}{2}\sum_\beta \sum_{\gamma\neq\beta} \frac{Z_\beta Z_\gamma}{|r_\beta - r_\gamma|}\right] \phi(\{r_\alpha\}) = \mathcal{E}\phi(\{r_\alpha\})
$$

for the nuclei function $\phi(\{r_\alpha\})$. We have thus successfully separated the movement of the electrons from that of the nuclei; this allows us simulate the movement of nuclei with Newtonian mechanics as described earlier in this subsection. With that, we can perform simulations for molecule dynamics or search for local minima on the potential surface generated by the position of the nuclei.

2.3 Tight-binding model

There are several variants of what is referred to as the *tight-binding* model or method. Generally, those methods are based on the use of a minimal local basis $\{\phi_{\alpha,l}(r - r_\alpha)\}$ of local orbitals, where l enumerates a small number of orbitals of the atom α [26, 16]. To this end, one considers a Hamiltonian matrix

$$
\Big(\mathcal{H}(\{r_\alpha\}\Big)^{lk}_{\beta,\gamma} = \int \phi_{\beta l}(r - r_\beta)\hat{H}\phi_{\gamma k}(r - r_\gamma)\,\mathrm{d}r
$$

with entries

$$
h^{lk}_{\beta,\gamma}(\{r_\alpha\}) \coloneqq \Big(\mathcal{H}(\{r_\alpha\}\Big)^{lk}_{\beta,\gamma}.
$$

In this thesis, we will present an *empirical tight-binding* model as originally described in [16, 17]. We will later see convergence of the moment tensor potential to this model in Section 4.3 as shown in [56].

From now on, we restrict our interest to systems of finitely many atoms, i.e.

$$\{r_\alpha\} = \{r_1, \dots, r_N\};$$

however the following can also be applied to periodic systems using Bloch waves [26]. The concrete tight-binding model we will use is a *two-center* tight-binding model, in which the matrix elements have the form

$$h_{\beta,\gamma}(\{r_\alpha\}) = \begin{cases} \varphi(r_\gamma - r_\beta) & \beta \neq \gamma \\ 0 & \beta = \gamma, \end{cases}$$

with $\varphi(r)$ a hopping term (see *hopping integral* in [26]) chosen empirically such that $\varphi(r) = 0$ if $\|r\| > R_c$ for some $R_c > 0$. Here, we have dropped the indices for the orbitals following [17] and set the on-site terms, i.e. the diagonal elements of $\mathcal{H}$, to 0, following [56].

The total energy of the system is now written as the sum of the *band energy* E^{band} and some *repulsive energy* E^{rep}. The repulsive energy can be handled using simple pair potentials. It is only mentioned here for completeness' sake and is of no further interest in this work. The band energy, henceforth referred to simply as E, is defined as follows: let ε_i the eigenvalues of $\mathcal{H}$ and ψ_i the corresponding normalized eigenvectors. Then

$$E(\{r_\alpha\}) = \sum_{i=1}^{N} \varepsilon_i f(\varepsilon_i) = \sum_{i=1}^{N} \mathfrak{f}(\varepsilon_i),$$

where

$$f(\varepsilon) = \left(1 + \exp\left(\frac{\varepsilon - \mu}{k_B T}\right)\right)^{-1}$$

is the Fermi-Dirac function and

$$\mathfrak{f}(\varepsilon) \coloneqq \varepsilon f(\varepsilon)$$

for ease of notation. The chemical potential μ is taken to be 0 in our case, still following [56]. The physical constants k_B and T are the Boltzmann constant and the electronic temperature respectively. We can then apply a common idea (e.g. [26]) and write the energy as sum of the local contributions of each site:

$$E(\{r_\alpha\}) = \sum_{i=1}^{N} \mathfrak{f}(\varepsilon_i) = \sum_{i=1}^{N} \mathfrak{f}(\varepsilon_i) \sum_{j=1}^{N} (\psi_i)_j^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathfrak{f}(\varepsilon_i)(\psi_i)_j^2,$$

i.e. the band energy is decomposed as

$$E(\{r_\alpha\}) = \sum_{i=1}^{N} V_i(\{r_\alpha\}),$$

with

$$V_i(\{r_\alpha\}) = \sum_{j=1}^{N} \mathfrak{f}(\varepsilon_j)(\psi_j)_i^2.$$

We call $V_i(\{r_\alpha\})$ the *site energy* at atom (site) i. In [56] and in the rest of this thesis, the alternate representation

$$V(\{r_\alpha\}) \coloneqq (\mathfrak{f}(\mathcal{H}))_{1,1}$$

of the site energy is used. The evaluation of the Fermi-Dirac function of a matrix is to be understood as defined in [35].

3 Interatomic potentials

3.1 Notation and basic properties

We now move from the quantum world to concrete methods for our mesoscale setting. Since the position of electrons is no longer relevant, we can elide them from the notation for clarity. With that, an atomic configuration is given by

$$X \coloneqq \left\{x_1^{(Z_1)}, \dots, x_N^{(Z_N)}\right\}, \tag{3.1}$$

where $x_i \in \mathbb{R}^3$ denotes the atom's Cartesian coordinates and (Z_i) its atomic number. For ease of notation, we will usually omit the latter.

Hence, we will assume the tight-binding approximation can be applied and write the total potential energy of the system given by X as

$$E(X) = \sum_{i=1}^{N} V_i(Dx_i), \tag{3.2}$$

with the local neighborhood of some atom i given by

$$Dx_i \coloneqq \{x_j - x_i \mid \|x_j - x_i\| < R_{\text{cut}}, i \neq j\}. \tag{3.3}$$

The finite cut-off radius assumes that the interaction between particles that are "far apart" is negligible. This is assumed to be true in our setting and enables efficient implementations, for example using linked-cell techniques. Note that long-range interactions between particles can exist, for example in charged systems. Algorithms and solvers for the treatment of such interactions exist, but are beyond the scope of this work.

For the frequently-needed distance vector from atom i to atom j, we write $u_{ij} \coloneqq x_j - x_i$ and $r_{ij} \coloneqq \|u_{ij}\|$ for its length, dropping the first index when the central atom is fixed.

We note some intuitive but important properties of the site energy functions V_i:

- Invariance under permutation of chemically equivalent atoms, i.e. for any pair of atoms $x_i^{Z_i}$, $x_j^{Z_j}$ with $Z_i = Z_j$ and $i < j$, we have

$$\begin{aligned} &V(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_N) \\ &\quad = V(x_1, \dots, x_{i-1}, x_j, x_{i+1}, \dots, x_{j-1}, x_i, x_{j+1}, \dots, x_N) \end{aligned}$$

- Invariance under Euclidean transformation of the atoms' positions: for any such transformation T, it holds

$$V(u_1, \dots, u_n) = V(T(u_1), \dots, T(u_n)).$$

- Smoothness with respect to atoms entering or leaving the neighborhood at R_{cut}.

As an additional property for an approximation $\tilde{V}_i$ of V_i, $\tilde{V}_i \in C^2$ when all r_{ij} are strictly positive is desired in order to guarantee conservation of energy in the model.

3.2 Existing approaches

We now go on to describe some empirical and non-empirical approaches to approximating the energy function $E(X)$.

3.2.1 Empirical potentials

The classical way of approximating the general energy term is using *empirical potentials*. Those potentials are usually written as sums of site energies $V(Dx_i)$ as defined in eq. (3.2). Normally, the site energies are represented as sums of pair-term, although sometimes as triple- or quadruple-term contributions. This also holds true for the two examples below. Note that although the use of pair- or even triple-terms suggest a complexity of $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$ in the number of particles, the overall complexity of the evaluation is $\mathcal{O}(N)$ because we assume that the number of atoms in each neighborhood is bounded. For demonstration purposes, we will present a simple and a more complex example of such potentials:

Lennard-Jones Potential First introduced in [42], it is one of the simplest empirical models. It has the form

$$E(X) = \sum_{i<j} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right],$$

where i and j run over all particles in the configuration. The Lennard-Jones potential is a classical pair potential using two adjustable parameters ε and σ per pair of particle types involved in the simulation. They influence the depth of the potential well and its position, respectively. This is clearly invariant under Euclidean transformations and under permutation of chemically equivalent atoms. In order to achieve smoothness at some finite R_{cut}, the potential can be multiplied with some taper function that smoothly goes from 1 to 0 on some interval $[R_{\text{taper}}, R_{\text{cut}}]$. Due to its simplicity, it can be evaluated extremely efficiently. However, the fact that it only allows for two adjustable parameters is rather restrictive; while the Lennard-Jones potential describes noble gases fairly well, many systems cannot be described sufficiently accurate with this model.

Modified Embedded Atom Method A more complex approach that is applicable to a large number of elements is the *Modified Embedded Atom Method* (MEAM), first described in [8]. It has been further extended, for

example to include second-nearest neighbor interaction in [41]. Here, we will present the formalism as described in [25]. The description is rather lengthy compared to the Lennard-Jones potential and its details are not needed in the rest of this thesis. However, it is reproduced here at length to demonstrate the big variety of empirical potentials at our disposal. In particular, it serves as a good example for a potential with a relatively large set of adjustable parameters. The MEAM energy for a system has the form

$$E(X) = \sum_i \left[F_i(\bar{\rho}_i) + \frac{1}{2} \sum_{i \neq j} \Phi_{ij}(r_{ij}) \right].$$

The function $F_i(\bar{\rho}_i)$ is some embedding energy term, while $\Phi_{ij}(r_{ij})$ is a normal pair potential. Following the notation in [25], we will use Latin subscripts i, j and k to denote particles and Greek subscripts α, β and γ for entries of vectors and matrices. The embedding function has the form

$$F_i(\bar{\rho}_i) = A_i E_i^0 \bar{\rho}_i \log \bar{\rho}_i,$$

with the element-dependent sublimation energy E_i^0 and some adjustable parameter A. $\bar{\rho}_i$ is background electron density and is given by

$$\bar{\rho}_i = \frac{\bar{\rho}_i^{(0)}}{\rho_i^0} G_i(\Gamma_i),$$

with

$$\Gamma_i = \sum_{k=1}^{3} t_i^{(k)} \left(\frac{\bar{\rho}_i^{(k)}}{\bar{\rho}_i^{(0)}} \right).$$

The function $G(\Gamma)$ is chosen empirically; common choices are $G(\Gamma) = \sqrt{1+\Gamma}$ and $G(\Gamma) = \exp(\Gamma/2)$. The densities $\bar{\rho}_i^{(k)}$ for $k = 0, \ldots, 3$ are given by

$$\begin{aligned}
\bar{\rho}_i^{(0)} &= \sum_{j \neq i} \rho_j^{a(0)}(r_{ij}) S_{ij} \\
\left(\bar{\rho}_i^{(1)}\right)^2 &= \sum_{\alpha=1}^{3} \left[\sum_{j \neq i} \rho_j^{a(1)} \frac{u_{ij\alpha}}{r_{ij}} S_{ij} \right]^2 \\
\left(\bar{\rho}_i^{(2)}\right)^2 &= \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} \left[\sum_{j \neq i} \rho_j^{a(2)} \frac{u_{ij\alpha} u_{ij\beta}}{r_{ij}^2} S_{ij} \right]^2 - \frac{1}{3} \left[\sum_{j \neq i} \rho_j^{a(2)} r_{ij} S_{ij} \right]^2 \\
\left(\bar{\rho}_i^{(3)}\right)^2 &= \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} \sum_{\gamma=1}^{3} \left[\sum_{j \neq i} \rho_j^{a(3)} \frac{u_{ij\alpha} u_{ij\beta} u_{ij\gamma}}{r_{ij}^3} S_{ij} \right]^2 - \frac{3}{5} \sum_{\alpha=1}^{3} \left[\sum_{j \neq i} \rho_j^{a(3)} \frac{u_{ij\alpha}}{r_{ij}} S_{ij} \right]^2,
\end{aligned}$$

where S_{ij} is a screening function that is explained later in this section and the composition-dependent electron density scaling ρ_i^0 is defined as

$$\rho_i^0 = \rho_{i0} Z_{i0} G_i(\Gamma_i^{\text{ref}}).$$

In this expression, ρ_{i0} is an element-dependent density scaling, Z_{i0} is the first-neighbor coordination of the reference system and Γ_i^{ref} is given by

$$\Gamma_i^{\mathrm{ref}} = \sum_{k=1}^{3} t_i^{(k)} \frac{s_i^{(k)}}{Z_{i0}^2},$$

where $s_i^{(k)}$ are shape factors depending on the reference structure for atom i. The atomic electron densities $\rho_i^{a(k)}$ are defined as

$$\rho_i^{a(k)}(r_{ij}) = \rho_{i0} \exp\left[-\beta_i^{(k)}\left(\frac{r_{ij}}{r_i^0} - 1\right)\right],$$

and the average weighting factors as

$$t_i^{(k)} = \frac{1}{\bar{\rho}_i^{(0)}} \sum_{j \neq i} t_{0,j}^{(k)} \rho_j^{a(0)} S_{ij},$$

with the nearest-neighbor distance in the reference structure r_i^0 and more element-dependent parameters $\beta_i^{(k)}$ and $t_{0,j}^{(k)}$.

