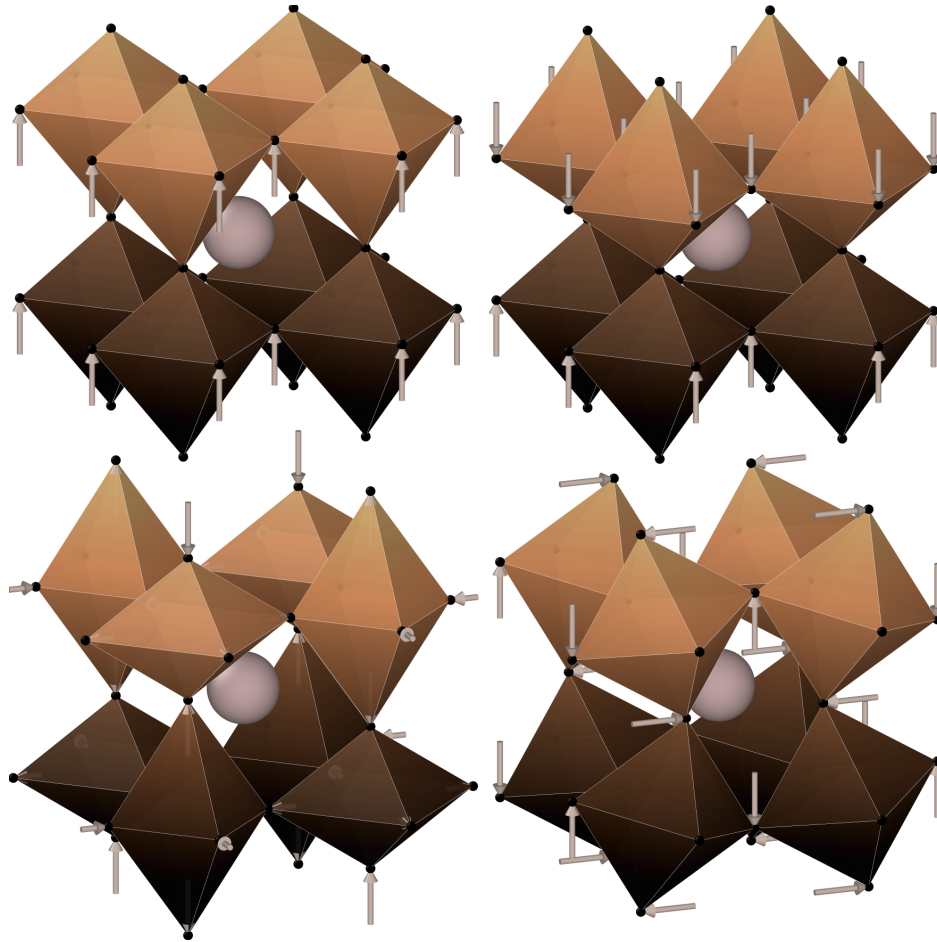




CHALMERS
UNIVERSITY OF TECHNOLOGY



A Foundational Machine-Learned Interatomic Potential for Perovskites

Trained on a Database of Generated DFT Data

Master's thesis in Physics

JACOB OLSSON

DEPARTMENT OF PHYSICS AND ASTRONOMY

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2026

www.chalmers.se

MASTER'S THESIS 2026

A Foundational Machine-Learned Interatomic Potential for Perovskites

Trained on a Database of Generated DFT Data

JACOB OLSSON



Department of Physics and Astronomy
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

A Foundational Machine-Learned Interatomic Potential for Perovskites
Trained on a Database of Generated DFT Data
JACOB OLSSON

© JACOB OLSSON, 2026.

Supervisor: Erik Fransson, Department of Physics and Astronomy
Examiner: Paul Erhart, Department of Physics and Astronomy

Master's thesis 2026
Department of Physics and Astronomy
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Four possible high-symmetry X-anion displacements in a perovskite.

Typeset in L^AT_EX
Gothenburg, Sweden 2026

A Foundational Machine-Learned Interatomic Potential for Perovskites
Trained on a Database of Generated DFT Data
JACOB OLSSON
Department of Physics and Astronomy
Chalmers University of Technology

Abstract

Perovskite materials cover a wide range of chemical compositions, all of which crystallize into a common structure. This structure admits a range of distortions and derivatives, from which a multitude of physical properties and applications arise. This project presents a foundational machine-learned model trained on 59 perovskite compositions. The ambition is that this model will facilitate perovskite research by allowing accurate atomistic simulations multiple orders of magnitude faster than density functional theory (DFT). The model uses the MACE architecture and the training data is generated via DFT computations. The training data includes relaxed structures at 0 K as well as finite temperature structures generated by running molecular dynamics (MD) simulations. For the total energy, the model achieves a root mean square error on the order of 0.1 meV to 1 meV/atom, which is small compared to the room temperature thermal energy scale of order 10 meV/atom. The model is tested by running MD simulations and determining the structural phase transitions of BaTiO_3 , BaZrO_3 , BaZrS_3 , CsGeBr_3 , CsPbI_3 , NaNbO_3 and SrTiO_3 . The model is generally able to determine the structural phases within about 100 K of previously presented experimental and computational results. One exception is NaNbO_3 , where the model is unable to recognize the multitude of fast structural transitions. It is, however, able to recognize that complex dynamics take place, directing further study to this material. Possible improvements to the model include ensuring maximally diverse training sets, training on more chemical compositions and fine-tuning more universal models with the perovskite data.

Keywords: perovskite, foundational model, machine-learning, DFT, MD, MACE, phase transitions.

Thank you to my friends and family for making the late nights and early mornings over the last five years bearable, and thank you to Paul Erhart and Julia Wiktor for believing in me and giving me the opportunity to work at the division of Condensed Matter and Materials Theory division at Chalmers. And a special thank you to Erik Fransson and Paul Erhart for all their assistance, all their ideas, and for making this project possible.

Jacob Olsson, Gothenburg, June 2026

Contents

1	Introduction	1
1.1	Background	1
1.1.1	The perovskite structure	2
1.1.2	The perovskite applications	3
1.2	Purpose and limitations	3
1.2.1	LLM disclaimer	4
2	Theory	5
2.1	The lattice	5
2.2	Electronic and nuclear dynamics	5
2.2.1	Quantization of lattice vibrations; phonons	6
2.2.2	Density functional theory	7
2.2.3	Molecular dynamics	8
2.3	Symmetry	9
2.3.1	Mathematical formalism of symmetry	9
2.3.2	Symmetries of the lattice	11
2.3.3	Symmetries of phonons and phase transitions	11
2.3.4	The perovskite structure, again	12
2.4	The MACE architecture	15
2.4.1	Message-passing neural network	15
2.4.2	Equivariant message passing neural network (MPNN)	15
2.4.3	atomic cluster expansion (ACE) message-passing functions	15
2.4.4	Accuracy and computational cost	17
3	Methods	19
3.1	Code design and structure generation	20
3.2	DFT calculations	20
3.2.1	Choice of functional	20
3.2.2	Single-point and relaxation	21
3.3	MD heating	22
3.4	Datasets and model training	22
4	Results and Discussion	25
4.1	DFT relaxation and MD simulations	25
4.2	Foundational models	27
4.2.1	Model 64x300K	31
4.2.2	Model 64x1000K	32
4.2.3	Model 128x1000K	33

4.3	Phases	34
4.3.1	BaTiO ₃	34
4.3.2	BaZrO ₃	35
4.3.3	BaZrS ₃	36
4.3.4	CsGeBr ₃	37
4.3.5	CsPbI ₃	37
4.3.6	SrTiO ₃	39
4.3.7	NaNbO ₃	39
4.4	Improvements	41
4.5	Future work	41
5	Conclusion	43
	Bibliography	45
A	Settings and parameters	I

1

Introduction

This project concerns perovskites, a term covering a wide range of materials with the commonality that they all crystallize into the perovskite structure. This structure, albeit rather simple, admits a wide range of distortions and derivations which offer a plethora of physical properties and applications [1, 2, 3, 4, 5]. The study of perovskites has become extremely popular, with much effort directed at finding chemical compositions with desired properties [1].

Computational problem-solving offers a unique alternative to analytic and experimental approaches. It is able to solve difficult many-body problems with high accuracy [6], where analytical solutions are impossible without heavy approximations. It also offers advantages compared to experimental research, as it does not require the synthesis of the materials in question.

Because of the many-body nature of condensed matter physics, computational methods offer a powerful tool to effectively study materials. This project will utilize and iterate upon these methods to create a foundational machine-learned model for perovskites. This model will serve as the starting point for future perovskite research, allowing fast and accurate predictions which can guide efforts to the most promising materials.

In this introduction, some background will first be given about perovskites, what they are and why they are interesting. After this a more detailed purpose will be presented, as well as limitations. Finally, the methodology will be laid out, which also serves as an overview of the whole report.

1.1 Background

The mineral CaTiO_3 was discovered in the Ural mountains in 1839 by Gustav Rose, who named it perovskite after the Russian mineralogist L. A. Perovski [7]. The crystal structure of CaTiO_3 can be found in other materials of the form ABX_3 (or some variation), and the term perovskite is now used to denote any such structure. There are many chemical compositions with the perovskite structure, the primary ones found in nature being various oxides [1], such as CaTiO_3 or SrTiO_3 . It is also possible to substitute the oxygen for other elements in the chalcogen family, such as S and Se. Another group of perovskites are the halide perovskites, where the X-anion is taken from the halide family, such as CsGeBr_3 .

