universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

The Structure Factor as a Global Descriptor in Machine-Learned Force Fields

verfasst von | submitted by

Georg Leonard Gentner BSc

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

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 876
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Physics
Betreut von | Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Georg Kresse

Acknowledgements

First, I would like to express my gratitude to my supervisor, Georg Kresse, for his continuous guidance and support throughout this thesis. Our regular meetings provided invaluable opportunities to discuss progress and clarify ideas. His insights, particularly regarding the underlying physical concepts, greatly shaped my understanding of the topic.

Michael Sahre also took on a supervisory role on a day-to-day basis, and I am grateful for his ongoing support throughout the project. His willingness to discuss ideas, track project progress, plan next steps and work through questions at the whiteboard was instrumental in developing my understanding and in identifying both conceptual issues and implementation bugs.

I would also like to thank Jonathan Lahnsteiner and Ferenc Karsai for their contributions to the initial stages of developing the code, and for guiding me in structuring a project of this scale. Thanks also go to Tobias Hilpert for performing the MACE calculations and to Angela Rittsteuer for her careful proofreading of this thesis and her consistently valuable input.

Furthermore, I would like to thank Angela, Michael and Tobias once again, this time in their role as office colleagues. Their humor and support created a welcoming and enjoyable working environment. In particular, Angela's thoughtful encouragement, sense of humor, and genuinely welcoming attitude kept me motivated and positive, especially during the most challenging phases of the thesis.

I am also grateful to my father, Gernot Singer, whose background in computer science was invaluable for thinking about how to represent the structure of the code using component diagrams. More generally, I am very grateful to my family. Although they are not physicists and were therefore unable to assist with the scientific aspects, they have supported me in many other ways throughout this thesis and my studies leading up to it.

Finally, I owe a special debt of gratitude to my friends and my community—whatever the precise defintion of any community is when considering intersectionality. While it's impossible to thank everyone individually, their encouragement, positivity, and support give me the confidence and backing I need every day to complete projects like this thesis. You always have my back, and I appreciate it so much.

Abstract

English

For machine learned force fields (MLFFs) to achieve *ab initio* accuracy at greatly reduced computational cost, the choice of descriptor for representing atomic structures is critical. While most approaches rely on local descriptors, this work explores the structure factor as a global representation in reciprocal space. Its performance is evaluated for silicon and sodium chloride, where it consistently outperforms the local descriptors used in the Vienna *ab initio* simulation package MLFF. Energy errors of 0.1–0.2 meV atom^{-1} and force errors of 10–20 meV Å^{-1} are achieved using on the order of 10–100 reference configurations. The two-body formulation is already highly expressive, while the three-body extension further improves accuracy at increased cost. Due to its dependence on a fixed reciprocal lattice, the descriptor is limited to fixed-cell simulations. Its accuracy and robustness, however, make it ideal for applications such as phonon calculations and heat transport. This demonstrates the potential of the structure factor as a competitive descriptor for MLFFs.

Deutsch

Damit maschinell gelernte Kraftfelder (MLFFs) bei deutlich reduziertem Rechenaufwand eine *ab initio*-Genauigkeit erreichen, ist die Wahl eines geeigneten Deskriptors zur Beschreibung atomarer Strukturen entscheidend. Während die meisten Ansätze auf lokalen Deskriptoren basieren, untersucht diese Arbeit den Strukturfaktor als globale Darstellung im reziproken Raum. Die Methode wird für Silizium und Natriumchlorid evaluiert und übertrifft dabei konsistent die im Vienna *ab initio* simulation package verwendeten lokalen MLFF-Deskriptoren. Es werden Energiefehler von 0.1–0.2 meV atom^{-1} sowie Kraftfehler von 10–20 meV Å^{-1} mit etwa 10–100 Referenzkonfigurationen erreicht. Bereits die Zwei-Körper-Formulierung erweist sich als hochgradig ausdrucksstark, während die Drei-Körper-Erweiterung die Genauigkeit bei erhöhtem Rechenaufwand weiter verbessert. Aufgrund seiner Abhängigkeit von einem festen reziproken Gitter ist der Deskriptor auf Simulationen mit festen Zellen beschränkt. Seine Genauigkeit und Robustheit machen ihn jedoch ideal für Anwendungen wie Phononenberechnungen und Wärmetransport. Dies verdeutlicht das Potenzial des Strukturfaktors als wettbewerbsfähiger Deskriptor für MLFFs.

Preface

A preface serves to establish a personal connection to the work. In the natural sciences, this is considerably less common than in fields such as the social sciences, where the objects of study are often more closely tied to the researchers' everyday experience. From the perspective of good scientific practice, however, I believe that such positionality should be made explicit in all disciplines. In particular, in a time where academic careers are increasingly precarious, contracts are becoming shorter, and publication pressure is steadily increasing, it would be revealing if physics papers explicitly stated that their main motivation was simply to produce publications for career-related reasons, highlighting how such developments can be detrimental to good science.

This is where I position myself: as someone who values and practices scientific methodology, while remaining aware of the historical entanglements and responsibilities of science, including the ways in which it has been misused and has failed to meet them. At the same time, these shortcomings have often been uncovered and critically examined through scientific work itself across different disciplines.

Therefore, my primary perspective in this work is that of a scientist engaging with computational methods in a broader physical context. Within physics, my initial interest lies in condensed matter physics, in particular in the mathematical elegance of the description of periodic crystal lattices. Together with a strong affinity for programming, which naturally led me towards computational materials physics. From there, my interest developed further through a more general curiosity about approximations: how accurate they can be, where their limits lie, and how they might be improved. In parallel, the rapid development of machine learning in science and its increasing impact more broadly provided a timely motivation to explore these ideas within a modern computational framework. This topic was ultimately shaped by the combination of these interests and the suggestion of my supervisor, Georg Kresse.

As already stated, however, my general enthusiasm for science and its practice takes precedence over any specific topic within physics. Article 17 of the Austrian Basic Law on the General Rights of Citizens states: *„Die Wissenschaft und ihre Lehre ist frei.“* (“Science and its teaching is free.”) This formulation is interesting in that it uses the singular form, implying that science and teaching are inseparably connected and should be understood as one unified practice. This aspect is also important to me, as throughout my studies I have placed great emphasis on not only acquiring knowledge

but also being able to communicate and pass it on. Learning something solely for myself has never been sufficient motivation; the freedom provided by a master's thesis is therefore an opportunity to engage with this aspect more fully.

Accordingly, chapter 1 motivates the central research question of whether the structure factor is a suitable descriptor from two perspectives: first, starting from basic physical principles accessible even to a general audience, and second, from the perspective of current research in machine-learned force fields for readers familiar with the field.

Chapter 2 develops the theoretical background, beginning with the fundamental question of what machine learning is and why it is useful, with the aim of providing an accessible entry point even for readers outside the field. At the same time, the presentation remains concise in accordance with scientific standards and introduces all necessary concepts, including the structure factor, which is central to this thesis.

Chapter 3 describes the methodology and the implementation of the computational framework used to assess the potential of the descriptor. Chapter 4 then investigates model and descriptor properties in more detail, providing the basis for the main results presented in chapter 5. By comparison with benchmark systems, the performance of the structure factor is ultimately evaluated. These findings are summarized in chapter 6, which also outlines possible future research directions and applications.

A final note concerns notation. This thesis alternates between the use of "I", "we", and the passive voice. These choices are deliberate and carry specific meaning: "I" is used when referring to methodological decisions and organizational aspects of the work; "we" is used in the context of standard derivations, in particular to denote intermediate mathematical steps; and the passive voice is employed when presenting results and observations, in order to emphasize their objective character.

Contents

1 Research Question

In this work, I investigate whether the structure factor can serve as a suitable descriptor for machine learning force fields (MLFFs). This introduction aims to motivate the research question and outline the conceptual framework for assessing descriptor suitability in the context of MLFFs. The motivation is developed along two complementary lines of argument. First, the question is approached from the perspective of the underlying physical concepts and thus situated within the broader context of physics. Second, it is motivated by its relation to current developments in the field of MLFFs.

1.1 Physical Motivation

In this section, I aim to motivate the research question starting from the very basics of physics such as Newton's equation of motion $F = ma$, where F is the force on an object, m its mass, and a its acceleration. Integrating the acceleration twice over time yields the object's trajectory and thus its time evolution. This principle extends to systems of many interacting bodies, ranging from macroscopic objects such as planets to microscopic ones such as atoms. For many-body systems the eqautions of motion cannot be solved analytically and therefore require iterative numerical integration of Newton's equations. In the atomic case, this numerical procedure is referred to as molecular dynamics (MD). MD enables, at least over short timescales, the simulation of systems composed of many atoms such as solids or liquids, allowing the study of their atomic trajectories and thermodynamic behavior. At each MD step, it is crucial to know the interatomic forces, which depend on the atomic positions. A function mapping positions to forces is commonly referred to as a force field (FF). Within classical mechanics, the forces acting on atoms are obtained as the negative gradient of a potential energy surface (PES), which describes the system's energy as a function of atomic positions.

In computational applications, such as evaluating the PES and its derivatives, accuracy and computational cost must be balanced. At one end of this spectrum lies *density functional theory* (DFT), which is an accurate yet computationally expensive method [1, 2]. DFT is an *ab initio* quantum mechanical method, i.e., it applies the fundamental laws of quantum mechanics and is, in principle, exact. Within the Born–Oppenheimer

approximation, where electronic and nuclear motion are decoupled, it provides an explicit electronic structure description, from which interatomic forces are derived. In practice, however, DFT relies on approximate exchange–correlation functionals. Its cost grows rapidly with system size becoming prohibitive for finite-temperature MD, where thermal atomic displacements reduce symmetry and require large supercells.

At the opposite end of the spectrum are empirical force fields, such as the well-known Lennard–Jones (LJ) potential [3, 4], which approximates the interatomic potential using a simple combination of inverse-power terms with a small number of parameters. In contrast to quantum mechanical methods such as DFT, empirical force fields do not explicitly describe the electronic structure, but instead approximate interatomic interactions through predefined functional forms. While computationally efficient, such empirical models typically fail to capture complex, many-body effects essential for an accurate physical description.

The rise of machine learning (ML) has opened new possibilities in physics and offers a promising route to bridge this gap. A natural question arises: can we leverage ML to train a machine-learned force field (MLFF) from a limited number of accurate but expensive *ab initio* data points, such as those obtained from DFT, such that the MLFF reproduces the reference forces at a fraction of the computational cost? In contrast to empirical force fields with predefined functional forms, ML models can represent highly flexible and complex functional relationships. This enables the description of intricate many-body interactions. Recent research [5] has shown that this is indeed possible, and several successful MLFF approaches have been developed, as discussed in section 1.2.

Training an ML model to predict a force field requires an appropriate representation, called a *descriptor*, that encodes the atomic environment of different configurations. In this work, the two- and three-body structure factor are investigated as potential descriptors. The structure factor depends on the atomic positions $\mathbf{r}_j$ and $\mathbf{r}_k$ and the wave vector (or reciprocal lattice vector) $\mathbf{q}$. For simplicity, only the two-body form is introduced here,

$$S(\mathbf{q}) \propto \sum_{j=1}^{N} \sum_{k=1}^{N} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)}\,, \tag{1.1}$$

whereas the three-body form will be treated in section 4.3. Originating from scattering theory, the structure factor captures spatial correlations and symmetry features of atomic arrangements. This makes it a physically motivated and potentially efficient descriptor for MLFFs.

After positioning the research question within the field of physics, we can now return to its core. Is the structure factor a suitable descriptor for MLFFs? This raises the follow-

up question of how we can assess such suitability. The above motivation suggests that an ML model should reproduce the *ab initio* forces as accurately as possible while relying on a minimal number of computationally expensive *ab initio* training data. To evaluate the structure factor in this regard, it must be compared to existing ML models that serve as established benchmarks. It is also crucial to determine whether differences in performance can be attributed solely to the choice of the descriptor, or whether they originate from other elements of the model itself. Therefore, the following section reviews current MLFF approaches that define the methodological framework for both comparison and evaluation of the proposed model.

1.2 Current Research

The modern development of MLFFs is often traced back to the work of Behler and Parrinello, who in 2007 introduced high-dimensional neural network potentials (HDNNPs) [5–7]. In their framework, the total energy of a system is expressed as a sum of atomic contributions predicted from symmetry-invariant descriptors of the local atomic environment [8]. This formulation allows complex many-body interactions to be represented while maintaining size extensivity and transferability [5, 8]. These works demonstrated that neural network models combined with suitable descriptors can reproduce DFT energies and forces with high accuracy for materials such as bulk silicon, while being several orders of magnitude faster than direct DFT calculations [5, 6].

A parallel and equally influential line of research was the introduction of kernel-based methods, most notably the Gaussian Approximation Potential (GAP) developed by Bartók *et al.* [9]. GAP employs Gaussian process regression to predict energies and forces from the similarity of local atomic environments, demonstrating that kernel regression models can achieve accuracy comparable to neural network potentials. Early implementations relied on bispectrum descriptors [10], which provide permutation-, translation-, and rotation-invariant representations through spherical harmonic expansions. This approach was later refined by the Smooth Overlap of Atomic Positions (SOAP) descriptor [11], which represents environments via continuous atomic densities and measures the rotationally averaged overlap between them. The transition from bispectrum to SOAP illustrates the crucial role that descriptor design plays in determining the accuracy and efficiency of MLFF models.

All MLFF approaches require reference training data, typically obtained from *ab initio* calculations such as DFT. Since these calculations are computationally expensive, an important challenge is to select a representative set of configurations that efficiently covers the relevant regions of configuration space. One solution is on-the-fly or active learning, where the training set is constructed adaptively during molecular dynamics simulations. Such schemes are implemented in the MLFF framework of the Vienna *ab initio* Simulation Package (VASP) [12–14], developed by Jinnouchi *et al.* [15–

17]. In this framework, atomic environments are described using local descriptors based on radial basis functions (spherical Bessel functions) and angular correlations (Legendre polynomials), which are conceptually similar to SOAP-type descriptors in that they encode local atomic environments in a symmetry-invariant manner, albeit with a different mathematical construction.

In parallel, neural network architectures have advanced rapidly, with modern equivariant message-passing models achieving state-of-the-art accuracy [5, 18]. A prominent example is the Message Passing Atomic Cluster Expansion (MACE), which extends conventional message-passing networks by incorporating higher-body interactions. Rather than starting from scratch, MACE builds upon the Atomic Cluster Expansion (ACE) framework and learns the corresponding expansion coefficients during training [18, 19]. While such models often achieve very high accuracy, their computational cost can be higher than that of simpler kernel-based approaches, though the exact cost depends on the dataset size and complexity.

The aforementioned models rely on *local* atomic descriptors that encode the environment of each atom within a finite cutoff. This locality assumption enforces size-extensivity and enables efficient modeling, but can make the accurate treatment of non-local effects, such as long-range electrostatics or collective correlations, challenging. While increasing the local cutoff or stacking message-passing layers can in principle capture longer-range interactions, these approaches are computationally costly and may still fail for truly non-local interactions. Modern strategies explicitly address long-range physics using partitioning approaches or latent representations. Notable examples include the Long-Distance Equivariant (LODE) descriptor [20], latent-variable Ewald approaches [21, 22], and recent equivariant long-range message-passing networks [23], which extend local or message-passing models to capture electrostatics, dispersion, and other non-local effects efficiently.

In contrast, *global* descriptors represent the entire atomic configuration as a single feature vector and can therefore encode long-range correlations more directly. However, this often comes at the expense of higher computational cost and reduced scalability with system size. A well-known example is the Coulomb matrix [24], which encodes pairwise Coulomb interactions between all atoms in a molecule or cluster. Other global or semi-global descriptors commonly used in molecular chemistry include Bag of Bonds [25] and Faber–Christensen–Huang–Lilienfeld (FCHL) representation [26, 27], which are mainly applied to small molecules and chemical space exploration.

This trade-off between locality and the efficient representation of long-range correlations motivates the exploration of alternative descriptors that can capture global structural information while remaining computationally tractable.

Global kernel approaches have also been developed for learning interatomic force fields. The Gradient-Domain Machine Learning (GDML) and symmetric GDML (sGDML)

frameworks [28, 29] achieve high accuracy for molecular systems while requiring relatively small training datasets by explicitly incorporating physical symmetries within a kernel-based force-learning framework. Recent developments such as Bravais-Inspired Gradient-Domain Machine Learning (BIGDML) [30] further demonstrate that accurate force fields can be constructed from as few as 10–200 reference geometries by additionally exploiting crystalline symmetries and periodicity.

Importantly, descriptors that encode similar physical correlations do not necessarily lead to the same performance in practice. The specific mathematical representation of the atomic environment strongly influences how efficiently a machine learning model can learn the underlying interactions [19]. Consequently, the design of descriptors remains an active area of research.

Despite the large variety of existing descriptors, the structure factor has not been widely used as a descriptor for MLFFs. Closely related density-based or Fourier-space representations aimed at incorporating long-range interactions have recently been explored, for example by Faller *et al.* [31], yet the structure factor itself has received comparatively little attention in this context. Since the structure factor naturally encodes spatial correlations, periodicity, and collective ordering in atomic configurations, it provides a physically well-motivated representation of the structure. In particular, its formulation in reciprocal space makes it a natural candidate for capturing long-range correlations in periodic systems.

Motivated by this perspective, the present work systematically investigates the structure factor, including its two- and three-body variants, as a potential global descriptor for MLFFs. By comparing its performance and data efficiency with established approaches such as VASP-MLFF and MACE, this study aims to assess whether a global Fourier-space representation can provide an accurate and efficient alternative to conventional descriptors.

2 Theoretical Background

This chapter introduces the theoretical concepts relevant to this work. I begin from a computer science perspective by discussing the basic principles of machine learning. In the second part, I bridge the gap to physics through machine-learned force fields, while the final section focuses on the structure factor as a purely physical quantity.

2.1 Machine Learning

As a starting point, this section introduces the fundamental concepts and core terminology of machine learning (ML). Subsequently, I apply these ideas to different model classes. For a more comprehensive and formal treatment, see Bishop [32], on which the notation and structure of this section are loosely based.

When modeling relationships between quantities, we can use either a rule-based or a data-driven approach. In science, we typically aim to discover the underlying laws and rules that govern these relationships. For highly complex systems with many interdependent variables, however, this becomes exceedingly difficult or even impossible. In such cases, data-driven models offer a powerful alternative.

A simple illustrative example is an email spam filter. It would be practically impossible to manually specify all rules that distinguish spam from non-spam messages. Instead, it is far more effective to provide a dataset of labeled example emails marked as "spam" or "not spam" and allow the computer to learn the relevant underlying patterns directly from the data. This approach of modeling from examples rather than predefined rules is what we refer to as *machine learning*.

Machine learning can be broadly divided into two categories. *Supervised learning* aims to learn a mapping from inputs to known target values based on labeled training data. In contrast, *unsupervised learning* seeks to uncover hidden patterns or clusters in data without access to explicit labels or target values. We will focus on supervised learning, whose objective can be written as follows

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \sim \{(\mathbf{x}_i, t_i)\}_{i=1}^{N}\,, \tag{2.1}$$

where the symbol $\sim$ indicates that the model output y_i approximates the true target t_i as close as possible according to a performance metric. If t_i is discrete, it is called *label* and the task is referred to as a *classification problem* (such as the example of spam detection). If t_i is continuous, however, we refer to it as *target* within a *regression problem*. This work focuses on regression problems. The target t_i, and therefore its approximation y_i, can be a vector. Here, we consider scalars for simplicity. $\{\mathbf{x}_i\}$ represents the input data, where the index i denotes the i-th of the N data points. The set of data points or *observations* $\{(\mathbf{x}_i, t_i)\}$ is assembled during the *data collection* and represents the *ground truth* for the ML model. The general modeling pipeline can be written as

$$\mathbf{x}_i \;\mapsto\; \boldsymbol{\varphi}(\mathbf{x}_i) \;\mapsto\; y_i = f\big(\boldsymbol{\varphi}(\mathbf{x}_i);\, \mathbf{w},\, \boldsymbol{\theta}\big)\,. \tag{2.2}$$

The first mapping represents *pre-processing* or *feature extraction*, where the raw input data are transformed into features that provide a more suitable representation for the learning algorithm. Since the function $\boldsymbol{\varphi}(\mathbf{x}_i)$ serves a descriptive purpose, it is referred to as a *descriptor* or *feature map*. The function f denotes the model itself, parameterized by weights $\mathbf{w}$ and hyperparameters $\boldsymbol{\theta}$. The weights are optimized with respect to a chosen *error* or *loss function*[1] on a subset of the data, called the *training set*, while the hyperparameters, which control the model complexity, are typically selected based on a separate *validation set*.

Finally, a third independent subset (the *test set*) is used to evaluate the model performance by comparing the predicted outputs y_i to the true targets t_i. A model that performs well on both the training data and on the previously unseen test data is said to *generalize*. In contrast, if it performs well on the training data but poorly on new data, it exhibits *overfitting*. It has learned noise rather than the underlying pattern. Its counterpart, *underfitting*, occurs when the model is too simple to capture the true relationship, for instance when it predicts a nearly constant output (such as the mean value) regardless of the input.

It should be noted that terminology in the literature is not always used consistently. Certain concepts are defined more broadly or narrowly depending on the source, which can be confusing, particularly for readers new to the field. In this work, I therefore follow a consistent usage and interpretation of terms as introduced above.

[1] The terms error and loss function are often used interchangeably, although they are conceptually distinct. The error function quantifies the deviation between prediction and truth according to a chosen metric, whereas the loss function may additionally assign different weights to specific types of errors (e.g., false positives vs. false negatives in medical tests). This distinction, however, is not relevant for the present work.

2.1.1 Linear Regression

Previously, I introduced the general concepts of machine learning. We now examine these ideas more closely by considering specific model classes, starting with the simplest case: *linear regression*.

For a single input vector $\mathbf{x} \in \mathbb{R}^d$ with d features, the output y is computed using a linear function of the form

$$y = w_0 + \sum_{i=1}^{d} w_i x_i = w_0 + \mathbf{w}^\top \mathbf{x}\,, \tag{2.3}$$

where $\mathbf{w} \in \mathbb{R}^d$ contains the weights associated with each feature, and w_0 is a bias term that allows a constant offset to be fitted into the data. When we have N input vectors $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$, we can collect them in a matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, where each row corresponds to one data point. The corresponding outputs can be collected in a vector $\mathbf{y} \in \mathbb{R}^N$. Neglecting the bias term for now, the model can be written compactly as

$$\mathbf{y} = \mathbf{X}\mathbf{w}\,, \tag{2.4}$$

where each output y_i is obtained from the linear combination of features of the corresponding row of $\mathbf{X}$. The linearity in the input variables $\mathbf{X}$ of the model in Eq. (2.4) significantly limits the complexity of our model. A solution to increase the flexibility is to map each input vector onto a set of nonlinear features:

$$\boldsymbol{\varphi}\colon \mathbb{R}^d \to \mathbb{R}^M\,, \quad \mathbf{x}_i \mapsto \boldsymbol{\varphi}(\mathbf{x}_i) \tag{2.5}$$

or in the compact notation in which all input data points are processed at once:

$$\boldsymbol{\Phi}\colon \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times M}\,, \quad \mathbf{X} \mapsto \begin{bmatrix} \boldsymbol{\varphi}(\mathbf{x}_1)^\top \\ \vdots \\ \boldsymbol{\varphi}(\mathbf{x}_N)^\top \end{bmatrix}\,, \tag{2.6}$$

where $\boldsymbol{\varphi}$ is composed of multiple *basis functions* $\varphi_j(\mathbf{x}_i)$. The mapping increases the effective dimensionality from d to M, thereby enabling the model to capture more complex relationships. Substituting this feature map into Eq. (2.4) gives

$$\mathbf{y} = \boldsymbol{\Phi}(\mathbf{X})\mathbf{w}\,. \tag{2.7}$$

The so-called *design matrix* $\mathbf{\Phi}$ with elements $\Phi_{ij} = \varphi_j(\mathbf{x}_i)$ collects the feature representations of all input vectors, where each row corresponds to one data point and each column to one basis function.

To include a constant offset in the model, we can extend the design matrix to $\mathbf{\Phi} \leftarrow [\mathbf{1}, \mathbf{\Phi}]$, which is equivalent to introducing an additional constant basis function $\varphi_0(\mathbf{x}_i) = 1$. By redefining the weight vector accordingly, $\mathbf{w} \leftarrow [w_0, w_1, \ldots, w_M]^\top$, the bias term w_0 can be absorbed into $\mathbf{w}$. Alternatively, the bias term can be eliminated by centering the data, i.e., subtracting the mean from both the input features and the target values, so that the model is fitted around zero and no explicit offset is required.

This model is linear in the newly mapped feature space and remains so in the weights. The optimal weights are obtained by minimizing a specific loss function. A common choice is the least-squares loss (for the motivation of this choice see, e.g., [32]), which measures the ℓ_2-norm between the predicted outputs $f(\boldsymbol{\varphi}(\mathbf{x}_i); \mathbf{w})$ and the targets t_i

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \left(t_i - \mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}_i) \right)^2 = \frac{1}{2} \|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2 . \tag{2.8}$$

To minimize this function, we set its gradient with respect to $\mathbf{w}$ to zero:

$$\nabla \mathcal{L}(\mathbf{w}) = -\mathbf{\Phi}^\top (\mathbf{t} - \mathbf{\Phi}\mathbf{w}) \stackrel{!}{=} 0 . \tag{2.9}$$

Solving for the weight vector yields the so-called *normal equation*:

$$\mathbf{w}^* = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{t} , \tag{2.10}$$

where $\mathbf{w}^*$ denotes the specific set of parameters that minimizes the loss. Although this equation provides a closed-form solution for $\mathbf{w}^*$, explicitly computing the matrix inverse can be numerically unstable, particularly if $\mathbf{\Phi}^\top \mathbf{\Phi}$ is ill-conditioned or close to singular. While methods such as singular value decomposition (SVD, see section A.1) or Cholesky decomposition are computationally comparable or even more expensive, they provide significantly improved numerical stability. Alternatively, iterative optimization schemes such as gradient descent can be used for large-scale problems (see, e.g., [33]).

We still need to define the model complexity by setting the dimensionality of the feature space through the hyperparameter M in Eq. (2.5). If M is too small, the model lacks the capacity to capture the underlying relationships in the data, resulting in underfitting. If M is too large, the model adapts too closely to the training data and fails to generalize, resulting in overfitting. A common approach to prevent

overfitting is to introduce a regularization term into the loss function that penalizes overly large weight magnitudes. This effectively constrains the model's flexibility. Intuitively, large weights make the model overly sensitive to noise in the input data, causing oscillatory behavior and overfitting. Thus, the regularization term acts as a smoothness constraint, promoting simpler models that generalize better to unseen data (see section A.1).

$$\mathcal{L}_{\text{total}}(\mathbf{w}) = \mathcal{L}(\mathbf{w}) + \lambda\,\mathcal{R}(\mathbf{w})\,, \tag{2.11}$$

where $\mathcal{L}(\mathbf{w})$ is the original loss function, $\mathcal{R}(\mathbf{w})$ is a regularization function, and $\lambda > 0$ is the regularization parameter controlling the strength of the regularization.