The term $\Phi_{ij}(r_{ij})$ is a simple pair potential (after the screening has been computed) and is given by

$$\begin{aligned}
\Phi_{ij} &= \bar{\Phi}_{ij}(r_{ij}) S_{ij} \\
\bar{\Phi}_{ij}(r_{ij}) &= \frac{1}{Z_{ij0}}[2E_i^{\mathrm{u}}(r_{ij}) - F_i(\hat{\rho}_i(r_{ij})) - F_j(\hat{\rho}_j(r_{ij}))] \\
E_i^{\mathrm{u}}(r_{ij}) &= -E_{ij}^0(1 + \alpha_{ij}^*(r_{ij})) \exp\Big(-\alpha_{ij}^*(r_{ij})\Big) \\
\alpha_{ij}^*(r_{ij}) &= \alpha_{ij}\left(\frac{r_{ij}}{r_{ij}^0} - 1\right).
\end{aligned}$$

The remaining undefined quantities above, namely E_{ij}^0, r_{ij}^0 and α_{ij}, are again element-dependent parameters. Last, we define the screening function as

$$\begin{aligned}
S_{ij} &= \bar{S}_{ij} f_c\left(\frac{R_{\mathrm{cut}} - r_{ij}}{\Delta r}\right) \\
\bar{S}_{ij} &= \prod_{k \neq i,j} S_{ikj} \\
S_{ikj} &= f_c\left(\frac{C_{ikj} - C_{\mathrm{min},ikj}}{C_{\mathrm{max},ikj} - C_{\mathrm{min},ikj}}\right) \\
C_{ikj} &= 1 + 2\frac{r_{ij}^2 r_{ik}^2 + r_{ij}^2 r_{jk}^2 - r_{ij}^4}{r_{ij}^4 - (r_{ik}^2 - r_{jk}^2)^2} \\
f_c(x) &= \begin{cases} 1 & x \geq 1 \\ \left[1 - (1-x)^4\right]^2 & 0 < x < 1 \\ 0 & x \leq 0 \end{cases}
\end{aligned}$$

The cutoff radius R_{cut} and Δr, which determines the length of the region on which the radial cutoff is smoothed near $r = R_{\text{cut}}$, are type-independent parameters, while the parameters $C_{\text{min},ikj}$ and $C_{\text{max},ikj}$ depend on the particle types of the involved atoms. Note how this screening function makes the MEAM a three-body potential, in contrast to the two-body Lennard-Jones potential presented above. Unsurprisingly, this more complex potential cannot match simpler ones such as Lennard-Jones in speed of computation. However the large amount of adjustable parameters (in the MEAM variant presented here, i.e. 12 per element plus the parameters for the pair-term Φ and the screening function S) allows the MEAM to be fitted for a large variety of applications. While this makes the MEAM a very powerful tool, actually fitting it for a new set of problems requires significant effort. Indeed, finding new and improved parameter sets for the MEAM is still subject of recent and ongoing research, i.e. [1, 38], despite the popular second-nearest neighbor MEAM being published almost two decades ago at the time of writing.

The key advantage of potentials of this form is the possibility to evaluate them with massively parallel, high-performance implementations, for example using the linked-cell method [29] which can be used to simulate systems of very big numbers of particles (from the view of our mesoscale problems) [37]. On the other hand, those potentials are *parametric* (using the terminology from [6]) by nature. This means that they rely on some fixed set of parameters that have to be somehow determined "by hand"; in the sense that there is not systematic way to improve an existing parameter set or find an optimal one in general. While there exist software solutions as presented for example in [24] that help in fitting an empirical potential to some dataset, caveats like having to choose some starting point for the non-convex problem of minimizing the error in forces and energy with respect to the potential parameter still apply. Also, even an optimal parameter set for some potential may not be able to produce results of sufficient accuracy. This core limitation leads to the need for non-parametric potentials as presented in the following section.

3.2.2 Machine learning potentials

The limitations of empirical potentials as described in Section 3.2.1 lead to the need of non-parametric alternatives. In current research, this problem has been approached using machine learning techniques such as kernel learning or neural networks.

The first key ingredient for any machine learning potential are so-called *descriptors* or *fingerprints* which map atomic positions to a tuple of real numbers; i.e. a descriptor is a tuple of scalar functions. This make their investigation an active area of recent research [3, 19]. On these tuples, some regression model can be applied to obtain the final potential. As this is a core

concept of machine learning potentials and the analysis of the relative merits of different approaches is a broad field, an in-depth description of the topic is beyond the scope of this thesis. For instance, in general different strategies have to be applied for describing atomic environments in condensed matter compared to fingerprinting molecules. Here, we will present a short overview of the basic requirements for descriptors based on [6] before we move on to some concrete examples in actual use.

There are two important conditions for useful descriptors. Firstly, they should guarantee the invariants stated in Section 3.1. Of course, the requirement of being smooth with regard to particles leaving the neighbor at R_{cut} only makes sense for descriptors that describe atomic environments, rather than whole molecules. Second, the descriptors should be true to the ensemble they are applied to, in the sense that they should map chemically different inputs to different outputs. As such, we say a descriptor $(q_1, \dots, q_M)$ is complete if it is a one-to-one mapping from the space of atomic environments into $\mathbb{R}^n$. We call a descriptor over-complete if a proper subset of $(q_1, \dots, q_M)$ is complete. In this case, all chemically different environments will still map to different values, but the same may also happen to chemically equivalent environments.

From here on, we will focus on descriptors for atomic environments as those are needed to approximate the site energy V from eq. (3.2). For a fixed central atom with n neighbors with relative positions $u_1, \dots, u_n$, a simple first idea for a descriptor is the symmetric matrix

$$\Sigma \coloneqq \begin{pmatrix} u_1 \cdot u_1 & u_1 \cdot u_2 & \cdots & u_1 \cdot u_n \\ u_2 \cdot u_1 & u_2 \cdot u_2 & \cdots & u_2 \cdot u_n \\ \vdots & \vdots & \ddots & \vdots \\ u_n \cdot u_1 & u_n \cdot u_2 & \cdots & u_n \cdot u_n \end{pmatrix},$$

which forms an over-complete array of invariants with respect to Euclidean transformations due to results from [63]. Of course, it is by itself not a suitable descriptor because it lacks invariance with respect to permutation. For example, swapping the described order of the first and the second atom also swaps the corresponding rows and columns:

$$\begin{pmatrix} u_2 \cdot u_2 & u_2 \cdot u_1 & u_1 \cdot u_3 & \cdots & u_2 \cdot u_n \\ u_1 \cdot u_2 & u_1 \cdot u_1 & u_2 \cdot u_3 & \cdots & u_1 \cdot u_n \\ u_3 \cdot u_2 & u_3 \cdot u_1 & u_3 \cdot u_3 & \cdots & u_1 \cdot u_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_n \cdot u_2 & u_n \cdot u_1 & u_n \cdot u_3 & \cdots & u_n \cdot u_n \end{pmatrix},$$

thus yielding a different description of a chemically equivalent environment. However, permutation invariance can be recovered by using functions of this matrix. Indeed, we will later see in Lemma 4.4 that our moment tensor basis

can be written in terms of its elements. Another, simpler way to recover permutation invariance while maintaining differentiability would be using the sorted eigenvalues of Σ. However, this is not only hard to compute for large n, it also loses completeness as the spectrum of Σ has dimension n, compared to the $3n-3$ degrees of freedom in the atomic environment modulo rotations.

We now move on to two popular examples of machine learning potentials; one based on Gaussian kernels and one based on neural networks.

Gaussian Approximation Potential The Gaussian approximation potential (GAP) was introduced by Bartók et al. in [7]. We will give a short overview based on this paper; a more thorough introduction including some words about practical implementations and the treatment of data noise can be found in [5].

The core idea of the GAP is to construct some permutation-invariant local atomic density from a neighborhood, represent it with 4D spherical harmonics and then apply Gaussian process regression [53]. For that, we fix a central atom i and define a local atomic density from its neighbors by

$$\rho_i(r) = \delta(r) + \sum_j \delta(r - r_j) f_{\text{cut}}(r_j), \tag{3.4}$$

where

$$f_{\text{cut}}(r) \coloneqq \begin{cases} \frac{1}{2}\Big[1 + \cos\Big(\frac{\pi r}{r_{\text{cut}}}\Big)\Big] & r < r_{\text{cut}} \\ 0 & r \geq r_{\text{cut}} \end{cases}$$

is a somewhat arbitrary cutoff function; any smooth function with compact support could be used here. The density is already invariant under permutation and translation. Invariance under rotation is achieved using the *bispectrum* [22] which can provide an almost one-to-one representation of the atomic neighborhood.

The atomic density is projected onto the surface of the four-dimensional unit sphere using the transformation

$$r \equiv \begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \begin{aligned} \phi &= \arctan(y/x) \\ \theta &= \arccos(z/\|r\|) \\ \theta_0 &= \|r\|/r_0 \end{aligned} \tag{3.5}$$

for some $r_0 > r_{\text{cut}}/\pi$. This has the advantage that the 4D surface contains all the information from the 3D spherical region inside the cutoff, including the radial dimension. Thus 4D spherical harmonics constitute a natural complete basis for the interior of the 3D sphere without the need for radial basis functions. Using Wigner matrices $U^j_{m'm}$ [61], the projection of the atomic density onto the 4D sphere can therefore be expanded in 4D spherical harmonics using coefficients

$$c^j_{m'm} = \left\langle U^j_{m'm} \middle| \rho \right\rangle,$$

dropping the atomic index i for clarity. The bispectrum built from these coefficients is given by

$$B_{j_1 j_2 j} = \sum_{m_1', m_1 = -j_1}^{j_1} \sum_{m_2', m_2 = -j_2}^{j_2} \sum_{m', m = -j}^{j} \left(c^{j}_{m' m}\right)^* C^{jm}_{j_1 m_1 j_2 m_2} C^{j m'}_{j_i m_1' j_2 m_2'} c^{j_1}_{m_1' m_1} c^{j_2}_{m_2' m_2},$$

where $C^{jm}_{j_1 m_1 j_2 m_2}$ are the ordinary Clebsch-Gordan coefficients. We write $\mathbf{b}_i$ for the three-index array $B_{j_1 j_2 j}$ that belongs to atom i and note that all its elements are invariant with respect to permutations of atoms and rotations of 4D, and thus also 3D space. In practice, the $\mathbf{b}_i$ are truncated by restricting $j, j_1, j_2 \leq J_{\max}$, which corresponds to a limit in the special resolution with which the neighborhood is described. We can now use the bispectrums $\mathbf{b}$ as descriptors and apply Gaussian process regression to interpolate the site energy. For that, we define the kernel

$$G(\mathbf{b}_i, \mathbf{b}_j) \coloneqq \exp\left[-\frac{1}{2} \sum_l \left(\frac{\mathbf{b}_{i,l} - \mathbf{b}_{j,l}}{\theta_l}\right)^2\right]$$

with some adjustable hyper-parameter $\{\theta_l\}$, where l indexes the bispectrum components. Then the site energy at atom i is given by

$$V(\mathbf{b}_i) = \sum_j \alpha_j G(\mathbf{b}_i, \mathbf{b}_j). \tag{3.6}$$

The kernel G is a *universal kernel* and can thus, loosely stated, be used to approximate any continuous function with a sum of the form in eq. (3.6)[57]. Unlike simple linear regression, Gaussian approximation uses a *covariance matrix*

$$C_{nn'} = \delta^2 G(\mathbf{b}_n, \mathbf{b}_{n'}) + \sigma^2 \mathbf{1},$$

where $\mathbf{1}$ is the identity matrix and δ and σ are two more hyper-parameters. With this, we obtain our coefficients $\alpha = (\alpha_n)_n$ as

$$\alpha = C^{-1} y,$$

where $y = (y_n)_n$ is the set of reference energies. This simple expression for α has been derived in [45]. We have thus arrived at a closed form for the site energy function with eq. (3.6).

The GAP is a popular candidate for a modern non-parametric potential, as it solves the issues of empirical potentials not being systematically improvable. However, expanding the atomic densities in spherical harmonics can be computationally expensive. Indeed, [4] notes that while GAP potentials are order of magnitudes faster to compute than Density Functional Theory, it is still about a hundred times more expensive than empirical potentials.

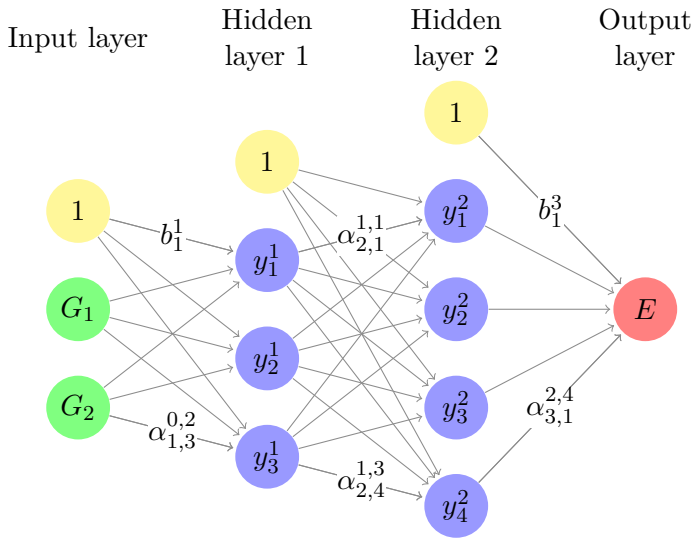

Figure 1: A simple feed-forward NN

Neural Network Potentials Neural networks have a long history, with the idea of using artificial neurons first proposed in 1943 by McCulloch and Pitts [46]. This idea was generalized to more complex neural networks in the following decades, eventually arriving at *feed-forward neural networks* (NN), which can represent every Borel-measurable function with arbitrary accuracy [36]. In the more recent past, the drastic increase in available computing power and further advances in the underlying theory made NN a powerful tool for a wide area of problems. This makes them a natural candidate for building a potential function, hence referred to as *neural network potential*. We will now give a short overview based on [11].

Feed-forward neural networks consist of nodes y_i^j called *artificial neurons* which are arranged in layers as shown in Figure 1. The number of layers and the number of neurons per layer define the analytic form of the NN. The goal is to construct a function from the input vector $\mathbf{G} = \{G_i\}$, which describes the atomic configuration, to the corresponding energy E. The hidden layers consisting of the nodes y_i^j have no physical meaning, they merely define the functional form of the NN. Their number and the number of nodes they contain determine the network's flexibility to approximate more complex functions. The arrows connecting the nodes represent the flow of information through the NN, and their *weights* $\alpha_{i,j}^{k,l}$ or b_i^j are the fitting parameters of the model. With this, the values in layer i are given as function of the values in layer $i-1$ as

$$y_i^j = f_i^j\left(x_i^j\right) \coloneqq f_i^j\left(b_i^j + \sum_{k=1}^{N_{j-1}} \alpha_{k,i}^{j-1,j} y_k^{j-1}\right), \tag{3.7}$$

where N_j is the number of neurons in layer j. Each node but the input nodes is connected to a bias node with constant value 1. They contribute with a bias weight b_i^j, acting as an adjustable offset to shift the linear combination if needed.

To the shifted linear combination, a so-called non-linear *activation function* f_i^j is applied in order to introduce non-linearity to the model. This is needed to allow the NN to approximate arbitrary measurable functions. Often, f_i^j is chosen to converge to small numbers for inputs of large magnitude; common choices are $f(x) = 1/(1+\exp(-x))$, $f(x) = \tanh(x)$ or, most popular, the *ReLU* function $f(x) = \max(0, x)$. In the output layer, a linear activation function is applied in order to not restrict the range of the output energy E.