1.1.1 The perovskite structure

The perovskite structure can be described by four corner-sharing BX_6 octahedra around a central A-cation, where the highest symmetry version is cubic [1], and is shown in Figure 1.1. Lower symmetry versions are also possible, for example where the BX_6 octahedra are tilted or the B-cation is displaced from the centre of the octahedra [8]. We will call the cubic structure the parent structure, and consider the lower symmetry structures as distorted versions of it.

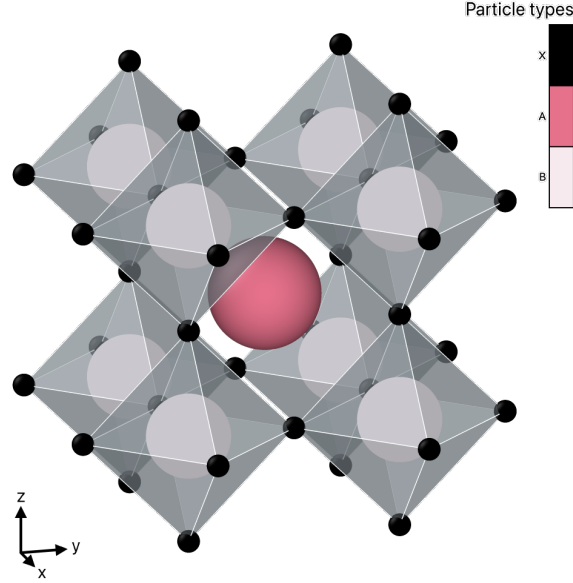


Figure 1.1: The highest symmetry perovskite structure, consisting of one A-cation surrounded by eight corner-sharing BX_6 octahedra.

There are several derivatives of this structure, two common examples being B-cation substitutions, resulting in structures like double perovskites $A_2B(I)B(II)X_6$, and organic-inorganic perovskites, where the A-cation has been replaced by a small organic molecule [1]. The distortions and derivations offer up a rich space of physical properties, and there is much ongoing research into finding functional perovskite materials.

The symmetry of the perovskite structure is often dependent on factors like temperature and pressure. As an example, take $BaTiO_3$, whose structure starts as cubic above 400 K and subsequently transitions to tetragonal, orthorhombic and rhombohedral as the temperature decreases [9]. The phases of this material will be determined as part of this project, and the impatient reader can jump ahead to the results chapter to examine them in detail. The structural changes are called structural phase transitions. Since the type of structure influences the material properties (the non-cubic phases of $BaTiO_3$ are ferroelectric, while the cubic phase is not [9]), finding materials that are functional at operating temperatures requires knowledge of the structural phases. In the next section, we will cover some applications, with special emphasis on the importance of the structural phases.

1.1.2 The perovskite applications

Perovskite solar cells have emerged as one of the most promising alternatives to standard Si and GeAs solutions [1], offering rivalling efficiencies [10] as well as cheaper and simpler production processes [4]. The halide perovskites excel in this application, and much research is dedicated to them [1]. One of the main challenges is stability. Current perovskite photovoltaic cells are prone to efficiency degradation under exposure to light and heat [4]. These problems need to be addressed (as it is very hard to use solar cells in the dark) before perovskite solar cells can be put into large-scale commercial use. Another issue is that many perovskite candidates contain Pb, which is toxic [4]. Finding alternative, non-toxic materials is therefore also a challenge that needs addressing. The search for more tolerant, non-toxic perovskites is ongoing, where organic-inorganic types show promise [4]. Finding materials with desired physical properties, which is influenced by the structural phase, is therefore crucial.

These same advantages also apply to perovskite-based light-emitting diodes, where again the halide perovskites shine (no pun intended). A mixture of CsGeBr₃ and CH₃NH₃ has been shown to reach an efficiency of 20% [5]. Some perovskites exhibit superconducting properties, such as MgCNi₃ below 8 K [2], and the perovskite-derivative structure Ba₂YCu₃O_{9- δ} ($\delta = 2.1$) at 93 K, making it a high-temperature superconductor [3]. Of course the temperature dependency of the structure is crucial here.

Hopefully, you are now convinced of the importance that perovskites play in functional materials, and how structural transitions are crucial to understanding them. It will come as no surprise that this project concerns itself with structural transitions, and we are now ready to present the purpose and goal of this project, as well as introduce some limitations.

1.2 Purpose and limitations

The goal is to train a foundational machine-learned interatomic potential for perovskites. The word foundational refers to the model being trained on a wide range of materials, and its future use as a source for generating perovskite data inexpensively. State-of-the-art calculations use DFT. They do, however, carry hefty computational costs, making large atom counts and heating simulations infeasible. In an attempt to reduce this cost, many machine-learning descriptors and architectures have been proposed. Examples relevant for this project include the ACE [11] descriptor, used in the MACE architecture [12], and the neuro-evolution potential (NEP) descriptor, used in the NEP architecture [13]. The foundational model will use the MACE architecture, as it in general produces more accurate results, albeit with a higher computational cost compared with the lightweight NEP architecture [12][14][13].

To train the model a dataset is required, which will be generated by DFT calculations. An auxiliary goal of this project is therefore the construction of a perovskite database, on which the model will be trained. This project is limited to regular perovskites (that is, no perovskite derivatives) and the generation of three common distortions, in-phase octahedral tilting, out-of-phase octahedral tilting and B-cation displacement. A total of 59 perovskites will be part of the training set, with 24 halide and 27 chalcogen perovskites

created from all the possible combination of elements presented in Table 1.1. Included are also eight common perovskites, which results in a total of 59 chemical compositions.

Table 1.1: The perovskites considered in this thesis. Included are the possible ion combinations of the first two rows, as well as 8 additional perovskites given in the third row.

	A-cations	B-cations	X-anions	Count
Halides	Cs, Rb	Ge, Pb, Sn	Br, Cl, F, I	24
Chalcogens	Ba, Ca, Sr	Hf, Ti, Zr,	O, S, Se	27
Common perovskites	KNbO ₃ , KTaO ₃ , LaAlO ₃ , LaMnO ₃ NaNbO ₃ , NaTaO ₃ , PbTiO ₃ , PbZrO ₃			8
				tot. = 59

This project can be divided into two major parts, data generation and model training. The data generation is performed using DFT calculations run on CPU computing clusters. This data is carefully organized into databases, and the lowest energy structural phase at 0 K is determined. The model is then trained on this database, evaluated and retrained until satisfactory accuracy is achieved. The final part of this project is to use the model to run MD simulations to determine the phase transitions of seven perovskites: BaTiO₃, BaZrO₃, BaZrS₃, CsGeBr₃, CsPbI₃, NaNbO₃, and SrTiO₃. This will then be compared with previous experimental and computational results to determine the validity of the model.

To appreciate the methods and results, some theory has to be presented first. This is done in chapter 2, and begins by outlining the basic properties of crystalline materials and their thermal vibrations. From this the phonon eigenvector equation is derived, DFT is outlined as well as the theory of MD. Afterwards, a mathematically heavy section about symmetry is presented, as this is the tool which will be used to study phonons and structural phase transitions. Finally, the MACE architecture is presented and explained, as well as why this choice is motivated for this project.

1.2.1 LLM disclaimer

Large language models have been used to assist with writing code, formatting L^AT_EX and providing insight into physical concepts. No text has been written by AI and all facts have been verified in other sources, all of which are referenced in this thesis.

2

Theory

This chapter outlines the theory needed to understand the work presented in this report, including the basics of solid state systems, the computational methods utilized in this project and the formalism behind structural phase transitions. All of this will then be applied to perovskites.

2.1 The lattice

A crystalline material can be represented by a lattice and a basis [15]. A lattice is a set of mathematical points that are defined by a set of N lattice vectors, where N is the spatial dimensionality. That is, the lattice points are given as superpositions of these vectors; in three dimensions

$$\mathbf{R}_n = n_1 \mathbf{a}_1 + n_2 \mathbf{a}_2 + n_3 \mathbf{a}_3$$

where $\mathbf{a}_{k \in \{1,2,3\}}$ are the lattice vectors and $n_{k \in \{1,2,3\}} \in \mathbb{Z}$. Because of the long-range order in crystals, we can partition the lattice into smaller cells, unit cells, and reduce the study of the whole lattice to the study of such cells. A primitive unit cell is a unit cell which only contains one lattice point. The basis is the set of atoms, and their relative positions, which are placed at each lattice point. A primitive unit cell always contains one lattice point, but will contain multiple atoms if the basis consists of more than one atom.

The position of atom μ in unit cell n is given as $\mathbf{R}_{n\mu} = \mathbf{R}_n + \mathbf{b}_\mu$, where $\mathbf{R}_n$ is the position of unit cell n and $\mathbf{b}_\mu$ is the position of atom μ relative to the origin of unit cell n . The lattice and basis are chosen such that the atoms are in the positions, called equilibrium positions, that minimize the potential energy.

2.2 Electronic and nuclear dynamics

A large part of this project is concerned with determining the ground state energy of an atomic system. This is done by solving the Schrödinger equation $\hat{H}\Psi = E\Psi$ with the full Hamiltonian given by [16]

$$\hat{H} = \underbrace{\hat{H}_{e,\text{kin}}}_{\text{electron kinetic energy}} + \underbrace{\hat{H}_{ee}}_{\text{electron-electron interaction}} + \underbrace{\hat{H}_{i,\text{kin}}}_{\text{nuclei kinetic energy}} + \underbrace{\hat{H}_{ii}}_{\text{nuclei-nuclei interaction}} + \underbrace{\hat{H}_{ei}}_{\text{electron-nuclei interaction}}.$$

This is a difficult problem to solve, with the main difficulties arising from the interaction terms.