The so-called *ridge regression* [34] or *Tikhonov regularization* [35] corresponds to the specific choice of an ℓ_2-penalty for the weights:

$$\mathcal{R}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2\,. \tag{2.12}$$

Consequently, our total loss function in Eq. (2.11) becomes

$$\mathcal{L}_{\text{ridge}}(\mathbf{w}) = \frac{1}{2}\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2\,, \tag{2.13}$$

which modifies the normal equation (Eq. (2.10)) of linear regression to

$$\mathbf{w}^* = \left(\mathbf{\Phi}^\top\mathbf{\Phi} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^\top\mathbf{t}\,. \tag{2.14}$$

In this section, we addressed the limitations of a linear model by introducing a feature mapping (Eq. (2.5)) to a potentially higher-dimensional space, allowing the modeling of more complex relationships. Then, we incorporated regularization in the form of ridge regression, leading to the solution for the optimal weights in Eq. (2.14). The introduction of regularization served two purposes. First, it addressed the numerical instability of the original normal equation (Eq. (2.10)). Second, it controlled model complexity.

Neither objective, however, is fully resolved by regularization alone. Regarding numerical stability, regularization does not truly resolve the problem but rather transforms it into a more stable one. Mathematically, the original problem remains unchanged, but for practical applications, especially in physics, this approach is reasonable, and the solution converges to the original problem as $\lambda \to 0$. Regarding model complexity, regularization only partially addresses the issue. The fundamental problem of model

selection is not eliminated but shifted, from choosing the number of basis functions M to determining an appropriate value for the regularization parameter λ. Simply selecting a large M and relying on regularization to prevent overfitting is unsatisfactory. These limitations can be partly overcome by considering another model class, *kernel regression*, which will be the focus of the next section.

2.1.2 Kernel Regression

To motivate this approach, we start by revisiting the regularized normal equation in Eq. (2.14) and analyzing their dimensionality. For the design matrix is $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$ (see Eq. (2.6)) and therefore $\mathbf{\Phi}^\top \mathbf{\Phi} \in \mathbb{R}^{M \times M}$, where N denotes the number of observations and M the number of features. The motivation for introducing feature mappings was to project the data into a higher-dimensional space to capture more complex relationships. However, if we wish to map into a very high-dimensional space or even an infinite-dimensional one, computing $\mathbf{\Phi}^\top \mathbf{\Phi}$ becomes infeasible, since its dimensionality grows with M. In such cases it would be more convenient to work with an expression such as $\mathbf{\Phi}\mathbf{\Phi}^\top \in \mathbb{R}^{N \times N}$ instead.

To achieve this, we apply an algebraic reformulation known as the *kernel trick* by expressing the optimal weights as

$$\mathbf{w}^* = \mathbf{\Phi}^\top \boldsymbol{\alpha} \,. \tag{2.15}$$

Substituting this into the regularized normal equation (Eq. (2.14)) yields

$$(\mathbf{\Phi}^\top \mathbf{\Phi} + \lambda \mathbf{I})\, \mathbf{\Phi}^\top \boldsymbol{\alpha} = \mathbf{\Phi}^\top \mathbf{t} \,. \tag{2.16}$$

Using associativity and the commutation $\lambda \mathbf{I} \mathbf{\Phi}^\top = \mathbf{\Phi}^\top \lambda \mathbf{I}$ gives

$$\mathbf{\Phi}^\top (\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda \mathbf{I})\, \boldsymbol{\alpha} = \mathbf{\Phi}^\top \mathbf{t} \,. \tag{2.17}$$

Finally, multiplying both sides from the left by the (Moore–Penrose) pseudo-inverse $(\mathbf{\Phi}\mathbf{\Phi}^\top)^{-1}\mathbf{\Phi}$ [33, 36] leads to a new but equivalent formulation of the regularized normal equation

$$\boldsymbol{\alpha} = (\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda \mathbf{I})^{-1}\, \mathbf{t} \,, \tag{2.18}$$

where the so-called *Gram matrix* or *kernel matrix* is defined as

$$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top, \quad \text{with} \quad K_{ij} = \boldsymbol{\varphi}(\mathbf{x}_i)^\top \boldsymbol{\varphi}(\mathbf{x}_j) =: k(\mathbf{x}_i, \mathbf{x}_j)\,. \tag{2.19}$$

Besides being computationally advantageous when $N < M$, this formulation provides two key conceptual benefits. First, a major advantage is that we no longer need to compute the feature maps $\boldsymbol{\varphi}(\mathbf{x})$ explicitly. Instead, it suffices to evaluate their inner products directly via a *kernel function* $k(\mathbf{x}_i, \mathbf{x}_{i_b})$. This enables computations in high-dimensional or even infinite-dimensional feature spaces without explicitly constructing them, thereby allowing for models of substantially higher complexity at moderate computational cost. In this sense, the kernel formulation represents the so-called *dual representation* of the regression problem, in contrast to the *primal representation*, which directly solves for the weight vector $\mathbf{w}$ in feature space.

Second, the kernel framework conceptually distinguishes between the training set $\{\mathbf{x}_i\}$ and the basis set $\{\mathbf{x}_{i_b}\}$, even though in practical applications these sets are often identical, as data acquisition is typically computationally or experimentally expensive. To emphasize this distinction, the index j in the kernel notation is replaced by i_b. Extending this idea, the kernel function can be understood as a similarity measure between a training sample $\mathbf{x}_i$ and a basis point $\mathbf{x}_{i_b}$. This interpretation also provides a direct link back to linear regression, where the kernel function can be viewed as a basis function evaluated at $\mathbf{x}_i$, i.e., $k(\mathbf{x}_i, \mathbf{x}_{i_b}) \equiv \tilde{\boldsymbol{\varphi}}_{i_b}(\mathbf{x}_i)$. In this sense, each kernel function implicitly defines a feature space spanned by the basis functions $\tilde{\boldsymbol{\varphi}}_{i_b}$ associated with the chosen kernel function.

Typical choices for kernel functions include the polynomial kernel

$$k_{\text{poly}}(\mathbf{x}_i, \mathbf{x}_{i_b}) = (\mathbf{x}_i^\top \mathbf{x}_{i_b} + c)^\chi\,, \tag{2.20}$$

where $c \geq 0$ is a bias term and χ the polynomial degree, and the *Gaussian* or *radial basis function (RBF)* kernel

$$k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_{i_b}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_{i_b}\|^2}{2\sigma^2}\right), \tag{2.21}$$

where σ controls the width of the Gaussian. The Gaussian kernel is particularly interesting since it can be interpreted as representing an infinite-dimensional feature space spanned by all polynomial orders. This is possible given that a Gaussian function can be expanded into an infinite series of polynomial terms.

As shown in Eq. (2.15), primal and dual representation are closely related and can be transformed into each other. This becomes particularly clear for the simple linear

kernel

$$k(\mathbf{x}_i, \mathbf{x}_{i_b}) = \mathbf{x}_i^\top \mathbf{x}_{i_b} \,, \tag{2.22}$$

for which, neglecting regularization, the model prediction becomes

$$y_i(\mathbf{x}_i) = \sum_{i_b=1}^{N} \alpha_{i_b} \, k(\mathbf{x}_i, \mathbf{x}_{i_b}) = \sum_{i_b=1}^{N} \alpha_{i_b} \, \mathbf{x}_i^\top \mathbf{x}_{i_b} = \sum_{i_b=1}^{N} \alpha_{i_b} \sum_{j=1}^{d} x_{i,j} x_{i_b,j} = \sum_{j=1}^{d} w_j x_{i,j} \,, \tag{2.23}$$

where we rewrote $w_j = \sum_{i_b=1}^{N} \alpha_{i_b} x_{i_b,m}$. The equation above is equivalent to Eq. (2.3). Thus, kernel regression with a linear kernel reduces to the standard linear regression model in Eq. (2.3). This equivalence in the above equation also gives insight into the relationship between the number of features vs. the size of the basis set and therefore, the number of fit parameters, which will be discussed in section 4.1.

Since this formulation was derived from ridge regression, the resulting model is referred to as *kernel ridge regression*. The same dualization principle, however, applies to other regularized or unregularized linear models. For the kernel formulation to be well-defined, the kernel function $k(\cdot, \cdot)$ is typically required to be symmetric and positive semidefinite, ensuring that the resulting kernel matrix $\mathbf{K}$ defines a valid inner product in feature space.

In summary, kernel regression replaces the explicit feature mapping, $\mathbf{\Phi}$, with a kernel function, $k(\mathbf{x}_i, \mathbf{x}_{i_b})$, that implicitly defines inner products in a (possibly infinite-dimensional) feature space. The computational benefit depends on the relative sizes of N and M. If $N < M$, working in the dual (kernel) form is advantageous, and if $N > M$, solving the primal form directly is often more efficient.

At the end of the linear regression section, I mentioned that kernel methods can *partly overcome* the limitations of linear regression. This characterization is deliberate. Kernel regression addresses model complexity more effectively, since it allows mapping into high, potentially infinite, dimensional feature spaces without explicitly defining M basis functions. It also improves numerical stability and efficiency by operating on the $N \times N$ kernel matrix instead of the $M \times M$ matrix $\mathbf{\Phi}^\top \mathbf{\Phi}$. The approach, however, does not eliminate all fundamental challenges. The choice of kernel type and its hyperparameters (e.g., polynomial degree or Gaussian width) remains crucial, and overfitting can still occur if these parameters are not properly controlled. Thus, kernel regression offers a more flexible and often more stable framework than linear regression, but it still requires careful model selection and regularization.

2.1.3 Neural Networks

In the previous model classes, a central question has been how to represent complex relationships while maintaining generalization and avoiding overfitting. In other words, how can we strike a balance between simplicity and necessary complexity? In the regression methods, we introduced non-linearity through either basis functions (feature mapping) or kernel functions. In both cases, however, the type of non-linearity is predefined by the chosen functions, and the model merely learns a linear combination of them. Neural networks (NNs), another class of models, aim to make this process more adaptive, enabling the representation of highly non-linear and complex relationships. Nevertheless, NNs remain subject to the same fundamental considerations regarding model complexity, hyperparameter selection, and regularization. These aspects are reflected in the architectural design of neural networks. The following section will provide a general introduction to NNs and outline these aspects. Since NNs themselves are not the focus of this work, but the MACE model, a neural-network-based MLFF method, is used as a benchmark, the following overview will remain brief.

To achieve the idea of adaptive non-linearity, two key steps are taken. The first is the introduction of a non-linear *activation function* $\sigma(\cdot)$ into a linear model, forming a so-called *perceptron*:

$$y = \sigma\left(\sum_{i=1}^{d} w_i x_i + b\right) , \tag{2.24}$$

where x_i are the input variables (for simplicity scalars are considered), w_i the corresponding weights, and b a bias term. Conceptually, this is similar to linear regression, except that the output is transformed by a non-linear function σ, enabling the model to approximate complex, non-linear relationships. In addition, the activation function should be differentiable and computationally efficient to evaluate.

The second step is to stack several such layers, where the outputs of one layer become the inputs of the next:

$$\mathbf{h}^{(l)} = \sigma^{(l)}\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) , \tag{2.25}$$

with $\mathbf{h}^{(0)} = \mathbf{x}$.

Each layer l contains multiple computational units, called *neurons*, each connected to all neurons in the previous layer $l-1$ via the weights in $\mathbf{W}^{(l)}$. Hence, $\mathbf{W}^{(l)}$ can be interpreted as the weight matrix encoding the connectivity pattern between adjacent layers. This fully connected structure can be illustrated as a topological mapping

chain:

$$\mathbf{x} \xmapsto{\mathbf{W}^{(1)},\sigma^{(1)}} \mathbf{h}^{(1)} \xmapsto{\mathbf{W}^{(2)},\sigma^{(2)}} \mathbf{h}^{(2)} \xmapsto{\cdots} \mathbf{h}^{(L-1)} \xmapsto{\mathbf{W}^{(L)},\sigma^{(L)}} \mathbf{y}\,. \tag{2.26}$$

Each arrow represents a linear transformation followed by a non-linear activation. By chaining several such transformations, the network can adaptively learn representations of the input data without relying on manually designed basis or kernel functions.

The optimal parameters, denoted by $\mathbf{W}^*$ and $\mathbf{b}^*$, are determined by minimizing a loss function analogous to the least-squares loss in Eq. (2.8). Due to the non-linear composition of layers, however, no closed-form solution such as the normal equation exists. Instead, the parameters are updated iteratively using gradient descent, where gradients are efficiently computed via the *backpropagation algorithm*, which propagates the error from the output layer backward through the network to compute partial derivatives with respect to each parameter. Training proceeds iteratively. To avoid excessive computational demands when using all observations at once, the data are typically divided into *mini-batches*, with model parameters updated after processing each batch. The model parameters are then updated after processing each mini-batch. Once all mini-batches have been processed, one *epoch* is completed, and this process of iterative optimization is repeated until convergence.

Since NNs contain a large number of parameters and can easily overfit, regularization remains crucial. A common approach is *weight decay*, which adds a penalty term proportional to the squared magnitude of the weights, analogous to ridge regularization. Another widely used technique is *dropout*, where, during training, individual neurons are randomly deactivated (their output set to zero). This prevents the network from relying too heavily on specific neurons and therefore becoming too sensitive to small changes, encouraging more robust and redundant feature representations.

In summary, NNs provide a flexible and powerful framework for modeling complex non-linear relationships. Compared to regression-based methods, they offer greater expressive power and can autonomously learn relevant features from data. This advantage, however, comes at the cost of higher computational demands, more hyperparameters, and a challenging optimization landscape.

This chapter emphasized the key challenges of machine learning: choosing an appropriate model class, controlling model complexity, and ensuring generalization through regularization and optimization. These principles form the foundation for applying machine learning to physical systems. The next chapter illustrates how these concepts extend to MLFFs, which aim to approximate interatomic potentials with high accuracy while retaining computational efficiency.

2.2 Machine Learned Force Fields

While the previous section introduced the general principles of machine learning, we now apply these concepts to MLFFs (machine-learned force fields). The objective of MLFFs is to learn the PES, and equivalently the FF, from atomic configurations in order to reduce the computational cost of expensive *ab initio* calculations. The data obtained from *ab initio* simulations thus serve as our ground truth or observations. The mapping that MLFFs aim to learn can be written, in analogy to Eq. (2.2), as

$$\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\} \;\mapsto\; \big(E,\, \{\mathbf{F}_i\}_{i=1}^{N}\big) = f\big(\boldsymbol{\varphi}(\mathbf{R});\, \mathbf{w},\, \boldsymbol{\theta}\big)\,, \tag{2.27}$$

where $\mathbf{R}$ denotes the atomic positions, $\boldsymbol{\varphi}(\mathbf{R})$ represents the descriptor or representation of the atomic configuration, E is the system's energy, F_i the force on atom i and $\mathbf{w}$ and $\boldsymbol{\theta}$ are the model parameters as before. Two aspects of the output quantities are important to note. First, the total energy is defined only up to an additive constant. Consequently, when referring to configuration energies, we effectively mean energy differences with respect to a reference configuration. Second, the PES and FF are not independent quantities but are connected via the negative gradient,

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} E\,, \tag{2.28}$$

where $\nabla_{\mathbf{r}_i}$ indicates the gradient with respect to $\mathbf{r}_i$. If both quantities are to be learned simultaneously, this relationship must be consistently embedded into the model, which will be discussed later in subsection 2.2.3.

Considering the input variables, i.e., the atomic positions, their representation is captured by the intermediate mapping $\boldsymbol{\varphi}(\mathbf{R})$ introduced in Eq. (2.2). In section 2.1, this mapping was introduced interchangeably as a *feature map* or a *descriptor*. In the subsequent discussion, however, I deliberately continued using only the term *feature map* to now establish a conceptual distinction between the two. In section 2.1, the purpose of the intermediate mapping was primarily to increase model flexibility, for instance by introducing basis functions in linear regression to capture nonlinear relationships. Accordingly, in this work the term *feature map* refers to transformations arising purely from structural or mathematical considerations related to model complexity.

In the context of MLFFs, however, the motivation for introducing such a mapping is more fundamental. The raw atomic coordinates do not inherently encode the system's physical symmetries and invariances. Both energies and forces, however, are invariant under translation, rotation, and permutation of identical atoms. This work therefore defines a *descriptor* as a mapping that transforms the atomic coordinates into a representation reflecting these physical invariances and symmetries. In doing

so, the atomic position vector $\mathbf{R}$ is transformed into a descriptor vector $\boldsymbol{\varphi}(\mathbf{R})$, whose individual components collectively encode the relevant configurational information. Ideally, the descriptor is sufficiently expressive so that no additional feature mapping is required. Naturally, the distinction between the two terms is not absolute but rather serves as a conceptual guide throughout this work.

After establishing this distinction, the key question becomes how to represent atomic environments in a compact yet informative way. The following sections discuss different types of descriptors, distinguishing between *local* and *global* approaches.

2.2.1 Local Descriptors

For local descriptors, the total energy E is expressed as a sum of local atomic contributions E_i,

$$E = \sum_i E_i = \sum_i f(\mathbf{D}_i)\,, \tag{2.29}$$

where $\mathbf{D}_i$ denotes the descriptor vector of the local environment of atom i. Ideally, $\mathbf{D}_i$ is invariant under translations and rotations, identical under permutation of atoms of the same species, and varies smoothly with respect to small structural changes. The local environment of atom i is evaluated within a cut-off radius r_c. A smooth cut-off function $g_\mathrm{cut}(r)$ ensures that atomic contributions continuously vanish as $r \to r_\mathrm{c}$, thereby preventing discontinuities in energy and forces. Within this region, the descriptor can be constructed as a sum over two-body terms

$$\mathbf{D}^{(2)}_{i,\alpha} = \sum_{j:Z_j=\alpha} f(r_{ij})\, g_\mathrm{cut}(r_{ij})\,, \tag{2.30}$$

and/or three-body terms

$$\mathbf{D}^{(3)}_{i,\alpha,\beta} = \sum_{j:Z_j=\alpha} \sum_{k:Z_k=\beta} f(r_{ij})\, f(r_{ik})\, g(\theta_{jik})\, g_\mathrm{cut}(r_{ij})\, g_\mathrm{cut}(r_{ik})\,, \tag{2.31}$$

where Z_j and Z_k denote the atomic species of atoms j and k, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the interatomic distance, and θ_{jik} is the bond angle between atoms j, i, and k. By increasing the body order of the descriptor, higher-order structural correlations can be captured. For instance, including three-body terms introduces angular dependencies that are absent in purely pairwise (two-body) descriptions, albeit at increased computational

cost. A common choice for the radial basis function $f(r)$ in the two-body term is a Gaussian,

$$f(r) = \exp\left[-\eta\,(r - r_s)^2\right] , \tag{2.32}$$

where η controls the width and r_s the center of the Gaussian. For the angular dependence in the three-body term, for example Legendre polynomials can be used,

$$g(\theta) = P_l(\cos\theta) = \frac{1}{2^l l!}\frac{d^l}{d(\cos\theta)^l}\left[(\cos^2\theta - 1)^l\right] , \tag{2.33}$$

with P_l being the Legendre polynomial of order l. These functional forms illustrate that each descriptor feature evaluates the atomic environment according to a specific combination of basis function parameters: η (Gaussian width), r_s (center), and l (Legendre polynomial order).

Local descriptors' main strength lie in their universality and transferability. A model trained on a system with a relatively small number of atoms can often be applied to much larger systems. Local descriptors, however, also introduce conceptual approximations. The decomposition of the total energy into atomic contributions and the introduction of a cut-off radius are artificial constructs, since the true physical energy of a system cannot be exactly decomposed into strictly local parts. In particular, rotational invariance is imposed on truncated local environments that are effectively treated as independent of their embedding in the surrounding bulk material. While this approximation is justified when interactions beyond the cut-off are sufficiently weak, it can complicate the description of long-range effects such as electrostatics in MLFFs.

2.2.2 Global Descriptors

As an alternative to local decompositions, global descriptors represent the entire atomic configuration within a single descriptor that captures all interatomic interactions. A prominent example is the *Coulomb matrix* [24], defined as

$$M_{ij} = \begin{cases} \dfrac{1}{2}Z_i^{2.4}, & i = j, \\ \dfrac{Z_i Z_j}{|\mathbf{r}_i - \mathbf{r}_j|}, & i \neq j\,, \end{cases} \tag{2.34}$$

where Z_i and Z_j denote the atomic numbers and $\mathbf{r}_i$ the atomic positions. The Coulomb matrix thus serves as one possible realization of a global descriptor, $\mathbf{D}_{\text{global}}$, from which the total energy can be modeled as

$$E_{\text{tot}} = g(\mathbf{D}_{\text{global}})\,. \tag{2.35}$$

Global descriptors, however, suffer from poor scalability. As shown by the Coulomb matrix, the computational cost increases as $\mathcal{O}(N^2)$ with the number of atoms, since each atom interacts with all the others. Furthermore, global descriptors are system-specific. They are tied to a fixed number and type of atoms. As a result, transferring them to systems with a different composition or atom count is not straightforward and may require techniques such as padding. This lack of universality severely limits their applicability, despite their conceptual simplicity.

2.2.3 Joint Energy and Force Design Matrix

As previously discussed, when both the PES and the corresponding forces are to be learned simultaneously, the functional relationship between energy and forces given in Eq. (2.28) must be incorporated explicitly. To illustrate this, we first consider the simpler case where only total energies are used as training targets, using a kernel-based regression model as an example. The standard (unregularized) normal equation, similar to Eq. (2.18), can then be written in compact form as

$$\mathbf{y}_E = \mathbf{K}\,\mathbf{w}\,, \tag{2.36}$$

where the target vector $\mathbf{y}_E = [E_1, E_2, \ldots, E_N]^\mathsf{T}$ contains the reference energies of the training configurations, $\mathbf{K}$ is the kernel matrix, and $\mathbf{w}$ denotes the regression weights. In expanded form, Eq. (2.36) reads

$$\begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix} = \begin{bmatrix} k(\mathbf{D}(\mathcal{C}_1), \mathbf{D}(\mathcal{C}_{1_b})) & \cdots & k(\mathbf{D}(\mathcal{C}_1), \mathbf{D}(\mathcal{C}_{N_b})) \\ \vdots & \ddots & \vdots \\ k(\mathbf{D}(\mathcal{C}_N), \mathbf{D}(\mathcal{C}_{1_b})) & \cdots & k(\mathbf{D}(\mathcal{C}_N), \mathbf{D}(\mathcal{C}_{N_b})) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_b} \end{bmatrix}, \tag{2.37}$$

where $\mathbf{D}(\mathcal{C}_i)$ denotes the descriptor of configuration i, which may correspond either to a local atomic environment or a global descriptor. $\mathcal{C}_i$ encodes the set of all atomic positions $\{\mathbf{r}_{i,j}\}_{j=1}^{N_A}$ of configuration i, containing N_A atoms. If we now wish to include atomic forces as training data, the system of equations must be extended to account for the derivatives of the energy with respect to the atomic positions, while simultaneously

fulfilling the physical relation given in Eq. (2.28). This leads to the augmented system

$$\begin{bmatrix} \mathbf{y}_E \\ \mathbf{y}_F \end{bmatrix} = \begin{bmatrix} \mathbf{K} \\ -\nabla_{\mathcal{C}}\mathbf{K} \end{bmatrix} \mathbf{w}\,, \tag{2.38}$$

where $\mathbf{y}_F = [F_{1,1,x}, F_{1,1,y}, \ldots, F_{N,N_A,z}]^\mathsf{T}$ collects all force components for N training configurations. The additional block $-\nabla_{\mathcal{C}}\mathbf{K}$ represents the derivatives of the kernel function with respect to all atomic coordinates, obtained via the chain rule,

$$\nabla_{\mathcal{C}} k = \frac{\partial k}{\partial \mathbf{D}} \frac{\partial \mathbf{D}}{\partial \mathcal{C}}\,. \tag{2.39}$$

Using the short notation $k_{i,i_b} = k\big(\mathbf{D}(\mathcal{C}_i), \mathbf{D}(\mathcal{C}_{i_b})\big)$, Eq. (2.38) reads in expanded form

$$\begin{bmatrix} E_1 \\ \vdots \\ E_N \\ F_{1,1,x} \\ \vdots \\ F_{N,N_A,z} \end{bmatrix} = \begin{bmatrix} k_{1,1_b} & \cdots & k_{1,N_b} \\ \vdots & \ddots & \vdots \\ k_{N,1_b} & \cdots & k_{N,N_b} \\ -\dfrac{\partial k_{1,1_b}}{\partial \mathbf{D}(\mathcal{C}_1)} \dfrac{\partial \mathbf{D}}{\partial r_{1,1,x}}(\mathcal{C}_1) & \cdots & -\dfrac{\partial k_{1,N_b}}{\partial \mathbf{D}(\mathcal{C}_1)} \dfrac{\partial \mathbf{D}}{\partial r_{1,1,x}}(\mathcal{C}_1) \\ \vdots & \ddots & \vdots \\ -\dfrac{\partial k_{N,1_b}}{\partial \mathbf{D}(\mathcal{C}_N)} \dfrac{\partial \mathbf{D}}{\partial r_{N,N_A,z}}(\mathcal{C}_N) & \cdots & -\dfrac{\partial k_{N,N_b}}{\partial \mathbf{D}(\mathcal{C}_N)} \dfrac{\partial \mathbf{D}}{\partial r_{N,N_A,z}}(\mathcal{C}_N) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_b} \end{bmatrix}. \tag{2.40}$$

Here, $\partial/\partial r_{i,j,\alpha}$ denotes differentiation with respect to the Cartesian coordinate α of atom j in configuration i. Including the forces increases the number of equations from N to $N(3N_A{+}1)$. If the size of the basis set, and thus the number of fitting parameters, is chosen as $N_b = N$, the resulting linear system becomes highly overdetermined. A suitable solution method in this case is SVD as described in section A.1.