The NN is evaluated layer by layer, starting by evaluating the first hidden layer from the input layer $\mathbf{G}$ using eq. (3.7). This is repeated until the output layer is reached, giving us, for example, the full functional form of the NN in Figure 1 as

$$E = f_i^3\left[b_i^3 + \sum_{k=1}^{4} \alpha_{k,1}^{2,3} f_k^2\left[b_k^2 + \sum_{j=1}^{3} \alpha_{j,k}^{1,2} f_j^1\left[b_j^1 + \sum_{i=1}^{2} \alpha_{i,j}^{0,1} G_i\right]\right]\right].$$

The choice of the NN's structure is very important: a NN with too many nodes and thus too much flexibility can cause spurious features to appear in the approximated potential energy surface, while too few neurons leave the network too restricted to properly represent it.

It now remains to apply this general technique to our atomistic problem setting. Directly feeding the Cartesian coordinates of the nuclei into a NN fails for two reasons. Firstly, the number of input nodes of some given NN is fixed, while adding or removing an atom would add or remove three input variables. As training a new NN for every interesting system size is neither practical nor useful, we need some way to transform systems of different size into a fixed size input vector $\mathbf{G}$. Furthermore, we cannot expect the NN to learn the aforementioned invariants with respect to Euclidean transformations and permutation of chemically equivalent atoms, so those invariants must already be encoded in $\mathbf{G}$. Additionally, handling the entire configuration in a single pass through a NN demands an unwieldy number of input nodes and, to provide the necessary flexibility, hidden nodes, making both the "training" (i.e. finding an optimal or at least good set of weight parameters $\alpha_{i,j}^{k,l}$ and b_i^j) as well as the evaluation of the trained network too expensive. To combat this last issue, we again move to site energy terms like in eq. (3.2) and apply machine learning with NN to the site energy function $V(Dx_i)$. We can then feed descriptors for each local environment into the neural net separately and recover the total energy as the sum of the site energies as presented in Figure 2 for four environments. As the site energy of an atom is not observable and thus cannot be extracted from quantum

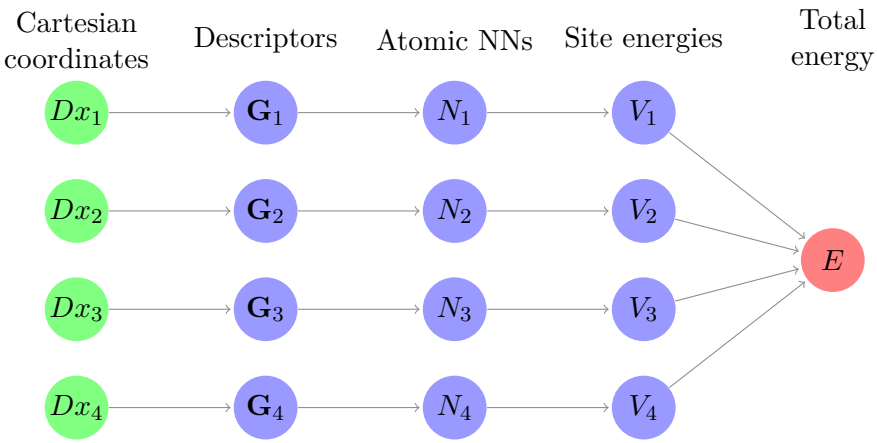

Figure 2: A neural network potential, in this example with four atomic sites

mechanical calculations directly, it is important to note that the local NNs can still be trained using the total energy as target quantity. If a neural network potential (NNP) is applied to systems containing chemically non-equivalent particles (i.e. atoms of different elements), one NN is trained for each species involved.

As for other machine learning potentials, finding a suitable descriptor for the atomic sites is the key challenge. In addition to the requirements for descriptors stated at the beginning of this section, descriptors for use in an NNP must also produce a fixed number of values independent of the number of atoms in the environment they describe, as the number of input nodes in the NNs is also fixed. This is here achieved by picking some fixed-size set of single-valued descriptor functions and using the vectors containing their values as the descriptors $\mathbf{G}_i$. Typically, they consist of 50 to 100 values.

To construct the descriptor functions, in this context called *symmetry functions*, we again start by picking a cutoff function f_{cut}. As for GAP above, different cutoff functions can be used, but

$$f_{\text{cut}}(r) \coloneqq \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{\pi r}{r_{\text{cut}}}\right)\right] & r < r_{\text{cut}} \\ 0 & r \geq r_{\text{cut}} \end{cases}$$

as used in GAP can also be employed here.

With this, we can define *atom-centered symmetry functions* as a set of descriptor functions. They explicitly take the angle

$$\Theta_{ijk} = \frac{u_{ij} \cdot u_{ik}}{r_{ij} r_{ik}}$$

into account to distinguish between different environments that are equal by absolute distances alone. Two types of such are angular functions are used,

namely[3]

$$G_i^1 = 2^{1-\zeta} \sum_{j \neq i} \sum_{k \neq i,j} \Big[(1 + \lambda \cos \Theta_{ijk})^\zeta \\ \exp\Big(-\eta(r_{ij}^2 + r_{ik}^2 + r_{jk}^2)\Big) f_{\text{cut}}(r_{ij}) f_{\text{cut}}(r_{ik}) f_{\text{cut}}(r_{jk}) \Big]$$

and

$$G_i^2 = 2^{1-\zeta} \sum_{j \neq i} \sum_{k \neq i,j} \Big[(1 + \lambda \cos \Theta_{ijk})^\zeta \exp\Big(-\eta(r_{ij}^2 + r_{jk}^2)\Big) f_{\text{cut}}(r_{ij}) f_{\text{cut}}(r_{ik}) \Big].$$

The parameter $\lambda \in \{-1, 1\}$ determines the location of the maximum of the cosine term, ζ controls the angular resolution, and η allows for scanning the angular distribution at various distances. The main difference between the two is that G^2 does not take r_{jk} into account, while G^1 only considers angles if all three pairwise distances are small enough. A more detailed description of these functions can be found in [11]. As distances and angles are invariant under Euclidean transformations and the sum guarantees invariance under permutation of equivalent atoms, functions of this type fulfill the requirements to be used as descriptors. Instead of using atom-centered symmetry functions, pair-centered symmetry functions or even bigger tuple-centered functions can also be used, but this is outside the scope of this thesis.

All that remains is to pick some set of parameters λ, ζ and η to produce the required set of scalar descriptor functions. Choosing the parameter set is currently an empirical process; it should yield functions that are not too similar, in order to avoid redundant information, and should be able to differentiate between chemically different environments. As noted about, a set of 50 to 100 such descriptor functions seems to produce a good fingerprint of the environment while keeping the required NNs sufficiently small.

It has been observed that NNPs do indeed yield results of good accuracy and extensibility if constructed from a training set containing a sufficiently large variety of atomic configurations. It also shows linear scaling in the size of the atomic configuration, as desired. However, it is about two orders of magnitude slower than classical empirical potentials.

[3]The numbering has been adjusted in this summary; in the original paper [11], these are referred to as G_i^4 and G_i^5.

4 Moment tensor potentials

4.1 Invariant polynomials

The following approach was first presented in [56], which we follow throughout this entire section. It is based on invariant polynomials and the fact that any sufficiently smooth potential $V^*(u)$ can be approximated by some polynomial $p(u) \approx V(u)$. Such polynomials can always be found. Indeed, given any polynomial $p(u)$, we can set

$$p_{\text{sym}}(u_1, \dots, u_n) \coloneqq \frac{1}{n!} \sum_{\sigma \in S_n} p(u_{\sigma_1}, \dots, u_{\sigma_n}) \tag{4.1}$$

to achieve invariance under permutation. The other invariances can be recovered similarly. With that, one can theoretically construct invariant polynomials $b_\nu(u)$ and set

$$V^*(u) \approx V(u) \coloneqq \sum_\nu c_\nu b_\nu(u).$$

This approach has been implemented for small system of up to ten atoms [13], but is computationally infeasible for larger systems, due to the fact that the number of permutations grows as $n!$, making the evaluation of eq. (4.1) impossible.

We will now present a way to construct polynomials invariant under permutation, translation, and rotation in $\mathcal{O}(n)$. They will be used later in Section 5 to construct a basis also satisfying smoothness at R_{cut}, thus yielding a promising candidate for efficient non-parametric interaction potentials.

The moment tensors $M_{\mu,\nu}$ We start by constructing a set of polynomials that guarantee the first two of the aforementioned invariants and spans the set $\mathbb{P}_{\text{perm}} \cap \mathbb{P}_{\text{rot}}$. With some set of relative positions of neighbors $u = \{u_i, \dots, u_n\}$, we define the *moment tensors* $M(u)$ as

$$M_{\mu,\nu}(u) \coloneqq \sum_{i=1}^{n} \|u_i\|^{2\mu} u_i^{\otimes \nu} \tag{4.2}$$

for some integers $\mu, \nu \geq 0$. Here,

$$w^{\otimes \nu} \coloneqq \underbrace{w \otimes \ldots \otimes w}_{\nu \text{ times}}$$

is the ν-fold Kronecker product of a vector with itself. Thus, $M_{\mu,\nu}$ has dimension ν, i.e. we can index its elements with indices $i_1, \dots, i_\nu$ with $0 \leq i_k \leq 3$. Each of the tensors has 3^ν entries, of which $\binom{\nu+2}{\nu}$ are unique. If we bound ν by some relatively small number, the moment tensors can be evaluated efficiently in $\mathcal{O}(n)$.

The moment tensors can be interpreted physically as follows: the zero-dimensional tensor $M_{0,0}$ just counts the number of atoms in the neighborhood, the one-dimensional $M_{0,1}$ gives the center of mass of those atoms (scaled by the mass), $M_{0,2}$ the tensor of second moments of inertia, and so on. For $\mu > 0$, $M_{\mu,\nu}$ can be interpreted as moments of inertia where the i-th atom has weight $\|u_i\|^{2\mu}$.

The basis polynomials $B_{\hat{\alpha},\alpha}$ From the entries of these tensors, we now construct our basis polynomials B_α as characterized by a symmetric matrix $\alpha \in \mathbb{N}^{n\times n}$ with zeros on the diagonal and a vector $\hat{\alpha} \in \mathbb{N}^n$ as follows: let α'_i the sum of the entries of the i-th row. With that, we define a tensor contraction between the n tensors $M_{\hat{\alpha}_i,\alpha'_i}$ by an Einstein summation

$$B_{\hat{\alpha},\alpha} \coloneqq (M_{\hat{\alpha}_1,\alpha'_1})_{i_1,\dots,i_{\alpha'_1}} \dots (M_{\hat{\alpha}_n,\alpha'_n})_{i_1,\dots,i_{\alpha'_n}}, \tag{4.3}$$

where the i-th and the j-th tensor share α_{ij} indices. As the moment tensors are symmetric in every dimension, the choice of the indices being shared is irrelevant. Note that the $B_{\hat{\alpha},\alpha}$ are invariant under simultaneous permutation of the rows and columns of the pairs $(\hat{\alpha}, \alpha)$, i.e. if for $(\hat{\alpha}, \alpha)$, $(\hat{\beta}, \beta)$ there exists some $\sigma \in S_n$ such that for all $i, j = 1, \dots, n$:

$$\hat{\alpha}_i = \hat{\beta}_{\sigma(i)} \tag{4.4}$$

and

$$\alpha_{ij} = \beta_{\sigma(i)\sigma(j)}, \tag{4.5}$$

then $B_{\hat{\alpha},\alpha} = B_{\hat{\beta},\beta}$.

For a more rigorous definition, we introduce some notation: we write $M^{(i)} \coloneqq M_{\hat{\alpha}_i,\alpha'_i}$ for the moment tensor corresponding to the i-th line (or column) of α. For multi-indices $\beta_1 \in \mathbb{N}^l$ and $\beta_2 \in \mathbb{N}^m$ let $\beta_1 \oplus \beta_2 \in \mathbb{N}^{l+m}$ denote their concatenation. Let

$$\mathcal{B} \coloneqq \prod_{i=1}^{k} \prod_{j=i}^{k} \{1, 2, 3\}^{\alpha_{i,j}} \tag{4.6}$$

be a set[4] of tuples (each indexed by (i, j) such that $i \leq j$) of multi-indices. Then for $\beta \in \mathcal{B}$,

$$\beta^{(i)} \coloneqq \bigoplus_{j=1}^{i} \beta_{j,i} \oplus \bigoplus_{j=i+1}^{k} \beta_{i,j} \tag{4.7}$$

is a multi-index that fits the dimensions of $M^{(i)}$. Note that for now, $\alpha_{ii} = 0$ for all $i = 1, \dots, k$ and thus $\beta_{i,i}$ is always an empty tuple. This restriction will later be lifted in Section 4.4 for intermediate results needed for an

[4]The product is to be understood as Cartesian product of sets.

computationally efficient evaluation of the B_α, but this is not relevant for the theoretical analysis in Sections 4.2 and 4.3. We can now write our basis polynomials as

$$B_\alpha(u) = \overset{\alpha}{\prod_{i=1}^{k}} M^{(i)}(u) \coloneqq \sum_{\beta \in \mathcal{B}} \prod_{i=1}^{k} M^{(i)}_{\beta^{(i)}}(u). \tag{4.8}$$

From eqs. (4.6) and (4.7) it is clear, as mentioned earlier, that the matrix entry α_{ij} for $i \neq j$ corresponds to the number of dimensions contracted between tensors $M^{(i)}$ and $M^{(j)}$ in the Einstein sum in eq. (4.3).