The problem can be simplified by adopting the Born-Oppenheimer approximation, which allows us to separate the wave function of the nuclei and the electrons [16]. This is plausible as the nuclei are much more massive than the electrons, and can thus be considered stationary with respect to the electrons. We therefore split the problem into a nuclear and an electronic part that we then solve separately. The electronic problem takes the form

$$(\hat{H}_{e,\text{kin}} + \hat{H}_{ee} + \hat{H}_{ei} + E_{ii})\Phi_e = E_e\Phi_e, \quad (2.1)$$

where Φ_e is the electronic wave-function and E_e the electronic energy [16]. The $\hat{H}_{i,\text{kin}}$ term has been neglected and $\hat{H}_{ii} = E_{ii}$ is constant under the approximation. The nuclear part of the equation is

$$(\hat{H}_{i,\text{kin}} + V)\chi_i = E_i\chi_i, \quad (2.2)$$

where χ_i is the nuclear wave-function, E_i is the nuclear energy, and $V \equiv E_e$ (written in this form to emphasise that it acts as a potential energy surface for the nuclei) [16].

We can retrieve the forces $\mathbf{F}$ that are acting on the nuclei via the Feynman-Hellmann theorem, which states that

$$\mathbf{F} = -\frac{\partial V}{\partial \mathbf{R}}, \quad (2.3)$$

where $\mathbf{R}$ is the position of the nucleus under consideration [16]. Often the nuclear motion is described classically by combining Eq. (2.3) with Newton's second law of motion

$$m\ddot{\mathbf{R}} = \mathbf{F} = -\frac{\partial V}{\partial \mathbf{R}}. \quad (2.4)$$

2.2.1 Quantization of lattice vibrations; phonons

The treatment of phonons here is based on W. Nolting and W. D. Brewer (2009) [17].

The displacement of atom μ in unit cell n is denoted by $\mathbf{u}_{n\mu}$ and the momentum as $\mathbf{p}_{n\mu} = m_\mu \dot{\mathbf{u}}_{n\mu}$. The Hamiltonian for the nuclei is given by Eq. (2.2). If we expand the kinetic term we get

$$\hat{H} = \hat{H}_{i,\text{kin}} + V = \frac{1}{2} \sum_{n\mu i} \frac{\hat{p}_{n\mu i}^2}{2m_\mu} + V.$$

Now assume that the displacements of the atoms from their equilibrium positions are small relative to the length of the lattice vectors. We can thus expand the potential energy around the equilibrium positions

$$V = V_0 + \sum_{n\mu i} \underbrace{\frac{\partial V}{\partial R_{n\mu i}} \Big|_0}_{=0} u_{n\mu i} + \sum_{n\mu i} \sum_{m\nu j} \underbrace{\frac{\partial^2 V}{\partial R_{n\mu i} \partial R_{m\nu j}} \Big|_0}_{\equiv \phi_{n\mu i m\nu j}} u_{n\mu i} u_{m\nu j} + \dots \quad (2.5)$$

where i and j denote cardinal directions. The first term sets an overall energy scale and is irrelevant, the second term is zero since we expand around the energy minimum, and terms of order three and above are neglected. This is called the harmonic approximation. We use Eq. (2.4) and the expansion in Eq. (2.5) to write

$$m_{n\mu}\ddot{u}_{n\mu i} = -\frac{\partial V}{\partial u_{n\mu i}} = -\sum_{m\nu j} \phi_{n\mu i m\nu j} u_{m\nu j}, \quad (2.6)$$

where $\phi_{n\mu i m\nu j}$ are the so-called force constants. Via the solution ansatz and definition of the dynamical matrix D ,

$$u_{n\mu i} = \frac{\varepsilon_{n\mu i}}{\sqrt{m_{n\mu}}} e^{-i\omega t}, \quad D_{n\mu i m\nu j} \equiv \frac{\phi_{n\mu i m\nu j}}{\sqrt{m_{n\mu} m_{m\nu}}},$$

the differential equation in Eq. (2.6) reduces to the eigenvalue equations

$$\sum_{m\nu j} D_{n\mu i m\nu j} \varepsilon_{m\nu j} = \omega^2 \varepsilon_{n\mu i} \quad \xrightarrow{\text{Fourier transform}} \quad \sum_{\nu j} D_{\mu i \nu j \mathbf{q}} \varepsilon_{\nu j \mathbf{q}} = \omega_{\mathbf{q}}^2 \varepsilon_{\mu i \mathbf{q}}.$$

The solution is given by

$$u_{n\mu i}(t, \mathbf{q}) = \frac{1}{\sqrt{m_{n\mu}}} \varepsilon_{\mu i \mathbf{q}} e^{i(\mathbf{q} \cdot \mathbf{R}_n - \omega_{\mathbf{q}} t)},$$

where $\varepsilon_{\mu i \mathbf{q}}$ are the eigenvectors of the oscillations, $\mathbf{q}$ is the wave-vector, and $\omega_{\mathbf{q}}$ is the frequency. Written in vector form, the key equations for the remainder of this thesis are

$$D_{\mathbf{q}} \varepsilon_{\mathbf{q}} = \omega_{\mathbf{q}}^2 \varepsilon_{\mathbf{q}}, \quad (2.7a)$$

$$\mathbf{u}_{n\mu}(\mathbf{q}) = \mathbf{u}_{0\mu}(\mathbf{q}) e^{i\mathbf{q} \cdot \mathbf{R}_n}, \quad (2.7b)$$

$$H = \frac{1}{2} \left(\frac{\mathbf{p}}{\sqrt{2\mathbf{m}}} \right)^2 + \mathbf{u}^T \phi \mathbf{u}. \quad (2.7c)$$

2.2.2 Density functional theory

DFT is a method for determining the properties of the ground state of a system without explicitly solving the Schrödinger equation [6]. At the foundation of DFT lie the two Hohenberg-Kohn theorems.

Theorem 1 (First Hohenberg-Kohn theorem) *It is possible to describe all ground state properties via the ground state electron density. More specifically, there is a unique functional of the ground state electron density which recovers the ground state energy solution [6].*

Theorem 2 (Second Hohenberg-Kohn theorem) *If the functional of the first Hohenberg-Kohn theorem is known, then the ground state electron density is the electron density which minimizes the functional [6].*

Denote the electron density by $n(\mathbf{r})$, where $\mathbf{r}$ is the electronic position vector. It is defined by the electron wave function as

$$n(\mathbf{r}) = \frac{\langle \Phi_e | \sum_i \delta(\mathbf{r} - \mathbf{r}_i) | \Phi_e \rangle}{\langle \Phi_e | \Phi_e \rangle}. \quad (2.8)$$

We can write the Hamiltonian in Eq. (2.1) as

$$E[n] = E_{e,\text{kin}}[n] + E_{ee}[n] + E_{ei}[n] + E_{ii},$$

where the square brackets denote functionals. To solve the problem in practice, the Kohn–Sham ansatz is employed, which assumes that the ground state electron density η is equal to that of some non-interacting ground state electron density η_0 [16]. The functional of the non-interacting ground state electron density η takes the form

$$E_{KS}[\eta] = \underbrace{E_{e,\text{kin}}^s[\eta]}_{\text{independent kinetic energy}} + \underbrace{E_{ei}[\eta]}_{\text{static nucleus interaction}} + \underbrace{E_{\text{Hartree}}[\eta]}_{\text{electron density self-interaction}} + \underbrace{E_{ii}}_{\text{constant}} + \underbrace{E_{xc}[\eta]}_{\text{exchange-correlation}},$$

where all the difficult interaction terms are kept in the exchange–correlation term $E_{xc}[\eta]$, which is the only term not known exactly [16] [18]. Approximative exchange–correlation functionals are therefore used, and many have been proposed [18]. The functional used in this project is presented in subsection 3.2.1.

With a given functional, the goal is to find n_0 which is equal to η_0 , from which the ground state energy can be determined. By looking at the variation of E_{KS} it is possible to arrive at a Schrödinger-like equation, called the Kohn–Sham equation, for the independent electrons

$$(\hat{H}_{i,\text{kin}} + \hat{H}_{ie} + \hat{H}_{\text{Hartree}} + \hat{H}_{xc})\Psi_{e,\text{indep}} = E_{e,\text{indep}}\Psi_{e,\text{indep}}, \quad (2.9)$$

where $\Psi_{e,\text{indep}}$ and $E_{e,\text{indep}}$ are the wave functions and the energy of the non-interacting electrons, respectively [16]. Now η_0 can be determined via a self-consistency loop: make an initial guess of η_0 ; determine $\Psi_{e,\text{indep}}$ from Eq. (2.9); determine η'_0 from Eq. (2.8) using $\Psi_{e,\text{indep}}$; if $\eta_0 = \eta'_0$, then the guess was correct and you are done, otherwise alter the guess and start over [16].

2.2.3 Molecular dynamics

To determine the dynamics of the atoms at finite temperatures MD simulations will be employed. We treat the nuclei classically and solve Eq. (2.4) [19]. By integration it is then possible to find the position and velocity at any other time. This has to be done numerically and Eq. (2.4) will have to be integrated sequentially for some small time step Δt . The choice of integration algorithm is crucial to ensure accurate and fast simulations [19]. One widely used algorithm is the Verlet algorithm which gives the new atomic position according to [19]

$$\mathbf{R}(t + \Delta t) = 2\mathbf{R}(t) - \mathbf{R}(t - \Delta t) - \frac{1}{m} \frac{\partial V}{\partial \mathbf{R}} \Delta t^2 + \mathcal{O}(\Delta t^4).$$

Note that this algorithm does not give the velocities explicitly. There are several methods which allow for calculation of velocities, such as immediate calculation from the calculated position (which gives an approximation up to second order) or the velocity–Verlet algorithm (which gives an approximation up to fourth order) [19].