In summary, MLFFs transform the atomic coordinates $\mathcal{C}_i$ of each configuration into numerical features via a descriptor $\mathbf{D}(\mathcal{C}_i)$. These descriptors may then pass through a kernel or nonlinear embedding $\tilde{\boldsymbol{\varphi}}\big(\mathbf{D}(\mathcal{C}_i)\big)$. The model $f(\cdot)$ subsequently predicts the corresponding energy and forces using the learned parameters $\mathbf{w}$ and hyperparameters $\boldsymbol{\theta}$. Analogous to Eq. (2.2), this structured mapping is illustrated below:

$$\mathcal{C}_i \;\mapsto\; \mathbf{D}(\mathcal{C}_i) \;\mapsto\; \tilde{\boldsymbol{\varphi}}\big(\mathbf{D}(\mathcal{C}_i)\big) \;\mapsto\; \big[E_i,\, \mathbf{F}_i\big] = f\big(\tilde{\boldsymbol{\varphi}}\big(\mathbf{D}(\mathcal{C}_i)\big);\, \mathbf{w},\, \boldsymbol{\theta}\big)\,. \tag{2.41}$$

2.3 Structure Factor

As discussed in the previous section, descriptors are fundamental building blocks for MLFFs. In this work, the structure factor is used as a global descriptor. Therefore, this section introduces the theoretical background of the structure factor. It is motivated from two complementary perspectives. First, through its experimental meaning in scattering experiments, and second, through its relation to the pair distribution function. Finally, I discuss several properties that are relevant when applying the structure factor within MLFFs for MD (molecular dynamics) simulations.

2.3.1 Introduction via Experimental Meaning

Consider a sample consisting of atoms located at positions $\mathbf{r}_j$, $j = 1, \ldots, N_A$, which is exposed to an incident beam that can be represented by a plane wave

$$\psi_{\text{in}}(\mathbf{r}) = e^{i\mathbf{k}_{\text{in}}\cdot\mathbf{r}}\,, \tag{2.42}$$

where $\mathbf{k}_{\text{in}}$ is the wave vector of the incoming beam and $\mathbf{r}$ denotes the observation point. Each atom in the sample elastically scatters the incoming wave into all directions. In elastic scattering, the magnitude of the incoming and outgoing wave vectors are identical, i.e.

$$|\mathbf{k}_{\text{in}}| = |\mathbf{k}_{\text{out}}| = k\,, \tag{2.43}$$

meaning that the photon (or particle) retains its energy while changing its propagation direction. This assumption is valid for typical X-ray or neutron diffraction experiments, where the energy transfer to the atomic nuclei is negligible.

According to Huygens' principle, each atom acts as a secondary source emitting a spherical wave. The scattered field can thus be written as

$$\psi_{\text{sc}}(\mathbf{r}) = \sum_j f_j\, e^{i\mathbf{k}_{\text{in}}\cdot\mathbf{r}_j} \frac{e^{ik|\mathbf{r}-\mathbf{r}_j|}}{|\mathbf{r}-\mathbf{r}_j|}\,, \tag{2.44}$$

where f_j denotes the atomic scattering amplitude associated with atom j. In general, f_j depends on the magnitude of the scattering vector $|k|$, but this dependence is neglected here for simplicity.

Since the detector is much farther away than the atomic distances within the sample ($r \gg |\mathbf{r}_j|$), the *far-field approximation* is justified. Using a Taylor expansion, one finds

$$|\mathbf{r} - \mathbf{r}_j| = r\sqrt{1 - 2\frac{\hat{\mathbf{r}} \cdot \mathbf{r}_j}{r} + \mathcal{O}(r_j^2/r^2)} \approx r - \hat{\mathbf{r}} \cdot \mathbf{r}_j \,, \tag{2.45}$$

with $\hat{\mathbf{r}} = \mathbf{r}/r$. Inserting this approximation into the scattered field yields

$$\psi_{\mathrm{sc}}(\mathbf{r}) \approx \frac{e^{ikr}}{r} \sum_j f_j \, e^{i\mathbf{k}_{\mathrm{in}} \cdot \mathbf{r}_j} e^{-ik\hat{\mathbf{r}} \cdot \mathbf{r}_j} \,. \tag{2.46}$$

In the far field, all scattered waves reaching the detector can be considered parallel, such that we may identify the outgoing wave vector as

$$\mathbf{k}_{\mathrm{out}} = k\hat{\mathbf{r}} \,. \tag{2.47}$$

Hence, the scattered wave becomes

$$\psi_{\mathrm{sc}}(\mathbf{r}) \approx \frac{e^{i\mathbf{k}_{\mathrm{out}} \cdot \mathbf{r}}}{r} \sum_j f_j \, e^{i(\mathbf{k}_{\mathrm{in}} - \mathbf{k}_{\mathrm{out}}) \cdot \mathbf{r}_j} \,. \tag{2.48}$$

The prefactor $e^{i\mathbf{k}_{\mathrm{out}} \cdot \mathbf{r}}/r$ determines the overall amplitude at the detector, while the interference pattern is governed entirely by the summation term. It depends only on the difference between the incoming and outgoing wave vectors, which motivates defining the scattering vector

$$\mathbf{q} = \mathbf{k}_{\mathrm{out}} - \mathbf{k}_{\mathrm{in}} \,. \tag{2.49}$$

The intensity observed at the detector is proportional to the absolute square of the scattered amplitude,

$$I(\mathbf{q}) \propto |\psi_{\mathrm{sc}}(\mathbf{q})|^2 \propto \sum_{j,k} f_j f_k^* e^{-i\mathbf{q} \cdot (\mathbf{r}_j - \mathbf{r}_k)} \,. \tag{2.50}$$

A normalized form of this quantity defines the *structure factor* as

$$S(\mathbf{q}) = \frac{1}{\sum_j |f_j|^2} \sum_{j,k} f_j f_k^* e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \,. \tag{2.51}$$

It is worth noting that two different naming conventions exist in the literature. In some cases, as adopted here, $S(\mathbf{q})$ refers to the intensity-like quantity proportional to $|T(\mathbf{q})|^2$, where

$$T(\mathbf{q}) = \sum_j f_j e^{-i\mathbf{q}\cdot\mathbf{r}_j} \,. \tag{2.52}$$

Other sources define $T(\mathbf{q})$ itself as the structure factor, leading to $I(\mathbf{q}) \propto S(\mathbf{q})S^*(\mathbf{q})$. Both conventions describe the same physical content.

It should further be noted that, in real scattering experiments, because atomic positions fluctuate thermally, the experimentally measured structure factor corresponds to an ensemble average, ensuring that $S(\mathbf{q})$ represents observable rather than instantaneous correlations. Considering now the interference pattern that would be observed in such an experiment, for a completely disordered system no constructive interference occurs, and $S(\mathbf{q})$ remains constant. In contrast, in an ordered (crystalline) system, constructive interference arises for specific $\mathbf{q}$ vectors whenever the Bragg condition

$$\mathbf{q} \cdot \mathbf{r}_j = 2\pi n \tag{2.53}$$

is satisfied. Thus, the structure factor contains information about the periodic arrangement of the sample's atoms and allows one to reconstruct structural order from the observed interference pattern. This property makes the structure factor an ideal candidate for a global descriptor that encodes the configurational information of an ordered atomic ensemble. Beyond its experimental interpretation, the structure factor also exhibits a direct and insightful connection to the pair distribution function, which provides a complementary real-space perspective on the same structural correlations.

2.3.2 Connection to the Pair Distribution Function

The structure factor is related to the pair distribution function $g(\mathbf{r})$. The latter describes the probability of finding an atom at a certain distance from another atom. This relationship provides a complementary and physically insightful link between reciprocal-space measurements and real-space atomic correlations, thereby further motivating the structure factor as a descriptor for MLFFs.

The pair distribution function is defined as

$$g(\mathbf{r}) = \frac{1}{\rho N_A} \Big\langle \sum_{j \neq k} \delta\big(\mathbf{r} - (\mathbf{r}_j - \mathbf{r}_k)\big) \Big\rangle \,, \tag{2.54}$$

where $\rho = N_A/V$ denotes the average atomic number density, and $\langle \cdot \rangle$ indicates an ensemble average over microscopic configurations.

For the structure factor defined in Eq. (2.51), we assume a monoatomic system ($f_j = 1$) for simplicity and consider the ensemble average. By separating the double sum, the expression can be rewritten as

$$S(\mathbf{q}) = \frac{1}{N_A} \Big\langle \sum_{j,k} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \Big\rangle \tag{2.55}$$

$$= \frac{1}{N_A} \Big\langle \sum_{j} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_j)} + \sum_{j \neq k} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \Big\rangle \tag{2.56}$$

$$= 1 \; + \; \frac{1}{N_A} \Big\langle \sum_{j \neq k} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \Big\rangle \,. \tag{2.57}$$

Using the identity

$$\sum_{j \neq k} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} = \sum_{j \neq k} \int d^3r \, e^{-i\mathbf{q}\cdot\mathbf{r}} \delta\big(\mathbf{r} - (\mathbf{r}_j - \mathbf{r}_k)\big) \;, \tag{2.58}$$

we can rewrite the ensemble average as

$$S(\mathbf{q}) = 1 + \frac{1}{N_A} \int d^3r \, e^{-i\mathbf{q}\cdot\mathbf{r}} \Big\langle \sum_{j \neq k} \delta\big(\mathbf{r} - (\mathbf{r}_j - \mathbf{r}_k)\big) \Big\rangle \tag{2.59}$$

$$= 1 + \rho \int d^3r \, g(\mathbf{r}) \, e^{-i\mathbf{q}\cdot\mathbf{r}} \,. \tag{2.60}$$

As shown in Eq. (2.60), the integral term represents the Fourier transform of the pair correlations encoded in $g(\mathbf{r})$. The prefactor ρ scales the correlation contribution with the average atomic number density, whereas the constant term can be interpreted as the sum of a uniform, uncorrelated background. While $S(\mathbf{q})$ is not strictly the Fourier transform of $g(\mathbf{r})$ alone, this formulation highlights that the structure factor

captures both the mean density and the deviations from it due to pairwise correlations, providing a reciprocal-space representation of the underlying real-space structure.

For isotropic materials, the structure factor can be written as

$$S(\mathbf{q}) = 1 + \rho \int d^3r \, g(|\mathbf{r}|) \, e^{-i\mathbf{q}\cdot\mathbf{r}} \,, \tag{2.61}$$

where $d^3r = r^2 \sin\theta \, dr \, d\theta \, d\phi$ in spherical coordinates. Since $g(r)$ depends only on the interatomic distance $r = |\mathbf{r}|$, we can choose the coordinate system such that $\mathbf{q}$ points along the z-axis, yielding $\mathbf{q} \cdot \mathbf{r} = qr\cos\theta$. The integral then becomes

$$S(\mathbf{q}) = 1 + \rho \int_0^\infty dr \, r^2 \int_0^\pi d\theta \, \sin\theta \int_0^{2\pi} d\phi \, g(r) \, e^{-iqr\cos\theta} \tag{2.62}$$

$$= 1 + 2\pi\rho \int_0^\infty dr \, r^2 g(r) \int_0^\pi d\theta \, \sin\theta \, e^{-iqr\cos\theta} \tag{2.63}$$

$$= 1 + 4\pi\rho \int_0^\infty dr \, r^2 g(r) \, \frac{\sin(qr)}{qr} \,. \tag{2.64}$$

From this final expression, one sees that the structure factor depends only on the magnitude $q = |\mathbf{q}|$, i.e., $S(\mathbf{q}) = S(q)$. Therefore, a machine-learned function f that depends exclusively on $S(q)$ effectively captures only the radial (pairwise) correlations contained in $g(r)$.

2.3.3 Partial Structure Factors

In multi-component systems, such as alloys or molecular mixtures, the total structure factor arises from contributions of all atomic species. To disentangle these effects, *partial structure factors* are introduced, which describe the correlations between specific pairs of atom types.

For a system with N_α atoms of type α and N_β atoms of type β, the partial structure factor is defined as

$$S_{\alpha\beta}(\mathbf{q}) = \frac{1}{\sqrt{N_\alpha N_\beta}} \sum_{j\in\alpha} \sum_{k\in\beta} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \,, \tag{2.65}$$

where $j \in \alpha$ and $k \in \beta$ indicate that the sums go over atoms of the respective types.

By construction, the partial structure factor satisfies

$$S_{\alpha\beta}(\mathbf{q}) = S^*_{\beta\alpha}(\mathbf{q}) = S_{\beta\alpha}(-\mathbf{q}) = S^*_{\alpha\beta}(-\mathbf{q})\,. \tag{2.66}$$

The total structure factor can then be expressed as a weighted sum over all pairs of atom types,

$$S(\mathbf{q}) = \sum_{\alpha,\beta} c_\alpha c_\beta f_\alpha f^*_\beta\, S_{\alpha\beta}(\mathbf{q})\,, \tag{2.67}$$

where c_α is the atomic fraction of species α, given by $c_\alpha = N_\alpha/N_A$ with N_α being the number of atoms of type α, and f_α is the corresponding scattering amplitude.

For isotropic systems, one can define the corresponding radial pair distribution functions $g_{\alpha\beta}(r)$, yielding the radial partial structure factor via

$$S_{\alpha\beta}(q) = \delta_{\alpha\beta} + 4\pi\rho \int_0^\infty r^2\, g_{\alpha\beta}(r) \frac{\sin(qr)}{qr} dr\,, \tag{2.68}$$

with ρ the total number density.

As shown in Eq. (2.40), fitting force data requires the derivative of the descriptor with respect to the atomic positions. For the partial two-body structure factor, this derivative is obtained by differentiating with respect to the Cartesian component $r_{i,\mu}$ of atom i, given by

$$\frac{\partial S_{\alpha\beta}(\mathbf{q})}{\partial r_{i,\mu}} = \frac{i q_\mu}{\sqrt{N_\alpha N_\beta}} \left[\mathbf{1}_{\{i\in\alpha\}} \sum_{k\in\beta} e^{\,i\mathbf{q}\cdot(\mathbf{r}_i - \mathbf{r}_k)} \;-\; \mathbf{1}_{\{i\in\beta\}} \sum_{j\in\alpha} e^{-i\mathbf{q}\cdot(\mathbf{r}_i - \mathbf{r}_j)} \right]. \tag{2.69}$$

Here, $\mathbf{1}_{\{i\in\alpha\}}$ denotes the indicator functions, defined as

$$\mathbf{1}_{\{i\in\alpha\}} = \begin{cases} 1\,, & \text{if atom } i \text{ belongs to species } \alpha\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{2.70}$$

A detailed derivation of this expression is provided in subsection A.2.1.

2.3.4 Perfect and Imperfect Crystals

So far, the type of system under investigation has remained unspecified but it was mentioned that only for crystals there is a distinct interference pattern. A perfect crystal can be described by a Bravais lattice with an (optionally multi-atomic) basis,

$$\mathbf{r}_j = \mathbf{R}_n + \boldsymbol{\tau}_s \,, \tag{2.71}$$

where the lattice vectors are given by

$$\mathbf{R}_n = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3 \,, \qquad n_i \in \mathbb{Z}, \tag{2.72}$$

and $\boldsymbol{\tau}_s$ denote the basis positions of the atoms inside the unit cell. Inserting this expression into the phase factor in Eq. (2.52) yields

$$T(\mathbf{q}) = \sum_{n,s} f_s \, e^{-i\mathbf{q}\cdot(\mathbf{R}_n+\boldsymbol{\tau}_s)} = \Big[\sum_n e^{-i\mathbf{q}\cdot\mathbf{R}_n}\Big]\Big[\sum_s f_s e^{-i\mathbf{q}\cdot\boldsymbol{\tau}_s}\Big] \,. \tag{2.73}$$

The first factor contains the interference from scattering by the periodic lattice and produces sharp peaks in $S(\mathbf{q})$ only for wave vectors satisfying the Bragg condition introduced in Eq. (2.53). These special wave vectors are precisely the reciprocal lattice vectors $\mathbf{G}$, defined by

$$\mathbf{G} = h\,\mathbf{b}_1 + k\,\mathbf{b}_2 + l\,\mathbf{b}_3 \,, \qquad h,k,l \in \mathbb{Z} \,, \tag{2.74}$$

where the reciprocal basis vectors are defined as

$$\mathbf{b}_i = 2\pi \, \frac{\mathbf{a}_j \times \mathbf{a}_k}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \,, \tag{2.75}$$

where (i,j,k) are cyclic permutations of $(1,2,3)$. Since these vectors satisfy

$$\mathbf{G}\cdot\mathbf{R}_n = 2\pi m \,, \qquad m \in \mathbb{Z} \,. \tag{2.76}$$

they are exactly the wave vectors that produce constructive interference, i.e.

$$e^{-i\mathbf{G}\cdot\mathbf{R}_n} = 1 \quad \text{for all } \mathbf{R}_n \,. \tag{2.77}$$

Hence, only for these $\mathbf{q}$-values constructive interference occurs, resulting in a discrete set of Bragg peaks. The second term depends on the atoms in the basis. It modulates the intensity of the Bragg peaks, which can even lead to systematic extinctions, and determines how strongly each reciprocal lattice point contributes. This also illustrates that the same crystal structure can equivalently be represented by different choices of the unit cell. Enlarging the real-space unit cell narrows the reciprocal lattice spacing, thereby increasing the density of Bragg peaks due to the first term. However, the corresponding larger multi-atomic basis suppresses redundant peaks, leaving the physically observable scattering pattern unchanged.

The dependence of the structure factor on atomic displacements has already been established in Eq. (2.69). For a more intuitive understanding, however, it is instructive to revisit this result using a simple Taylor expansion, which is done in the following. As a starting point consider finite temperatures, where atomic vibrations lead to deviations from perfect periodicity, which can be incorporated through the atomic displacements $\mathbf{u}_j$, At finite temperatures, atomic vibrations lead to deviations from perfect periodicity, which can be incorporated through the atomic displacements $\mathbf{u}_j$,

$$\tilde{\mathbf{r}}_j = \mathbf{r}_j + \mathbf{u}_j \,, \tag{2.78}$$

where $\mathbf{r}_j$ denotes the equilibrium lattice position at absolute zero and $\tilde{\mathbf{r}}_j$ the instantaneous atomic position at finite temperature. Clearly, the atomic displacements fluctuate in time due to thermal vibrations, but for simplicity we consider static snapshots of the atomic configuration. Since the structure factor is meant to describe the structure, it is instructive to examine how it responds to such small atomic displacements. For this, consider the case where all atoms except a single atom x remain at their equilibrium positions, and atom x is displaced by $\mathbf{u}$:

$$\tilde{\mathbf{r}}_x = \mathbf{r}_x + \mathbf{u}, \qquad \tilde{\mathbf{r}}_j = \mathbf{r}_j \text{ for } j \neq x \,. \tag{2.79}$$

Let the undisturbed phase factor be defined as $T_0 \equiv \sum_j f_j e^{-i\mathbf{q}\cdot\mathbf{r}_j}$. Upon displacing atom x, it changes by

$$\Delta T \equiv T(\{\tilde{\mathbf{r}}\}) - T_0 = f_x\big(e^{-i\mathbf{q}\cdot(\mathbf{r}_x+\mathbf{u})} - e^{-i\mathbf{q}\cdot\mathbf{r}_x}\big) = a_x\big(e^{-i\mathbf{q}\cdot\mathbf{u}} - 1\big) \,, \tag{2.80}$$

where $a_x \equiv f_x e^{-i\mathbf{q}\cdot\mathbf{r}_x}$. For small $\mathbf{u}$ a Taylor expansion yields

$$\Delta T \approx a_x\Big[- i(\mathbf{q}\cdot\mathbf{u}) - \tfrac{1}{2}(\mathbf{q}\cdot\mathbf{u})^2 + \mathcal{O}(u^3)\Big] \,. \tag{2.81}$$

The (unnormalized) structure factor is defined as $S(\mathbf{q}) = T^*(\mathbf{q})T(\mathbf{q}) = |T(\mathbf{q})|^2$. Hence, the change ΔS due to a displacement can be written as

$$\Delta S = |T_0 + \Delta T|^2 - |T_0|^2 = T_0^* \Delta T + \Delta T^* T_0 + |\Delta T|^2 \,. \tag{2.82}$$

Keeping only the leading (linear) order in $\mathbf{u}$ yields

$$\begin{aligned} \Delta S^{(1)} &\approx T_0^* \big(- i a_x \, \mathbf{q}\cdot\mathbf{u} \big) + \big(i a_x^* \, \mathbf{q}\cdot\mathbf{u} \big) T_0 + \mathcal{O}(u^2) \\ &\approx -2 \, \mathbf{q}\cdot\mathbf{u} \, \mathrm{Im}\big(T_0^* a_x \big) + \mathcal{O}(u^2) \\ &= -2 \, \mathbf{q}\cdot\mathbf{u} \, \mathrm{Im}\Big(f_x \sum_j f_j^* e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_x)} \Big) + \mathcal{O}(u^2) \,. \end{aligned} \tag{2.83}$$

so that

$$\Delta S \sim \mathcal{O}(u) \,. \tag{2.84}$$

The linearized term in Eq. (2.83) can be viewed as the contribution of atom x to the structure factor, modulated by its interference with all other atoms in reciprocal space. The relation in Eq. (2.84) is in so far interesting as the energy for small displacements scales with $\Delta E \sim \mathcal{O}(u^2)$. This shows that the structure factor is linearly sensitive to small displacements, while the energy scales quadratically, making $S(\mathbf{q})$ a sensitive descriptor for small structural changes.

The linearized term in Eq. (2.83) can be interpreted as the contribution of atom x to the structure factor, modulated by its interference with all other atoms in reciprocal space. Eq. (2.84) shows that the structure factor responds linearly to small atomic displacements, i.e. $\Delta S \sim \mathcal{O}(u)$. This behavior is particularly relevant for MLFFs trained on forces. For small displacements around equilibrium, the energy changes quadratically, $\Delta E \sim \mathcal{O}(u^2)$, while the corresponding forces follow Hooke's law and therefore scale linearly with the displacement, $\mathbf{F} \sim \mathcal{O}(u)$. Since the structure factor exhibits the same leading-order scaling behavior, it provides a descriptor that is naturally sensitive to the physically relevant variations governing the force response.

3 Methods

While the previous chapter laid the theoretical groundwork for this study, this chapter details the computational methodology used to develop and train the MLFF. The focus is on translating these concepts into practical, efficient implementations. First, the test systems and reference data used to evaluate the MLFF are specified, thereby defining this study's computational setting. Next, a high-level overview of the general workflow is presented, outlining the main steps from reading atomic configurations to predicting energies and forces. For a detailed discussion of the specific methodological decisions, see chapter 4.

3.1 Data Acquisition

To provide a basis for training and evaluating the MLFF, reference data for crystalline systems were generated. As test materials, crystalline silicon (Si) and sodium chloride (NaCl) were considered. For both materials, cubic supercells containing 64 atoms were used.

The reference configurations were generated using VASP (Vienna *ab initio* Simulation Package) [12–14] within DFT (density functional theory). Finite-temperature molecular dynamics simulations were performed to sample diverse atomic configurations. For both test systems, ionic motion was propagated using MD with a Nosé–Hoover thermostat [37, 38], covering a temperature range from 10 K to 500 K. A time step of 1.5 fs was used, and each simulation comprised up to 100 000 ionic steps. All simulations were carried out in the canonical (NVT) ensemble with fixed lattice vectors. This choice ensures that the reciprocal lattice—and, consequently, the points at which the structure factor is evaluated—remains the same across all configurations. This makes the resulting descriptors directly comparable.

Projector-augmented wave (PAW) potentials [39, 40] and a plane-wave cutoff energy of 500 eV were used to ensure converged energies and forces. For each configuration, total energies and atomic forces were extracted and used as reference targets for training.

Using this protocol, a total of 911 configurations were generated for Si and 1039 configurations for NaCl.

3.2 General Implementation and Workflow

The overall goal of this work is to employ the structure factor (Eq. (2.51)) as a descriptor for an MLFF. A general mapping chain for MLFF problems has already been introduced in Eq. (2.41). The specific realization of this mapping in the present work is summarized in the component diagram shown in Figure 3.1 and described in the following section.

The workflow begins with reading the input files. Here, two types of input are distinguished. The first type consists of files that provide computational settings and model parameters (also called hyperparameters), which are systematically analyzed and varied in chapter 4. Second, the *ab initio* data, i.e., the atomic configuration data as inputs as well as the corresponding total energies and atomic forces are read in as reference targets.

This dataset is then randomly split into training and test sets, typically using an 80%/20% split. In this work, no separate validation set is employed, since kernel regression does not require procedures such as early stopping that are commonly used in neural network training. For kernel-based models, the basis set introduced in subsection 2.1.2 consists of the reference configurations used to construct the kernel expansion. In the present work, the basis set generally coincides with the training set. If desired, both the training and basis sets can be enlarged via symmetry-based data augmentation. In this procedure, additional structures are generated by applying symmetry operations to the *ab initio* reference configurations (see section 4.2). Prior to descriptor evaluation, the data are centered on the training set, as described in subsection 2.1.1, in order to eliminate the bias term in the regression. Based on the lattice vectors of the atomic configurations, a grid of reciprocal vectors $\{\mathbf{q}\}$ is constructed, defining the wave vectors at which the structure factor is evaluated. The specific choice of the reciprocal vector grid is analyzed in detail in section 4.1.

For each configuration in the training, basis, and test sets, the partial structure factors $S_{\alpha\beta}(\mathbf{q})$ according to Eq. (2.65) (omitting the normalization pre-factor) are computed at all specified $\mathbf{q}$-vectors. These partial contributions are concatenated into a single descriptor vector for each configuration. Analytical derivatives of the descriptors of the training and test set with respect to all atomic positions are also calculated according to Eq. (2.69), as they are required for force predictions. The descriptors and their derivatives are used to construct the design matrices for both the training and test sets. This is done by evaluating the chosen kernel function for all pairs of training and basis configurations according to Eq. (2.40), where the descriptor $\mathbf{D}$ for each configuration is given by the concatenated partial structure factors. Generally, a simple linear kernel function is employed (see Eq. (2.22)), while a brief analysis of polynomial kernels (see Eq. (2.20)) is presented in section 4.4.

Once the design matrix for the training set is constructed, the regression parameters (weights) of the MLFF are obtained using the SVD representation of the ridge regression solution given in Eq. (A.6). The determined weights are then used to predict energies and forces for both training and test configurations, including de-centering. Model performance is subsequently assessed by comparing the predictions with the *ab initio* reference targets using appropriate error metrics, such as root mean square errors (RMSE) for energies and forces, as detailed in section A.4.

The computation of the descriptors and their derivatives was implemented using the `JAX` library in `Python`, whereas the remaining components of the MLFF code were developed primarily with `NumPy`. The use of `JAX` enables efficient vectorization and parallelization, thereby significantly accelerating the evaluation of descriptors and their analytical derivatives. Despite `JAX` providing automatic differentiation, the derivatives for the two-body descriptors were implemented manually, and automatic differentiation was only employed for the three-body case. Initially, all descriptors and derivatives are computed and stored in memory. While this approach provides high computational speed, it requires substantial memory. Limitations associated with this strategy and alternative implementations, particularly for the three-body descriptors, are discussed in detail in section 4.3. Furthermore, all computations are performed using double precision to ensure numerical accuracy.