Examples We will start with two examples involving only two moment tensors, thus $k = 2$. Let

$$\alpha \coloneqq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and

$$\hat{\alpha} \coloneqq \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}.$$

Both row sums are 1, thus the two tensors both have only one dimension. As $\alpha_{1,2} = 1$, one dimension is contracted between the first and the second tensor, and thus we have

$$B_{\hat{\alpha},\alpha} = \sum_{i=1}^{3} (M_{\mu_1,1})_i (M_{\mu_2,1})_i.$$

Similarly, with

$$\alpha \coloneqq \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}$$

and

$$\hat{\alpha} \coloneqq \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix},$$

we get two-dimensional tensors (i.e. matrices) and contract two dimensions between them, hence

$$B_{\hat{\alpha},\alpha} = \sum_{i=1}^{3} \sum_{j=1}^{3} (M_{\mu_1,2})_{i,j} (M_{\mu_2,2})_{i,j}.$$

Last, an example with three tensors, given by

$$\alpha \coloneqq \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

and

$$\hat{\alpha} \coloneqq \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}.$$

The sums of the rows and thus the dimensions of the corresponding tensors are 3, 2, and 3 respectively. We contract two dimensions between the first and the third tensor, because $\alpha_{1,3} = 2$ and one dimension of the second with each the first and the third tensor, as $\alpha_{1,2} = \alpha_{2,3} = 1$. This provides

$$B_{\hat{\alpha},\alpha} = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3}(M_{\mu_1,3})_{i,j,k}(M_{\mu_2,2})_{i,l}(M_{\mu_3,3})_{j,k,l}.$$

4.2 Representability

The key statement is that the set of the B_α spans the space of polynomials with our desired invariants:

Theorem 4.1 (Shapeev [56])**.** *The polynomials B_α form a spanning set of the linear space $\mathbb{P}_{\text{rot}} \cap \mathbb{P}_{\text{perm}} \subset \mathbb{P}$, in the sense that any $p \in \mathbb{P}_{\text{rot}} \cap \mathbb{P}_{\text{perm}}$ can be represented by a (finite) linear combination of B_α (but the combination is, in general, not unique).*

The proof of this theorem closely follows [56]. It uses several lemmas, which we also cite (including the proofs) from the aforementioned publication. First, we state the First Fundamental Theorem for the orthogonal group $O(d)$ (where $d = 3$ in our case) [63]:

Theorem 4.2. *$p \in \mathbb{P}$ is rotation invariant if and only if it can be represented as a polynomial of $n(n+1)/2$ scalar variables of the form $r_{ij} \coloneqq u_i \cdot u_j$, where $1 \le i \le j \le n$.*

We introduce $\mathbb{Q}$ as the set of polynomials of $n(n+1)$ scalar variables $r = (r_{ij})_{1 \le i,j \le n}$. The restriction $i \le j$ is dropped for ease of notation. With this, we can hence identify each polynomial $p = p(u) \in \mathbb{P}_{\text{rot}}$ with the respective polynomial $q = q(r) \in \mathbb{Q}$. We write

$$\mathcal{N} \coloneqq \{1, \dots, n\}$$

and define a notation of composition of tuples by

$$a_b \coloneqq (a_{b_1}, \ldots, a_{b_m}) \qquad \text{for} \quad b \in \mathcal{N}^m \quad \text{and} \quad a = (a_1, \ldots, a_n). \tag{4.9}$$

The permutation invariant subset of $\mathbb{Q}$ is given by

$$\mathbb{Q}_{\text{perm}} \coloneqq \{q \in \mathbb{Q} \mid \forall \sigma \in \mathcal{S}_n \ : \ q(r) \equiv q(r_{\sigma\sigma})\}$$

with $r_{\sigma\sigma} \coloneqq (r_{\sigma_i\sigma_j})_{1\leq i,j\leq n}$, which corresponds to $\mathbb{P}_{\text{perm}}$. The following lemma essentially states that $\mathbb{Q}_{\text{perm}}$ can be spanned by symmetrizing all monomials of r:

Lemma 4.3.

$$\mathbb{Q}_{\text{perm}} = \operatorname{span}\left\{\sum_{\sigma\in\mathcal{S}}\prod_{i=1}^{m}\prod_{j=1}^{m}(r_{\sigma_i\sigma_j})^{\alpha_{ij}} \mid m \in \mathbb{N}, m \leq n, \alpha \in \mathbb{N}^{m\times m}\right\}.$$

Proof. If $q \in \mathbb{Q}_{\text{perm}}$, then

$$\frac{1}{n!}\sum_{\sigma\in\mathcal{S}_n} q(r_{\sigma\sigma}) = \frac{1}{n!}q(r) = q(r).$$

Applying this identity to all monomials

$$q(r) = \prod_{i=1}^{m}\prod_{j=1}^{m} r_{ij}^{\alpha_{ij}}$$

yields the result. □

Lemma 4.4. *For $m \in \mathbb{N}$, $m \leq n$ and $\alpha \in \mathbb{N}^{m\times m}$,*

$$B_\alpha(u) = \sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m}\prod_{j=1}^{m}(u_{\gamma_i}\cdot u_{\gamma_j})^{\alpha_{ij}}.$$

Proof. In this proof, we will use the distributive law of addition and multiplication in the following form:

$$\prod_{j=1}^{m}\sum_{i\in\mathcal{I}} f(i,j) = \sum_{\gamma\in\mathcal{I}^m}\prod_{j=1}^{m} f(\gamma_j, j).$$

For some $\beta \in \mathcal{B}$ we write $\beta^{i,j}$ for its element indexed by i and j and $\beta_l^{i,j}$ for the l-th entry of said element. Then

$$\begin{aligned} M^{(i)}_{\beta^{(i)}}(u) &= \sum_{\gamma\in\mathcal{N}} |u_\gamma|^{2\hat{\alpha}_i}\left(\prod_{j=1}^{i-1}\left(u_\gamma^{\otimes\alpha_{ji}}\right)_{\beta^{j,i}}\right)\left(\prod_{j=i+1}^{m}\left(u_\gamma^{\otimes\alpha_{ij}}\right)_{\beta^{i,j}}\right) \\ &= \sum_{\gamma\in\mathcal{N}} |u_\gamma|^{2\hat{\alpha}_i}\left(\prod_{j=1}^{i-1}\prod_{l=1}^{\alpha_{ji}} u_{\gamma,\beta_l^{j,i}}\right)\left(\prod_{j=i+1}^{m}\prod_{l=1}^{\alpha_{ij}} u_{\gamma,\beta_l^{i,j}}\right). \end{aligned}$$

With the notation $\mathcal{B}$ from eq. (4.6), we finally get

$$
\begin{aligned}
{}^{\alpha}\prod_{i=1}^{m} M^{(i)}(u) &= \sum_{\beta\in\mathcal{B}}\prod_{i=1}^{m} M^{(i)}_{\beta^{(i)}} \\
&= \sum_{\beta\in\mathcal{B}}\prod_{i=1}^{m}\sum_{\gamma\in\mathcal{N}} |u_\gamma|^{2\hat{\alpha}_i}\left(\prod_{j=1}^{i-1}\prod_{l=1}^{\alpha_{ij}} u_{\gamma,\beta_l^{j,i}}\right)\left(\prod_{j=i+1}^{m}\prod_{l=1}^{\alpha_{ij}} u_{\gamma,\beta_l^{i,j}}\right) \\
&= \sum_{\beta\in\mathcal{B}}\sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m} |u_{\gamma_i}|^{2\hat{\alpha}_i}\left(\prod_{j=1}^{i-1}\prod_{l=1}^{\alpha_{ji}} u_{\gamma_i,\beta_l^{j,i}}\right)\left(\prod_{j=i+1}^{m}\prod_{l=1}^{\alpha_{ij}} u_{\gamma_i,\beta_l^{i,j}}\right) \\
&= \sum_{\beta\in\mathcal{B}}\sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m} |u_{\gamma_i}|^{2\hat{\alpha}_i}\prod_{j=i+1}^{m}\prod_{l=1}^{\alpha_{ij}}\left(u_{\gamma_i,\beta_l^{i,j}}\right)\left(u_{\gamma_j,\beta_l^{i,j}}\right) \\
&= \sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m} |u_{\gamma_i}|^{2\hat{\alpha}_i}\prod_{j=i+1}^{m}\sum_{\beta\in\{1,2,3\}^{\alpha_{ij}}}\prod_{l=1}^{\alpha_{ij}}(u_{\gamma_i,\beta_l})(u_{\gamma_j,\beta_l}) \\
&= \sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m} |u_{\gamma_i}|^{2\hat{\alpha}_i}\prod_{j=i+1}^{m}\prod_{l=1}^{\alpha_{ij}}\sum_{\beta=1}^{3}(u_{\gamma_i,\beta})(u_{\gamma_j,\beta}) \\
&= \sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m} |u_{\gamma_i}|^{2\hat{\alpha}_i}\prod_{j=i+1}^{m}\prod_{l=1}^{\alpha_{ij}} u_{\gamma_i}\cdot u_{\gamma_j} \\
&= \sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m} |u_{\gamma_i}|^{2\hat{\alpha}_i}\prod_{j=i+1}^{m}\left(u_{\gamma_i}\cdot u_{\gamma_j}\right)^{\alpha_{ij}} \\
&= \sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m}\prod_{j=1}^{m}\left(u_{\gamma_i}\cdot u_{\gamma_j}\right)^{\alpha_{ij}}.
\end{aligned}
$$

□

With the above lemma, it is clear that Theorem 4.1 holds for

$$\tilde{\mathbb{Q}}_{\text{perm}} = \text{span}\left\{\sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m}\prod_{j=1}^{m}\left(r_{\gamma_i\gamma_j}\right)^{\alpha_{ij}} \mid m\in\mathbb{N}, \alpha\in\mathbb{N}^{m\times m}\right\}.$$

All that remains is to show that $\tilde{\mathbb{Q}}_{\text{perm}} = \mathbb{Q}_{\text{perm}}$. This follows directly from the following lemma:

Lemma 4.5. *For $m\in\mathbb{N}$, $m\le n$, denote*

$$\mathbb{Q}^{(m)}_{\text{perm}} \coloneqq \text{span}\left\{\sum_{\sigma\in\mathcal{S}_n}\prod_{i=1}^{m}\prod_{j=1}^{m}(r_{\sigma_i\sigma_j})^{\alpha_{ij}} \mid \alpha\in\mathbb{N}^{m\times m}\right\} \quad \textit{and}$$

$$\tilde{\mathbb{Q}}^{(m)}_{\text{perm}} \coloneqq \text{span}\left\{\sum_{\gamma\in\mathcal{N}^m}\prod_{i=1}^{m}\prod_{j=1}^{m}(r_{\gamma_i\gamma_j})^{\alpha_{ij}} \mid \alpha\in\mathbb{N}^{m\times m}\right\}.$$

Then $\tilde{\mathbb{Q}}^{(m)}_{\text{perm}} = \mathbb{Q}^{(m)}_{\text{perm}}$.

The proof of Lemma 4.5 requires two more auxiliary results:

Lemma 4.6. *We equip $\mathcal{N}^m$ with the lexicographical order and, with the tuple composition defined in eq.* (4.9)*, and denote by $\Gamma \coloneqq \{\gamma \in \mathcal{N}^m \mid \gamma = \min\{\sigma_\gamma \mid \sigma \in \mathcal{S}_n\}\}$ the set of representatives of the equivalence classes $\{\sigma_\gamma | \sigma \in \mathcal{S}_n\}$, and $C_\gamma \coloneqq \#\{\sigma_\gamma | \sigma \in \mathcal{S}_n\}$. Then*

$$\sum_{\gamma\in\mathcal{N}^m} \prod_{i=1}^m \prod_{j=1}^m \left(r_{\gamma_i\gamma_j}\right)^{\alpha_{ij}} = \sum_{\gamma\in\Gamma} \frac{C_\gamma}{n!} \sum_{\sigma\in\mathcal{S}_n} \prod_{i=1}^m \prod_{j=1}^m \left(r_{\sigma_{\gamma_i}\sigma_{\gamma_j}}\right)^{\alpha_{ij}}.$$

Proof. Any $\sigma \in \mathcal{S}_n$ induces a bijection $\gamma \mapsto \sigma_\gamma$ on $\mathcal{N}^m$. Thus for all $\sigma \in \mathcal{S}_n$

$$\sum_{\gamma\in\mathcal{N}^m} \prod_{i=1}^m \prod_{j=1}^m \left(r_{\gamma_i\gamma_j}\right)^{\alpha_{ij}} = \sum_{\gamma\in\mathcal{N}^m} \prod_{i=1}^m \prod_{j=1}^m \left(r_{\sigma_{\gamma_i}\sigma_{\gamma_j}}\right)^{\alpha_{ij}}$$

and therefore

$$\sum_{\gamma\in\mathcal{N}^m} \prod_{i=1}^m \prod_{j=1}^m \left(r_{\gamma_i\gamma_j}\right)^{\alpha_{ij}} = \sum_{\gamma\in\mathcal{N}^m} \frac{1}{n!} \sum_{\sigma\in\mathcal{S}_n} \prod_{i=1}^m \prod_{j=1}^m \left(r_{\sigma_{\gamma_i}\sigma_{\gamma_j}}\right)^{\alpha_{ij}}.$$

Grouping together identical terms yields the result. □

The other auxiliary result expresses the elementary combinatorial idea that elements of Γ rather than $(1, \dots, m) \in \Gamma$ have repeated values.

Proposition 4.7. *Let $m \geq 1$. Then $\Gamma = \{(1, \dots, m)\} \cup \Gamma'$, where $\Gamma' \coloneqq \{\gamma \in \Gamma \mid \max_i \gamma_i \leq m - 1\}$.*

Proof of Lemma 4.5. The proof uses induction over m. The base case $m = 0$ is trivially true as $\mathbb{Q}_{\mathrm{perm}}^{(0)} = \tilde{\mathbb{Q}}_{\mathrm{perm}}^{(0)} = \mathrm{span}\{1\}$. For the induction step, let $m \in \mathbb{N}$, $m \leq n$ and $\alpha \in \mathbb{N}^{m\times m}$ arbitrary. The functions

$$\begin{aligned} q(r) &\coloneqq \sum_{\sigma\in\mathcal{S}_n} \prod_{j,k=1}^m (r_{\sigma_j\sigma_k})^{\alpha_{jk}} && \in \mathbb{Q}_{\mathrm{perm}}, \quad \text{and} \\ \tilde{q}(r) &\coloneqq \frac{n!}{C_{\{1,\dots,m\}}} \sum_{\gamma\in\mathcal{N}^m} \prod_{j,k}^m (r_{\gamma_j\gamma_k})^{\alpha_{jk}} && \in \tilde{\mathbb{Q}}_{\mathrm{perm}} \end{aligned}$$

span $\mathbb{Q}_{\mathrm{perm}}^{(m)}$ and $\tilde{\mathbb{Q}}_{\mathrm{perm}}^{(m)}$, respectively. We will now show that $q(r) - \tilde{q}(r) \in \mathbb{Q}_{\mathrm{perm}}^{(m-1)} = \tilde{\mathbb{Q}}_{\mathrm{perm}}^{(m-1)}$. The result will follow from this because $\mathbb{Q}_{\mathrm{perm}}^{(m-1)} \subset \mathbb{Q}_{\mathrm{perm}}^{(m)}$ and $\tilde{\mathbb{Q}}_{\mathrm{perm}}^{(m-1)} \subset \tilde{\mathbb{Q}}_{\mathrm{perm}}^{(m)}$ by definition. Using the previous two results, we write

$$\begin{aligned} \tilde{q}(r) - q(r) &= \frac{n!}{C_{\{1,\dots,m\}}} \sum_{\gamma\in\Gamma} \frac{C_\gamma}{n!} \sum_{\sigma\in\mathcal{S}_n} \prod_{i=1}^m \prod_{j=i}^m (r_{\sigma_{\gamma_i}\sigma_{\gamma_j}})^{\alpha_{ij}} - \sum_{\sigma\in\mathcal{S}_n} \prod_{i=1}^m \prod_{j=i}^m (r_{\sigma_i\sigma_j})^{\alpha_{ij}} \\ &= \sum_{\gamma\in\Gamma'} \frac{C_\gamma}{C_{\{1,\dots,m\}}} \sum_{\sigma\in\mathcal{S}_n} \prod_{i=1}^m \prod_{j=i}^m (r_{\sigma_{\gamma_i}\sigma_{\gamma_j}})^{\alpha_{ij}}. \end{aligned}$$