2.3 Symmetry

Symmetries are important in the study of condensed matter and can reduce the complexity of many problems. It is an integral part of this project as it helps us to understand and study phase transitions in perovskites. This section will therefore first present the mathematical formalism of symmetries, before applying it to crystalline materials.

2.3.1 Mathematical formalism of symmetry

In this section, Definitions 1-4 are based on J. R. Durbin (2009) [20], while Definition 5 and 6 as well as Theorems 1 and 2 are based on J. P. Serre (1977) [21].

Consider for a moment what properties a symmetry transformation should have. First of all, it needs to leave the object invariant after it has been applied, or it would not be a symmetry at all. Some other properties are also desirable; we can always reverse a symmetry transformation and get back to the original structure, the identity (i.e. not transforming the structure at all) is always a symmetry operation, and applying two symmetry operations is equivalent to applying some other symmetry operation. This all leads directly to the definition of a group.

Definition 1 (Group) *A group is a set of objects G together with a binary operation $\otimes : G \times G \rightarrow G$, which fulfil the following three properties*

1. *The binary operation is associative, i.e. $a \otimes (b \otimes c) = (a \otimes b) \otimes c$, where $a, b, c \in G$.*
2. *There exists an identity $e \in G$ such that $a \otimes e = e \otimes a$ for all $a \in G$.*
3. *There exists an inverse $a^{-1} \in G$ for every $a \in G$ such that $a^{-1}a = aa^{-1} = e$.*

Note that the binary operation ensures that the group is closed. Using these properties it is easy to show that the identity is unique and that every element has a single unique inverse. Going forward, the binary operation will be written as a contraction, so that $a \otimes b$ will be written as ab . The binary operation will simply be called multiplication, remembering to not confuse it with arithmetic multiplication, which is just one specific example of a binary operation. When groups are considered in the context of symmetries, the group is often referred to as the symmetry group, and its elements as symmetry operations. One also needs to define a subgroup.

Definition 2 (Subgroup) *A subset $S \subset G$ is a subgroup if S is closed under the multiplication of G , that is $ab \in S$ for all $a, b \in S$.*

Every group has two trivial subgroups, namely the group itself and the group only consisting of the identity element.

The labelling of the elements in the group is of no importance, since any relabelling will result in the exact same structure. It is the structure of the group that is important, and a multiplication table includes all information about a group. This introduces the important concept of isomorphism, which are bijective mappings from one group to another which preserves multiplication.

Definition 3 (Isomorphism) *An isomorphism is a map $\rho : G \rightarrow H$ between groups G and H that is bijective and which preserves the group structure, i.e. $\rho(ab) = \rho(a)\rho(b)$ for all $a, b \in G$.*

This is a precise mathematical definition of relabelling. If we remove the restriction of bijectivity, we arrive at another important concept, namely homomorphisms.

Definition 4 (Homomorphism) *A homomorphism is a map $\rho : G \rightarrow H$ between groups G and H which preserves the group structure.*

It is often useful to represent a group by something concrete, that is to find a group with the same multiplication structure with simple and well-studied elements. For our purposes this will be the general linear group $GL(V)$, which consists of $N \times N$ invertible matrices over some vector space V .

Definition 5 (Representation) *A representation of a group G is a group homomorphism $\rho : G \rightarrow GL(V)$. The vector space V is called the representation space.*

In practice this results in the ability to map any element $g \in G$ to a matrix $\rho(g) \in GL(V)$, called the representation of g .

Suppose now that there is a linear subspace $W \subseteq V$ such that $\rho(a)w \in W$ for all $a \in G$ and all $w \in W$. Then $GL(W)$ is itself a subgroup of $GL(V)$, and ρ restricted in the codomain to W is called a subrepresentation. Every representation has two trivial subrepresentations, where $W = V$ or $W = 0$. The following definition and theorems tell us that we can decompose a representation.

Definition 6 (Irreducible and reducible representations) *A representation that has no non-trivial subrepresentations is called an irreducible representation (irrep); otherwise it is called a reducible representation.*

Theorem 3 (Representation decomposition) *Let V be the representation space of a finite group G . If V has a subrepresentation space W , then there exists another subrepresentation space U , such that V is their direct sum, $U \oplus W = V$.*

Theorem 4 (Decomposition into irreducible representations) *Every representation can be decomposed into a direct sum of irreducible representations.*

To summarize, we can take any group and represent it in a vector space, where the elements are mapped to matrices. This representation space can be decomposed into a direct sum of irrep spaces. This means that they are completely separated, and any representation matrix can be written in block diagonal form where each block corresponds to one irrep space. Therefore, each irrep can be studied separately without losing any information. This makes the study of symmetries easier as we can break down all the possible symmetry operations into a set of smaller operations.

2.3.2 Symmetries of the lattice

In three dimensions there are a total of 14 possible lattices, called the Bravais lattices, which arise from the constriction of translational symmetry that the lattice must possess [15, 22].

It is also possible to rotate and reflect the lattice around some line or plane, respectively. Of course, for these to be symmetry operations they have to be compatible with the translational symmetry of the lattice. These symmetry operations form a so-called point group, and it contains the possible rotations and reflections which leave the lattice invariant [15]. Now, if we consider the 14 Bravais lattices from before, there are exactly 32 possible point groups which leave one or more of these lattices invariant. These are called the crystallographic point groups [15, 23].

So given any lattice, it is now natural to ask what translation and point group symmetries it has. This is encoded in the lattice's space group, which is simply a label that explains which combinations of the Bravais lattice and the crystallographic point groups that are symmetry operations [23]. This classification leads to a total of 230 space groups, and any crystalline structure has the symmetry operations of one of these groups [23]. Now recall our treatment of groups and their representations from before. For any such space group, we can find a set of irreducible representations, which are matrices, which together yield all possible symmetry operations.

2.3.3 Symmetries of phonons and phase transitions

Some materials, and often perovskites, have what is called soft phonon modes [24]. These are phonon modes where the frequency ω approaches zero at some temperature. Recalling Eq. (2.5) we can see what happens at such a transition. The force matrix changes sign, which implies that $\omega^2 < 0$, so the solutions become imaginary [24]. This is, of course, entirely unphysical, but is simply an artifact from the harmonic approximation; higher order terms in V cannot be neglected, which introduces new minima. Following the potential energy landscape, the atoms move into some new energy minimum, which becomes the new equilibrium position. Such a transition will be called a structural phase transition, referring to the permanent displacement of the atoms [24].

Suppose that a crystal has some space group G , and that we introduce a displacement arising from the lattice vibrations of some phonon. This of course breaks the original space group symmetry in some way, and left is the symmetry group $G(\mathbf{q})$, where $\mathbf{q}$ is the wave vector of the phonon. Because it is a symmetry operation, any element $g \in G(\mathbf{q})$ must commute with Hamiltonian in Eq. (2.7c), and as such also the dynamical matrix $D(\mathbf{q})$. By choosing a representation space we can block-diagonalize g by finding the irreps. As it commutes with D , they are simultaneously diagonalizable. Each block can be studied separately, and the eigenvalue equation in Eq. (2.7a) can be solved for each block. The eigenvectors $\boldsymbol{\epsilon}$ thus live in the same vector space as the irrep, and can be expressed as a superposition of the irrep's basis $\{\mathbf{E}_i\}_{i=1}^D$, where D is the dimension of the irrep [25]. That is,

$$\boldsymbol{\epsilon} = \sum_{i=1}^D c_i \mathbf{E}_i,$$

where c_i are coefficients [25].

The coefficients c_i are order parameters of the structural phase transitions [25]. Above the transition temperature, $\omega_{\mathbf{q}} > 0$ and the atoms oscillate around their equilibrium positions. The average amplitude of the distortions is therefore zero, and so the time average of c_i becomes $\langle c_i \rangle_t = 0$. Below the transition a constant displacement from the equilibrium position is present, and so $\langle c_i \rangle_t \neq 0$. It is also possible to look at structure parameters as order parameters since the symmetry constraints also disallow certain strains. When the symmetry of the structure is changed, the constraints on the lattice parameters also change.

2.3.4 The perovskite structure, again

We are now ready to apply this theory to perovskites. We will consider the distortions arising from the high-symmetry wave-vectors

$$\mathbf{q}_\Gamma = (0, 0, 0), \quad \mathbf{q}_X = \frac{1}{2}(0, 1, 0), \quad \mathbf{q}_M = \frac{1}{2}(1, 1, 0), \quad \mathbf{q}_R = \frac{1}{2}(1, 1, 1).$$

These give rise to 120 distortions, which can be decomposed into irreps. The distortions used in this project has been generated by ISODISTORT [26]. One eigenvector from each irrep is shown in Fig. 2.1, 2.2 and 2.3. The dimensions of these irreps ranges from one to six. We will index the eigenvectors by a, b, c, d, e, f , which denote the directions in the irrep space, and they do not necessarily align with the axes of the position space. Three of these distortions are especially common in perovskites, namely B-cation displacement, in-phase octahedral tilting and out-of-phase octahedral tilting [8]. These correspond to Γ_4^- in Fig. 2.2 and M_2^+ , R_5^- in Fig. 2.3 (Γ_4^- , M_3^+ and R_4^+ in Stokes et al. [8]). Systematically calculating all inequivalent combinations of these results in a total of 61 possible distortions [8]. Some of these have the same space group, which does not imply that they are equivalent since different atoms can displace but have the same symmetry.