In this way, the theoretical concepts introduced in the previous chapter are systematically translated into a practical, efficient, and reproducible MLFF implementation. The code developed for this work is available upon request and serves as the basis for the detailed investigations presented in the following chapters.

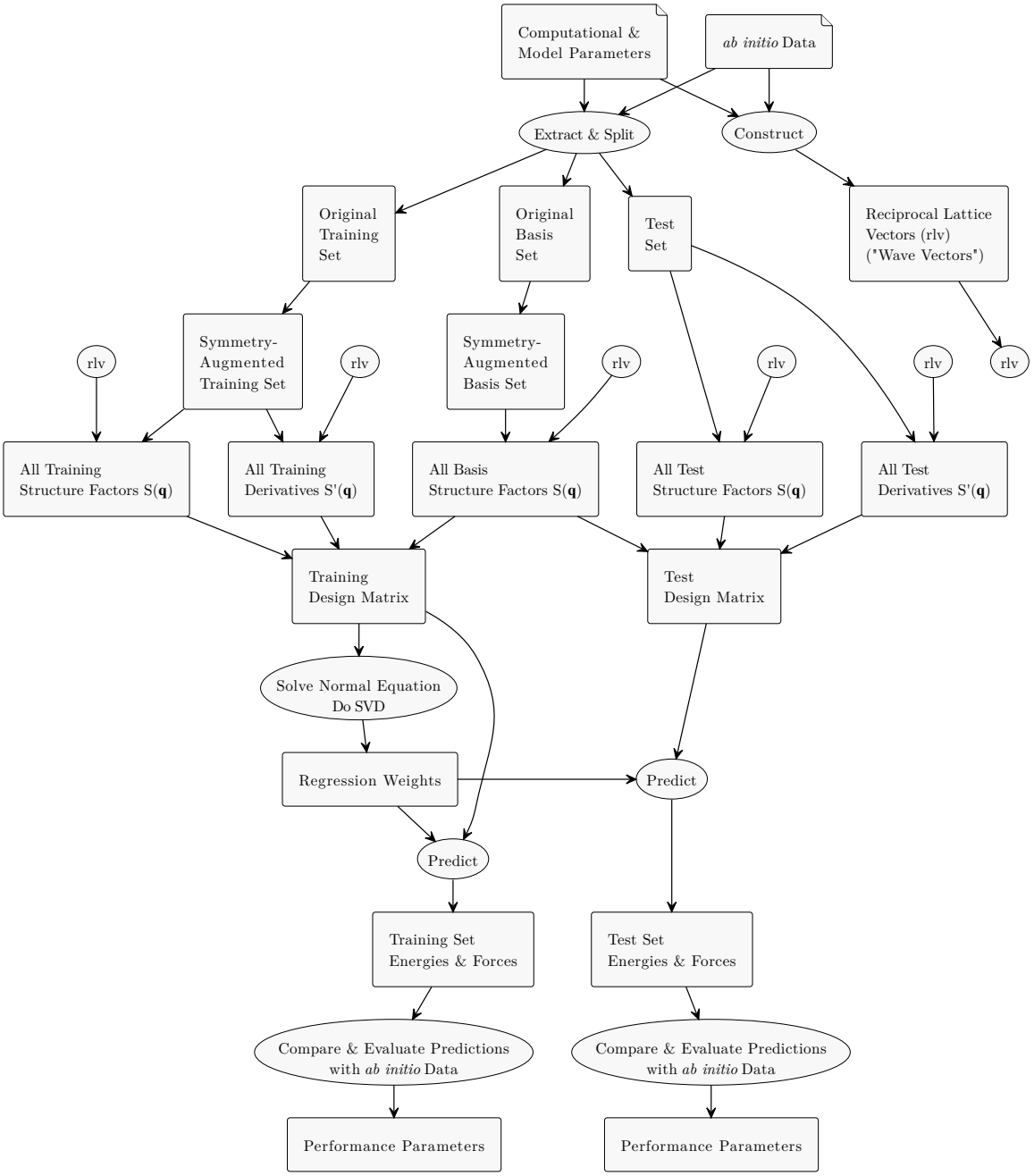

Figure 3.1: Component diagram of the MLFF workflow used in this work. Here, "rlv" is short for reciprocal lattice vectors.

4 Model Parameter Study

Having established the computational methodology and workflow for constructing and training the MLFF in the previous chapter, this chapter is devoted to a systematic analysis of the model parameters. To clearly distinguish them from the regression coefficients (fit parameters or weights), the terms *model parameters* or *hyperparameters* are used throughout this chapter.

The performance of the MLFF depends critically on several key factors. These include the number of descriptor features, i.e. the number and selection of reciprocal lattice vectors used in the structure factor representation, as well as the number of fit parameters controlled by the size of the basis set (see section 4.1). Further important aspects are the application of symmetry-based data augmentation (see section 4.2). In this procedure, additional configurations are generated by applying symmetry operations to the *ab initio* reference structures. Further considerations include methodological choices such as extending the descriptor from two-body to three-body contributions (see section 4.3) or selecting an appropriate kernel function (see section 4.4).

The following sections examine each of these aspects separately. In each case, progressing from theoretical considerations to methodological implementation and numerical results, including their implications for model parameter selection.

4.1 Number of Reciprocal Vectors

The first question is at which, and at how many, wave vectors the structure factor should be evaluated. From Eq. (2.76) it follows that a perfect infinite crystal exhibits constructive interference only at reciprocal lattice vectors. Evaluating the structure factor at arbitrary wave vectors $\mathbf{q} \notin \{\mathbf{G}\}$ yields primarily finite-size contributions. These values depend on the artificial truncation of the lattice sum rather than on true configurational information. It is therefore natural to restrict the descriptor to reciprocal lattice vectors.

In a perfect infinite crystal, however, the structure factor at all wave vectors is not independent. Due to lattice periodicity it satisfies $S(\mathbf{q}) = S(\mathbf{q} + \mathbf{G})$, so that all physically distinct information is contained within the first Brillouin zone.

For a finite-temperature or defect-containing configuration this periodicity is no longer exact, and therefore $S(\mathbf{q}) \neq S(\mathbf{q}+\mathbf{G})$. This raises the question of whether it is meaningful to evaluate the structure factor at periodic replicas of the reciprocal lattice, or, equivalently, whether $S(\mathbf{q})$ and $S(\mathbf{q}+\mathbf{G})$ contain different physical information at finite temperature. To analyse this, we compare the finite-temperature structure factor $S(\mathbf{q})$ with its zero-temperature reference $S_0(\mathbf{q})$ (see section A.3 for the derivation). Writing the thermal displacement of atom j from its equilibrium position as $\mathbf{u}_j$, one obtains the following expansion:

$$\begin{aligned}&\big(S(\mathbf{q})-S_0(\mathbf{q})\big)-\big(S(\mathbf{q}+\mathbf{G})-S_0(\mathbf{q}+\mathbf{G})\big)\\ &\quad\approx\sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)}\Big[-i\,\mathbf{G}\cdot\Delta\mathbf{u}_{jk}+(\mathbf{q}\cdot\mathbf{G})\,(\Delta\mathbf{u}_{jk})^2+\tfrac{G^2}{2}\,(\Delta\mathbf{u}_{jk})^2+\mathcal{O}\big((\mathbf{G}\cdot\Delta\mathbf{u}_{jk})^3\big)\Big]\,,\end{aligned} \tag{4.1}$$

where $\Delta\mathbf{u}_{jk} = \mathbf{u}_j - \mathbf{u}_k$. The sum converges provided that $|\mathbf{G}\cdot\Delta\mathbf{u}_{jk}| \lesssim 1$. For a typical reciprocal lattice magnitude of $|\mathbf{G}| \sim 2\pi/10\,\text{Å}$ and thermal displacements of order $|\Delta\mathbf{u}_{jk}| \sim 0.3\,\text{Å}$, the product $|\mathbf{G}\cdot\Delta\mathbf{u}_{jk}| \sim 0.2$ remains below unity. Thus, for such $|\mathbf{G}|$-values, the expansion is well-behaved, although this does not need to hold for larger $|\mathbf{G}|$.

In general, the expression is non-zero, and its leading contribution scales approximately linearly with $|\mathbf{G}|$. Consequently, the structure factor becomes increasingly sensitive at reciprocal lattice vectors with greater length. This indicates that $S(\mathbf{q})$ and $S(\mathbf{q}+\mathbf{G})$ generally encode distinct configurational information at finite temperature.

As a result, no strict upper cutoff in reciprocal space can be defined in the sense that "all relevant information is contained up to a certain $|\mathbf{G}|$." Rather, increasing the number of evaluated wave vectors increases the amount of structural information captured by the descriptor, but requires more computational effort. The cutoff $q_{\max}$ thus constitutes a central hyperparameter of the model.

At this point, it is useful to clarify the distinction between the different dimensionalities entering the model. The number of training configurations is denoted by N and refers to the number of independent *ab initio* configurations. Each configuration is mapped to a descriptor vector of dimension d, which depends on the number of reciprocal lattice vectors N_G and, for multi-component systems, on the number of partial structure factors. For a system with N_s atomic species, this yields

$$d = N_G \cdot N_s^2\,, \tag{4.2}$$

corresponding to all pairwise combinations of atomic species. Diagonal pairs ($\alpha = \beta$) contribute one real feature per $\mathbf{q}$-vector, while off-diagonal pairs ($\alpha \neq \beta$) contribute one complex feature, which correspond to two independent real-valued features. As a result, the simple N_s^2 scaling already accounts for the symmetries $S_{\alpha\beta}(\mathbf{q}) = S^*_{\alpha\beta}(-\mathbf{q})$ (see Eq. (2.66)) between partial structure factors. For n-body terms, the number of features scales as $N_G \cdot N_s^n$, which will become relevant in section 4.3.

To fully capture the information contained in the descriptor, the regression model requires a sufficient number of fit parameters, which equals the size of the basis set N_b. As shown in Eq. (2.23), the complexity that can be represented by a linear kernel model is directly linked to the number of fit parameters relative to the number of features. In particular, for a simple linear kernel, the number of fit parameters must be at least equal to the number of features, i.e.,

$$N_b \geq d\,. \tag{4.3}$$

Consequently, increasing the number of reciprocal lattice vectors N_G, and thus the descriptor dimension d, requires a corresponding increase in the number of fit parameters. Accordingly, the basis set size N_b, which directly determines the number of fit parameters, should generally exceed the number of training configurations N. This formally departs from the dual representation derived in subsection 2.1.2, where the basis set coincides with the training set. In the present approach, the regression model is parametrized by an explicitly chosen set of basis descriptors in the high-dimensional descriptor space, rather than directly by the training configurations themselves. The model thus remains linear in these kernel-induced basis functions, and regularization (according to Eq. (2.12) or Eq. (2.18)) ensures numerical stability despite $N_b > N$. Conceptually, this can be understood as a kernel-based linear regression with an explicitly chosen, overcomplete set of basis descriptors.

In the implementation, the basis set is enlarged using symmetry-based data augmentation, which is described in detail in section 4.2. Furthermore, the symmetry relation $S_{\alpha\beta}(\mathbf{q}) = S^*_{\alpha\beta}(-\mathbf{q})$, derived in Eq. (2.66), is explicitly exploited to reduce the dimensionality of the descriptor by evaluating it only for one representative of each pair of wave vectors $\pm\mathbf{q}$. The wave vectors $\mathbf{q}$ are restricted to reciprocal lattice vectors of the form $\mathbf{G} = h\,\mathbf{b}_1 + k\,\mathbf{b}_2 + l\,\mathbf{b}_3$ (see Eq. (2.74)), with integer indices h, k, l. To obtain a finite descriptor, these indices are restricted to $|h|, |k|, |l| \leq h_{\max}$. This index cutoff corresponds to an effective cutoff $q_{\max}$ in reciprocal space.

Figures 4.1 and 4.2 show the RMSE of the MLFF relative to the reference data for Si and NaCl, respectively. Along each curve, the number of fit parameters N_b is kept fixed, while the number of features N_G is continuously increased. For Si, the expected behavior is observed. As long as $N_G \leq N_b$, the size of the basis set does not limit the model performance, and the accuracy is determined solely by the number of features

included in the descriptor. The prediction improves as more reciprocal lattice vectors are added, reflecting the increasing amount of structural information captured by the descriptor. Once the threshold $N_G > N_b$ is exceeded, the error rises again, indicating that the information contained in the features can no longer be fully represented by the available fit parameters, in agreement with the analysis in Eq. (2.23).

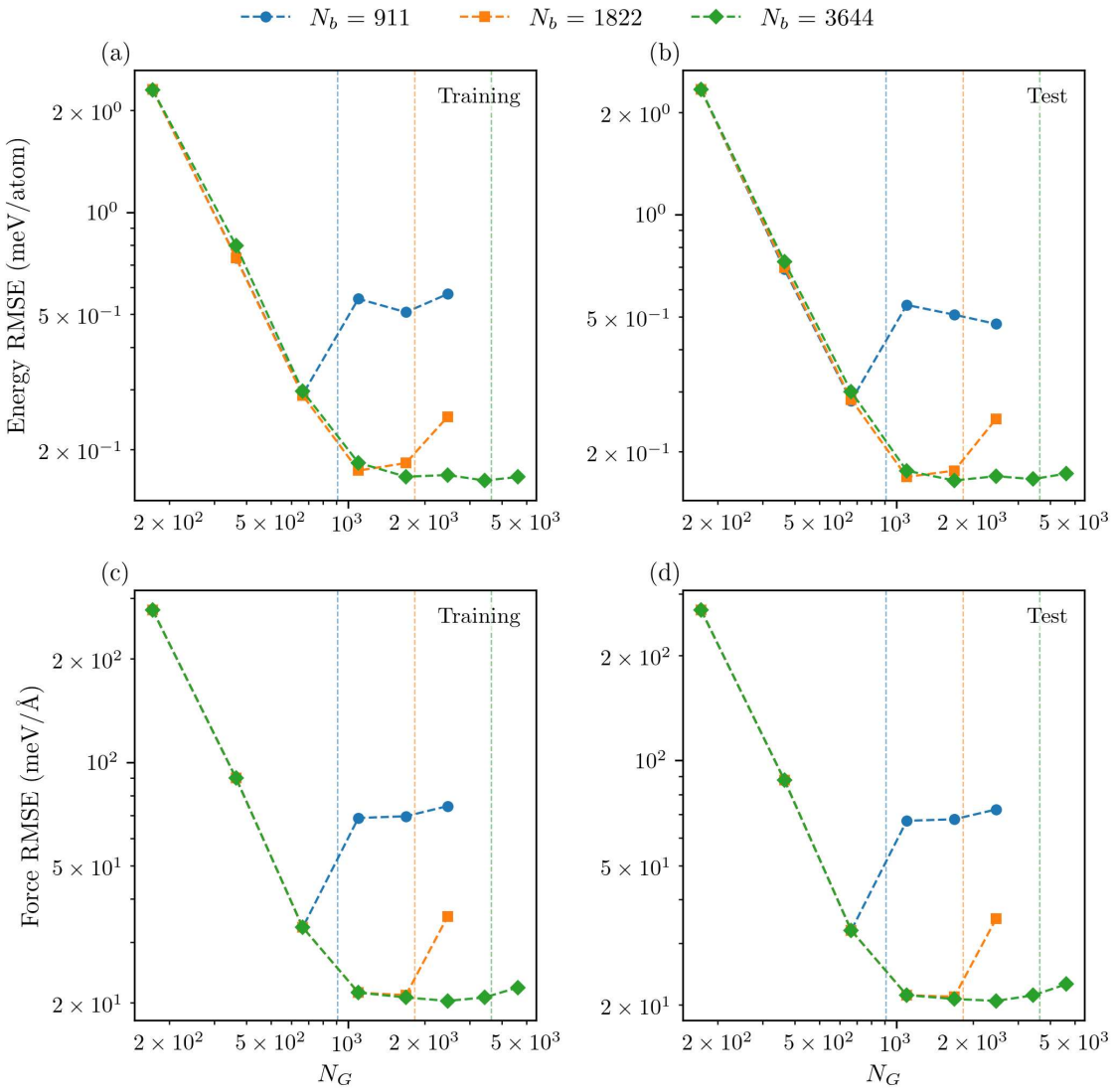

Figure 4.1: Dependence of the MLFF error on the number of features $d = N_G$ for Si. Each curve corresponds to a fixed number of fit parameters N_b, while the x-axis denotes the total number of descriptor features. Vertical lines mark the points where $d = N_b$. The subplots show (a) training energy RMSE, (b) test energy RMSE, (c) training force RMSE, and (d) test force RMSE.

A saturation in accuracy is also apparent. For Si, this saturation occurs at approximately 1688 features, corresponding to $h_{\max} = 15$, with an energy prediction RMSE (test set) of 0.164 meV atom^{-1} and a force prediction RMSE of 20.8 meV Å^{-1}. Here and throughout the text, all force errors are reported as component-wise RMSE values (see section A.4).

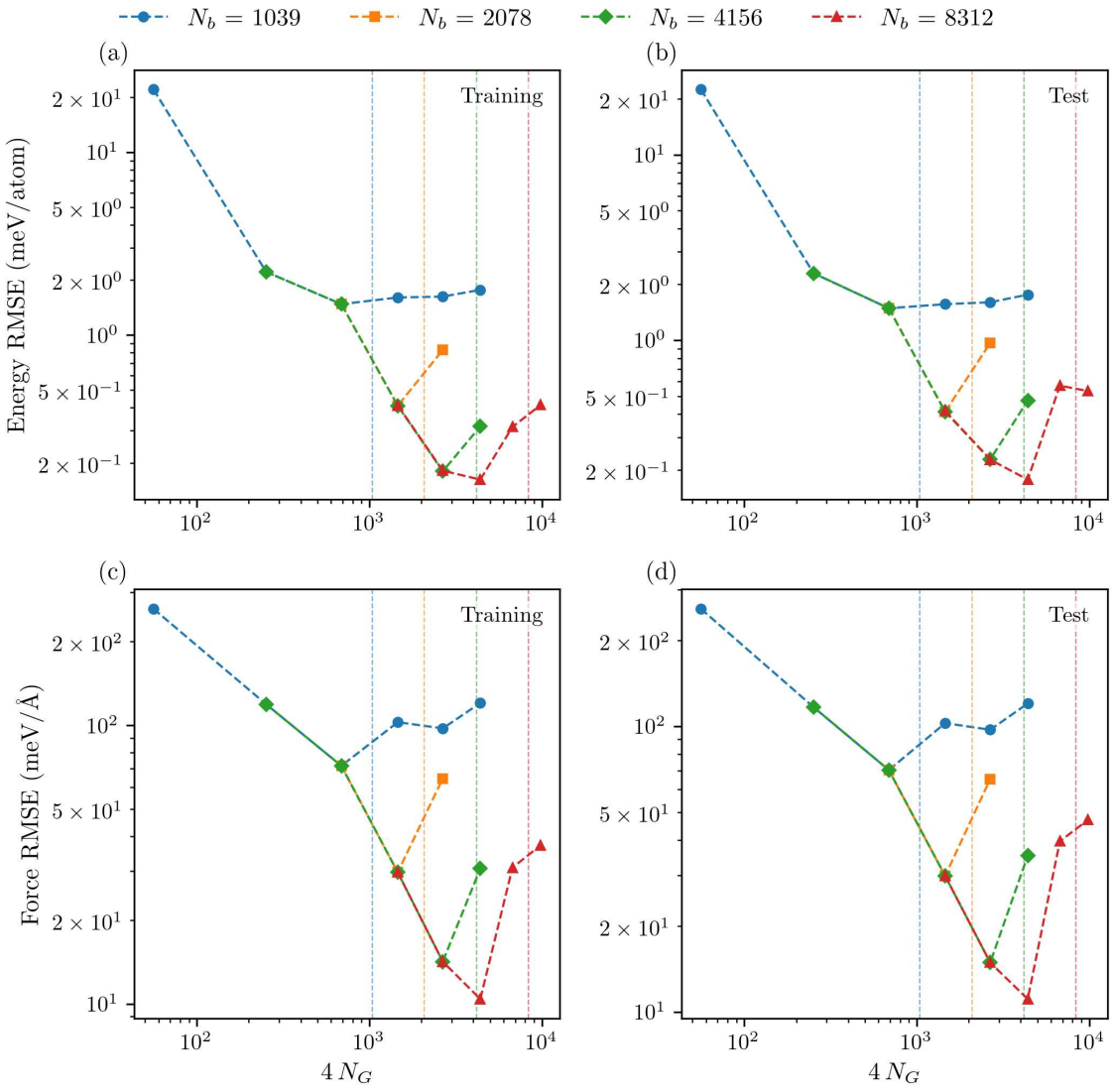

Figure 4.2: Dependence of the MLFF error on the number of features $d = 4\,N_G$ for NaCl. Each curve corresponds to a fixed number of fit parameters N_b, while the x-axis denotes the total number of descriptor features. Vertical lines mark the points where $d = N_b$. The subplots show (a) training energy RMSE, (b) test energy RMSE, (c) training force RMSE, and (d) test force RMSE.

For NaCl, which is a diatomic material, the descriptor consists of four partial structure factors, resulting in a total feature dimension of $d = 4\,N_G$. Analogous to Si, a similar trend is generally observed in Figure 4.2. However, the data point at $N_b = 8312$ and $d = 6752$ (corresponding to $N_G = 1688$) deviates from this trend. The energy and force RMSE increase from $0.178\,\mathrm{meV\,atom^{-1}}$ and $11.1\,\mathrm{meV\,Å^{-1}}$ at $d = 4396$ to $0.570\,\mathrm{meV\,atom^{-1}}$ and $39.6\,\mathrm{meV\,Å^{-1}}$ even though $d < N_b$. This behavior can be explained by the effective rank r_{eff} of the design matrix being lower than N_b. r_{eff} quantifies how many independent modes of the basis are effectively utilized by the regression solution. It is computed from the singular values S_i of the design matrix, where regularization suppresses modes associated with small singular values (see section A.1 for details).

In practice, the number of active modes is quantified using a threshold criterion,

$$r_{\mathrm{eff}} = \sum_i \Theta\left(\frac{S_i^2}{S_i^2 + \lambda} - \tau\right), \tag{4.4}$$

where $\Theta(\cdot)$ denotes the Heaviside step function, defined as $\Theta(x) = 1$ for $x > 0$ and $\Theta(x) = 0$ otherwise. $\tau = 0.5$ is used to identify modes that contribute significantly to the regression, and λ denotes the regularization parameter. Small singular values $S_i \lesssim \sqrt{\lambda}$ are effectively suppressed by the regularization and contribute little to the solution. In this definition, they are therefore not counted as active modes. Using this measure, the effective ranks and the corresponding energy and force RMSE values for selected calculations are summarized in Table 4.1.

From these values, it becomes evident that for $N_b = 8312$ and $d = 6752$, the effective rank is substantially smaller than the nominal number of features. This indicates that not all descriptor directions are effectively utilized by the regression, even though the basis size formally exceeds d, providing a plausible explanation for the observed increase in RMSE. Such a reduction in effective dimensionality may arise from linear dependencies or strong correlations among the symmetry-augmented basis descriptors.

In contrast, for $N_b = 8312$ and $d = 4396$, the effective rank is $r_{\mathrm{eff}} = 4392$, i.e., only slightly smaller than d, while the RMSE remains essentially unchanged. This suggests that the missing modes contribute only marginally and that the deviation of r_{eff} from d is mainly due to the threshold of 0.5 used to classify active modes, rather than a genuine loss of relevant information.

A qualitatively different behavior is observed when increasing the basis size to $N_b = 10390$ for $d = 6752$, where nearly all modes become active ($r_{\mathrm{eff}} \approx N_b$). Remarkably, this increase in effective rank occurs although the basis is constructed by augmenting the original 1039 configurations with seven symmetry operations to obtain 8312 vectors, and subsequently with only two additional operations to reach 10390 vectors.

Table 4.1: Effective rank r_{eff}, energy RMSE_E ($\text{meV}\,\text{atom}^{-1}$), and force RMSE_F ($\text{meV}\,\text{Å}^{-1}$) on the test set for NaCl, evaluated for different regularization parameters λ.

N_b	$d = 4N_G$	metric	$\lambda = 10^{-1}$	$\lambda = 10^{-4}$	$\lambda = 10^{-7}$	$\lambda = 10^{-9}$	$\lambda = 10^{-11}$	$\lambda = 10^{-13}$
8312	4396	r_{eff}	4325	4373	4387	4392	8311	8312
		RMSE_E	0.289	0.189	0.168	0.168	0.934	3.52
		RMSE_F	23.2	12.9	11.1	11.1	24.7	93.9
8312	6752	r_{eff}	5771	5834	5837	5841	8311	8312
		RMSE_E	0.583	0.570	0.567	0.579	0.643	0.716
		RMSE_F	39.6	39.6	39.5	39.6	39.6	39.6
10390	6752	r_{eff}	6613	6710	6741	10388	10389	10390
		RMSE_E	0.212	0.170	0.150	0.153	0.155	0.240
		RMSE_F	15.5	11.6	10.6	10.6	10.7	13.7

One possible explanation is that the smaller basis contains nearly redundant or strongly correlated directions. Additional symmetry operations may introduce slightly more independent contributions, enabling a better use of the descriptor space. The exact origin of this behavior, however, remains unclear and warrants further analysis.

Given that the effective rank is defined through the singular values of the design matrix, a natural question is to what extent the number of active modes can be influenced by the choice of the regularization parameter. The regularization parameter λ directly controls how strongly modes associated with small singular values are suppressed in the regression. Decreasing λ increases the effective rank by progressively activating these modes. As shown in Table 4.1, however, an increased effective rank does not necessarily lead to improved predictive accuracy. In particular, for $N_b = 8312$ and $d = 6752$, reducing λ results in a higher r_{eff} without a corresponding decrease in RMSE. This indicates that the additionally activated modes do not necessarily encode additional physically relevant information, but may instead correspond to numerically ill-conditioned directions or contributions dominated by numerical noise. This behavior is characteristic of Tikhonov (ridge) regularization.

A similar analysis for Si reveals a considerably more benign behavior. For all tested combinations of N_b and d, the effective rank remains close to the basis size. This is consistent with the trends observed in Fig. 4.1, where no unexpected increase in RMSE is found. The corresponding effective ranks are summarized in Table 4.2. Owing to this stable behavior, a more detailed effective-rank study was not pursued for Si.