Next, we denote

$$\alpha_{kl}^{(\gamma)} \coloneqq \sum_{\substack{1\le i\le j\le m \\ \gamma_i=k,\, \gamma_j=l}} \alpha_{ij}$$

and with that write

$$\tilde{q}(r) - q(r) = \sum_{\gamma\in\Gamma'} \frac{C_\gamma}{C_{\{1,\dots,m\}}} \sum_{\sigma\in\mathcal{S}_n} \prod_{k=1}^{m-1} \prod_{l=k}^{m-1} (r_{\sigma_k\sigma_l})^{\alpha_{kl}^{(\gamma)}}.$$

Using the definition of Γ', we now have $m-1$ as the upper bound of the products. Also note that we used the fact that $\alpha_{kl}^{(\gamma)} = 0$ whenever $k > l$. Since by definition

$$\sum_{\sigma\in\mathcal{S}_n} \prod_{k=1}^{m-1} \prod_{l=k}^{m-1} (r_{\sigma_k\sigma_l})^{\alpha_{kl}^{(\gamma)}} \in \mathbb{Q}_{\text{perm}}^{(m-1)}$$

for every γ, we finally have

$$\tilde{q}(r) - q(r) \in \mathbb{Q}_{\text{perm}}^{(m-1)} = \tilde{\mathbb{Q}}_{\text{perm}}^{(m-1)},$$

which completes the proof of the induction step. □

Proof of Theorem 4.1. With the above lemmas, all the work is done, so we summarize: with Lemma 4.4 we see that the B_α span $\tilde{\mathbb{Q}}_{\text{perm}}$ and Lemma 4.5 shows, using lemma 4.3, that $\mathbb{Q}_{\text{perm}} = \tilde{\mathbb{Q}}_{\text{perm}}$. Thus, the B_α span $\mathbb{Q}_{\text{perm}}$. □

4.3 Approximation error estimate

In this section, we present the core convergence result, showing that invariant polynomials, and thus the moment tensor potential, can approximate the tight-binding quantum model as introduced in Section 2.3. Note that in the following analysis, we take the number of neighbors n to be fixed. This is essentially equivalent to assuming a finite R_{cut} and that the atoms do not get "too close" to each other. These assumptions fit the real-world scenario we want to model, so they are not problematic for our actual application. However, as Shapeev mentions in [56], performing the analysis for variable n is not trivial.

To this end, we assume that the hopping term $\varphi(u)$ is analytically extended to some box

$$\mathcal{V}_\delta \coloneqq \Big\{\zeta\in\mathbb{C} \mid \mathrm{Re}(\zeta)\in[-R-\delta, R+\delta]^d \wedge \mathrm{Im}(\zeta)\in[-\delta,\delta]^d\Big\}.$$

For models in which φ has a singularity in $\zeta = 0$, like $\varphi(\zeta) = \beta_0\exp(-q|\zeta|)$ or $\varphi(\zeta) = \beta_0 R^{-n}$ [26, eq. (7.24),(7.25)], we assume φ can be approximated by some analytical function, for example using Hermite polynomials [56]. The proof of the convergence theorem requires some results from approximation theory. First, an error estimate for interpolation polynomials from [58] as presented in [32]:

Definition 4.8 (Bernstein's regularity ellipse)**.** A function $f \in C^\infty([-1,1])$ has *Bernstein's regularity ellipse* $\mathcal{E}_\rho$ if it admits an analytic extension to the closed ellipse

$$\mathcal{E}_\rho \coloneqq \left\{\zeta \in \mathbb{C} \mid |z-1| + |z+1| \leq \rho + \rho^{-1}\right\}$$

for some $\rho > 1$. For $f \in C^\infty([-1,1]^n)$, we define

$$\mathcal{E}_\rho^{(j)} \coloneqq [-1,1]^{j-1} \times \mathcal{E}_\rho \times [-1,1]^{n-j-1}$$

equivalently.

Proposition 4.9. *For some function $f \in C^\infty([-1,1])$ with regularity ellipse $\mathcal{E}_{\rho_0}$, let $p_N \in \mathbb{P}_N([-1,1])$ be the interpolation polynomial of f with respect to the Chebyshev-Gauss-Lobatto nodes*

$$\xi_j = \cos\frac{\pi j}{N}, \quad j = 1, \ldots, N.$$

Then the following approximation error estimate holds:

$$\|f - p_N\|_{L^\infty} \leq cN\frac{\rho^N}{\rho - 1}\max_{\zeta \in \mathcal{E}_\rho}|f(\zeta)| \tag{4.10}$$

for all $\rho \in (0, \rho_0)$.

This is extended to functions in several variables in [32] as follows: for multi-dimensional $f \in C^\infty([-1,1]^n)$ we get the Chebyshev-Gauss-Lobatto nodes in n dimensions as the n-fold Cartesian product of the one-dimensional nodes.

Proposition 4.10. *Let $f \in C^\infty([-1,1]^n)$ such that f extends to $\mathcal{E}_{\rho_0}^{(j)}$ for some $\rho_0 > 1$ and all $j = 1, \ldots, n$. Let p_N be the interpolating polynomial with respect to the Chebyshev-Gauss-Lobatto nodes in n dimensions. Let*

$$M_\rho(f) \coloneqq \max_{1 \leq j \leq n}\max_{\zeta \in \mathcal{E}_\rho^{(j)}}|f(\zeta)|.$$

Then for $\rho \in (0, \rho_0)$ it holds

$$\|f - p_N\|_{L^\infty} \leq cN\log^{n-1}N\frac{\rho^{-N}}{\rho - 1}M_\rho(f). \tag{4.11}$$

Proof. Let $I_N^i f$ be the polynomial interpolating f with respect to ζ_i and $I_N f$ the interpolation polynomial p_N. Furthermore, let

$$X_{-i} \coloneqq \{\zeta_1, \ldots, \zeta_{i-1}, \zeta_{i+1}, \ldots, \zeta_n\}$$

for $i = 1, \dots, n$. Then the triangle inequality and eq. (4.10) yield

$$\begin{aligned}
\|f - I_N f\|_{L^\infty} &\leq \|f - I_N^1 f\|_{L^\infty} + \|I_N^1(f - I_N^2 \dots I_N^n f)\|_{L^\infty} \\
&\leq \|f - I_N^1 f\|_{L^\infty} \\
&\quad + \|I_N^1(f - I_N^2 f)\|_{L^\infty} + \|I_N^1 I_N^2(f - I_N^3 f)\|_{L^\infty} \\
&\quad + \dots + \|I_N^1 \dots I_N^{n-1}(f - I_N^n f)\|_{L^\infty} \\
&\overset{\text{eq. (4.10)}}{\leq} c \Bigg[\max_{X_{-1}} \max_{\zeta_1 \in \mathcal{E}_\rho^{(1)}} |f(\zeta)| + \log N \max_{X_{-2}} \max_{\zeta_2 \in \mathcal{E}_\rho^{(2)}} |f(\zeta)| \\
&\quad + \dots + \log^{n-1} N \max_{X_{-n}} \max_{\zeta_n \in \mathcal{E}_\rho^{(n)}} |f(\zeta)| \Bigg] N \frac{\rho^{-N}}{\rho - 1},
\end{aligned}$$

where we use an upper bound on the single-variable interpolant similar to [40]:

$$\|I_N^i f\|_{L^\infty} \leq c \log N \|f\|_{L^\infty}$$

This completes the proof of eq. (4.11). □

We can now formulate and prove the main convergence result from [56]:

Theorem 4.11. *Define*

$$M_{\delta_0} \coloneqq \max_{\zeta \in \mathcal{V}_{\delta_0}} \varphi(\zeta).$$

Then there exists $C > 0$ and $\rho > 1$, both depending only on n, δ_0, M_{δ_0} and T, such that for any $m \in \mathbb{N}$ there exists some $p_m \in \mathbb{P}_{\text{perm}} \cap \mathbb{P}_{\text{rot}}$ such that

$$\max_{u : \max_i |u_i| \leq R} |V^q(u) - p_m(u)| < C\rho^{-m}. \tag{4.12}$$

Proof. In this proof, we will first obtain bounds on $\varphi(u)$, on $H(u)$, and finally on the interpolating polynomial. Then we symmetrize this polynomial as a final step.

For $\delta \leq \delta_0/2$, the Cauchy integral formula bounds

$$|\varphi'(z)| \leq \left| \int_{\partial \mathcal{V}_{2\delta}} \frac{\varphi(\zeta)}{(\zeta - z)^2} \, \mathrm{d}\zeta \right| \leq \frac{(2R + 8\delta) M_{2\delta}}{\delta^2} =: M'_\delta \qquad \forall z \ : \ |z| \leq \delta.$$

So for $\delta \in (0, \delta_0/2]$ and $z \in \mathcal{V}_\delta$, the intermediate value theorem gives us

$$|\operatorname{Im}(\varphi(z))| \leq \operatorname{Im}(\varphi(\operatorname{Re}(z))) + M'_{\delta_0} \operatorname{Im}(z) \leq M'_{\delta_0} \delta.$$

Let

$$\mathcal{U}_\delta \coloneqq \bigcup_{i=1}^{n} [-R, R]^{i-1} \times \mathcal{V}_\delta \times [-R, R]^{n-i-1}.$$

$H(u)$ is symmetric for $u \in \mathcal{U}_\delta$ with at most $2n$ elements being non-real. We can thus estimate for the Frobenius norm:

$$\|\operatorname{Im}(H(u))\| \leq \sqrt{2n} M'_\delta \delta \qquad \forall u \in \mathcal{U}_\delta$$

By the Bauer-Fike theorem [9], we can bound the imaginary part of every element $\sigma \in \operatorname{Sp}(H(u))$ of the spectrum of H:

$$|\operatorname{Im}(\sigma)| \leq \sqrt{2n} M'_\delta \delta \qquad \forall u \in \mathcal{U}_\delta$$

The real part of the spectrum elements is bounded directly by the norm:

$$|\operatorname{Re}(\sigma)| \leq \|H(u)\| \leq n M_{\delta_0} \qquad \forall u \in \mathcal{U}_\delta,\ \sigma \in \operatorname{Sp}(H(u))$$

We now fix δ such that

$$\sqrt{2n} M'_\delta \delta < \frac{\pi}{3} k_B T$$

and use the representation for the matrix function from [35, Definition 1.11]:

$$V(u) = -\frac{1}{2\pi i} \int_{\partial\Omega} \mathfrak{f}(z) \Big((H(u) - zI)^{-1} \Big)_{1,1} \,\mathrm{d}z$$

where

$$\Omega \coloneqq \left\{ z \in \mathbb{C} \;\middle|\; |\operatorname{Re}(z)| \leq n M_{\delta_0} + \frac{\pi}{3} k_B T,\ |\operatorname{Im}(z)| \leq \frac{2\pi}{3} k_B T \right\}.$$

Note that Ω is chosen such that it contains the spectrum of $H(u)$ for all $u \in \mathcal{U}_\delta$ and that $\partial\Omega$ is separated from every eigenvalue λ of $H(u)$ by the distance

$$|z - \lambda| \geq \frac{\pi}{3} k_B T.$$

Also, $\mathfrak{f}$ is regular on $\overline{\Omega}$. Hence we can estimate

$$\|(H(u) - zI)^{-1}\| \leq \left(\frac{\pi}{3} k_B T \right)^{-1},$$

and thus

$$\max_{u \in \mathcal{U}_\delta} |V(u)| \leq \max_{z \in \Omega} |\mathfrak{f}(z)| \operatorname{meas}(\partial\Omega) \left(\frac{\pi}{3} k_B T \right)^{-1} \leq C$$

where C depends only on n, δ_0, M_{δ_0} and T as desired. It remains to notice that we can pick some $\rho > 1$ such that

$$\{ z \in \mathbb{C} \mid |\, z/R - 1| + |\, z/R + 1| \leq \rho + \rho^{-1} \} \subset \mathcal{V}_\delta.$$

Thus, we can apply Proposition 4.10 to the function $f(z) \coloneqq V(Rz)$ to obtain an interpolating polynomial $\tilde{p}_m(u)$ such that

$$\max_{u : \max_i |u_i| \leq R} |V(u) - \tilde{p}_m(u)| \leq C \rho^{-m}.$$

All that remains is to recover the desired symmetries. First, we note that $\tilde{p}_m \in \mathbb{P}_{\text{perm}}$ by construction; indeed, the function $f(z) = V(Rz)$ is symmetric with respect to permutation of variables, and so are the Chebyshev-Gauss-Lobatto nodes on $[-1, 1]^n$. Thus uniqueness of interpolation yields the permutation symmetry of $\tilde{p}_m$. Finally, we set

$$p_m(u) \coloneqq \frac{\int_{Q \in O(3)} \tilde{p}_m(Qu)\, \mathrm{d}Q}{\int_{Q \in O(3) \mathrm{d}Q}} \in \mathbb{P}_{\text{perm}} \cap \mathbb{P}_{\text{rot}},$$

(where $\mathrm{d}Q$ denotes the Haar measure) and with the rotation invariance of V, we get eq. (4.12). □

It is worth noting that integration with respect to rotation is only needed in the proof as a technical device. In reality, we construct our approximation of V as a linear combination of the B_α polynomials. It is thus directly rotation invariant by construction, without using Chebyshev-Gauss-Lobatto interpolation as an intermediate step.

4.4 Efficient evaluation

Evaluating the basis polynomials naively as defined in eq. (4.3) requires the addition of $3^{\sum \alpha_i'/2}$ summands. This quickly becomes prohibitively expensive, but can be avoided by using the symmetry of the moment tensors and the fact that many of the sub-products of each summand will be used repeatedly across our collection of basis functions. This will allow us to separate the evaluation into some off-line precomputation step and a computationally efficient on-line evaluation step. During the precomputation step, we build a recursive representation of the entire basis set, carefully grouping together terms we know to be equal due to symmetry and short-circuiting the recursive decomposition of each basis function whenever values from another function's decomposition can be reused. With that, the actual evaluation of all functions in our basis set can be written as a simple loop which resembles the element-wise product of arrays. In this section, we will describe how to obtain this decomposition and present the on-line evaluation step.