The the cubic perovskite primitive cell contains 5 atoms. We can represent the positions as a $(5, 3)$ vector where each row is the position of an atom. The vector (in fractional coordinates) is then

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{matrix} \leftarrow \text{A} \\ \leftarrow \text{B} \\ \leftarrow \text{X}_1 \\ \leftarrow \text{X}_2 \\ \leftarrow \text{X}_3 \end{matrix} \quad (2.10)$$

For the phonon modes at X , M and R , a larger supercell is required as the phase in Eq. (2.7b) only becomes apparent when the supercell contains more than one primitive cell [25]. To see this effect, let $\mathbf{R}_n = n_1 \mathbf{a}_1 + n_2 \mathbf{a}_2 + n_3 \mathbf{a}_3$ be some lattice vector, where $n_1, n_2, n_3 \in \mathbb{Z}$. The phases for Γ , X , M and R become 1, $(-1)^{n_2}$, $(-1)^{n_1+n_2}$ and $(-1)^{n_1+n_2+n_3}$, respectively. Adjacent cell distortions have opposite signs in the y direction for X , in the x and y direction for M , and in all three directions for R . In the Γ mode all cells move in phase. To account for this, all distortions are generated for the $2 \times 2 \times 2$ structure.

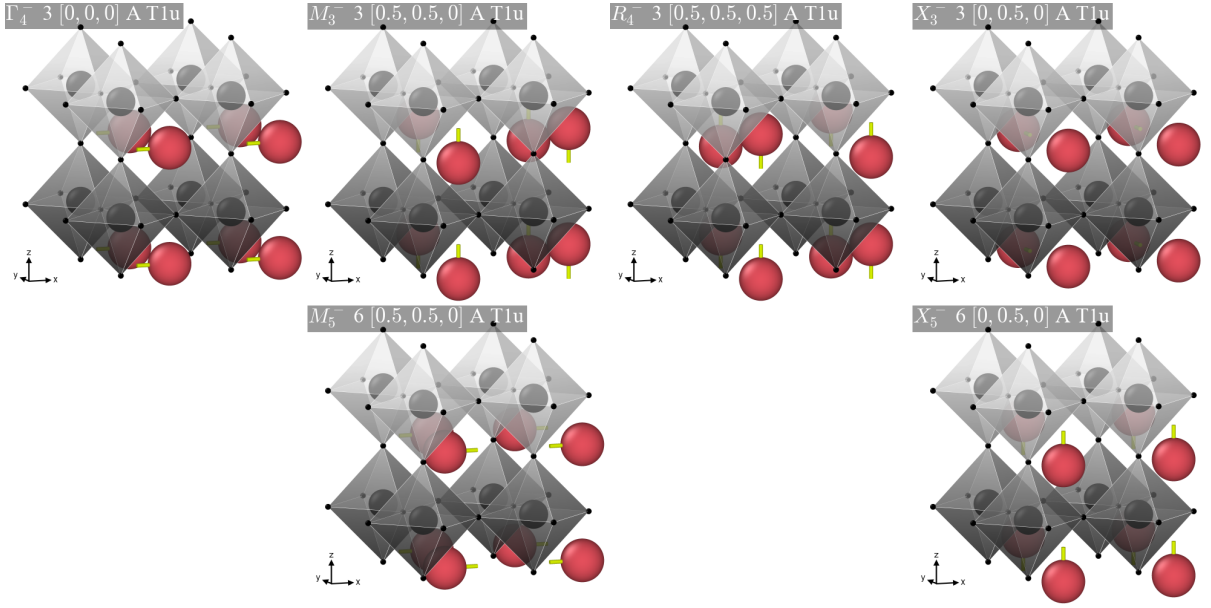


Figure 2.1: One eigenvector from each irrep which displaces the A atoms. The labels describe the irreps name, the dimension of the irrep, the wave-vector, and the displaced species, respectively.

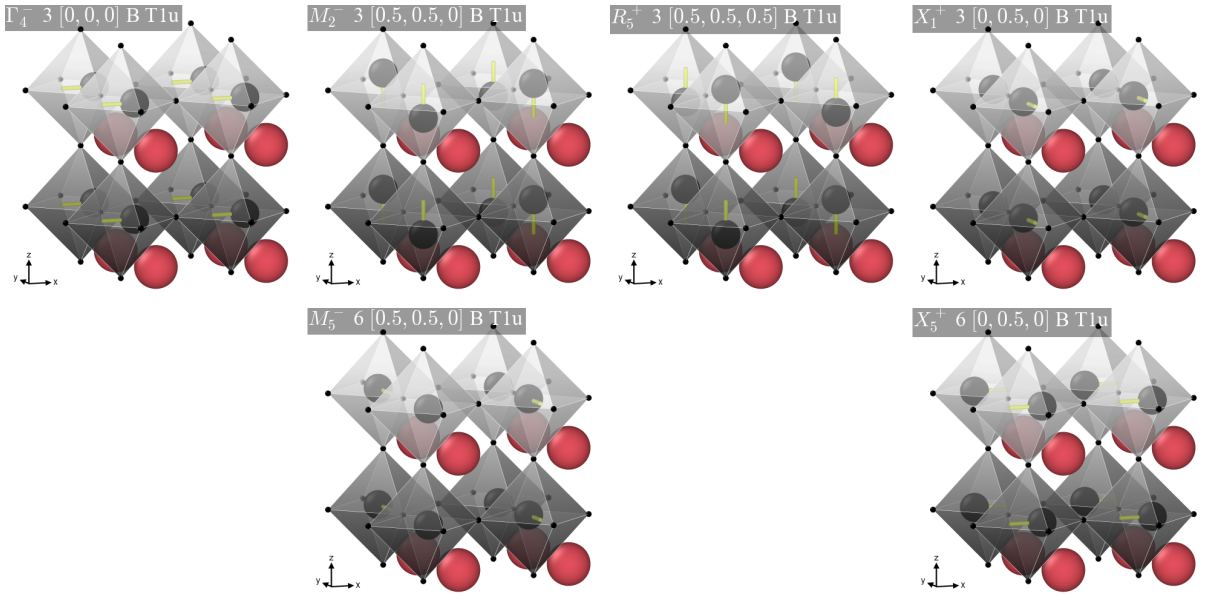


Figure 2.2: One eigenvector from each irrep which displaces the B atoms. The labels describe the irreps name, the dimension of the irrep, the wave-vector, and the displaced species, respectively.

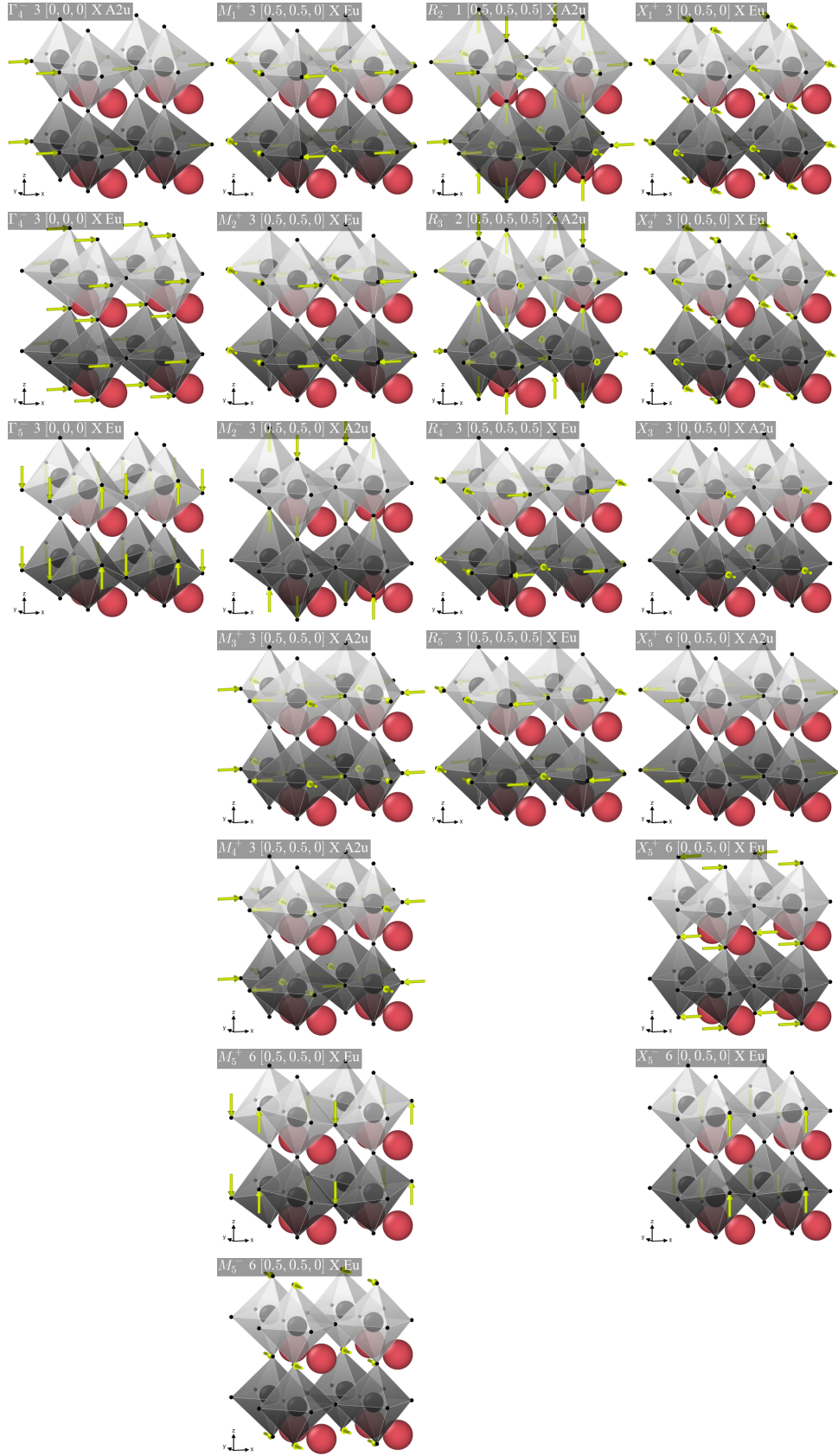


Figure 2.3: One eigenvector from each irrep which displaces the O atoms. The labels describe the irreps name, the dimension of the irrep, the wave-vector, and the displaced species, respectively.