Table 4.2: Effective rank r_{eff} for selected combinations of basis size N_b and descriptor dimension d for Si using regularization parameter $\lambda = 10^{-9}$ and a threshold of 0.5.

N_b	d	r_{eff}
3644	3430	3642
3644	4631	3643
6377	6084	6377

Based on these observations, all subsequent calculations follow the general guideline $N_b > d$, ensuring that the number of fit parameters is sufficient to represent the descriptor space. At the same time, N_b is chosen as small as possible in order to limit the computational cost of the regression. In practice, this requires monitoring the effective rank r_{eff}, since cases with $N_b > d$ can still exhibit an insufficient effective model capacity if r_{eff} is substantially smaller than d, as observed for $N_b = 8312$ and $d = 6752$ for NaCl. In such situations, an anomalous increase in RMSE serves as an indicator that the descriptor information is not fully captured by the regression solution.

A regularization parameter of $\lambda = 10^{-9}$ is used throughout, as it provides a good compromise between numerical stability and predictive accuracy. Overall, this analysis highlights that for multi-component systems the relationship between the number of features, the number of fit parameters, and the effective rank is nontrivial. In these cases, the effective rank offers a more reliable measure of the model capacity than the nominal values of d or N_b alone.

4.2 Symmetry Properties

In the previous section, it was shown that increasing the number of reciprocal lattice vectors enhances the expressive power of the structure factor descriptor. It also requires a corresponding increase in the number of fit parameters, however. This increase was achieved by enlarging the basis set using symmetry-based data augmentation. This procedure highlights a second central aspect that affects both model accuracy and data efficiency: the treatment of physical symmetries. As discussed in section 2.2, a suitable descriptor must encode the fundamental invariances of the underlying potential energy surface. In the following, we analyze how the structure factor behaves under these symmetry operations and how the resulting properties can be systematically exploited through symmetry-based data augmentation.

First, permutations of atoms belonging to the same chemical species are considered.

For the partial structure factor

$$S_{\alpha\beta}(\mathbf{q}) = \sum_{j\in\alpha}\sum_{k\in\beta} f_\alpha f_\beta \, e^{-i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)} \tag{4.5}$$

exchanging two atoms within the same set, e.g. $\mathbf{r}_j \leftrightarrow \mathbf{r}_{j'}$ with $j, j' \in \alpha$, only reorders the summation. Thus the value of $S_{\alpha\beta}(\mathbf{q})$ remains unchanged.

Second, the structure factor is invariant under global translations. Shifting all positions by the same vector $\mathbf{a}$,

$$\mathbf{r}_j \to \mathbf{r}_j + \mathbf{a} \tag{4.6}$$

leads to

$$e^{-i\mathbf{q}\cdot[(\mathbf{r}_j+\mathbf{a})-(\mathbf{r}_k+\mathbf{a})]} = e^{-i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)} \tag{4.7}$$

so that the entire expression for $S_{\alpha\beta}(\mathbf{q})$ remains unchanged.

Finally, to analyze the rotational behaviour of the structure factor as a descriptor in MLFFs, it is important to distinguish two types of rotations. A rigid rotation of the entire crystal, acting both on the atomic positions and the lattice vectors, leaves all physical observables strictly invariant. In contrast, one may also consider an internal rotation applied only to the atomic coordinates inside the unit cell, which is periodically continued along the fixed lattice vectors. In the following, we focus on the latter case and consider a rotation $R \in SO(3)$ acting solely on the atomic positions,

$$\mathbf{r}_j \to R\mathbf{r}_j \,, \tag{4.8}$$

while the lattice remains fixed. Such a transformation generally changes the energy of the system, except when R belongs to the point-group symmetry of the lattice, in which case the energy remains invariant, $E(\{\mathbf{r}_j\}) = E(\{R\mathbf{r}_j\})$.

For the structure factor, the transformed phase factor becomes

$$T'(\mathbf{q}) = \sum_j f_j \, e^{-i\mathbf{q}\cdot(R\mathbf{r}_j)} = \sum_j f_j \, e^{-i(R^{\mathrm{T}}\mathbf{q})\cdot\mathbf{r}_j} \,, \tag{4.9}$$

which directly yields the covariance relation

$$S(\mathbf{q}; \{R\mathbf{r}_j\}) = S(R^{\mathrm{T}}\mathbf{q}; \{\mathbf{r}_j\})\,. \tag{4.10}$$

Eq. (4.10) shows that the structure factor is not rotationally invariant but transforms covariantly. Rotating all atomic coordinates is equivalent to an inverse rotation of the wave vectors. For a full rigid rotation of the crystal (i.e. rotating both the lattice and the atomic positions), this covariance ensures that the structure factor remains unchanged.

The situation is different when only the internal coordinates are rotated while the lattice is held fixed. For rotations belonging to the point group of the lattice, the reciprocal lattice is mapped onto itself. Consequently, the discretized set of wave vectors used to define the descriptor, $\{\mathbf{q}_1, \ldots, \mathbf{q}_M\}$, is mapped onto itself as a set, i.e. for each i there exists an index $\sigma(i)$ such that

$$R^{\mathrm{T}}\mathbf{q}_i = \mathbf{q}_{\sigma(i)}\,. \tag{4.11}$$

Using Eq. (4.10), the structure factor therefore transforms as

$$S(\mathbf{q}_i; \{R\mathbf{r}_j\}) = S(\mathbf{q}_{\sigma(i)}; \{\mathbf{r}_j\})\,. \tag{4.12}$$

Thus, point-group operations leave the set of descriptor values invariant; however, they permute the individual components. If the descriptor is used in the ML model as an ordered feature vector with a fixed indexing of the $\mathbf{q}$-points, these permutations lead to physically equivalent configurations being represented by different feature vectors.

Since the ML model does not enforce rotational invariance by construction, this mismatch must be addressed explicitly. In the present implementation, this is achieved through symmetry-based data augmentation. All training and/or basis configurations are augmented using the 48 rotational operations of the cubic point group O_h (since the systems' super cells are cubic). For each rotation, only the atomic coordinates are transformed, while the $\mathbf{q}$-grid is kept fixed. All augmented configurations share the same energy target, while the corresponding force targets are rotated accordingly. This forces the regression model to identify symmetry-equivalent descriptor representations as physically equivalent.

The impact of symmetry-based data augmentation on model accuracy and data efficiency is illustrated in Figures 4.3 and 4.4. The figures show learning curves, reporting the RMSE on the test set as a function of the number of training configurations for both systems. Different curves correspond to different augmentation levels, ranging from 1× (no symmetry operations applied beyond the identity) to 48× (all cubic symmetry operations included).

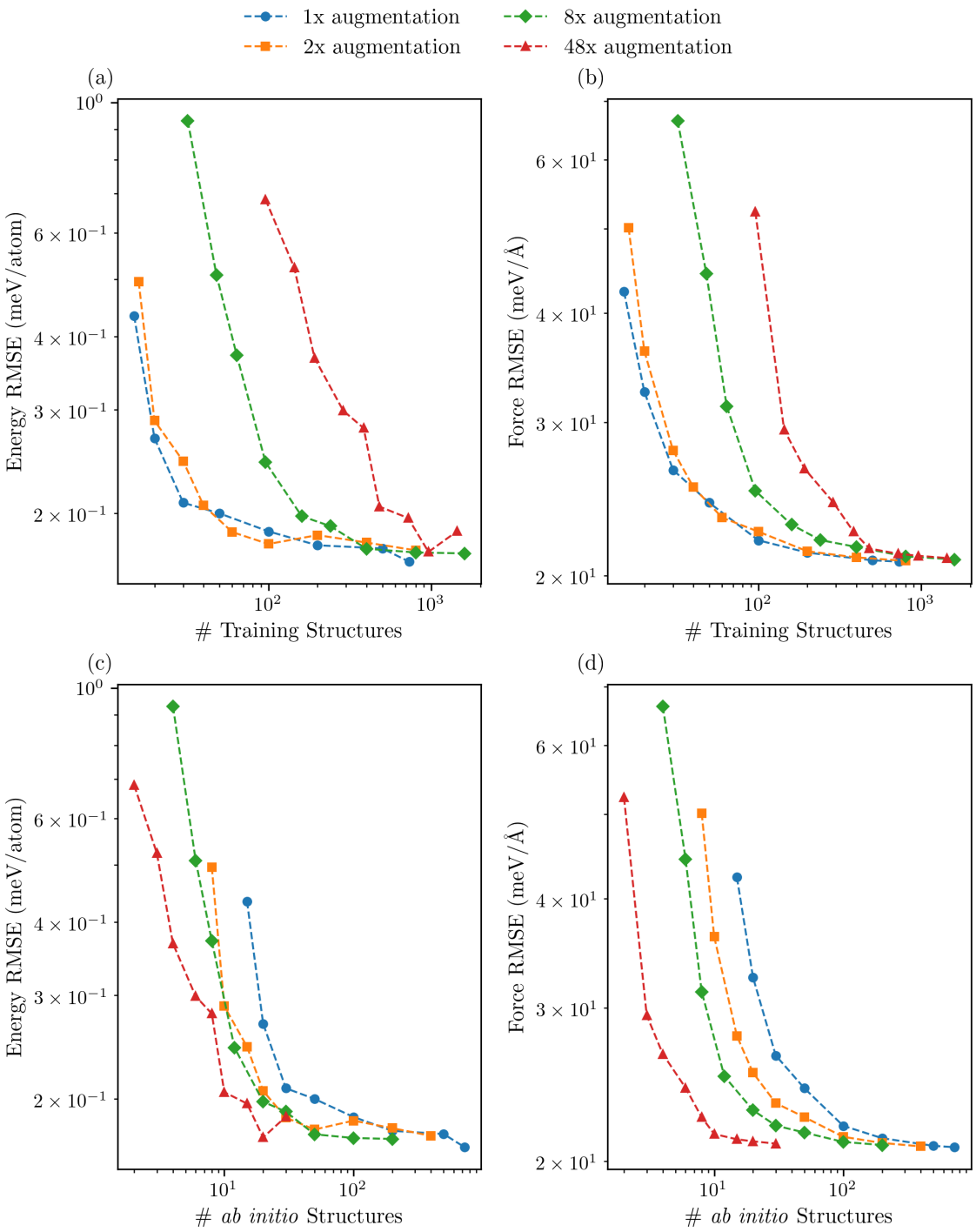

Figure 4.3: Learning curves for Si with different levels of symmetry-based data augmentation. (a) and (b) show test set RMSE for energies and forces versus the number of training configurations, calculated *ab initio* structures times number of augmentation. In (c) and (d), the x-axis is rescaled to the number of independent *ab initio* structures.

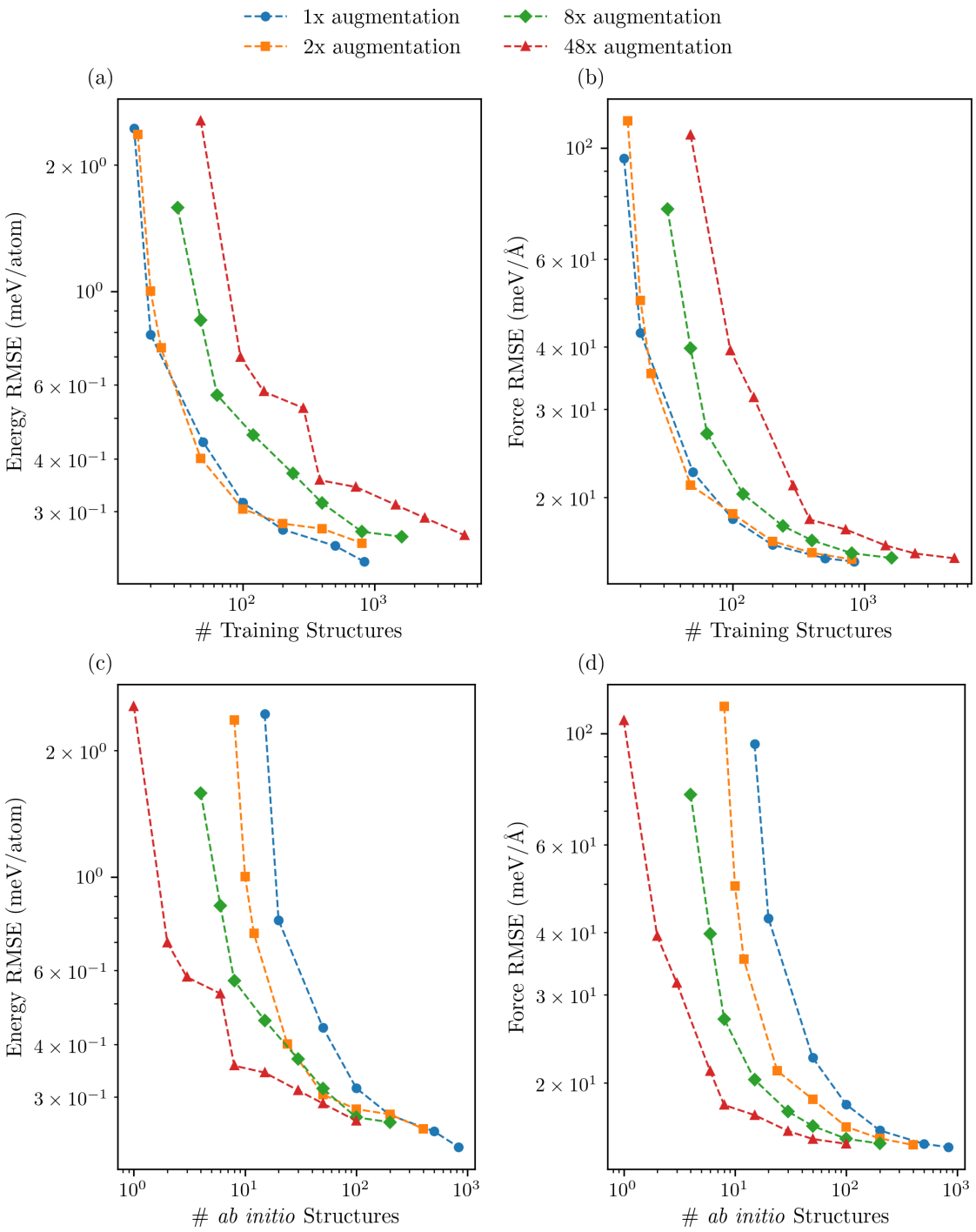

Figure 4.4: Learning curves for NaCl with different levels of symmetry-based data augmentation. (a) and (b) show test set RMSE for energies and forces versus the number of training configurations, calculated *ab initio* structures times number of augmentation. In (c) and (d), the x-axis is rescaled to the number of independent *ab initio* structures.

The learning-curve results in Figures 4.3 and 4.4. are obtained using a regularization parameter of $\lambda = 10^{-9}$. For Si, a feature dimension of $d = 1688$ ($h_{\max} = 15$) and a basis size of $N_b = 3644$ are employed, while for NaCl $d = 2664$ ($h_{\max} = 11$) and $N_b = 4156$ are used.

For each material and for both energy and force predictions, two representations are shown. In the first, the RMSE is plotted against the total number of augmented training configurations. In the second, the x-axis is rescaled to the number of original *ab initio* structures. These are obtained by dividing the number of augmented configurations by the corresponding augmentation factor and rounding up to the nearest integer. Training structures are added sequentially in random order. For augmented data, the original configuration is always included first, followed by its symmetry-generated replicas.

The learning curves reveal that stronger symmetry augmentation generally slows down convergence when measured in terms of the total number of (augmented) training configurations. This indicates that the original *ab initio* structures contain a larger amount of independent information per configuration. This behavior is expected, since the original structures correspond to distinct points in configuration space with different target energies, whereas the augmented configurations are symmetry-related copies. While these share identical energy labels with the original structure, the associated forces are merely rotated or permuted according to the applied symmetry operation.

For computational efficiency, the relevant quantity is the number of original *ab initio* structures, since generating them dominates the computational cost. Rescaling the horizontal axis accordingly highlights the practical benefit of data augmentation. For Si, applying the full set of cubic symmetry operations allows near-saturation of both energy and force accuracy with on the order of only ~ 20 *ab initio* structures, compared to several hundred structures required in the absence of augmentation.

For NaCl, a qualitatively similar trend to Si is observed, but with notable differences in the learning behavior of energies. While for Si the energy learning curves already exhibit stronger fluctuations and a slower convergence than the force curves—reflecting the generally higher difficulty of learning energies—this effect is significantly more pronounced for NaCl. For force predictions, full symmetry-based augmentation leads to a rapid reduction of the RMSE, reaching a near-saturation level already after approximately 30–50 *ab initio* structures. In contrast, without augmentation, a comparable accuracy is only achieved after roughly 200–500 *ab initio* configurations. This demonstrates a substantial gain in data efficiency for forces through symmetry augmentation.

The situation is less clear-cut for energy predictions. When additional *ab initio* structures are added, the energy RMSE initially decreases rapidly, up to about 8 *ab initio*

structures for the 48× augmented dataset and about 100–200 structures without augmentation. Beyond this point, no clear saturation is observed for the energy RMSE. Instead, the improvement becomes markedly slower, and the relative benefit of further augmentation gradually diminishes. This indicates that, for NaCl, symmetry-generated configurations provide less additional independent information for energy learning than for forces, and that energies require a substantially larger set of distinct *ab initio* configurations to approach saturation.

4.3 Three-Body Structure Factor

Up to this point, the MLFF has been constructed exclusively using the two-body structure factor $S(\mathbf{q})$ as a descriptor. While this representation captures pairwise atomic correlations, it is inherently limited in its ability to encode higher-order structural information. In particular, angular correlations and cooperative effects involving three atoms cannot be represented explicitly. To address this limitation, the descriptor features are extended by a three-body structure factor, which encodes correlations among atomic triplets.

Analogously to the two-body case (see Eq. (2.52)), the three-body structure factor, neglecting normalization, can be expressed in terms of phase factors associated with each atom as

$$S^{(3)}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \propto \sum_{j,k,l} e^{i\mathbf{q}_1 \cdot \mathbf{r}_j} e^{i\mathbf{q}_2 \cdot \mathbf{r}_k} e^{i\mathbf{q}_3 \cdot \mathbf{r}_l}. \tag{4.13}$$

Physical observables are invariant under a uniform translation of the entire system. Therefore, the three-body structure factor must remain unchanged under the transformation $\mathbf{r}_j \to \mathbf{r}_j + \mathbf{a}$, $\mathbf{r}_k \to \mathbf{r}_k + \mathbf{a}$, $\mathbf{r}_l \to \mathbf{r}_l + \mathbf{a}$. Applying this to the expression above yields

$$S^{(3)}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \to e^{i(\mathbf{q}_1+\mathbf{q}_2+\mathbf{q}_3)\cdot \mathbf{a}}\, S^{(3)}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3). \tag{4.14}$$

For the structure factor to be invariant under translation, the exponential prefactor must equal unity for any $\mathbf{a}$, which is only satisfied if

$$\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 = 0\,. \tag{4.15}$$

This condition is formally equivalent to momentum conservation in a scattering process and reflects the translational invariance of the system. As a result, the three-body

structure factor effectively depends only on two independent wave vectors, e.g.,

$$S^{(3)}(\mathbf{q}_1, \mathbf{q}_2) \propto \sum_{j,k,l} e^{i\mathbf{q}_1(\mathbf{r}_j - \mathbf{r}_l)} e^{i\mathbf{q}_2(\mathbf{r}_k - \mathbf{r}_l)} \,. \tag{4.16}$$

In the species-resolved formulation, the partial three-body structure factor $S^{(3)}_{\alpha\beta\gamma}(\mathbf{q}_1, \mathbf{q}_2)$ is obtained by restricting the indices j, k, and l to atoms of species α, β, and γ, respectively. Due to the asymmetric role of the three indices in Eq. (4.16), the resulting descriptor does not exhibit full permutation symmetry. Instead, only the following restricted symmetry relations hold

$$S^{(3)}_{\alpha\beta\beta}(\mathbf{q}_1, \mathbf{q}_2) = S^{(3)*}_{\alpha\beta\beta}(-\mathbf{q}_1, \mathbf{q}_2) \tag{4.17}$$

$$S^{(3)}_{\beta\alpha\beta}(\mathbf{q}_1, \mathbf{q}_2) = S^{(3)*}_{\beta\alpha\beta}(\mathbf{q}_1, -\mathbf{q}_2) \tag{4.18}$$

$$S^{(3)}_{\alpha\beta\gamma}(\mathbf{q}_1, \mathbf{q}_2) = S^{(3)}_{\beta\alpha\gamma}(\mathbf{q}_2, \mathbf{q}_1) = S^{(3)*}_{\alpha\beta\gamma}(-\mathbf{q}_1, -\mathbf{q}_2) \,. \tag{4.19}$$

For force training, derivatives of the three-body structure factor with respect to atomic positions are required. The derivative with respect to the Cartesian component μ of atom i reads (see subsection A.2.2 for a detailed derivation)

$$\begin{aligned} \frac{\partial S^{(3)}_{\alpha\beta\gamma}(\mathbf{q}_1, \mathbf{q}_2)}{\partial r_{i,\mu}} \propto\ & \mathbf{1}_{\{i\in\alpha\}}\, i q_{1,\mu} \sum_{k\in\beta} \sum_{l\in\gamma} \mathrm{e}^{\,i\mathbf{q}_1\cdot(\mathbf{r}_i - \mathbf{r}_l)}\ \mathrm{e}^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k - \mathbf{r}_l)} \\ & +\ \mathbf{1}_{\{i\in\beta\}}\, i q_{2,\mu} \sum_{j\in\alpha} \sum_{l\in\gamma} \mathrm{e}^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j - \mathbf{r}_l)}\ \mathrm{e}^{\,i\mathbf{q}_2\cdot(\mathbf{r}_i - \mathbf{r}_l)} \\ & -\ \mathbf{1}_{\{i\in\gamma\}}\, i(q_{1,\mu} + q_{2,\mu}) \sum_{j\in\alpha} \sum_{k\in\beta} \mathrm{e}^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j - \mathbf{r}_i)}\ \mathrm{e}^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k - \mathbf{r}_i)} \,. \end{aligned} \tag{4.20}$$

Having established the formal definition and symmetry properties of the three-body structure factor, the focus now shifts to its practical realization as a descriptor and to the corresponding modifications required in the MLFF implementation.

For the three-body contribution, the descriptor is constructed from a finite set of reciprocal-space wave vectors. Specifically, the $N_q^{(3)}$ vectors with the smallest $|\mathbf{q}|$-magnitudes are selected, and all ordered pairs $(\mathbf{q}_1, \mathbf{q}_2)$ are formed from this set, with each pair defining one three-body feature. This rank-based selection effectively implements a spherical truncation in reciprocal space without specifying an explicit cutoff

radius. By controlling $N_q^{(3)}$, the dimensionality of the three-body part of the descriptor is directly controlled, as it results in ${N_q^{(3)}}^2$ ordered wave-vector pairs, each defining one three-body feature.

Inversion symmetry is preserved by enforcing the inclusion of $\pm\mathbf{q}$ pairs in the wave-vector set; however, the symmetry relations discussed in Eq. (4.19) are not exploited to reduce the number of features. Instead, all wave-vector pairs are treated as independent, and the full descriptor is obtained by concatenating all features of the partial two-body and three-body structure factors.

In the original workflow illustrated in Figure 3.1, descriptors and their derivatives are computed in full for all configurations and subsequently assembled into the design matrix. This approach is well suited for two-body descriptors and is particularly convenient for parallel execution, as all configurations and wave-vector contributions can be processed simultaneously. However, once three-body features are included, this strategy becomes impractical. Owing to the quadratic scaling of the number of three-body descriptors with the number of wave vectors, the memory consumption increases rapidly and can exceed available system memory.

To address this limitation, the original workflow is replaced by the adapted strategy shown in Figure 4.5. The computation is instead performed in batches of configurations. For each batch, partial two-body and three-body structure factors and their derivatives are calculated and concatenated into descriptor vectors. These vectors are then used to construct the corresponding blocks of the design matrix. After processing a batch, temporary arrays are released, and the full design matrix is obtained by concatenating all blocks.

This batching strategy substantially reduces the peak memory footprint but comes at the expense of reduced parallelism, as parallel execution is limited to configurations within a batch. Consequently, the overall computational time increases. Analytical expressions are retained for the derivatives of the two-body descriptors, while the derivatives of the three-body descriptors are computed using automatic differentiation in the `JAX` library. Due to the resulting computational cost, systematic three-body parameter studies are restricted to Si. For NaCl, the presence of two chemical species introduces partial structure factors $S^{(3)}_{\alpha\beta\gamma}(\mathbf{q}_1, \mathbf{q}_2)$, which increase the number of three-body features further by a factor of eight. Despite the adapted implementation, this leads to substantial memory requirements. Therefore, no comprehensive parameter study was performed for NaCl. Instead, only selected comparative calculations were carried out.