For that, we first note that the symmetric moment tensors $M_{\mu,\nu}$ have $\binom{\nu+2}{\nu}$ unique entries. This allows us to simplify the contraction between n indices of two moment tensors M_1, M_2 as

$$\sum_{\beta \in B} c_\beta (M_1)_\beta (M_2)_\beta, \tag{4.13}$$

with

$$B \coloneqq \{(\beta_1, \ldots, \beta_n) \subset \{1, 2, 3\}^n \mid \beta_1 \leq \ldots \leq \beta_n\} \tag{4.14}$$

being the set of sorted multi-indices of length n and c_β the multiplicity of the index β. That is, c_β is the number of permutations $\sigma \in S_n$ such that

$$(\beta_1, \ldots, \beta_n) \neq (\beta_{\sigma(1)}, \ldots, \beta_{\sigma(n)}).$$

In this manner, we reduce the number of summands in this basic contraction from the exponential 3^n to the quadratic $\binom{n+2}{n}$.

Next, we decompose the long contraction between all moment tensors needed for some B_α recursively into contractions between two intermediate tensors. This serves two purposes:

a) Many intermediate pairs can be reused for several different B_α, thus further reducing computation time.

b) It is relatively easy to compute the derivative of these two-term contractions, which is needed to compute the forces acting on individual particles.

To this end, we use contractions between two tensors as defined in eq. (4.8), yielding first our intermediate tensors and finally our scalar basis functions (which, in the following notation, will appear as zero-dimensional tensors).

To characterize the intermediate tensors, we use notation as in Section 4.1, but allow non-zero entries for α to correspond with their non-zero dimensions. This yields tensors $B_{\hat{\alpha},\alpha}$ with dimension $\sum \alpha_{ii}$, where the $B_{\hat{\alpha},\alpha}$ with dimension zero are our final basis functions.

In order to decompose some given $B_{\hat{\alpha},\alpha}$, we split $\hat{\alpha}$ in two vectors $\hat{\gamma} = (\alpha_1, \ldots, \alpha_k)$ and $\hat{\eta} = (\alpha_{k+1}, \ldots, \alpha_n)$ for some $1 < k < n$ and α into

$$\gamma = \begin{pmatrix} \alpha_{11} + \sum_{i=k+1}^{n} \alpha_{i1} & \alpha_{12} & \cdots & \alpha_{1k} \\ \alpha_{12} & \alpha_{22} + \sum_{i=k+1}^{n} \alpha_{i2} & \cdots & \alpha_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{1k} & \alpha_{2k} & \cdots & \alpha_{kk} + \sum_{i=k+1}^{n} \alpha_{ik} \end{pmatrix} \text{ and}$$

$$\eta = \begin{pmatrix} \alpha_{k+1,k+1} + \sum_{i=1}^{k} \alpha_{i,k+1} & \alpha_{k+1,k+2} & \cdots & \alpha_{k+1,n} \\ \alpha_{k+1,k+2} & \alpha_{k+2,k+2} + \sum_{i=1}^{k} \alpha_{i,k+2} & \cdots & \alpha_{k+2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{k+1,n} & \alpha_{k+2,n} & \cdots & \alpha_{nn} + \sum_{i=1}^{k} \alpha_{i,n} \end{pmatrix}$$

To find an optimal decomposition for some $B_{\hat{\alpha},\alpha}$, we choose some permutation of $(\hat{\alpha}, \alpha)$ as defined in Section 4.1 and some k such that $B_{\hat{\gamma},\gamma}$ and $B_{\hat{\eta},\eta}$ have minimal dimension. Now, $B_{\hat{\alpha},\alpha}$ can be represented as contraction between $B_{\hat{\gamma},\gamma}$ and $B_{\hat{\eta},\eta}$ over the indices from the block matrix

$$\begin{pmatrix} \alpha_{1,k+1} & \cdots & \alpha_{1,n} \\ \vdots & \ddots & \vdots \\ \alpha_{k,k+1} & \cdots & \alpha_{k,n} \end{pmatrix}.$$

The entries of the $\sum_i \alpha_{ii}$-dimensional tensors are then given by

$$\left(B_{\hat{\alpha},\alpha}(u)\right)_{\tilde{\beta}} = \left({}^{\alpha}\prod_{i=1}^{k} M^{(i)}(u) \right)_{\tilde{\beta}} \coloneqq \sum_{\beta \in \mathcal{B}} \prod_{i=1}^{k} M^{(i)}_{\beta^{(i)}}(u) \tag{4.15}$$

with the definitions from Section 4.1 and

$$\tilde{\beta} \coloneqq \bigoplus_{i=1}^{k} \beta_{i,i}$$

fixed for every entry (that is, while computing some given entry of $B_{\hat{\alpha},\alpha}$, every $\beta_{i,i}$ in the right hand side of eq. (4.15) is also fixed). Note that, unlike the moment tensors, the tensors $B_{\hat{\alpha},\alpha}$ need not be symmetric in every dimension. However, the dimensions corresponding to the same diagonal entry of α are still equivalent, so we can exploit multiplicity of indices among equivalent dimensions again for computational performance. Another potential source of speedup is the fact that some decomposition components such as $B_{\hat{\gamma},\gamma}$ can be used in the computation of several different $B_{\hat{\alpha},\alpha}$. We now formalize the above considerations:

Precomputation The $B_{\hat{\alpha},\alpha}$ are decomposed as functions, i.e. the values u are not used in this step. The precomputation step consists of the following substeps:

1. Starting with the initial set of $(\hat{\alpha}, \alpha) \in A$, decompose them into $(\hat{\gamma}, \gamma)$ and $(\hat{\eta}, \eta)$. Continue this recursively until only $\alpha \in \mathbb{N}^{1\times 1}$ remain. At every decomposition step, choose a permutation of $(\hat{\alpha}, \alpha)$ as defined in eq. (4.4) and (4.5), and k such that the resulting intermediate tensors have minimal dimensions. Recall that we can choose the permutation of $(\hat{\alpha}, \alpha)$ freely as it does not affect the corresponding function. Also, make sure not to produce separate equivalent $(\hat{\alpha}, \alpha)$, but reuse existing once where possible. This allows for intermediate results to be used for several different basis functions in the end.

2. Next, we enumerate all unique entries of the resulting $B_{\hat{\alpha},\alpha}$ as b_i, taking into account the symmetries in the different dimensions where possible.

3. The tensors corresponding to each of the basic $(\hat{\alpha}, \alpha)$ with $\alpha \in \mathbb{N}^{1\times 1}$ are equal to the moment tensors $M_{\hat{\alpha}_1,\alpha_{11}}$; by definition their entries b_i are determined by $\mu_i = \hat{\alpha}_1$ and multi-indices $\beta_i \in \mathbb{N}^3$ with $|\beta_i| = \alpha_{11}$ (during the evaluation acting on the $u_j \in \mathbb{R}^3$ as exponents $u_j^{\beta_i}$). For those entries, we write $m_{\mu_i,\beta_i} \coloneqq b_i$.

4. The entries of the other tensors are contractions between elements of tensors they are decomposed into. We can thus write them as

$$b_i = \sum_{j=1}^{J_i} c_j b_{l_j} b_{k_j}, \tag{4.16}$$

   where c_j reflects the multiplicity of the summand respecting symmetries and the vectors l and k hold the indices of the entries of the decomposing tensors.

5. Finally, output the resulting

    - indices of the b_i corresponding to the scalar-valued basis functions from A,
    - indices of the moment tensor entries and the corresponding μ_i and β_i,
    - tuples (i, c, l, k) defining the sums in eq. (4.16).

Evaluation In the evaluation step, the basis functions are evaluated for some input u using the decomposition obtained through the precomputation step.

1. Calculate all the basic moment tensor entries $m_{\mu_i,\beta_i}(u)$.
2. Calculate the other $b_i(u)$ from the tuples (i, c, l, k). This can be implemented as a simple loop over contiguous arrays, which is favorable for performance.
3. Pick the final $b_i(u)$ corresponding to the desired basis set.

The final site energy can then be evaluated as

$$V(u) = \sum_{(\hat{\alpha},\alpha)\in A} c_\alpha B_{\hat{\alpha},\alpha}(u)$$

with some coefficient vector c_α obtained from training the model on some dataset.

5 Numerical results

5.1 Training

From here on, we will omit the $\hat{\alpha}$ in the notation and simply write B_α for $B_{\hat{\alpha},\alpha}$ for simplicity of notation unless we need to refer to $\hat{\alpha}$ explicitly.

Up until now, our basis functions B_α have been polynomials in the entries of the distance vectors. As such, they do not satisfy the smoothness condition at R_{cut}. We thus replace the $\|u\|^{2\mu}$ term in eq. (4.2) by some cutoff function $f_{\mu,\nu}(r)$ that guarantees $B_\alpha(u) = 0$ for $\|u\| > R_{\text{cut}}$ in a smooth manner. For that, we define

$$\hat{f}_{\mu,\nu}(r) \coloneqq \begin{cases} r^{-\nu-2+\mu}(R_{\text{cut}} - r)^2 & r < r_{\text{cut}} \\ 0 & r \geq R_{\text{cut}} \end{cases} \tag{5.1}$$

The $r^{-\nu}$ compensates for the degree of the $u^{\otimes\nu}$ and r^{-2} favors closer atoms over more distant ones. In [56], this function is then normalized on the interval $[R_{\text{min}}, R_{\text{cut}}]$ with the weight $r^{2\nu}(r - R_{\text{min}})(R_{\text{cut}} - r)$. R_{min} is chosen such that any two atoms never approach closer than $\mathbb{R}_{\text{min}}$. The concrete values used in both [56] and this work are $R_{\text{min}} \coloneqq 1.9\text{Å}$ and $R_{\text{cut}} \coloneqq 4.9\text{Å}$. The normalized functions $f_{\mu,\nu}$ are then used to compute the moment tensors as

$$M_{\mu,\nu}(u) \coloneqq \sum_{i=1}^{n} f_{\mu,\nu}(\|u_i\|) u_i^{\otimes\nu}. \tag{5.2}$$

However, we found in our experiments that simply setting

$$f_{\mu,\nu}(r) \coloneqq C_\mu \hat{f}_{\mu,\nu}(r)$$

with C_μ being the μ-th Chebyshev polynomial yields a model of almost the same quality as the normalizing approach; in the experiments we conducted it even performed very slightly better. Thus, the Chebyshev variant was used to produce the plots below. We saw that the choice of the cutoff function can have a major influence on the quality of the results. Using the function $\hat{f}$ without further modification increased the error by about a factor of two in the tungsten experiments in Section 5.3.1. Further optimizing the cutoff function might thus be an interesting task for future work.

For some fixed set A of α, we train our model using regularized linear regression as follows: for some database of atomic configurations $X = \{x^{(k)} \mid k = 1, \dots, K\}$, where $x^{(k)}$ consists of $N^{(k)}$ atoms, and their energies $E(x^{(k)}) = E^{(k)}$ and forces $-\nabla E(x^{(k)}) = f^{(k)}$, we form an overdetermined

system of linear equations by

$$\begin{aligned} \sum_{i=1}^{N^{(k)}} \sum_{\alpha \in A} c_\alpha B(Dx_i^{(k)}) &= E^{(k)} \\ \frac{\partial}{\partial x_j^{(k)}} \sum_{i=1}^{N^{(k)}} \sum_{\alpha \in A} c_\alpha B(Dx_i^{(k)}) &= -f_j^{(k)}, \end{aligned} \tag{5.3}$$

which can be written in matrix form as

$$Mc = g. \tag{5.4}$$

A short analysis regarding the accuracy of least squares approximation will be given in the next subsection.

Cross-validation As the system in eq. (5.4) may be ill-conditioned, and to avoid overfitting, some regularization scheme must be employed. For the results in this work, l_2-regularization was used, giving the problem

$$\min_c \|Mc - g\|_{l_2} + \gamma \|c\|_{l_2}.$$

The parameter γ was determined by 16-fold cross-validation, following the work of Shapeev. For this, the training dataset X is divided into 16 random and equally large subsets $\tilde{X} \coloneqq \{X_1, \ldots, X_{16}\}$. Then, for each $i = 1, \ldots, 16$, a model is trained on $\tilde{X} \setminus X_i$ using different values for γ and then evaluated on X_i. The value for γ which produces the lowest error is selected to train the model on the entire set X. In our experiments with tungsten, a good value for γ typically ended up being in the order of 10^{-10} to 10^{-9}. The cross-validation and the least squares computation was performed using the implementation in the scikit-learn package [50].

5.2 Least squares approximations

We will now summarize some basis results regarding the stability and accuracy of least square approximations based on [18].

Let ρ_Ω be a probability measure on some domain $\Omega \subset \mathbb{R}^d$. The goal is to estimate an unknown function $f : \Omega \to \mathbb{R}$ from a set of samples $(y_i)_{i=1,\ldots,n}$. These are are observations of f at the points $(x_i)_{i=1,\ldots,n}$ which are i.i.d. with respect to ρ_Ω. The error between f and its estimator $\tilde{f}$ is given in the $L^2(\Omega, \rho_\Omega)$ norm

$$\|v\|_\rho \coloneqq \left(\int_\Omega |v(x)|^2 \, \mathrm{d}\rho_\Omega(x) \right)^{1/2}.$$

We write $\langle \cdot, \cdot \rangle_\rho$ for the associated inner product.

Let V_m be a subspace of $L^2(\Omega, \rho_\Omega)$ with $\dim(V_m) = m$. We want to find the best approximation of f in V_m, which is given by the orthogonal

projection of f onto V_m. This can generally not be computed directly as we cannot access ρ_Ω or f outside the set $(x_i)_{i=1,\dots,n}$. Therefore, we consider the least squares problem

$$w = \arg\min_{v \in V_m} \sum_{i=1}^{n} |y_i - v(x_i)|^2$$

instead. Assuming noise-free data, we have $y_i = f(x_i)$ and can thus write

$$w = P_m^n f \coloneqq \arg\min_{v \in V_m} \|f - v\|_n,$$

where

$$\|v\|_n \coloneqq \left(\frac{1}{n} \sum_{i=1}^{n} |v(x_i)|^2 \right)^{1/2}$$

is the L^2 norm with respect to the empirical measure and $\langle \cdot, \cdot \rangle$ the associated inner product.