2.4 The MACE architecture

The MACE architecture is a type of MPNN, which is in turn a type of graph neural network (GNN). A GNN is a neural network which takes a graph as its input, that is a set of nodes and edges. It is readily apparent why this type of network is suitable for atomic structures, as each atom can be represented by a node and its interactions by edges.

2.4.1 Message-passing neural network

In an MPNN, as described by J. Gilmer et al. (2017) [27], each node i is represented by a vector of known information and learnable features V_i . A message is constructed as an expansion for each node according to

$$m_i = \sum_j M^{(1)}(V_i, V_j) + \sum_{jk} M^{(2)}(V_i, V_j, V_k) + \dots, \quad (2.11)$$

where $M^{(j)}$ are learnable message functions and the sum runs over the atoms of the lattice. Then the node vector is updated

$$V_i^{\text{updated}} = U(V_i, m_i)$$

by some learnable update function U . These message and update steps are then repeated for a specified number of steps. Finally, the node vectors are projected to a state vector via a learnable readout function. For atomic simulation purposes, each node i in the graph represents an atom and the final state vector is the energy at atom site i [12].

The MACE architecture expands upon this idea in two main ways: equivariant MPNNs and higher-order atomic cluster expanded message passing functions.

2.4.2 Equivariant MPNN

The idea here is to utilize some underlying symmetry of the system to restrict the possibilities of the learnable atomic features V_i . For atomistic purposes, the symmetry is $O(3)$, which represents rotations and reflections of the atom, neither of which should influence the physics of the atom. This method allows for higher performance, but suffers limitations from high computational costs and difficulty in parallelising over multiple GPUs [12].

2.4.3 ACE message-passing functions

The inclusion of all interactions quickly grows unfeasible as the number of atoms increases, and it is therefore required to introduce a cutoff radius r_c . Naïvely, one can truncate the message function at some point which yields a sufficient balance between accuracy and computational cost. This series, however, converges as N_c^{n-1} , where N_c is the number of neighbours of each atom and n is the order of the leading term [11]. This quickly becomes computationally heavy as r_c increases.

To remedy this problem, we want to find some expansion which converges faster. One solution is ACE, introduced by R. Drautz (2019) [11]. The description here is based on

their derivation.

The goal is to determine some property of atom i , which we write as

$$\Lambda_i(\mathbf{R}_{1i}, \dots, \mathbf{R}_{Ni}),$$

where $\mathbf{R}_{ji}$ is the vector from the position of atom i to atom j , and N is the number of atoms. Introduce an orthogonal (both in index and position) basis $\phi_{\alpha \in \mathbb{N}}(\mathbf{R}_{ji})$ which crucially only depends on a single vector: the vector from atom i to atom j .

We now construct a cluster around atom i , which is a collection of atoms fulfilling some criteria when related to atom i , such as lying within some cutoff radius. Let there be M atoms in the cluster. There are then $M - 1$ position vectors from atom i to the atoms in the cluster, which we denote as $\mathbf{R}_{j_1i}, \dots, \mathbf{R}_{j_Mi}$. We can group the index pairs (called bonds) into a vector $\mathbf{b} = (j_1i, \dots, j_Mi)$. For each bond there is an orthogonal basis as described above, and we can arrange their indices into the vector $\mathbf{a} = (\alpha_1, \dots, \alpha_M)$. The basis for each bond can be combined into a cluster basis function

$$\Phi_{\mathbf{ab}} = \phi_{\alpha_1}(\mathbf{R}_{j_1i}) \dots \phi_{\alpha_M}(\mathbf{R}_{j_Mi}).$$

Any two cluster basis functions are orthogonal, which is a direct consequence of the orthogonality of the ϕ bases. We can now express the property Λ_i in this basis

$$\Lambda_i = \sum_{\mathbf{ab}} J_{\mathbf{ab}} \Phi_{\mathbf{ab}}, \quad \text{where} \quad J_{\mathbf{ab}} = \langle \Phi_{\mathbf{ab}} | \Lambda_i \rangle.$$

Assume that atom k and atom l are of the same species. Exchanging the bonds $ik \leftrightarrow il$ does not change the atomic configuration, and therefore not any physical property. Because of this invariance, the coefficients $J_{\mathbf{ab}}$ must only depend on the number of bonds to each unique species. Denote the number of bonds to species s by $\mu_{s \in \{1, \dots, S\}}$, where S is the number of unique species. We can then write $J_{\mathbf{ab}} = J_{\mathbf{a}\mu_1, \dots, \mu_S}$. We can now group the expansion in terms of number of bonds to each species. Let $J_{\mathbf{as}}^{(\sigma)}$ denote the coefficient with σ bonds to species s . We get

$$\Lambda_i(\mathbf{R}_{1i}, \dots, \mathbf{R}_{Ni}) = \frac{1}{1!} \sum_{\alpha, j, s} J_{\alpha s}^{(1)} \phi_{\alpha}(\mathbf{R}_{ji}) + \frac{1}{2!} \sum_{\substack{\alpha_1, j_1, s_1 \\ \alpha_2, j_2, s_2}}^{j_1 \neq j_2} J_{\alpha_1 \alpha_2 s_1 s_2}^{(2)} \phi_{\alpha_1}(\mathbf{R}_{j_1i}) \phi_{\alpha_2}(\mathbf{R}_{j_2i}) + \dots$$

The final step is to project these basis vectors onto a basis that is more convenient for our purposes, called the atomic basis

$$A_{i\alpha} = \sum_j \phi_{\alpha}(\mathbf{R}_{ji}).$$

The expansion then becomes

$$\Lambda_i(\mathbf{R}_{1i}, \dots, \mathbf{R}_{Ni}) = \frac{1}{1!} \sum_{\alpha, s} J_{\alpha s}^{(1)} A_{i\alpha} + \frac{1}{2!} \sum_{\substack{\alpha_1, s_1 \\ \alpha_2, s_2}}^{j_1 \neq j_2} J_{\alpha_1 \alpha_2 s_1 s_2}^{(2)} A_{i\alpha_1} A_{i\alpha_2} + \dots$$

The computational construction of the atomic basis is of order N_c , and the evaluation of Λ_i is invariant of N_c , which means that the energy evaluation scales linearly with N_c and is independent of the order of expansion [11]. The MACE architecture utilizes this by expanding the message function in Equation 2.11 using ACE [12].

2.4.4 Accuracy and computational cost

MACE was chosen for this project because of its demonstrated high-accuracy compared to other equivariant MPNNs and ACE architectures [12]. Moreover, it produces smaller errors on highly restricted datasets, compared to other architectures [12]. Other architectures, such as NEP [13], were not chosen as MACE displays more accurate results, albeit at higher computational costs [14]. Since the goal of this project is to train a foundational model, the accuracy of MACE is preferred. The data produced by the model can be used to train subsequent NEP models without relying on new DFT data, a process referred to as model distillation [14].

3

Methods

The methods used in this project are presented here, beginning with an overview before giving detailed information under the various sections. The workflow is presented schematically in Fig. 3.1. All settings and parameters used in the different computations are presented in App. A.