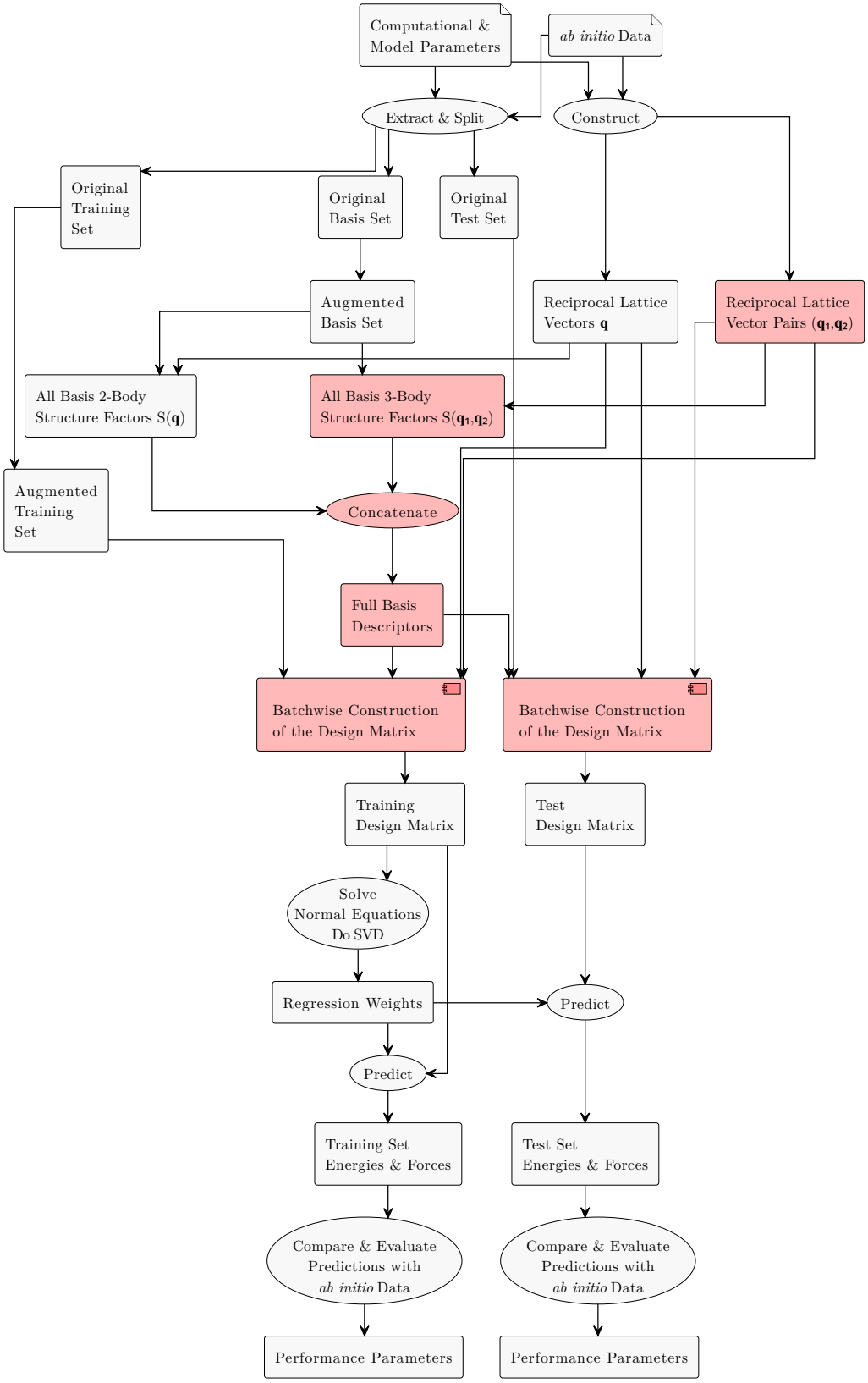

Figure 4.5: Component diagram of the adapted (red) MLFF workflow to incorporate three-body features.

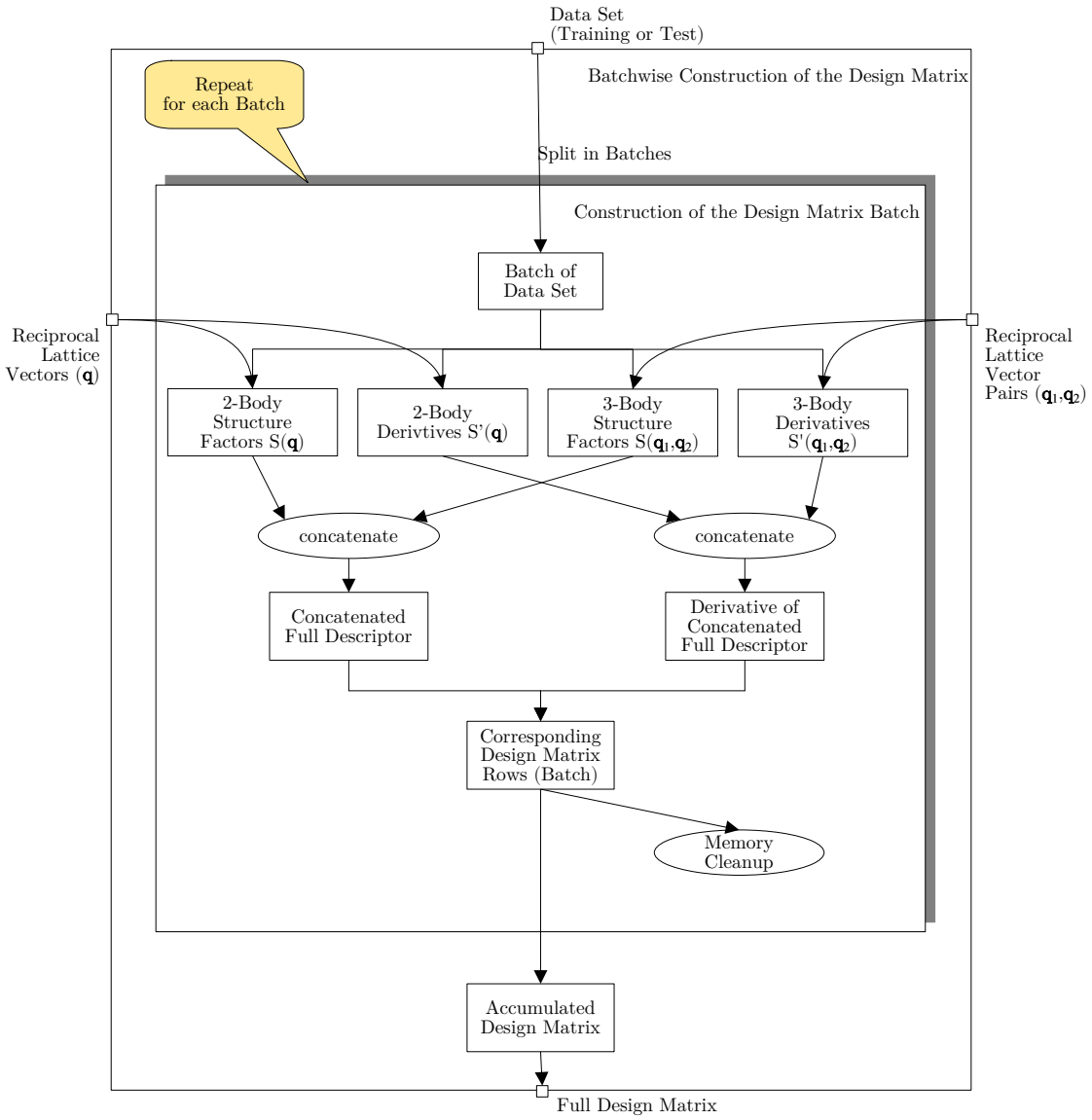

Figure 4.6: Component diagram for the batchwise construction of the design matrix.

Figure 4.7 illustrates the effect of expanding the descriptor with the three-body structure factor for the Si test system. All results were obtained using a regularization parameter of $\lambda = 10^{-5}$. This stronger regularization was necessary to stabilize the fit and achieve consistently lower errors, as the model contains a larger number of descriptor components and, correspondingly, more fit parameters (see section 4.1). No data augmentation was applied.

For a fixed number of two-body features, the number of three-body features is increased systematically. Each three-body feature corresponds to an ordered pair $(\mathbf{q}_1, \mathbf{q}_2)$ of reciprocal-space wave vectors, such that the square root of the number of three-body

features reflects the number of distinct reciprocal-space wave vectors employed in the three-body part of the descriptor.

The inclusion of three-body features leads to a systematic reduction of the RMSE, with an overall improvement of approximately $0.1\,\mathrm{meV\,atom^{-1}}$ for energies and 10–$12\,\mathrm{meV\,Å^{-1}}$ for forces, largely independent of the number of two-body features. However, an insufficient two-body representation cannot be compensated by simply adding three-body features, as evidenced by the model with only 666 two-body features (corresponding to $h_{\max} = 11$, i.e. a small cutoff in reciprocal space indices in Eq. (2.74)), which remains significantly less accurate even when many three-body features are included. This demonstrates that three-body information complements, but does not replace, an adequate two-body description.

The impact of additional three-body features is non-uniform. The first ~ 1000 features, corresponding to pairs for which both wave vectors lie close to the Γ point (pairs of roughly 32 distinct wave vectors), produce only a marginal reduction in RMSE. This suggests that these innermost pairs carry limited additional structural information. Beyond this range, RMSE decreases more noticeably and saturates around 5000–7500 three-body features (roughly 70–85 wave vectors). As with other aspects of the RMSE reduction, this behavior is consistent across all two-body feature sets, indicating that it reflects the intrinsic shape of the learning curve rather than specifics of a particular two-body representation.

For NaCl, the largest calculation employed $N_q^{(3)} = 50$ distinct three-body wave vectors, corresponding to 2500 $(\mathbf{q}_1, \mathbf{q}_2)$ pairs. Due to the partial structure factors of the biatomic material, this results in a total of 20 000 three-body features. In this configuration, the RMSE is $0.151\,\mathrm{meV\,atom^{-1}}$ for energies and $8.66\,\mathrm{meV\,Å^{-1}}$ for forces, compared to $0.173\,\mathrm{meV\,atom^{-1}}$ and $10.5\,\mathrm{meV\,Å^{-1}}$ for the corresponding two-body-only model. In both cases, the two-body descriptor uses $h_{\max} = 15$. Although no systematic parameter study was performed for NaCl, the results follow the same qualitative trend observed for Si in Figure 4.7, suggesting that further improvements are expected upon inclusion of additional three-body features.

Based on these observations, a practical strategy for model construction emerges. First, increase the number of two-body features until the RMSE saturates. If additional improvement is needed, three-body features can be added until a second saturation is achieved. Since the inclusion of three-body descriptors is computationally expensive, the results indicate that some optimizations may be possible: innermost pairs could be omitted, additionally the information content of the other pairs could be evaluated to retain only the most relevant contributions, and the symmetry relations in Eq. (4.19) could be exploited to reduce the dimensionality of the descriptor.

In summary, the three-body structure factor provides an improvement over two-body-only models. Whether and under which conditions these gains justify the additional

computational cost cannot be clearly determined based on the present study.

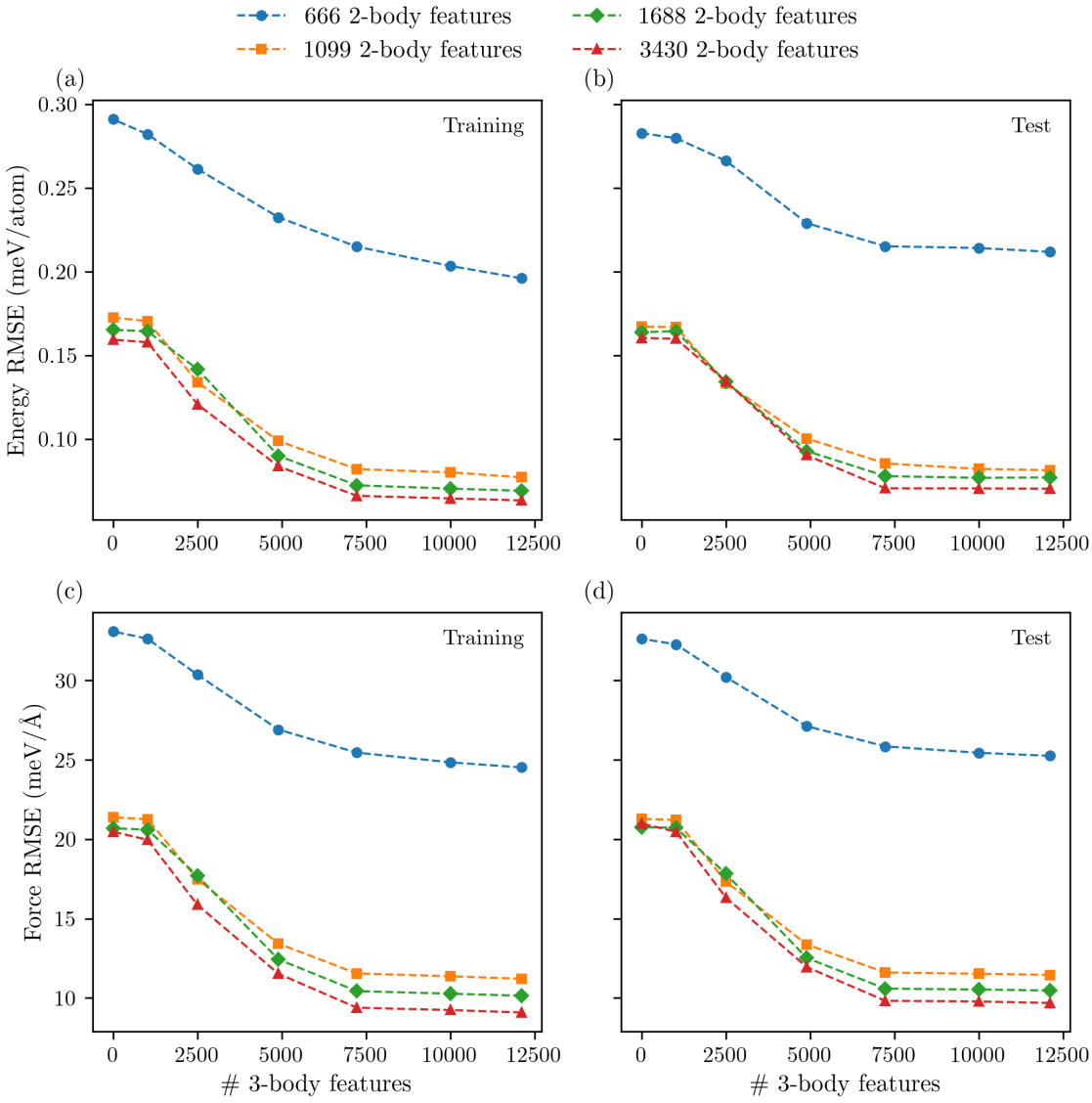

Figure 4.7: Impact of three-body features on the model accuracy for Si. The x-axis shows the number of three-body features (wave-vector pairs $(\mathbf{q}_1, \mathbf{q}_2)$), the y-axis the RMSE for energies (a,b) and forces (c,d). Panels (a,c) show training error, (b,d) test error. Different curves correspond to varying numbers of included two-body features.

4.4 Kernel

The machine-learning model employed in this work is formulated within a kernel regression framework, which is reflected, for example, in the construction of the design matrix according to Eq. (2.40). As discussed in subsection 2.1.2, kernel regression enables an efficient implicit mapping of input descriptors into higher-dimensional feature spaces, thereby allowing the model to represent complex, potentially nonlinear relationships between descriptors and target quantities.

In all preceding model parameter studies, a linear kernel defined by the scalar product in Eq. (2.22) was employed. This choice is equivalent to standard linear regression (see Eq. (2.23)) and does not exploit the nonlinear representational power of kernel methods. A natural question is whether introducing additional kernel-induced nonlinearity can further improve the predictive performance of the model. To address this question, polynomial kernels of the form

$$k_{\mathrm{poly}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^\chi \tag{4.21}$$

are considered in the following, where all symbols are defined in subsection 2.1.2. For $\chi = 1$ and $c = 0$, this expression reduces to the linear kernel. Unless stated otherwise, all kernel studies presented in this section were performed for the Si test system.

First, the influence of the polynomial degree χ is examined. For two representative descriptor sizes, the resulting energy and force RMSEs are summarized in Table 4.3. Increasing the polynomial degree beyond the linear case leads to a systematic deterioration of prediction accuracy on the test set. This trend is observed consistently for both energies and forces and for both investigated values of h_{max}. The linear kernel yields the lowest errors, and quadratic and quartic kernels perform increasingly worse, indicating that the additional nonlinearity introduced by higher-order polynomial kernels is not beneficial in this setting.

A possible explanation for this behavior is that polynomial kernels amplify correlations between already highly informative descriptor components. To investigate whether this degradation can be mitigated, both the kernel offset parameter c and the regularization parameter λ were varied over several orders of magnitude for the quadratic kernel ($\chi = 2$), while keeping all other parameters fixed.

The corresponding results, provided in Appendix B.1, show no systematic dependence on either parameter. This indicates that neither tuning the offset parameter nor increasing regularization improves performance, and that the reduced accuracy of polynomial kernels is not primarily caused by insufficient regularization or an unfavorable choice of c.

Table 4.3: Effect of the polynomial degree χ on the RMSE of energies (E) and forces (F) for Si. All results were obtained with $c = 0$ and $\lambda = 10^{-9}$.

χ	$h_{\max}$	N_G	$\mathrm{RMSE}_E^{\mathrm{train}}$ (meV atom^{-1})	$\mathrm{RMSE}_E^{\mathrm{test}}$ (meV atom^{-1})	$\mathrm{RMSE}_F^{\mathrm{train}}$ (meV Å^{-1})	$\mathrm{RMSE}_F^{\mathrm{test}}$ (meV Å^{-1})
1	13	1099	0.176	0.171	21.4	21.3
2	13	1099	0.348	0.387	48.7	50.1
4	13	1099	0.446	0.462	60.2	61.1
1	15	1688	0.167	0.169	20.7	20.8
2	15	1688	0.273	0.382	39.6	39.3
4	15	1688	0.398	0.524	50.2	50.3

Table 4.4: Comparison of linear ($\chi = 1$) and quadratic ($\chi = 2$) kernels for reduced descriptor sizes Si.

χ	$h_{\max}$	N_G	$\mathrm{RMSE}_E^{\mathrm{train}}$ (meV atom^{-1})	$\mathrm{RMSE}_E^{\mathrm{test}}$ (meV atom^{-1})	$\mathrm{RMSE}_F^{\mathrm{train}}$ (meV Å^{-1})	$\mathrm{RMSE}_F^{\mathrm{test}}$ (meV Å^{-1})
1	7	172	2.30	2.34	278	270
2	7	172	2.02	2.01	240	240
1	9	365	0.742	0.700	89.9	87.8
2	9	365	0.631	0.683	86.5	86.9
1	11	666	0.294	0.300	33.1	32.6
2	11	666	0.309	0.385	40.1	41.3

One remaining question is whether polynomial kernels might be advantageous in a regime where only a limited number of descriptor components is available. To test this hypothesis, kernels of degree $\chi = 1$ and $\chi = 2$ were compared for reduced values of $h_{\max}$. The results are reported in Table 4.4. While the performance gap between linear and quadratic kernels decreases for very small descriptor sizes, the polynomial kernel does not consistently outperform the linear kernel in any regime. In particular, for moderately sized descriptors, the linear kernel again yields superior generalization performance.

To assess whether these observations are specific to the Si system, polynomial kernels were additionally evaluated for NaCl. Representative results are shown in Table 4.5. As for Si, increasing the polynomial degree leads to a pronounced degradation of both energy and force prediction accuracy, confirming that the observed behavior is not system-specific.

Table 4.5: Effect of the polynomial degree χ for NaCl ($h_{\max} = 13$, $\lambda = 10^{-9}$).

χ	$\mathrm{RMSE}_{\mathrm{E}}^{\mathrm{train}}$ (meV atom^{-1})	$\mathrm{RMSE}_{\mathrm{E}}^{\mathrm{test}}$ (meV atom^{-1})	$\mathrm{RMSE}_{\mathrm{F}}^{\mathrm{train}}$ (meV Å^{-1})	$\mathrm{RMSE}_{\mathrm{F}}^{\mathrm{test}}$ (meV Å^{-1})
1	0.152	0.168	10.4	11.1
2	0.795	1.17	63.0	71.0
4	1.37	1.66	90.0	98.3

Overall, these results show that polynomial kernels offer no advantage over the linear kernel for the investigated systems, descriptor sizes, kernel parameters, or regularization strengths. Instead, increasing the kernel nonlinearity systematically worsens prediction accuracy. This behavior suggests that the structure factor-based descriptors employed in this work already constitute a highly expressive and physically informed representation of the atomic structures. The intrinsic nonlinearity of the structure factor with respect to atomic positions appears sufficient, such that additional kernel-induced nonlinearity primarily amplifies redundancies rather than introducing useful new information.

Consequently, the linear kernel does not represent a limitation of the present model, but rather the appropriate choice for the considered descriptors. Although polynomial kernels were not explicitly evaluated for three-body descriptors due to their substantially higher computational cost, a similar behavior is expected, as the underlying representation is already significantly more expressive.

Radial basis function (Gaussian) or related exponential kernels were not explicitly investigated in this study. However, the implemented framework would allow for their straightforward inclusion without substantial modification of the underlying codebase. It would be interesting to test whether such kernels become beneficial when a stronger emphasis on local structural differences or multiple length scales is required. A systematic comparison of Gaussian kernels combined with structure factor descriptors is therefore left for future work.

5 Performance Assessment

Having identified suitable model parameters in the previous chapter, now the central question of this thesis is addressed. Specifically, how well do the structure factor

$$S(\mathbf{q}) \propto \sum_{j=1}^{N} \sum_{k=1}^{N} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \,, \tag{5.1}$$

(variables defined in Eq. (2.51)) and its three-body extension

$$S^{(3)}(\mathbf{q}_1, \mathbf{q}_2) \propto \sum_{j,k,l} e^{i\mathbf{q}_1\cdot(\mathbf{r}_j - \mathbf{r}_l)} e^{i\mathbf{q}_2\cdot(\mathbf{r}_k - \mathbf{r}_l)} \,, \tag{5.2}$$

(see also Eq. (4.16)) perform as descriptors for MLFFs compared to more traditional approaches.

Theoretical aspects of the structure factor and its suitability as descriptor were discussed in section 2.3 and in chapter 4. Furthermore, prior results have already highlighted several key points. In section 4.2 it was shown that only a small number of *ab initio* reference structures are needed to train the model (for Si: ≈ 20 and for NaCl: ≈ 40), in section 4.3 that the three-body descriptor improves over the two-body variant (for Si: $0.1\,\mathrm{meV\,atom^{-1}}$ for energies and 10–$12\,\mathrm{meV\,\AA^{-1}}$ for forces), and in section 4.4 that the structure factor is already highly expressive and does not require additional high-dimensional kernel mappings.

In this chapter, the performance of the structure factor descriptor is quantitatively evaluated by comparison with established benchmark models. The primary reference is provided by the VASP-MLFF framework, as it shares the same kernel-based regression structure and thus enables a controlled assessment of descriptor quality [12, 15–17]. Results obtained with the MACE architecture are included for additional context [18].

As discussed in section 4.1, the number of reciprocal lattice vectors (or wave vectors) constitutes a key hyperparameter that controls the performance of the structure factor representation. For all results reported in this chapter, the following parameter settings were used: For Si, the two-body representation employs $h_{\max} = 15$ corresponding to

$N_G = 1688$ reciprocal lattice vectors, while the three-body extension uses an additional $N_q^{(3)} = 85$ wave vectors, resulting in $85^2 = 7225$ pair features. For NaCl, the two-body model is based on $h_{\max} = 17$ ($N_G = 2457$), which, due to the inclusion of partial structure factors, yields $4 \times 2457 = 9826$ effective features. The corresponding three-body model uses $h_{\max} = 15$ ($N_G = 1688$ and 6752 features) for the two-body part, combined with $N_q^{(3)} = 50$ wave vectors for three-body part, resulting in $50^2 = 2500$ pair features and, including partial structure factors, a total of $9 \times 2500 = 22500$ three-body features.

5.1 Kernel-Based Model Comparison

The VASP-MLFF model is a kernel-based machine learning force field implementation following the local descriptor philosophy introduced in subsection 2.2.1, where the total energy is decomposed into atomic contributions (see Eq. (2.29)). Atomic environments are evaluated within finite cut-off radii of 8.0 Å for radial (two-body) contributions and 5.0 Å for angular (three-body) contributions, consistent with the default settings of the VASP-MLFF framework. The descriptor construction is based on explicit radial and angular terms analogous to Eq. (2.30) and Eq. (2.31), where radial contributions are expressed in terms of normalized spherical Bessel functions, while angular correlations are represented through Legendre polynomials, ensuring rotational invariance consistent with the general framework of local rotationally invariant descriptors introduced in subsection 2.2.1. To enable a direct and fair comparison with the structure factor approach, separate reference models are constructed using either radial terms only (two-body) or combined radial and angular terms (two- and three-body).

All reference models are trained on the same *ab initio* dataset employed for the structure factor MLFFs. This ensures a controlled comparison in which differences in performance can be directly attributed to the underlying descriptor construction. The quantitative results summarized in Tables 5.1–5.4 demonstrate a clear and consistent advantage of the structure factor descriptor across both materials and target quantities.

For Si, the structure factor model outperforms the VASP reference. Notably, the two-body structure factor model even surpasses the three-body VASP reference, with an energy (test set) RMSE of 0.164 meV atom^{-1} compared to 0.287 meV atom^{-1}, and a force RMSE of 20.7 meV Å^{-1} versus 31.5 meV Å^{-1}. It should be noted that this comparison involves different descriptor orders and therefore does not constitute a strictly like-for-like comparison. Nevertheless, the three-body VASP model entails a substantially larger feature space, higher computational cost, and increased memory requirements, which limits its practical applicability. In this context, achieving comparable or better performance with the two-body structure factor is already indicative of its efficiency. Inclusion of three-body contributions in the structure factor

further improves the performance to 0.074 meV atom^{-1} for energies and 10.5 meV Å^{-1} for forces, representing an approximate upper bound for the achievable accuracy of the descriptor.

Table 5.1: Energy (meV/atom) prediction RMSE for Si. Relative errors with respect to the standard deviation of the reference data are given in parentheses.

		Structure Factor		VASP-MLFF	
		Training	Test	Training	Test
2-body	MAE	0.129 (0.8%)	0.129 (0.8%)	0.575 (3.4%)	0.587 (3.4%)
	RMSE	0.167 (1.0%)	0.164 (1.0%)	0.781 (4.6%)	0.781 (4.6%)
	MAX	0.772 (4.5%)	0.449 (2.6%)	3.30 (19%)	2.29 (13%)
3-body	MAE	0.058 (0.3%)	0.060 (0.3%)	0.189 (1.1%)	0.217 (1.3%)
	RMSE	0.077 (0.4%)	0.074 (0.4%)	0.243 (1.4%)	0.287 (1.7%)
	MAX	0.311 (1.8%)	0.214 (1.2%)	1.08 (6.3%)	0.856 (5.0%)

Table 5.2: Force (meV Å^{-1}) prediction RMSE for Si. Relative errors with respect to the standard deviation of the reference data are given in parentheses.

		Structure Factor		VASP-MLFF	
		Training	Test	Training	Test
2-body	MAE	16.3 (2.6%)	16.1 (2.6%)	51.3 (8.1%)	49.9 (7.9%)
	RMSE	20.9 (3.3%)	20.7 (3.3%)	66.8 (11%)	65.0 (10%)
	MAX	111 (17%)	102 (16%)	643 (102%)	425 (67%)
3-body	MAE	7.98 (1.3%)	7.82 (1.3%)	23.2 (3.7%)	24.3 (3.8%)
	RMSE	10.7 (1.7%)	10.5 (1.7%)	29.9 (4.7%)	31.5 (5.0%)
	MAX	96.5 (15%)	79.1 (13%)	226 (36%)	210 (33%)

For NaCl, a similar trend is observed, although the performance advantage is somewhat smaller. In particular, the two-body structure factor model achieves an energy RMSE of 0.178 meV atom^{-1} and a force RMSE of 10.5 meV Å^{-1}, and therefore slightly outperforms the three-body VASP reference, which yields 0.251 meV atom^{-1} and 12.4 meV Å^{-1} on the test set. In the two-body setting, however, the structure factor model clearly outperforms the radial VASP model, reducing the energy RMSE from 0.384 meV atom^{-1} and the force RMSE from 16.8 meV Å^{-1}. The present three-body model achieves an energy RMSE of 0.151 meV atom^{-1} and a force RMSE of 8.66 meV Å^{-1} on the test set. It should be noted, however, that the three-body structure factor model was not fully hyperparameter-optimized due to computational limitations (see section 4.3) and therefore does not represent the attainable lower bound. In analogy to Si, a further reduction of the error is expected upon full optimization, which would further increase the performance gap in favor of the structure factor approach.