Stability Let $(L_1, \dots, L_m)$ be a basis of V_m, $\mathbf{G} \coloneqq (\langle L_j, L_k \rangle_n)_{j,k=1,\dots,m}$ and $\mathbf{f} \coloneqq (\langle f, L_k \rangle_n)_{k=1,\dots,m}$. Then the solution of the least squares problem can be written as

$$w = \sum_{j=1}^{n} u_j L_j,$$

where $\mathbf{u} = (u_j)_{j=1,\dots,m}$ is the solution of the $m \times m$ system

$$\mathbf{G}\mathbf{u} = \mathbf{f}.$$

For the purpose of this analysis, we will assume $(L_1, \dots, L_m)$ is an orthonormal basis with respect to $L^2(\Omega, \rho_\Omega)$. This requirement is not needed for the actual computation as the solution w is independent of the basis used to compute it. Then, we have

$$\mathbb{E}[\mathbf{G}] = (\langle L_j, L_k \rangle)_{j,k=1,\dots,m} = \mathbf{1}$$

We want understand how the random matrix $\mathbf{G}$ deviates from $\mathbf{1}$ in probability as this allows us to compare the norms $\|\cdot\|_n$ and $\|\cdot\|_\rho$ by

$$\|\mathbf{G} - \mathbf{1}\|_2 \leq \delta \iff |\|v\|_n^2 - \|v\|_\rho^2| \leq \delta \|v\|_\rho^2 \quad \text{for all} \quad \delta \in [0,1],\ v \in V_m,$$

where $\|\cdot\|_2$ for matrices denotes the spectral norm. For this, we define

$$K(m) \coloneqq \sup_{x \in \Omega} \sum_{j=1}^{n} |L_j(x)|^2.$$

Note that this is independent of the choice of basis and only depends on the space V_m and ρ_Ω and that

$$K(m) \geq \sum_{j=1}^{n} \|L_j\|^2 = m.$$

With this, we can formulate

Theorem 5.1. *For $0 < \delta < 1$, the estimate holds that*

$$\mathbb{P}(\|\mathbf{G} - \mathbf{1}\|_2 > \delta) \leq 2m \exp\Big(-\frac{c_\delta}{K(m)}\Big),$$

where $c_\delta \coloneqq \delta + (1-\delta)\log(1-\delta) > 0$.

From this, the stability condition

$$K(m) \lesssim \frac{n}{\log(n)}$$

can be derived. However, due the constants involved, the estimate needed to satisfy the condition in Theorem 5.1 does generally not hold in our setting for e.g. $\delta = 1/2$. We thus cannot assume our least squares system eq. (5.4) to be well behaved and hence apply regularization as described in the previous subsection.

Accuracy Let $P_m f$ be the exact orthogonal projection of f onto V_m, independent of the samples $(y_i)_{i=1,n}$ and let

$$e_m(f) \coloneqq \|f - P_m f\|_\rho.$$

We assume that the function f is uniformly bounded by L. With the assumption that our particles do not get too close to each other, implying a lower bound on their relative distance and an upper bound for the number of atoms in each neighborhood, this is reasonable in our setting. Then, from [31, Thm. 11.3], we get an error estimate of the form

$$\mathbb{E}\Big[\|f - \tilde{f}\|^2\Big] \leq C\Big(L^2 \frac{(\log(n)+1)m}{n} + e_m(f)^2\Big).$$

In our experiments, n was constant and we observed algebraic convergence in m.

5.3 Numerical experiments

5.3.1 Systems of one element

In the original paper, the moment tensor potential was trained on the database GAP_6 from `http://www.libatoms.org/`, which contains the energy of 9,693 unique atomic configurations and the forces on a total of

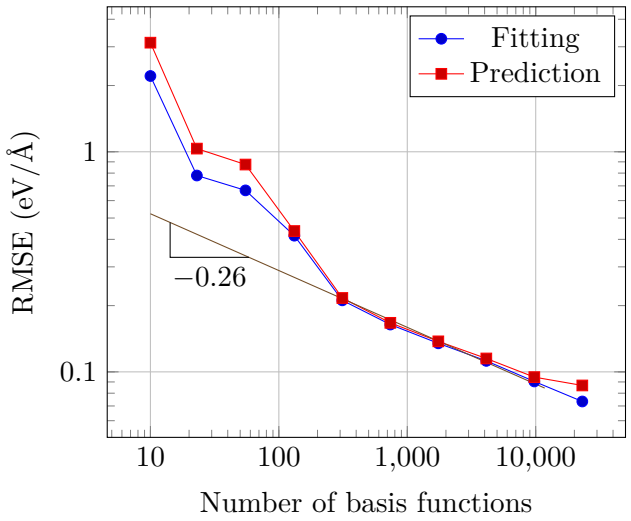

Figure 3: Convergence of fitting and prediction error for the tungsten dataset GAP_6.

158,526 atoms contained within. Most of our work, including the experiment in Figure 3, was also performed on this data set. The data in GAP_6 was generated using DFT; while we only proved convergence of our model to tight-binding. However, we will demonstrate convergence numerically [56].

With the "level" of a $B_{\hat{\alpha},\alpha}$ given by

$$\operatorname{lev}(B_{\hat{\alpha},\alpha}) \coloneqq \sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_{i,j} + 2\sum_{i=1}^{k} \hat{\alpha}_i,$$

we define our basis set as

$$A_N \coloneqq \{B_{\hat{\alpha},\alpha} \mid \operatorname{lev}(B_{\hat{\alpha},\alpha}) + 8k \leq N\}.$$

In the above, k is the length of the $\hat{\alpha} \in \mathbb{N}^k$. Note that this deviates from [56], but as we were not able to produce basis sets of the size reported there using their definition of A_N, we chose the above as it does show proper convergence behavior. The N is now our discretization variable in the sense that—given enough training data—we can expect our potential to converge as $N \to \infty$.

For the experiment in Figure 3, we chose, matching [56], $N = 62$ and further restricted $k \leq 4$, $\mu \leq 5$ and $\nu \leq 4$. Due to the unclear differences in the definition of A_N, this left us with 23,058 basis functions, while Shapeev reports $\#A_N = 11{,}133$ with the above choice of parameters. The performance of the models is measured via the root mean square error of the forces. For the prediction error, the models were trained on 60% of datasets and the plot shows the prediction error for the forces of remaining 40%. The fitting

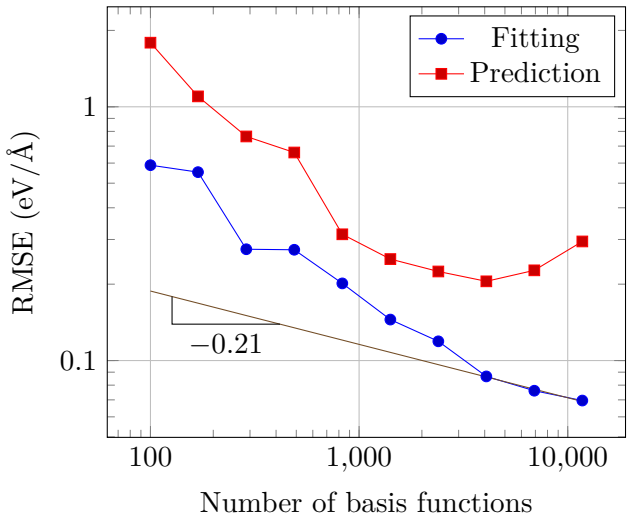

Figure 4: Convergence of fitting and prediction error for systems consisting of silver, copper, palladium and platinum using the atomic number to capture the different elements as described in eq. (5.5).

error was calculated after training the model on the entire dataset and then predicting all forces.

It can be seen that both the prediction error as well as the fitting error converge algebraically with a rate of about $\#A^{-0.26}$ after some pre-asymptotic regime with faster convergence. The value -0.26 is not a universal constant, but depends on the training data. We can also observe the predictive power to be quite close to the fitting error. In the original paper, Shapeev reports a slower rate of convergence with $\#A^{-0.227}$, but reaches a better root mean square error of 0.0427eV/Å compared to our 0.0733eV/Å despite the overall smaller basis set.

5.3.2 Systems of several elements

The moment tensor potential can easily be modified to handle systems consisting of chemically different atoms. One natural way to implement this is by modifying the cutoff function to reflect the difference between elements in the pair interaction, for example by setting

$$f_{\mu,\nu}^{(m)}(r_{ij}; i, j) \coloneqq Z_i Z_j f_{\mu,\nu}(r_{ij}), \tag{5.5}$$

where Z_i is the atomic number of the atom i. Using the product of the atomic numbers of the involved atoms to capture the difference in the pair interactions is a common technique used in machine learning, for example in Coulomb matrices [54, 48].

We investigated the fitting and prediction error as for the tungsten case for the dataset BFCC-13 from `http://qmml.org` [49], which consists of approximately four thousand systems of Ag-Pd, Cu-Pt and Ag-Pt alloys. The results are shown in Figure 4. In this setting, the choice of cutoff function proved to be more important than in the pure tungsten case, with the Chebyshev variant performing significantly better than the orthonormalized one. Due to the smaller dataset, we picked parameters yielding fewer basis functions: as before, we set $k \leq 4$, $\mu \leq 5$ and $\nu \leq 4$, but now restricted $\mathrm{lev}(B_\alpha) \leq 56$, yielding 11,748 basis functions. The model is able to reproduce the interactions between the chemically different atoms as in the single-element case. The rate or convergence is slightly slower with $\#A^{-0.22}$, but due to the longer pre-asymptotic regime with fast convergence, an error of similar magnitude as the tungsten error is reached, even with the relatively smaller basis set.

Unfortunately, the prediction error is much worse than the fitting error on this dataset. It appears the 60% of configurations the model was trained on did not contain sufficient information about the remaining 40% to properly predict the forces in the test set. The fact that the prediction error starts to grow again as the number of basis functions increases suggests we are overfitting the model to our training set. This highlights one of the caveats of machine learning potentials: the region of the potential energy surface in which properties are to be predicted must be represented in the available training data sufficiently well; thus a large computational effort to generate that data might be needed.

An alternative way to encode the particles' types is using the electronegativity χ instead of the atomic numbers, giving the cutoff function

$$f^{(m)}_{\mu,\nu}(r_{ij}; i, j) \coloneqq \chi_i \chi_j f_{\mu,\nu}(r_{ij}). \tag{5.6}$$

The hope would be that this quantity is a better label for the chemical and physical properties of the element than the atomic number; in the sense that we can expect elements with a similar electronegativity to exhibit similar behavior, something we cannot expect from atomic number. While elements with a similar atomic numbers often have fundamentally different properties (consider the direct neighbors fluorine and neon), the electronegativity is related to the elements actual interaction with surrounding electrons.

The direct comparison of the fitting error in Figure 5 shows that the electronegativity represents the ensembles significantly better for small numbers of basis functions. But as the number of basis functions grows, the representation based on atomic numbers manages to adapt to the training data similarly well.

More interesting is the comparison between the predictive power of the two variants, which is depicted in Figure 6. As in the experiment using atomic numbers, the accuracy of the predicted forces is considerably worse

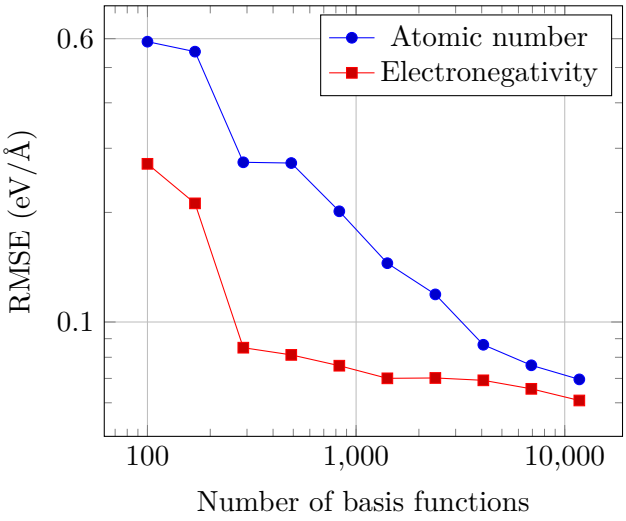

Figure 5: Comparison of the fitting error between capturing the particle's type by its atomic number (eq. (5.5)) and by its electronegativity (eq. (5.6)).

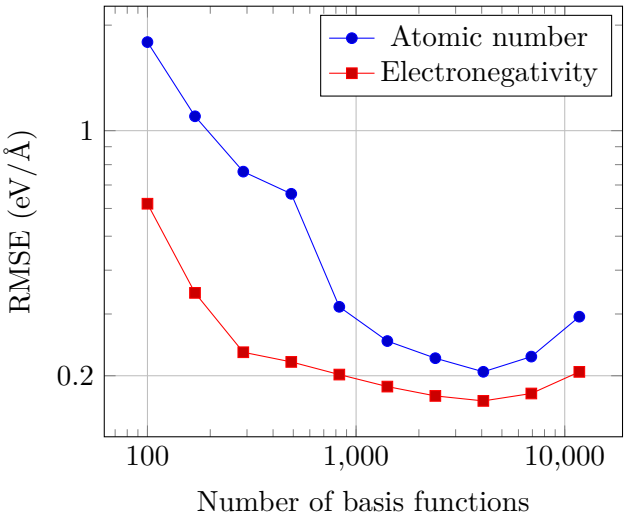

Figure 6: Comparison of the prediction error between capturing the particle's type by its atomic number (eq. (5.5)) and by its electronegativity (eq. (5.6)).

than the fitting precision. We can also again see the increase in the error as the number of basis functions gets too large and overfitting occurs. However, the predictions are consistently better than those based on the atomic numbers, a comparatively good accuracy is reached fairly early and the problems with overfitting are less severe. This allows for two observations:

a) The electronegativity of atoms is not just theoretically a better descriptor providing more relevant information, using it over atomic numbers yields a direct and measurable improvement of the resulting machine learning potentials. The increased predictive power is particularly noticeable and worthy of mention.

b) The quality of the cutoff function in general and its dependence on the particle types in a multi-element setting in particular play a crucial role in the quality of the resulting accuracy. This leaves room for further research, with the possibility of creating better models without a (significant) increase in the computational cost.

Of course, improving the moment tensor potential by changes to the cutoff function is an empirical process again. But as the model is systematically improvable regardless of the exact choice for the cutoff function, it provides a way to tweak the model's performance while retaining the core advantage of the non-parametric potential.

6 Conclusion

In this work, we gave an overview about particle methods for computational material design. We revised the quantum mechanical foundations such as the Schrödinger equation, followed be an summary of important approximations that make the simulation of quantum systems computationally feasible.

We then moved on to the specific problem setting this thesis focuses on: mesoscale atomistic simulations. We reviewed some state-of-the-art models for conducting such simulations and analyzed their advantages and disadvantages. The two main points of consideration were computationally efficiency and systematically improvability, i.e. the possibility to fit a model to quantum-mechanical or experimental data to arbitrary precision.