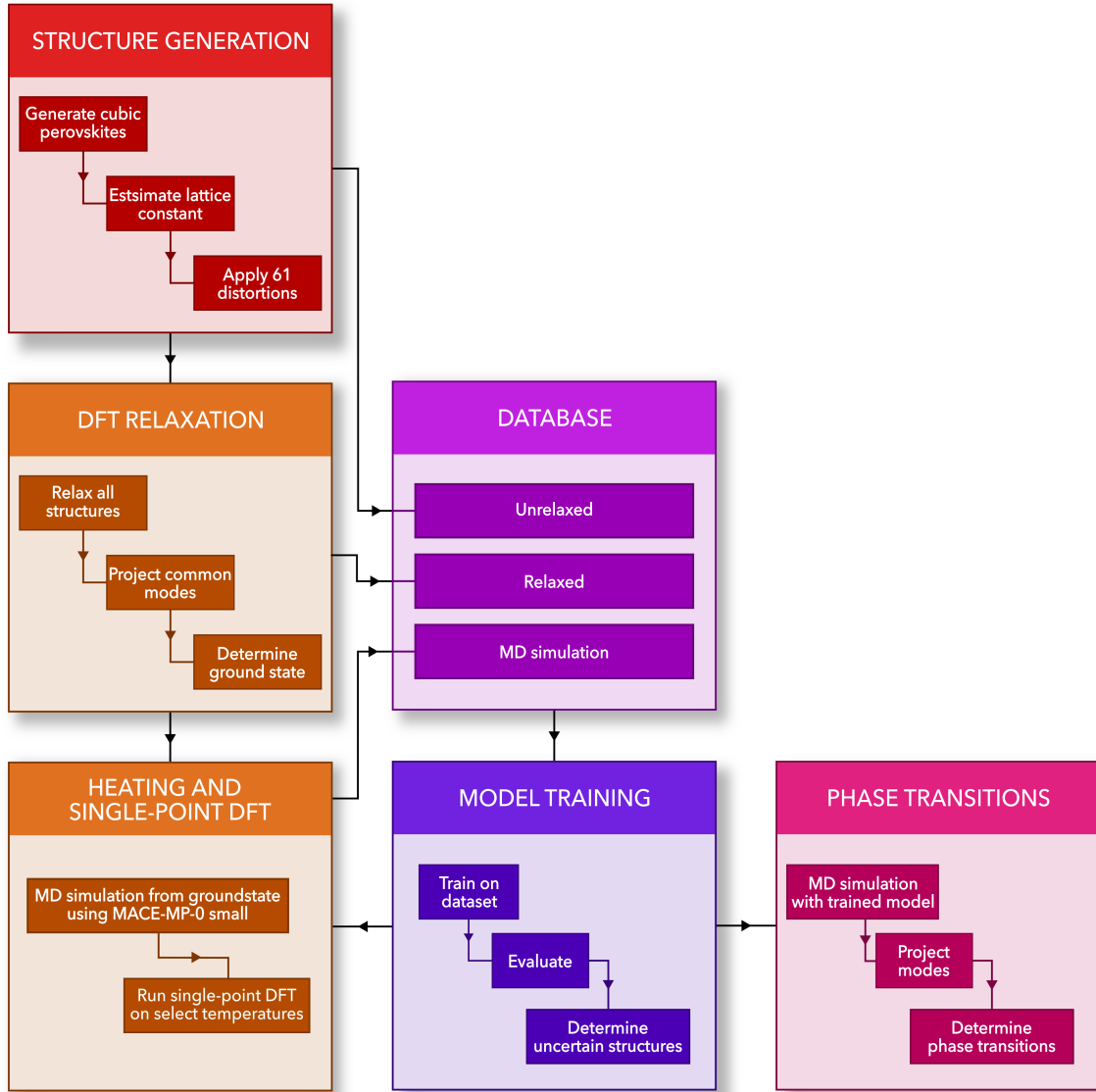


Figure 3.1: A schematic drawing of the methods employed in this thesis.

3.1 Code design and structure generation

All structures were handled via the atomic simulation environment (ASE) Python package, and the data was organized into ASE databases [28]. A small package, pfox (perovskite foundational executables), was constructed for this project, which serves as the interface from which the databases and various output files can be read, analysed and stored in a consistent way.

The cubic perovskite’s lattice parameter was first estimated by a quick relaxation with the general MACE-MP-0 foundational model [29]. The cubic structure was then tiled into a $2 \times 2 \times 2$ supercell. This was done as the in-phase and out-of-phase tilting patterns become distinguishable first at this size. The 61 possible distortions was then applied according to the relative eigenvector amplitudes, as presented by Stokes et al. (2002) [8]. An initial guess of the amplitudes was then made, which starts at $0.5 \text{ \AA}$ and decreases in steps of $0.05 \text{ \AA}$ for each subsequent mode added (this corresponds to $a = 0.5$, $b = 0.45, \dots$ in Stokes’ paper). The motivation of this descension of amplitudes lies in that if more modes are added, their overall amplitude decreases. When having multiple soft phonon modes active above the transition temperature, the thermal energy is distributed among them.

3.2 DFT calculations

All DFT calculations carried out in this project was done using the Vienna Ab initio Simulation Package (VASP) [30, 31, 32]. The full settings can be found in Tab. A.1.

3.2.1 Choice of functional

The first step in running DFT is choosing the exchange-correlation functional. To motivate the choice of functional in this project, some common ones are presented in Tab. 3.1. The local density approximation (LDA) considers the density at each point without regarding the surroundings [18]. A generalized gradient approximation (GGA) functional considers the electron’s immediate surrounding via the gradient, increasing accuracy [18]. A meta-GGA (mGGA) functional takes this further by including the second-order gradient and the kinetic energy density τ [18]. Finally, the Van der Waals (VDW) term adds long range corrections via a double integral of the positions, weighted by a kernel Ψ [33]. This functional is added as a correction to a semi-local functional. The expressions presented in Tab. 3.1 only describe the general form, and there is a wide range of implementations. Each functional in the table improves accuracy upon the last, but at higher computational costs [18].

Since this project required accurate data for the training of the foundational model, an mGGA functional with a VDW correction term was used. The specific mGGA that was used is $r^2\text{SCAN}$ [34]. This is based on the $r\text{SCAN}$ functional [35], itself a more stable version of the strongly constrained and appropriately normed (SCAN) functional [36]. The VDW functional that was used is $r\text{VV10}$ [37], an improved version of the Vydrov and van Voorhis (2010) (VV10) [38] functional. The functional that was used is therefore

r^2 SCAN + r VV10, as described by Ning et al. [39].

Table 3.1: Common functionals, their locality and general mathematical expressions. The descriptions are taken from the VASP wiki [18] [33].

Locality	Name	Form
Local	LDA	$E_{xc} = \int \epsilon_{xc}(n) \, dr^3$
Semi-local	GGA	$E_{xc} = \int \epsilon_{xc}(n, \nabla n) \, dr^3$
Semi-local	mGGA	$E_{xc} = \int \epsilon_{xc}(n, \nabla n, \nabla^2 n, \tau) \, dr^3$
Non-local	VDW	$E_{xc} = \int \int n(r) \Psi(r, r') n(r') \, dr^3 dr^3$

3.2.2 Single-point and relaxation

Single-point calculations entail finding the ground-state electron density η_0 , as described in Sec. 2.2.2, and the calculation is carried out via the self-consistency loop. As DFT is designed to handle precisely this type of problem, the settings used are rather standard and will not be discussed here. See Tab. A.1 for the settings used.

Structure relaxation (or optimization) refers to the process of finding the ground state when considering the positions of electrons and nuclei as dynamic. The description here is based on the VASP wiki [40]. The relaxation process entails multiple single-point calculations, one for each nuclear position arrangement. Because of this, such a problem is harder and computationally much more expensive. VASP carries out different approaches, as specified by the IBRION tag. Since the structures have been distorted according to three common irreps, it can be assumed that the true ground-state distortion is not very far from its minimum. Because of this, and the demand for highly accurate results, IBRION=1 was chosen. The symmetry was fixed via ISYM=1, which ensures that the space group symmetry is not broken during relaxation, with a tolerance SYMPREC= 10^{-5} . Atomic positions, cell lengths and cell angles were allowed to change by setting ISIF=3.

Finally, the relaxation was run multiple times, starting from the final relaxed structure of the previous run. This was done because the inclusion of wave vectors was created dynamically, i.e. depending on the cell volume of the initial structure. This was done by setting ENCUT=520, which then determines how many wave vectors that lie inside the energy cut-off. Starting a new run from the relaxed structure thus gives a re-scaled sampling, giving more accurate predictions which allow the structure to relax further. The restarts terminate when the resampled run agrees with the previous run, and so the relaxation terminates after one step. The maximum number of restarts was set to 15, in case DFT struggled to find the minimum.

3.3 MD heating

MD heating was performed using ASE [28] and the large-scale atomic/molecular massively parallel simulator (LAMMPS) [41] with the Kokkos package [42]. The settings and parameters are found in Tab. A.2 and A.3. The starting point of every run was the relaxed ground-state structure at 0 K, as determined by DFT. This was done as non-relaxed structures can be unstable, and so the start of the MD run might be unstable.

For the initial MD heating runs an already existing foundational model, MACE-MP-0 small [29], was used. This model was trained on a large database of materials (not specifically perovskites), and was used to give approximate structures at specific temperatures. The MACE-MP-0 model was used to heat $2 \times 2 \times 2$ (40 atoms) and $4 \times 4 \times 4$ (320 atoms) structures. As these are rather small, they do not require the parallelization offered by LAMMPS, and were therefore run in ASE. Larger structures are considered unfeasible as the DFT computations become too costly. For the 40 atom structures, strains were randomly applied to the initial structures to ensure more diversity.

After training of the foundational model, it was used to run MD on $12 \times 12 \times 12$ (8640 atoms) supercells. The large supercell is used to account for finite size effects which can arise in smaller cells. The size requires the use of more sophisticated MD implementations, and as such LAMMPS was used.

3.4 Datasets and model training

In Tab. 3.2 the datasets generated from DFT and MD are described and named. Included is also which datasets were used in the different models. In total $3 \times 5 = 15$ models will be presented in this project, where five models were trained for each set of input data and parameters. The grouping of models are called 64x300K, 64x1000K and 128x1000K, which describes the number of hidden irreps (see Tab. A.4) and the highest temperature structures included in the training. Such a grouping will frequently be referenced to as one model, even though it in actuality contains five models trained using different initial seeds (see below). The datasets included in the training data of each model is shown in Tab. 3.2. All parameters used in training are found in Tab. A.4.