Interestingly, an indication of overfitting is observed for the VASP-MLFF three-body model, as its test errors are consistently higher than the corresponding training errors (see Tables 5.3 and 5.4). A similar but less pronounced trend can also be seen for the structure factor model, although this should be interpreted with caution given the limited hyperparameter exploration.

A particularly decisive difference emerges in the maximum force errors. Across both materials, the VASP-MLFF models exhibit substantially larger force outliers, in some cases exceeding the structure factor maxima by several hundred meV Å^{-1}. Such extreme deviations can be critical in MD simulations, where single strongly mispredicted forces can lead to instability and unphysical trajectories. The consistently smaller maximum errors of the structure factor models therefore indicate enhanced robustness and improved stability for dynamical applications.

Table 5.3: Energy (meV/atom) prediction RMSE for NaCl. Relative errors with respect to the standard deviation of the reference data are given in parentheses.

		Structure Factor		VASP-MLFF	
		Training	Test	Training	Test
2-body	MAE	0.112 (0.4%)	0.115 (0.4%)	0.269 (0.9%)	0.274 (0.9%)
	RMSE	0.168 (0.6%)	0.178 (0.6%)	0.368 (1.3%)	0.384 (1.3%)
	MAX	0.852 (2.9%)	0.778 (2.6%)	2.38 (8.0%)	1.76 (6.0%)
3-body	MAE	0.083 (0.3%)	0.091 (0.3%)	0.103 (0.4%)	0.184 (0.6%)
	RMSE	0.123 (0.4%)	0.151 (0.5%)	0.145 (0.5%)	0.251 (0.9%)
	MAX	0.667 (2.3%)	1.19 (4.0%)	0.781 (2.6%)	1.07 (3.6%)

Table 5.4: Force (meV Å^{-1}) prediction RMSE for NaCl. Relative errors with respect to the standard deviation of the reference data are given in parentheses.

		Structure Factor		VASP-MLFF	
		Training	Test	Training	Test
2-body	MAE	7.17 (2.3%)	7.20 (2.4%)	11.8 (3.9%)	11.7 (3.8%)
	RMSE	10.2 (3.3%)	10.5 (3.5%)	16.8 (5.5%)	16.8 (5.5%)
	MAX	130 (43%)	107 (35%)	447 (146%)	447 (146%)
3-body	MAE	5.84 (1.9%)	6.02 (2.0%)	6.66 (2.2%)	8.10 (2.6%)
	RMSE	8.18 (2.7%)	8.66 (2.9%)	9.58 (3.1%)	12.4 (4.0%)
	MAX	92.2 (30%)	89.3 (29%)	291 (95%)	419 (137%)

5.2 Parity Plots

Parity plots for the training set are shown in Figures 5.1–5.4. Overall, the models exhibit strong agreement with the reference data. In particular, the energy parity plots show near-ideal correspondence without noticeable systematic deviations.

For the force predictions, however, a systematic deviation becomes visible at large force magnitudes, where the model tends to slightly underestimate the absolute values. This effect is most clearly illustrated in Figure 5.1, which provides a zoomed-in view of the two-body structure factor model for Si. The same tendency is consistently observed across different body orders (cf. Figure 5.2 vs. 5.3) as well as across different materials (cf. Figure 5.2 vs. 5.4). Despite this behavior, the overall deviations remain moderate and do not indicate a fundamental limitation of the model.

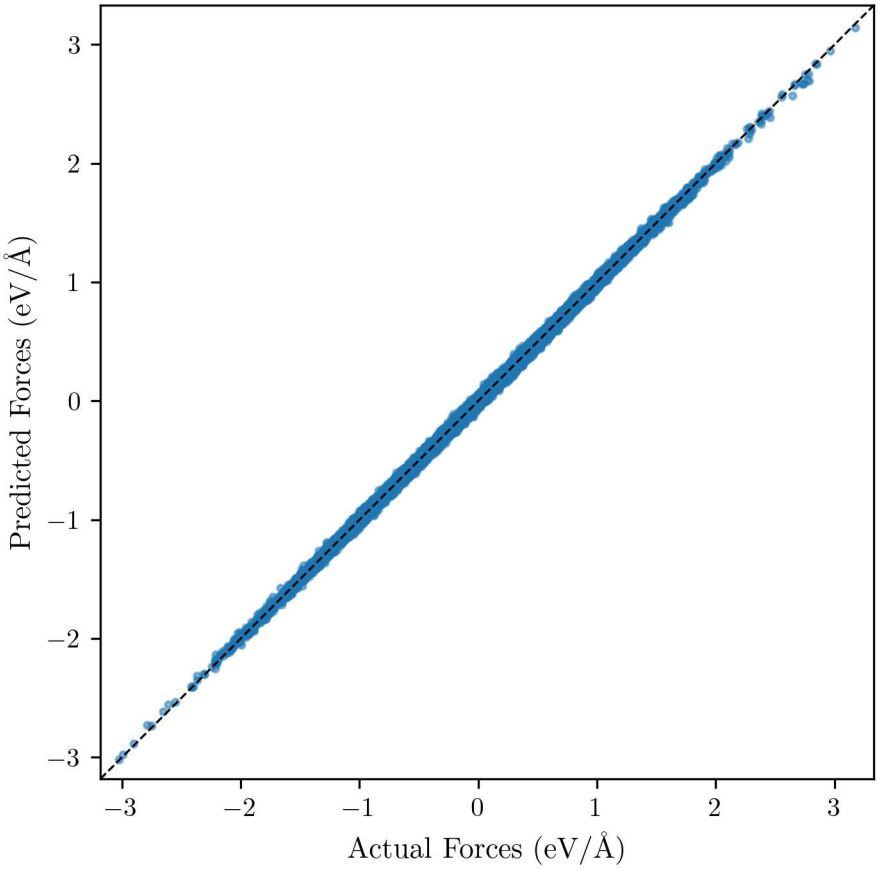

Figure 5.1: Parity plot (training set) for the two-body structure factor model in Si.

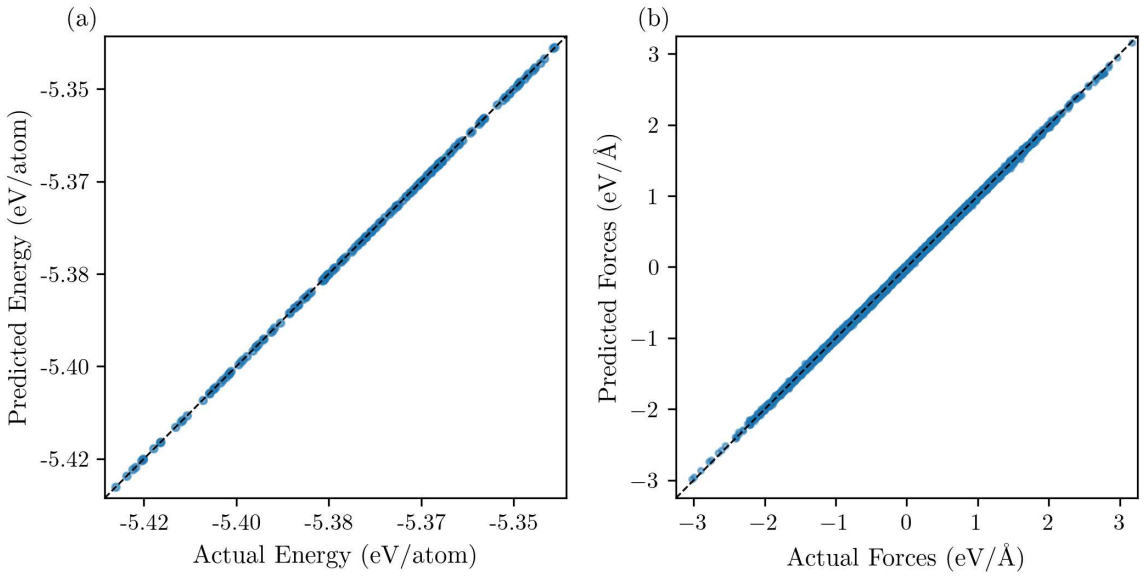

Figure 5.3: Parity plots (training set) of (a) energies and (b) forces for the three-body structure factor model in Si.

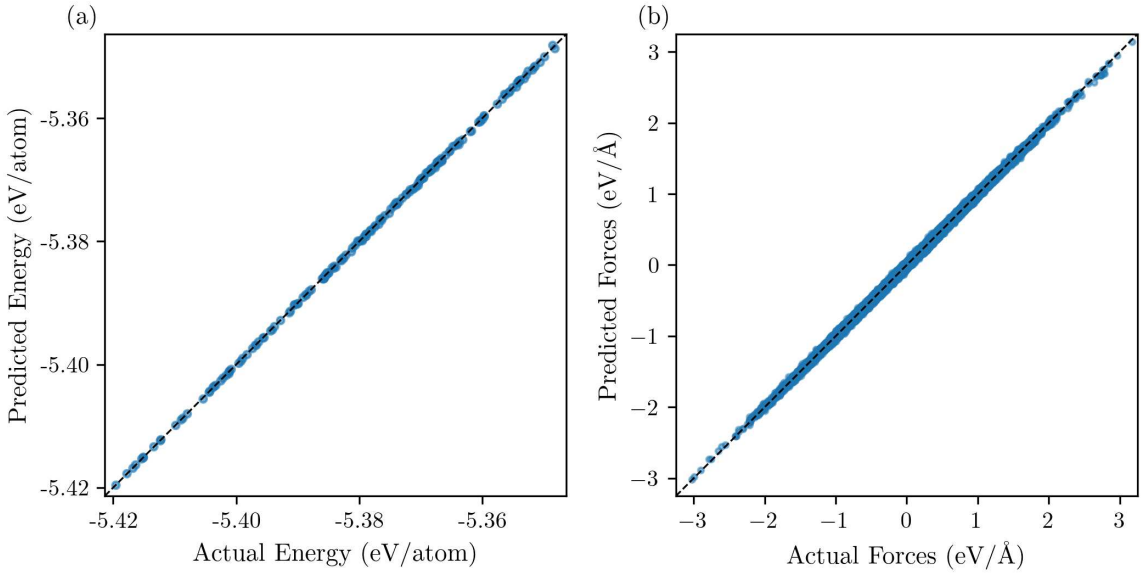

Figure 5.2: Parity plots (training set) of (a) energies and (b) forces for the two-body structure factor model in Si.

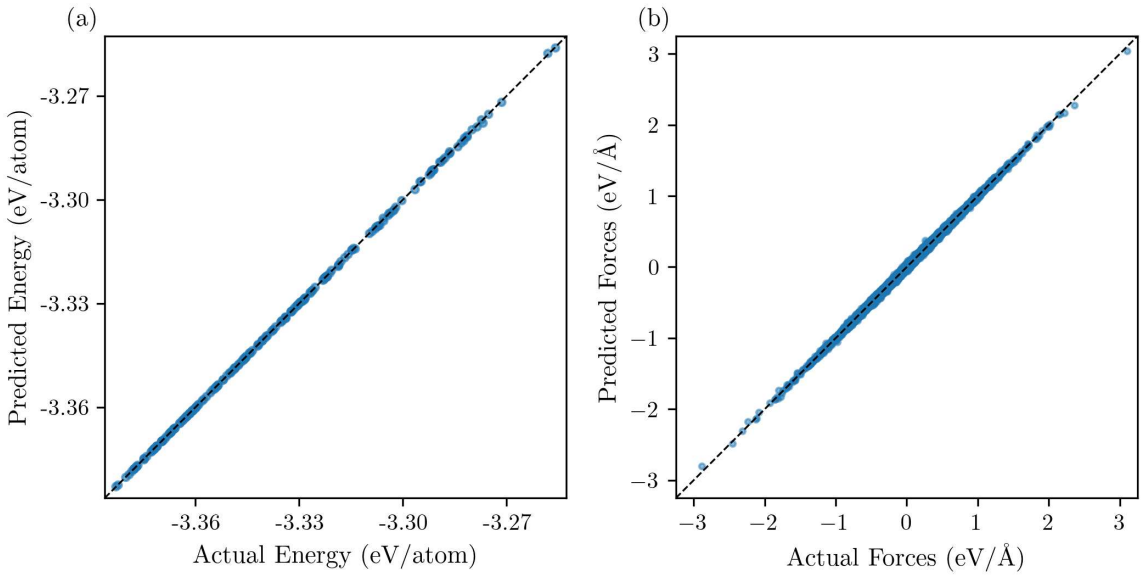

Figure 5.4: Parity plots (training set) of (a) energies and (b) forces for the two-body structure factor model in NaCl.

5.3 Learning Curves

The learning curves in Figures 5.5 and 5.6 illustrate the dependence of the prediction error on the number of *ab initio* training configurations for both structure factor and VASP-MLFF models, including their two- and three-body variants. All structure factor results were obtained using full symmetry-based data augmentation (cf. section 4.2), applying all 48 crystal point group operations to each configuration and thereby effectively enlarging the training dataset.

For Si, the VASP-based models exhibit a rapid decrease in error and reach a saturated regime already between 4 and 8 *ab initio* configurations. This comparatively fast convergence is consistent with the locality of the underlying descriptor, which generates a large number of training environments from each reference structure via its decomposition into local atomic environments. As a result, the model benefits from a high effective sample density even for small training sets.

In contrast, the structure factor-based models exhibit more gradual convergence and require more reference configurations to reach a comparable error plateau, achieved at around 32 configurations. This is consistent with the more global nature of the descriptor, where each configuration yields a single collective representation instead of multiple local environments. The two-body model shows slightly less regular error decay in the energy prediction, reflecting the increased sensitivity of energies.

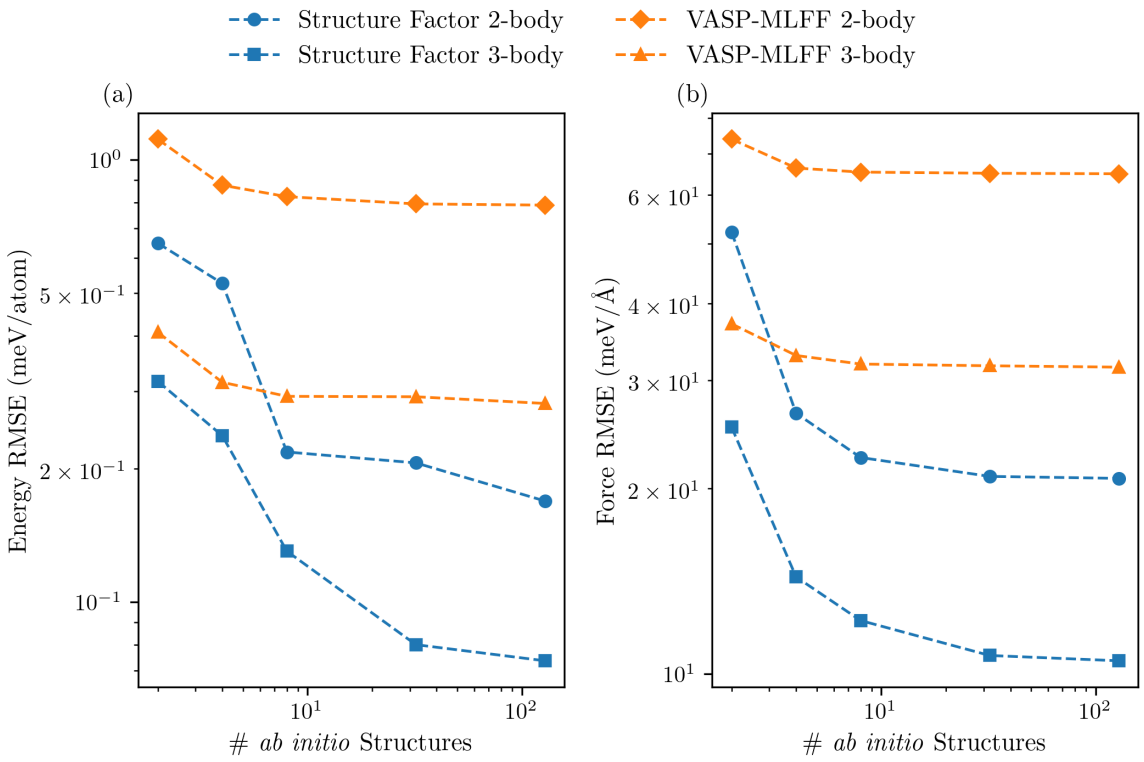

Figure 5.5: Learning curves of (a) energy and (b) force test RMSE versus number of *ab initio* training configurations for Structure Factor and VASP-MLFF models in Si.

For NaCl, a similar trend is observed; however, both model classes require more training configurations to reach saturation. The VASP-MLFF models converge earlier, between 8 and 16 configurations, whereas the structure factor models approach their asymptotic regime only at larger dataset sizes (32–128 configurations).

A comparison between both materials reveals consistent trends. NaCl yields lower errors but slower convergence, while Si converges faster but with slightly higher residual errors, reflecting a smoother ionic versus more information-dense covalent interaction landscape. Interestingly, the performance gap between the structure factor and the local VASP-MLFF reference is more pronounced for Si than for NaCl. At first glance, this appears counterintuitive, since NaCl, as an ionic system, is expected to exhibit more long-range electrostatic interactions, which should in principle favour global descriptors such as the structure factor. However, this effect is mitigated by the finite cutoff radii used in VASP-MLFF, extending up to 8.0 Å (radial) and 5.0 Å (angular). Given the relatively small supercells (64 atoms, with lattice vectors ≈ 11 Å), relevant correlations are already captured by the local descriptors. Consequently, the expected advantage of the global structure factor representation in describing long-range effects is not fully realised in this setup.

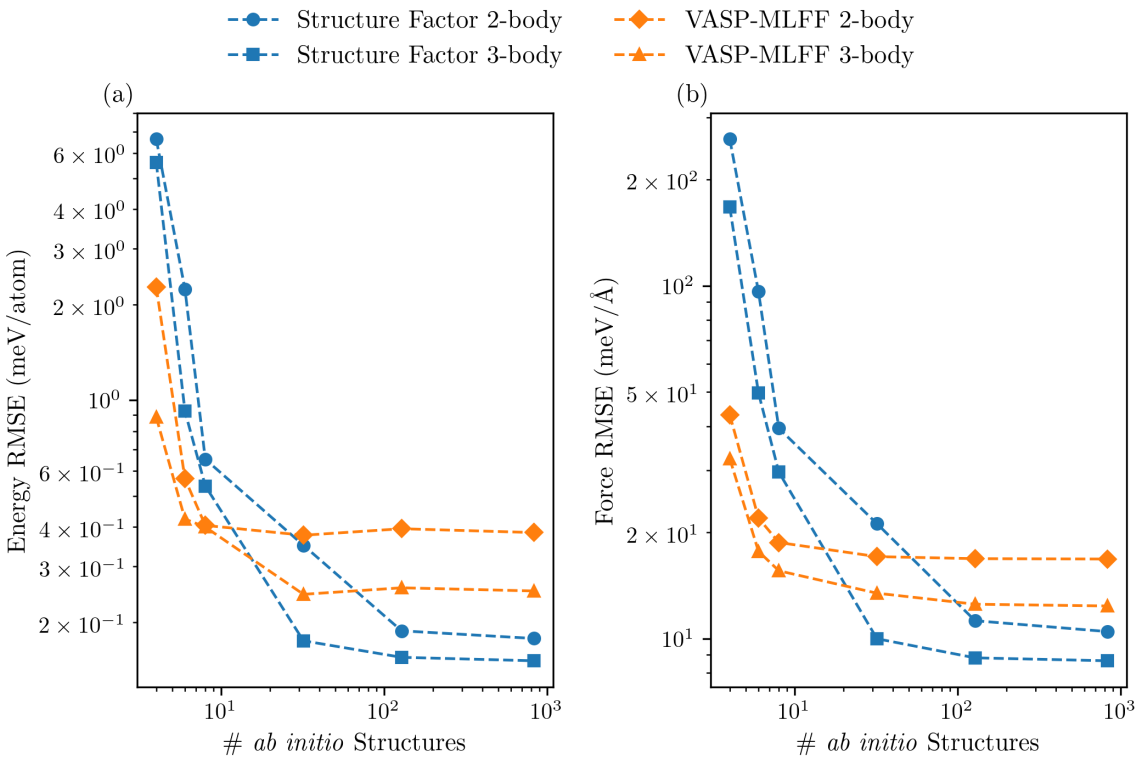

Figure 5.6: Learning curves of (a) energy and (b) force test RMSE versus number of *ab initio* training configurations for Structure Factor and VASP-MLFF models in NaCl.

This suggests that the observed comparison is partially limited by finite-size effects, and that larger supercells would be required to fully resolve differences in the treatment of long-range interactions between the local VASP-MLFF and the global structure factor descriptor approach.

5.4 Comparison with Neural Network Model

As an additional benchmark, the MACE (Message Passing Atomic Cluster Expansion) framework is considered. Unlike the explicitly constructed local basis functions used in VASP-MLFF or the structure factor models, MACE represents atomic environments using equivariant message-passing neural networks (MPNNs), enabling the adaptive learning of higher-order structural correlations beyond predefined basis expansions. The employed architecture consists of two interaction layers, a hidden feature dimension of 64 channels, and equivariance up to angular momentum order $L_{\max} = 1$. The atomic environment representation includes up to three-body order and is evaluated within a cut-off radius of 4.0 Å for Si and 6.0 Å for NaCl.

Table 5.5 compares the test RMSE of the best-performing models across all descriptor classes. MACE achieves the lowest energy and force errors in Si and NaCl due to the high flexibility of equivariant MPNNs, but represents a fundamentally different model class with substantially higher training and computational costs.

Interestingly, despite its significantly simpler functional form, the structure factor-based model achieves errors that are within the same order of magnitude as MACE for most quantities. The RMSE differs by approximately a factor of two. An exception is the energy prediction for NaCl, where MACE achieves substantially lower errors. As discussed above and in section 4.3, the three-body structure factor model was not fully hyperparameter-optimized in this case due to computational constraints and memory limitations. Therefore, the reported values do not represent the intrinsic lower bound of the descriptor, and further improvements are expected upon full optimization.

Table 5.5: Test RMSE comparison of the best-performing models for each descriptor.

	Energy (meV atom^{-1})		Force (meV Å^{-1})	
	Si	NaCl	Si	NaCl
Structure Factor (3-body)	0.074	0.151	10.5	8.66
VASP-MLFF (3-body)	0.287	0.251	31.5	12.4
MACE	0.033	0.019	6.42	4.29

The results in this chapter conclusively answer the central question of this thesis. The structure factor and its three-body extension constitute accurate and robust descriptors for MLFF construction across both Si and NaCl. Across all cases, they consistently outperform the VASP-MLFF reference models within the same regression framework, yielding lower RMSEs and, in particular, significantly reduced force outliers.

The learning curves highlight a trade-off between data efficiency and representation complexity: VASP-MLFF models converge rapidly in the low-data regime due to their local decomposition, whereas the structure factor requires more reference configurations but achieves superior accuracy at moderate dataset sizes. The parity plots indicate a weak systematic underestimation of large force magnitudes, which is consistently reduced, but not fully removed, by including three-body contributions.

Despite its much simpler and more constrained functional form compared to equivariant message-passing models, the structure factor reaches errors of the same order of magnitude as MACE for most quantities. This suggests that physically motivated reciprocal-space descriptors capture a large fraction of the relevant structural information without requiring highly flexible neural network architectures.

Building on these findings, the following chapter evaluates the broader significance of the structure factor as a descriptor for MLFFs and discusses its potential and limitations within the context of machine-learned interatomic potentials.

6 Conclusion and Outlook

This thesis set out to investigate whether the structure factor and its three-body extension constitute suitable descriptors for MLFFs. The results provide an affirmative answer. Across both a covalent system (Si) and an ionic system (NaCl), the structure factor representation yields high predictive accuracy for energies and forces, exhibits improved robustness with respect to force outliers, and consistently surpasses the VASP reference implementation within the same regression framework. The descriptor thus proves to be not only physically well motivated, but also highly expressive without requiring additional high-dimensional kernel mappings.

A central outcome of this study is the strong performance of the two-body structure factor representation. Even without explicit angular correlations, it already outperforms the three-body VASP reference. For both materials, it achieves energy errors on the order of 0.1–0.2 meV atom^{-1} and force errors of $\sim$10–20 meV Å^{-1}. The inclusion of three-body contributions further reduces these errors by approximately a factor of two, but at the cost of a substantially increased feature space, memory demand, and computational expense. From a practical perspective, the two-body model therefore represents an efficient and accurate baseline, while the three-body extension defines an upper performance bound whose cost–benefit ratio must be carefully evaluated.

An additional important result concerns data efficiency. The structure factor models require on the order of 10–30 reference configurations for Si and ~ 50 for NaCl to reach their asymptotic accuracy. In contrast, VASP-MLFF models converge already with $\lesssim 10$ configurations due to their decomposition into local atomic environments. At moderate dataset sizes, however, the structure factor models reach comparable or superior accuracy, highlighting the effectiveness of the global representation once sufficient coverage of the configurational space is achieved.

Several limitations of the present study should be noted. First, the structure factor formulation is restricted to fixed cell geometries, limiting its direct applicability to NpT conditions. As it is defined on a fixed reciprocal lattice, changes to the simulation cell modify the underlying feature space itself, preventing a straightforward transfer to variable-volume ensembles. This limitation could, in principle, be mitigated by incorporating explicit lattice vector information into the descriptor and employing a nonlinear regression scheme with exponential kernels, which were not investigated in the present work.

Second, the computational cost of the three-body extension becomes substantial, particularly for multi-component materials due to the rapid growth of feature dimensionality. However, the present implementation was not optimized for computational or memory efficiency, as the focus of this work was on assessing the fundamental potential of the descriptor. Consequently, part of the observed overhead arises from implementation aspects rather than intrinsic limitations.

A key finding of the parameter study is that achieving high accuracy requires a comparatively large number of reciprocal lattice vectors, resulting in feature spaces with several thousand two-body and up to tens of thousands of three-body features. A systematic analysis of feature relevance is therefore promising, as a reduced subset is likely sufficient to retain most of the predictive performance, enabling a significant reduction in computational cost.