The latter has been approached using machine learning techniques in the recent past. We found that so called "descriptors", which map an atomic environment to a tuple of numbers, are a key ingredient of modern machine learning potentials. Two popular examples were described at length.

Section 4 provided an in-depth analysis of a new class of descriptors for machine learning potentials based on moment tensors, due to Shapeev in 2016 [56]. We presented proofs for the fact that the moment tensor potentials built from the moment tensors are theoretically capable of approximating the quantum-mechanical tight-binding model to arbitrary precision.

The computationally efficient evaluation of basis polynomials used in the moment tensor potentials required some additional work. As their direct definition involves the summation of an exponentially growing number of terms, Shapeev's idea to separate the computation into an off-line and an on-line step was used: first, a network-like representation of the entire basis set is pre-computed, using the symmetry of the moment tensors and the repeated occurrence of intermediate terms. Then, the basis functions could be evaluated efficiently by feeding the atomic coordinates through the network. An independent implementation of this strategy was developed for numerical experiments.

With the above, we had all the tools in place to actually construct moment tensor potentials. Using a cutoff function, the invariant polynomials were restricted to local atomic environments. This cutoff function proved to be of further interest in our numerical experiments; more on that later.

Fitting the model to a training dataset was done using a least squares approximation. We investigated the stability and accuracy of such approximations from a theoretical point of view. As instability concerns arose from the theoretical considerations, which were confirmed numerically, a regularization scheme was employed.

Lastly, numerical experiments were conducted on datasets of metallic configurations. Those were created using density functional theory. While convergence of the moment tensor potentials was only proven to tight-binding models, convergence to density functional theory was demonstrated

numerically. Fitting the model to a large dataset of tungsten configurations showed the expected algebraic convergence of the error in the size of the basis set. It also allowed for the observation that, even with only one chemical element involved, the choice of cutoff function can have a major influence on the quality of the resulting model. We have not seen the importance of this choice for the moment tensor potentials being stated elsewhere.

In addition to the fitting error, which was also measured by Shapeev, we analyzed the potential's predictive power. On the tungsten dataset, this predictive power proved to be very promising, reaching accuracies close to the fitting error. This shows that the moment tensor potentials can exhibit good transferability to atomic configurations outside of their training data.

Further experiments were conducted on a dataset of metallic alloys. The presence of different chemical elements in the configurations required to encode the particles' types in the basis functions. This was done by introducing a dependency on the types to the cutoff function. Two natural approaches, using the atomic number and using the electronegativity of the involved elements, were implemented and compared. To the best of our knowledge, this was the first time such an analysis of the dependence of moment tensor potentials on the encoding of chemical elements was performed. In addition to the effect of the encoding of particle types, the choice of cutoff function proved to be substantially more important than in the tungsten experiments.

Like in the single-element setting, the predictive power of the model was assessed. For the multi-element configurations, the prediction error turned out to be significantly larger than the fitting error and even failed to converge. This highlighted one of the caveats of machine learning potentials: if the region of the potential energy surface in which predictions are to be made is not represented in the training data sufficiently well, the quality of the predictions suffers considerably.

The problem of finding out how good a given configuration is represented by an existing training set for the moment tensor potentials has been attacked in recent publications [52, 30]. Using a D-optimality criterion, the "novelty" of an atomic configuration with respect to the training data can be estimated. With this, the moment tensor potentials can be used for active learning methods and for learning on-the-fly, thus optimizing the number of quantum machine calculations needed to achieve good results.

Outlook Several potential improvements to the methods presented here come to mind. As our numerical experiments suggest, the choice of cutoff function plays a non-negligible role in the performance of the model's performance. This leaves room for reducing the error without the additional cost required by increasing the number of basis functions. Of course, the process of picking a cutoff function is empirical by its nature; however, for a fixed choice, the systematic improvability of the moment tensor potential

is preserved.

One can also use a norm other than the l_2 norm in the regularization of the least squares system. Sparse solutions can be generated with an l_0 regularization [56]. Our knowledge about the smoothness of the target function could also be exploited to improve the regularization: as we expect $V \in C^2$ for the exact site energy function $V(X)$, choosing a norm that punishes large magnitudes in higher-order derivatives might help to reduce spurious oscillations in the model.

Another angle of attack is the choice of machine learning method. Instead of using a least squares approximation, the basis functions B_α could for example be used as descriptors for a neural network as they already guarantee all relevant invariants.

In conclusion, the moment tensor potentials seem to be a promising candidate for developing non-parametric, but computationally efficient interatomic potentials. They can approximate both tight-binding and density functional theory models with an algebraic rate of convergence. As we have demonstrated, their transferability can be limited depending on the training data. However, by applying active learning schemes, problematic configurations can be identified and included in the training set using a quantum model [52].

Acknowledgments

I would like to thank Prof. Dr. Michael Griebel and Prof. Dr. Marc Alexander Schweitzer for advising this thesis and the Institute for Numerical Simulation for allowing me to conduct my numerical experiments on their Atacama high performance computing cluster.

Furthermore, a big thanks goes to the department for Virtual Material Design of the Fraunhofer Institute for Algorithms and Scientific Computing (SCAI). They supported this thesis with hardware resources and access to some helpful software. Two members of the department deserve special mention: First of all Dr. Jan Hamaekers, with whom I had many very fruitful discussions about the involved topics and who provided a huge amount of useful advice and guidance. Also, James Barker, who kindly proof-read an intermediate draft of this work. Any typographical mistake or language error was most certainly introduced by me afterwards.

Last, but not least, I would like to thank my family for supporting me throughout my years of studying.

References

[1] Ebrahim Asadi, Mohsen Asle Zaeem, Sasan Nouranian, and Michael I. Baskes. Two-phase solidliquid coexistence of ni, cu, and al by molecular dynamics simulations using the modified embedded-atom method. *Acta Materialia*, 86:169 – 181, 2015.

[2] Alan Aspuru-Guzik, Roland Lindh, and Markus Reiher. The matter simulation (r) evolution. *ACS central science*, 2018.

[3] James Barker, Johannes Bulin, Jan Hamaekers, and Sonja Mathias. *LC-GAP: Localized Coulomb Descriptors for the Gaussian Approximation Potential*, pages 25–42. Springer International Publishing, Cham, 2017.

[4] Albert Bartók. *The Gaussian Approximation Potential: An Interatomic Potential Derived from First Principles Quantum Mechanics*. Springer Science & Business Media, 2010.

[5] Albert P Bartók and Gábor Csányi. Gaussian approximation potentials: A brief tutorial introduction. *International Journal of Quantum Chemistry*, 115(16):1051–1057, 2015.

[6] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

[7] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

[8] MI Baskes, JS Nelson, and AF Wright. Semiempirical modified embedded-atom potentials for silicon and germanium. *Physical Review B*, 40(9):6085, 1989.

[9] Friedrich L Bauer and Charles T Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2(1):137–141, 1960.

[10] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics*, 134(7):074106, 2011.

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

[12] Max Born and Robert Oppenheimer. Zur quantentheorie der molekeln. *Annalen der Physik*, 389(20):457–484, 1927.

[13] Bastiaan J Braams and Joel M Bowman. Permutationally invariant potential energy surfaces in high dimensionality. *International Reviews in Physical Chemistry*, 28(4):577–606, 2009.

[14] Donald W. Brenner. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. *Phys. Rev. B*, 42:9458–9471, Nov 1990.

[15] Matt Challacombe. A simplified density matrix minimization for linear scaling self-consistent field theory. *The Journal of chemical physics*, 110(5):2332–2342, 1999.

[16] Huajie Chen and Christoph Ortner. Qm/mm methods for crystalline defects. part 1: Locality of the tight binding model. *Multiscale Modeling & Simulation*, 14(1):232–264, 2016.

[17] Huajie Chen and Christoph Ortner. Qm/mm methods for crystalline defects. part 2: Consistent energy and force-mixing. *Multiscale Modeling & Simulation*, 15(1):184–214, 2017.

[18] Albert Cohen, Mark A Davenport, and Dany Leviatan. On the stability and accuracy of least squares approximations. *Foundations of computational mathematics*, 13(5):819–834, 2013.

[19] Christopher R Collins, Geoffrey J Gordon, O Anatole von Lilienfeld, and David J Yaron. Constant size descriptors for accurate machine learning models of molecular properties. *The Journal of Chemical Physics*, 148(24):241718, 2018.

[20] National Research Council et al. *Getting up to speed: The future of supercomputing*. National Academies Press, 2005.

[21] Murray S. Daw and M. I. Baskes. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B*, 29:6443–6453, Jun 1984.

[22] Soheil A Dianat and Raghuveer M Rao. Fast algorithms for phase and magnitude reconstruction from bispectra. *Optical Engineering*, 29(5):504–513, 1990.

[23] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. Number 27. Oxford university press, 1981.

[24] Andrew Ian Duff, MW Finnis, Philippe Maugis, Barend J Thijsse, and Marcel HF Sluiter. Meamfit: A reference-free modified embedded atom method (rf-meam) energy and force-fitting code. *Computer Physics Communications*, 196:439–445, 2015.

[25] H Fang, Philip Michael Gullett, Alexander Slepoy, Mark F Horstemeyer, Michael I Baskes, Gregory John Wagner, and Mo Li. Numerical tools for atomistic simulations. Technical report, Sandia National Laboratories, 2004.

[26] Mike Finnis. *Interatomic forces in condensed matter*, volume 1. OUP Oxford, 2003.

[27] Peter Galison. *Image and logic: A material culture of microphysics*. University of Chicago Press, 1997.

[28] Michael Griebel, Jan Hamaekers, and Frederik Heber. Bossanova: A bond order dissection approach for efficient electronic structure calculations. *INS Preprint*, 704, 2008.

[29] Michael Griebel, Stephan Knapek, and Gerhard Zumbusch. *Numerical Simulation in Molecular Dynamics. Numerics, Algorithms, Parallelization, Applications, volume 5 of Texts in Computational Science and Engineering*. Springer Verlag, 2007.

[30] Konstantin Gubaev, Evgeny V. Podryabinkin, and Alexander V. Shapeev. Machine learning of molecular properties: Locality and active learning. *The Journal of Chemical Physics*, 148(24):241727, 2018.

[31] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[32] Wolfgang Hackbusch and Boris N Khoromskij. Towards hmatrix approximation of linear complexity. In *Problems and methods in mathematical physics*, pages 194–220. Springer, 2001.

[33] Gaute Hagen, Thomas Papenbrock, M Hjorth-Jensen, and David J Dean. Coupled-cluster computations of atomic nuclei. *Reports on Progress in Physics*, 77(9):096302, 2014.

[34] P Haynes. *Linear-scaling methods in ab initio quantum-mechanical calculations*. PhD thesis, University of Cambridge, 1998.

[35] Nicholas J Higham. *Functions of matrices: theory and computation*, volume 104. Siam, 2008.

[36] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[37] Chaofeng Hou, Ji Xu, Peng Wang, Wenlai Huang, Xiaowei Wang, Wei Ge, Xianfeng He, Li Guo, and Jinghai Li. Petascale molecular dynamics

simulation of crystalline silicon on tianhe-1a. *The International Journal of High Performance Computing Applications*, 27(3):307–317, 2013.

[38] Young-Kwang Kim, Hong-Kyu Kim, Woo-Sang Jung, and Byeong-Joo Lee. Development and application of ni-ti and ni-al-ti 2nn-meam interatomic potentials for ni-base superalloys. *Computational Materials Science*, 139:225–233, 2017.

[39] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

[40] Christian Lage and Christoph Schwab. Wavelet galerkin algorithms for boundary integral equations. *SIAM Journal on Scientific Computing*, 20(6):2195–2222, 1999.

[41] Byeong-Joo Lee and MI Baskes. Second nearest-neighbor modified embedded-atom-method potential. *Physical Review B*, 62(13):8564, 2000.

[42] John Edward Lennard-Jones. On the determination of molecular fields. ii. from the equation of state of gas. *Proc. Roy. Soc. A*, 106:463–477, 1924.

[43] X-P Li, RW Nunes, and David Vanderbilt. Density-matrix electronic-structure method with linear system-size scaling. *Physical Review B*, 47(16):10891, 1993.

[44] Zhenwei Li, James R Kermode, and Alessandro De Vita. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Physical review letters*, 114(9):096405, 2015.

[45] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[46] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[47] Chr. Møller and M. S. Plesset. Note on an approximation treatment for many-electron systems. *Phys. Rev.*, 46:618–622, Oct 1934.

[48] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *Advances in Neural Information Processing Systems*, pages 440–448, 2012.

[49] Lance J. Nelson, Vidvuds Ozoliņš, C. Shane Reese, Fei Zhou, and Gus L. W. Hart. Cluster expansion made easy with bayesian compressive sensing. *Phys. Rev. B*, 88:155105, Oct 2013.

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[51] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.

[52] Evgeny V Podryabinkin and Alexander V Shapeev. Active learning of linearly parametrized interatomic potentials. *Computational Materials Science*, 140:171–180, 2017.

[53] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian process for machine learning*. MIT press, 2006.

[54] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[55] Julian Schneider, Jan Hamaekers, Samuel T Chill, Søren Smidstrup, Johannes Bulin, Ralph Thesen, Anders Blom, and Kurt Stokbro. Atk-forcefield: a new generation molecular dynamics software package. *Modelling and Simulation in Materials Science and Engineering*, 25(8):085007, 2017.

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

[57] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of machine learning research*, 2(Nov):67–93, 2001.

[58] Eitan Tadmor. The exponential accuracy of fourier and chebyshev differencing methods. *SIAM Journal on Numerical Analysis*, 23(1):1–10, 1986.

[59] Jerry Tersoff. New empirical approach for the structure and energy of covalent systems. *Physical Review B*, 37(12):6991, 1988.

[60] Adri CT Van Duin, Siddharth Dasgupta, Francois Lorant, and William A Goddard. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A*, 105(41):9396–9409, 2001.

[61] Dmitriĭ Aleksandrovich Varshalovich, Anatolij Nikolaevič Moskalev, and Valerii Kel'manovich Khersonskii. *Quantum theory of angular momentum*. World Scientific, 1988.

[62] Hans-Joachim Werner, Frederick R Manby, and Peter J Knowles. Fast linear scaling second-order møller-plesset perturbation theory (mp2) using local and density fitting approximations. *The Journal of chemical physics*, 118(18):8149–8160, 2003.

[63] Hermann Weyl. *The Classical Groups: Their Invariants and Representations*, volume 1. Princeton University Press, 1997.