Enforcing rotational invariance more directly represents a further improvement. While rotational symmetry is currently handled via data augmentation, it could in principle be incorporated into the descriptor by constructing rotationally invariant combinations of reciprocal-space features, e.g. by averaging over symmetry-equivalent $\mathbf{q}$-vectors. This would yield a more compact and physically consistent representation. From a physical perspective, a key limitation of this study is the relatively small supercell sizes employed, which prevent full exploitation of the structure factor's ability to capture long-range correlations. As a result, the performance gap to local VASP-MLFF descriptors remains smaller than expected, and extending the analysis to larger systems is an important next step.

In terms of applications, the structure factor descriptor is particularly well suited for simulations at fixed cell geometries, such as phonon calculations in periodic cells with fixed lattice parameters, where accurate and stable force predictions are essential. Similarly, heat transport simulations in the NVT or NVE ensemble rely on long MD trajectories, where the low occurrence of force outliers is crucial for numerical stability. More generally, the global nature of the structure factor limits its transferability across system sizes and unit or supercells, restricting its applicability for intermediate-scale, but long-time, analyses. At the same time, this property makes it well suited for preliminary studies of new materials or datasets with low data requirements and a physically transparent formulation, thereby occupying a complementary position within MLFF methodologies.

In summary, the structure factor is an accurate and robust MLFF descriptor with clear potential for further applications. Within the same regression framework, it consistently outperforms local VASP-MLFF models and reaches accuracy close to modern neural network potentials such as MACE, while remaining physically interpretable. This shows that reciprocal-space representations can efficiently capture structural correlations without requiring highly complex architectures and are promising, in particular, for phonon and heat transport simulations.

A Mathematical Background and Derivations

A.1 Singular Value Decomposition

Linear regression problems can be written in matrix form as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}\,, \tag{A.1}$$

where $\mathbf{\Phi}$ is the design matrix, $\mathbf{w}$ the vector of weights (fitting parameters), and $\mathbf{y}$ the vector of model outputs approximating the target values $\mathbf{t}$. The optimal weights minimize the least-squares loss

$$\min_{\mathbf{w}} \|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2\,, \tag{A.2}$$

leading to the normal equation

$$\mathbf{w}^* = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{t}\,. \tag{A.3}$$

Explicitly forming the inverse, however, can be numerically unstable if $\mathbf{\Phi}^\top\mathbf{\Phi}$ is ill-conditioned. A robust approach is provided by the singular value decomposition (SVD), which decomposes the design matrix into orthogonal rotations and diagonal scaling:

$$\mathbf{\Phi} = \mathbf{U}\mathbf{S}\mathbf{V}^\top\,, \tag{A.4}$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\mathbf{S} \in \mathbb{R}^{m\times n}$ is a rectangular diagonal matrix containing the non-negative singular values S_i on on the diagonal (i.e. entries S_{ii}), with all off-diagonal entries equal to zero. The singular values are unique up to ordering, while the orthogonal matrices are determined up to signs and

rotations in degenerate subspaces. In this basis, the least-squares solution decomposes into independent contributions along the singular vectors:

$$\mathbf{w}^* = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^\top \mathbf{t}\,. \tag{A.5}$$

Small singular values correspond to directions that are poorly constrained by the data and can lead to large, unstable weights. Regularization (such as ridge regression introduced in Eq. (2.12)) suppresses these contributions, effectively damping these modes and acting as a smoothness constraint on the model. In the SVD basis, the ridge regression solution can be expressed as

$$\mathbf{w}^*_{\text{ridge}} = \sum_i \frac{S_i}{S_i^2 + \lambda} (\mathbf{u}_i^\top \mathbf{t})\, \mathbf{v}_i\,, \tag{A.6}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. This shows that modes with $S_i^2 \ll \lambda$ are strongly suppressed, while modes with $S_i^2 \gg \lambda$ remain largely unaffected.

The effective rank provides a measure of how many modes contribute substantially to the regression, and is defined as

$$r_{\text{eff}} = \sum_i \frac{S_i^2}{S_i^2 + \lambda}\,. \tag{A.7}$$

In practice, a threshold $\tau \approx 0.5$ is often applied to count only active modes (see section 4.1):

$$r_{\text{eff}} = \sum_i \Theta\!\left(\frac{S_i^2}{S_i^2 + \lambda} - \tau\right)\,, \tag{A.8}$$

where $\Theta(\cdot)$ denotes the Heaviside step function, defined as $\Theta(x) = 1$ for $x > 0$ and $\Theta(x) = 0$ otherwise.

Alternative approaches for solving linear least-squares problems include Cholesky decomposition of $\mathbf{\Phi}^\top\mathbf{\Phi}$ and iterative optimization methods such as gradient descent. These methods are discussed extensively in the numerical linear algebra literature (see, e.g., [33]).

A.2 Derivatives of the Structure Factor

A.2.1 Two-Body Structure Factor

This section derives the derivative of the partial two-body structure factor introduced in Eq. (2.65). For convenience, the definition is repeated here:

$$S_{\alpha\beta}(\mathbf{q}) = \frac{1}{\sqrt{N_\alpha N_\beta}} \sum_{j\in\alpha} \sum_{k\in\beta} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \,. \tag{A.9}$$

The derivative is taken with respect to the Cartesian component $r_{i,\mu}$ of the position vector of atom i. The derivative of the exponential term reads

$$\frac{\partial}{\partial r_{i,\mu}} e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} = \begin{cases} -iq_\mu \, e^{-i\mathbf{q}\cdot(\mathbf{r}_i - \mathbf{r}_k)} \,, & j = i \,, \\ \;\; iq_\mu \, e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_i)} \,, & k = i \,, \\ 0 \,, & \text{otherwise.} \end{cases} \tag{A.10}$$

Only terms in which atom i appears in the double sum contribute to the derivative. Substituting Eq. (A.10) into Eq. (A.9) therefore gives

$$\begin{aligned} \frac{\partial S_{\alpha\beta}(\mathbf{q})}{\partial r_{i,\mu}} = \frac{1}{\sqrt{N_\alpha N_\beta}} \Bigg[& \sum_{k\in\beta} (-iq_\mu) e^{-i\mathbf{q}\cdot(\mathbf{r}_i - \mathbf{r}_k)} \mathbf{1}_{\{i\in\alpha\}} \\ & + \sum_{j\in\alpha} (iq_\mu) e^{-i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_i)} \mathbf{1}_{\{i\in\beta\}} \Bigg] \,, \end{aligned} \tag{A.11}$$

where $\mathbf{1}_{\{i\in\alpha\}}$ denotes the indicator function

$$\mathbf{1}_{\{i\in\alpha\}} = \begin{cases} 1 \,, & \text{if atom } i \text{ belongs to species } \alpha \,, \\ 0 \,, & \text{otherwise.} \end{cases}$$

Rearranging the terms yields

$$\frac{\partial S_{\alpha\beta}(\mathbf{q})}{\partial r_{i,\mu}} = \frac{iq_\mu}{\sqrt{N_\alpha N_\beta}} \left[\mathbf{1}_{\{i\in\alpha\}} \sum_{k\in\beta} e^{\,i\mathbf{q}\cdot(\mathbf{r}_i-\mathbf{r}_k)} - \mathbf{1}_{\{i\in\beta\}} \sum_{j\in\alpha} e^{-i\mathbf{q}\cdot(\mathbf{r}_i-\mathbf{r}_j)} \right], \tag{A.12}$$

which corresponds to the expression used in Eq. (2.69).

A.2.2 Three-Body Structure Factor

The derivative of the three-body structure factor follows the same principle as for the two-body case, although the algebra becomes slightly more involved. Starting from the partial three-body structure factor introduced in Eq. (4.16):

$$S^{(3)}_{\alpha\beta\gamma}(\mathbf{q}_1,\mathbf{q}_2) = \frac{1}{\sqrt{N_\alpha N_\beta N_\gamma}} \sum_{j\in\alpha}\sum_{k\in\beta}\sum_{l\in\gamma} e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j-\mathbf{r}_l)}\, e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k-\mathbf{r}_l)}\,. \tag{A.13}$$

The derivative is taken with respect to the Cartesian component $r_{i,\mu}$ of the position vector of atom i. Writing the exponential product as

$$A_{jkl} = e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j-\mathbf{r}_l)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k-\mathbf{r}_l)}\,,$$

the derivative of a single term reads

$$\frac{\partial A_{jkl}}{\partial r_{i,\mu}} = \begin{cases} iq_{1,\mu}\, e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_i-\mathbf{r}_l)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k-\mathbf{r}_l)}\,, & j=i\,, \\ iq_{2,\mu}\, e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j-\mathbf{r}_l)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_i-\mathbf{r}_l)}\,, & k=i\,, \\ -i(q_{1,\mu}+q_{2,\mu})\, e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j-\mathbf{r}_i)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k-\mathbf{r}_i)}\,, & l=i\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{A.14}$$

Thus only terms in which atom i appears in the triple sum contribute to the derivative. Substituting Eq. (A.14) into Eq. (A.13) yields

$$\frac{\partial S^{(3)}_{\alpha\beta\gamma}(\mathbf{q}_1,\mathbf{q}_2)}{\partial r_{i,\mu}} = \frac{1}{\sqrt{N_\alpha N_\beta N_\gamma}}\Bigg[\mathbf{1}_{\{i\in\alpha\}} i q_{1,\mu} \sum_{k\in\beta}\sum_{l\in\gamma} e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_i-\mathbf{r}_l)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k-\mathbf{r}_l)} + \mathbf{1}_{\{i\in\beta\}} i q_{2,\mu} \sum_{j\in\alpha}\sum_{l\in\gamma} e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j-\mathbf{r}_l)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_i-\mathbf{r}_l)} - \mathbf{1}_{\{i\in\gamma\}} i(q_{1,\mu}+q_{2,\mu}) \sum_{j\in\alpha}\sum_{k\in\beta} e^{\,i\mathbf{q}_1\cdot(\mathbf{r}_j-\mathbf{r}_i)} e^{\,i\mathbf{q}_2\cdot(\mathbf{r}_k-\mathbf{r}_i)}\Bigg], \tag{A.15}$$

where $\mathbf{1}_{\{i\in\alpha\}}$ again denotes the indicator function. Up to the normalization constant, this expression corresponds to the result given in Eq. (4.20).

A.3 Reciprocal-Space Expansion of the Structure Factor

This appendix derives Eq. (4.1). At finite temperature the instantaneous atomic positions can be written as $\mathbf{r}_j + \mathbf{u}_j$, where $\mathbf{r}_j$ are equilibrium lattice positions and $\mathbf{u}_j$ denote thermal displacements. The corresponding structure factor reads

$$S(\mathbf{q}) = \sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j+\mathbf{u}_j-\mathbf{r}_k-\mathbf{u}_k)}\,. \tag{A.16}$$

The corresponding zero-temperature reference is

$$S_0(\mathbf{q}) = \sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)}\,. \tag{A.17}$$

Introducing the displacement difference $\Delta\mathbf{u}_{jk} = \mathbf{u}_j - \mathbf{u}_k$, the structure factor can be written as $S(\mathbf{q}) = \sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)} e^{\,i\mathbf{q}\cdot\Delta\mathbf{u}_{jk}}$. Hence

$$S(\mathbf{q}) - S_0(\mathbf{q}) = \sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)}\Big(e^{\,i\mathbf{q}\cdot\Delta\mathbf{u}_{jk}} - 1\Big)\,. \tag{A.18}$$

Analogously,

$$S(\mathbf{q}+\mathbf{G}) - S_0(\mathbf{q}+\mathbf{G}) = \sum_{j,k} e^{\,i(\mathbf{q}+\mathbf{G})\cdot(\mathbf{r}_j-\mathbf{r}_k)} \Big(e^{\,i(\mathbf{q}+\mathbf{G})\cdot\Delta\mathbf{u}_{jk}} - 1 \Big) \,. \tag{A.19}$$

For equilibrium lattice vectors $\mathbf{r}_j - \mathbf{r}_k$ and reciprocal lattice vectors $\mathbf{G}$, the identity

$$e^{\,i\mathbf{G}\cdot(\mathbf{r}_j-\mathbf{r}_k)} = 1 \tag{A.20}$$

holds. Consequently,

$$e^{\,i(\mathbf{q}+\mathbf{G})\cdot(\mathbf{r}_j-\mathbf{r}_k)} = e^{\,i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)} \,. \tag{A.21}$$

Using this relation yields

$$\begin{aligned} &\big(S(\mathbf{q}) - S_0(\mathbf{q})\big) - \big(S(\mathbf{q}+\mathbf{G}) - S_0(\mathbf{q}+\mathbf{G})\big) \\ &= \sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j-\mathbf{r}_k)} \Big[e^{\,i\mathbf{q}\cdot\Delta\mathbf{u}_{jk}} - e^{\,i(\mathbf{q}+\mathbf{G})\cdot\Delta\mathbf{u}_{jk}} \Big] \,. \end{aligned} \tag{A.22}$$

Assuming small thermal displacements, both exponentials can be expanded to second order,

$$e^{\,i\mathbf{q}\cdot\Delta\mathbf{u}} = 1 + i\,\mathbf{q}\cdot\Delta\mathbf{u} - \frac{1}{2}(\mathbf{q}\cdot\Delta\mathbf{u})^2 + \mathcal{O}(\Delta u^3), \tag{A.23}$$

$$e^{\,i(\mathbf{q}+\mathbf{G})\cdot\Delta\mathbf{u}} = 1 + i(\mathbf{q}+\mathbf{G})\cdot\Delta\mathbf{u} - \frac{1}{2}\big((\mathbf{q}+\mathbf{G})\cdot\Delta\mathbf{u}\big)^2 + \mathcal{O}(\Delta u^3) \,. \tag{A.24}$$

Subtracting the two expressions yields

$$e^{\,i\mathbf{q}\cdot\Delta\mathbf{u}} - e^{\,i(\mathbf{q}+\mathbf{G})\cdot\Delta\mathbf{u}} = -i\,\mathbf{G}\cdot\Delta\mathbf{u} + (\mathbf{q}\cdot\mathbf{G})(\Delta\mathbf{u})^2 + \frac{G^2}{2}(\Delta\mathbf{u})^2 + \mathcal{O}(\Delta u^3) \,. \tag{A.25}$$

Insertion into Eq. (A.22) leads to

$$\begin{aligned}&\big(S(\mathbf{q}) - S_0(\mathbf{q})\big) - \big(S(\mathbf{q}+\mathbf{G}) - S_0(\mathbf{q}+\mathbf{G})\big) \\ &\quad\approx \sum_{j,k} e^{\,i\mathbf{q}\cdot(\mathbf{r}_j - \mathbf{r}_k)} \Big[-i\,\mathbf{G}\cdot\Delta\mathbf{u}_{jk} + (\mathbf{q}\cdot\mathbf{G})\,(\Delta\mathbf{u}_{jk})^2 + \tfrac{G^2}{2}\,(\Delta\mathbf{u}_{jk})^2 + \mathcal{O}\big((\Delta\mathbf{u}_{jk})^3\big)\Big]\,,\end{aligned} \tag{A.26}$$

which corresponds to Eq. (4.1).

A.4 Error Metrics

To quantify the accuracy of a regression model (as in chapter 5), we compare the predicted values $y_i = f(\boldsymbol{\varphi}(\mathbf{x}_i); \mathbf{w})$ with the reference targets t_i. A general class of error measures can be defined as the p-norm error:

$$\mathrm{error}_p = \left(\frac{1}{N} \sum_{i=1}^{N} |y_i - t_i|^p \right)^{1/p}, \tag{A.27}$$

where N is the number of data points. Widely used metrics arise as special cases: the mean absolute error (MAE) for $p = 1$, the root mean square error (RMSE) for $p = 2$, and the maximum error (MAX) for $p \to \infty$, i.e., $\mathrm{MAX} = \max_i |y_i - t_i|$. Here, MAE emphasizes the average absolute deviation, RMSE penalizes larger deviations more strongly, and MAX identifies the largest single error.

For force predictions, the error metrics are evaluated component-wise. That is, each Cartesian component of the force vector is treated as an individual data point. The error measures defined in Eq. (A.27) are then applied to the set of all force components. This definition differs from an alternative approach based on force magnitudes, where the norm of the vector error is considered per atom.

B Additional Results

B.1 Additional Kernel Results

This appendix summarizes additional results of the kernel parameter study presented in section 4.4. Variations of the kernel offset parameter c and the regularization parameter λ show no significant impact on model performance. The corresponding results are reported in Tables B.1 and B.2, respectively.

Table B.1: Influence of the kernel offset parameter c for $\chi = 2$, $h_{\max} = 15$, and $\lambda = 10^{-9}$ Si.

c	$\mathrm{RMSE}_{\mathrm{E}}^{\mathrm{train}}$ ($\mathrm{meV\,atom^{-1}}$)	$\mathrm{RMSE}_{\mathrm{E}}^{\mathrm{test}}$ ($\mathrm{meV\,atom^{-1}}$)	$\mathrm{RMSE}_{\mathrm{F}}^{\mathrm{train}}$ ($\mathrm{meV\,Å^{-1}}$)	$\mathrm{RMSE}_{\mathrm{F}}^{\mathrm{test}}$ ($\mathrm{meV\,Å^{-1}}$)
0	0.273	0.382	39.6	39.3
0.1	0.273	0.380	39.6	39.3
1	0.273	0.380	39.6	39.3
10	0.272	0.381	39.6	39.3
1000	0.272	0.381	39.6	39.3

Table B.2: Influence of the regularization parameter λ for $\chi = 2$, $c = 0$, and $h_{\max} = 15$ for Si.

λ	$\mathrm{RMSE}_{\mathrm{E}}^{\mathrm{train}}$ ($\mathrm{meV\,atom^{-1}}$)	$\mathrm{RMSE}_{\mathrm{E}}^{\mathrm{test}}$ ($\mathrm{meV\,atom^{-1}}$)	$\mathrm{RMSE}_{\mathrm{F}}^{\mathrm{train}}$ ($\mathrm{meV\,Å^{-1}}$)	$\mathrm{RMSE}_{\mathrm{F}}^{\mathrm{test}}$ ($\mathrm{meV\,Å^{-1}}$)
10^{-12}	0.273	0.380	39.6	39.3
10^{-9}	0.273	0.382	39.6	39.3
10^{-5}	0.273	0.380	39.6	39.3
10^{-2}	0.273	0.380	39.6	39.3
10^{2}	0.273	0.380	39.6	39.3

Bibliography

[1] Pierre C. Hohenberg and Walter Kohn. "Inhomogeneous Electron Gas". In: *Phys. Rev.* 136 (3B Nov. 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864.

[2] Walter Kohn and L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". In: *Phys. Rev.* 140 (4A Nov. 1965), A1133–A1138. DOI: 10.1103/PhysRev.140.A1133.

[3] John Edward Jones. "On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 106.738 (1924), pp. 441–462. DOI: 10.1098/rspa.1924.0081.

[4] John Edward Lennard-Jones. "Cohesion". In: *Proceedings of the Physical Society* 43.5 (Sept. 1931), pp. 461–482. DOI: 10.1088/0959-5309/43/5/301.

[5] Oliver T. Unke et al. "Machine Learning Force Fields". In: *Chemical Reviews* 121.16 (2021), pp. 10142–10186. DOI: 10.1021/acs.chemrev.0c01111.

[6] Jörg Behler and Michele Parrinello. "Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces". In: *Phys. Rev. Lett.* 98 (14 Apr. 2007), p. 146401. DOI: 10.1103/PhysRevLett.98.146401.

[7] Jörg Behler. "Perspective: Machine learning potentials for atomistic simulations". In: *The Journal of Chemical Physics* 145.17 (Nov. 2016), p. 170901. DOI: 10.1063/1.4966192.

[8] Jörg Behler. "Atom-centered symmetry functions for constructing high-dimensional neural network potentials". In: *The Journal of Chemical Physics* 134.7 (Feb. 2011), p. 074106. DOI: 10.1063/1.3553717.

[9] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. "Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons". In: *Phys. Rev. Lett.* 104 (13 Apr. 2010), p. 136403. DOI: 10.1103/PhysRevLett.104.136403.

[10] Soheil A. Dianat and Raghuveer M. Rao. "Fast algorithms for phase and magnitude reconstruction from bispectra". In: *Optical Engineering* 29.5 (1990), pp. 504–512. DOI: 10.1117/12.55619.

[11] Albert P. Bartók, Risi Kondor, and Gábor Csányi. "On representing chemical environments". In: *Phys. Rev. B* 87 (18 May 2013), p. 184115. DOI: 10.1103/PhysRevB.87.184115.

[12] Georg Kresse and Jürgen Hafner. "Ab initio molecular dynamics for liquid metals". In: *Phys. Rev. B* 47 (1 Jan. 1993), pp. 558–561. DOI: 10.1103/PhysRevB.47.558.

[13] Georg Kresse and Jürgen Furthmüller. "Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set". In: *Computational Materials Science* 6.1 (1996), pp. 15–50. DOI: https://doi.org/10.1016/0927-0256(96)00008-0.

[14] Georg Kresse and Jürgen Furthmüller. "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set". In: *Phys. Rev. B* 54 (16 Oct. 1996), pp. 11169–11186. DOI: 10.1103/PhysRevB.54.11169.

[15] Ryosuke Jinnouchi, Ferenc Karsai, and Georg Kresse. "On-the-fly machine learning force field generation: Application to melting points". In: *Phys. Rev. B* 100 (1 July 2019), p. 014105. DOI: 10.1103/PhysRevB.100.014105.

[16] Ryosuke Jinnouchi, Jonathan Lahnsteiner, Ferenc Karsai, Georg Kresse, and Menno Bokdam. "Phase Transitions of Hybrid Perovskites Simulated by Machine-Learning Force Fields Trained on the Fly with Bayesian Inference". In: *Phys. Rev. Lett.* 122 (22 June 2019), p. 225701. DOI: 10.1103/PhysRevLett.122.225701.

[17] Ryosuke Jinnouchi, Ferenc Karsai, Carla Verdi, Ryoji Asahi, and Georg Kresse. "Descriptors representing two- and three-body atomic distributions and their effects on the accuracy of machine-learned inter-atomic potentials". In: *The Journal of Chemical Physics* 152.23 (June 2020), p. 234102. DOI: 10.1063/5.0009491.

[18] Ilyes Batatia, David Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. "MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields". In: (June 2022). DOI: 10.48550/arXiv.2206.07697.

[19] Ralf Drautz. "Atomic cluster expansion for accurate and transferable interatomic potentials". In: *Phys. Rev. B* 99 (1 Jan. 2019), p. 014104. DOI: 10.1103/PhysRevB.99.014104.

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

[21] Bingqing Cheng. "Latent Ewald summation for machine learning of long-range interactions". In: *npj Computational Materials* 11.1 (2025). Published: 26 March 2025, p. 80. DOI: 10.1038/s41524-025-01577-7.

[22] Peichen Zhong, Dongjin Kim, Daniel S. King, and Bingqing Cheng. "Machine learning interatomic potential can infer electrical response". In: *npj Computational Materials* 11.1 (2025). Published: 22 December 2025, p. 384. DOI: 10.1038/s41524-025-01911-z.

[23] Egor Rumiantsev, Marcel Langer, Tulga-Erdene Sodjargal, Michele Ceriotti, and Philip Loche. "Learning Long-Range Representations with Equivariant Messages". In: (July 2025). DOI: 10.48550/arXiv.2507.19382.

[24] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. "Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning". In: *Phys. Rev. Lett.* 108 (5 Jan. 2012), p. 058301. DOI: 10.1103/PhysRevLett.108.058301.

[25] Katja Hansen et al. "Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space". In: *The Journal of Physical Chemistry Letters* 6.12 (2015), pp. 2326–2331. DOI: 10.1021/acs.jpclett.5b00831.

[26] Felix A. Faber, Anders S. Christensen, Bing Huang, and O. Anatole von Lilienfeld. "Alchemical and structural distribution based representation for universal quantum machine learning". In: *The Journal of Chemical Physics* 148.24 (Mar. 2018), p. 241717. DOI: 10.1063/1.5020710.

[27] Anders S. Christensen, Lars A. Bratholm, Felix A. Faber, and O. Anatole von Lilienfeld. "FCHL revisited: Faster and more accurate quantum machine learning". In: *The Journal of Chemical Physics* 152.4 (Jan. 2020), p. 044107. DOI: 10.1063/1.5126701.

[28] Stefan Chmiela et al. "Machine learning of accurate energy-conserving molecular force fields". In: *Science Advances* 3.5 (2017), e1603015. DOI: 10.1126/sciadv.1603015.

[29] Stefan Chmiela et al. "Accurate global machine learning force fields for molecules with hundreds of atoms". In: *Science Advances* 9.2 (2023), eadf0873. DOI: 10.1126/sciadv.adf0873.

[30] Huziel E. Sauceda et al. "BIGDML: A general, scalable, symmetry-adapted, strict energy-conserving machine-learning framework for molecular dynamics simulations". In: *Nature Communications* 13 (2022), p. 3733.

[31] Carolin Faller, Merzuk Kaltak, and Georg Kresse. "Density-based long-range electrostatic descriptors for machine learning force fields". In: *The Journal of Chemical Physics* 161.21 (Dec. 2024), p. 214701. DOI: 10.1063/5.0245615.

[32] C.M. Bishop. *Pattern recognition and machine learning*. Vol. 4. Springer New York, 2006.

[33] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 3rd. Baltimore, MD: Johns Hopkins University Press, 1996.

[34] Arthur E. Hoerl and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1 (1970), pp. 55–67.

[35] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of Ill-Posed Problems*. Translation of the Russian edition by John Wiley & Sons. Washington, D.C.: V. H. Winston & Sons, 1977.

[36] Calyampudi Radhakrishna Rao and Satyendra Kumar Mitra. *Generalized Inverse of Matrices and Its Applications*. New York: Wiley, 1971.

[37] Shūichi Nosé. "A molecular dynamics method for simulations in the canonical ensemble". In: *Molecular Physics* 52.2 (1984), pp. 255–268. DOI: 10.1080/00268978400101201.

[38] William G. Hoover. "Canonical dynamics: Equilibrium phase-space distributions". In: *Phys. Rev. A* 31 (3 Mar. 1985), pp. 1695–1697. DOI: 10.1103/PhysRevA.31.1695.

[39] Peter E. Blöchl. "Projector augmented-wave method". In: *Phys. Rev. B* 50 (24 Dec. 1994), pp. 17953–17979. DOI: 10.1103/PhysRevB.50.17953.

[40] Georg Kresse and Dominique Joubert. "From ultrasoft pseudopotentials to the projector augmented-wave method". In: *Phys. Rev. B* 59 (3 Jan. 1999), pp. 1758–1775. DOI: 10.1103/PhysRevB.59.1758.