To evaluate the models they were cross-validated, where the different seed models all were set to evaluate some structure. The root mean square error (RMSE) of these models then estimate the model’s uncertainty, as the models will disagree on structures that are under-represented or complex to evaluate.

Table 3.2: The different datasets generated for this project. They are here given names to make referring to them easier, as well as a short description and the total count of structures they contain. Presented is also for which models, if any, the dataset was part of the training data.

Dataset	Description	Count	64x300K	64x1000K	128x1000K
UN-RELAXED	All generated structures and distortions, taken from first step of relaxation trajectory	3599	YES	YES	YES
RELAXED	All relaxed structures and distortions, taken from last step of relaxation trajectory (or lowest energy step)	3599	YES	YES	YES
MD300K40	MD heated 40 atom structures from 0 K to 300 K, taken at approx. temp. 60, 120, 180, 240, 300 K	295	YES	YES	YES
MD300K320	MD heated 320 atom structures from 0 K to 300 K, taken at approx. temp. 60, 120, 180, 240, 300 K	295	YES	YES	YES
MD1000K40	MD heated 40 atom structures from 0 K to 1000 K, taken at approx. temp. 500 and 1000 K	118	NO	YES	YES
MD1000K40VAL	MD heated 40 atom structures from 0 K to 1000 K, taken at approx. temp. 500 and 1000 K. Used to test the models.	118	NO	NO	NO

4

Results and Discussion

Here the main results of the projects are presented alongside discussions. The chapter closes by considering improvements and possible extensions of this project.

4.1 DFT relaxation and MD simulations

Here the results of the DFT relaxations are presented. The determined ground states are found in Tab. 4.1, as well as comparisons to the materials project database [43]. We see generally good agreement, with some discrepancies marked. Some correspond to missing reference data (*) in the materials project database. States with <0.01 eV/atom energy differences (**) are sensitive to the chosen functional since the differences are so miniscule. Symmetries that were not included in the 61 that were generated (***) sometimes appear as well, some not even being perovskites. The remaining discrepancies (†) are not as easy to explain, and can be considered erroneous. All these are converged, which may indicate that the structure only finds some local minimum. One probable explanation is that the initial structure was too different from the ground state, making the algorithm unable to locate it. Of note is that the majority of these contain Rb, indicating that the guessed modes are inaccurate for this element.

These will be used as the starting point for MD simulations. It is therefore not necessary to find the exact correct ground-state, as the MD simulation will allow the atoms to move into the correct state. It is important that the structure is close to the ground-state, as the MD heating may be unstable otherwise. As such, these results were considered sufficiently accurate for the remainder of the project.

Table 4.1: Presented here is the determined ground state space groups (SG), materials project references (SG ref.) [43], convergence status (Con.), and mode amplitudes ($G_{x,y,z}$, $M_{x,y,z}$, $R_{x,y,z}$). The Disc. column indicates discrepancies: (*) reference data unavailable; (**) energy difference is <0.01 eV/atom; (***) the symmetry was not considered; (†) unexplained.

Name	Con.	SG	SG ref.	Disc.	Gx	Gy	Gz	Mx	My	Mz	Rx	Ry	Rz
BaHfS3	True	Pnma	Pnma		0.00	0.00	-0.00	0.32	0.00	0.00	0.00	0.29	0.29
BaHfSe3	True	Pnma	NaN	*	0.00	-0.00	0.00	0.45	-0.00	-0.00	0.00	0.35	0.35
BaTiO3	True	R3m	Amm2	**	0.05	0.05	0.05	-0.00	-0.00	-0.00	-0.00	0.00	0.00
BaTiS3	True	I4cm	P6 ₃ /mmc	***	0.00	0.00	0.25	0.00	0.01	0.00	0.00	0.00	0.43
BaTiSe3	True	I4cm	P6 ₃ /mmc	***	0.00	0.00	0.38	0.00	0.00	-0.00	0.00	0.00	0.61
BaZrO3	True	Pm3m	I4/mcm	**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
BaZrS3	True	Pnma	Pnma		0.00	0.00	0.00	0.36	0.00	0.00	0.00	0.31	0.31
BaZrSe3	True	Pnma	Pnma		0.00	0.00	0.00	0.48	0.00	0.00	0.00	0.37	0.37
CaHfO3	True	Pnma	Pnma		0.00	-0.00	0.00	0.40	-0.00	-0.00	0.00	0.41	0.41
CaHfS3	True	Pnma	Pnma		0.00	-0.00	0.00	0.00	0.00	0.60	0.57	0.57	0.00
CaHfSe3	True	Pnma	NaN	*	-0.00	-0.00	0.00	-0.00	-0.00	0.67	0.62	0.62	0.00
CaTiO3	True	Pnma	Pnma		0.00	-0.00	0.00	0.32	-0.00	-0.00	0.00	0.30	0.30
CaTiS3	True	Pmn2 ₁	NaN	*	-0.07	0.07	0.00	0.00	0.00	0.57	0.49	0.49	0.00
CaTiSe3	True	Pmn2 ₁	NaN	*	-0.08	0.08	-0.00	0.00	0.00	0.65	0.55	0.55	0.00
CaZrO3	True	Pnma	Pnma		-0.00	0.00	-0.00	0.42	0.00	0.00	0.00	0.44	0.44
CaZrS3	True	Pnma	Pnma		0.00	-0.00	-0.00	0.62	0.00	0.00	0.00	0.60	0.60
CaZrSe3	True	Pnma	NaN	*	-0.00	0.00	0.00	0.00	-0.00	0.69	0.64	0.64	0.00
CsGeBr3	True	R3m	R3m		0.14	0.14	0.14	-0.00	-0.00	0.00	0.00	-0.00	-0.00
CsGeCl3	True	R3m	R3m		0.09	0.09	0.09	-0.00	0.00	-0.00	-0.00	0.00	-0.00
CsGeF3	True	R3c	NaN	*	-0.00	-0.00	-0.00	-0.00	0.00	-0.00	0.26	0.26	0.26
CsGeI3	True	Cm	R3m	†	0.19	0.19	0.19	0.14	0.14	-0.00	0.00	-0.00	0.00
CsPbBr3	True	Pnma	Pnma		0.00	-0.00	0.00	0.56	0.00	-0.00	0.00	0.41	0.41
CsPbCl3	True	Pnma	P4/mbm	†	0.00	0.00	-0.00	0.47	0.00	0.00	0.00	0.37	0.37
CsPbF3	True	R3c	R3c		0.02	0.02	0.02	0.00	-0.00	-0.00	0.25	0.25	0.25
CsPbI3	True	Pnma	Pnma		0.00	-0.00	-0.00	0.78	0.00	0.00	0.00	0.47	0.47
CsSnBr3	True	Pnma	Pnma		0.00	0.00	0.00	0.37	0.00	-0.00	0.00	0.30	0.30
CsSnCl3	True	Ima2	P12 ₁ /c1	***	0.04	0.04	-0.00	0.00	-0.00	0.00	0.43	0.43	0.00
CsSnF3	False	Pmn2 ₁	P12 ₁ /c1	***	0.02	-0.02	0.00	0.00	0.00	0.31	0.14	0.14	0.00
CsSnI3	True	Pnma	Pm3m	**	0.00	0.00	0.00	0.57	0.00	0.00	0.00	0.39	0.39
KNbO3	True	R3m	R3m		0.04	0.04	0.04	-0.00	0.00	-0.00	0.00	-0.00	0.00
LaAlO3	True	R3c	Imma	**	0.00	-0.00	0.00	0.00	0.00	-0.00	0.11	0.11	0.11
LaMnO3	True	I4/mcm	Pnma	†	-0.00	0.00	-0.00	0.00	-0.00	0.00	0.00	-0.00	0.32
NaNbO3	True	R3c	R3c		0.02	0.02	0.02	0.00	0.00	-0.00	0.33	0.33	0.33
NaTaO3	True	Pnma	Pnma		0.00	0.00	0.00	0.34	-0.00	0.00	0.00	0.31	0.31
PbTiO3	True	P4mm	P4mm		0.00	-0.00	-0.06	0.00	0.00	0.00	0.00	0.00	0.00
PbZrO3	True	R3c	Pbam	†	-0.08	-0.08	-0.08	0.00	-0.00	0.00	0.23	0.23	0.23
RbGeBr3	True	Pna2 ₁	P12 ₁ /m1	†	-0.00	-0.00	0.17	0.00	0.00	0.62	0.37	0.37	0.00
RbGeCl3	True	Pna2 ₁	P12 ₁ /m1	†	-0.00	0.00	0.14	-0.00	-0.00	0.59	0.36	0.36	-0.00
RbGeF3	True	R3c	NaN	*	0.00	0.00	0.00	0.00	0.00	-0.00	0.41	0.41	0.41
RbGeI3	False	Pmc2 ₁	P2 ₁ 2 ₁ 2 ₁	***	0.24	0.24	0.00	-0.00	0.00	0.65	0.41	0.41	0.00
RbPbBr3	True	Pnma	Cmcm	†	-0.00	0.00	-0.00	0.71	-0.00	-0.00	0.00	0.61	0.61
RbPbCl3	True	Pnma	Cmcm	†	0.00	0.00	0.00	0.62	-0.00	-0.00	0.00	0.56	0.56
RbPbF3	True	R3c	Pnma	**	0.03	0.03	0.03	0.00	0.00	0.00	0.39	0.39	0.39
RbPbI3	True	Pnma	Pnma		0.00	-0.00	-0.00	0.83	-0.00	-0.00	0.00	0.66	0.66
RbSnBr3	True	Pnma	Pnma		0.00	-0.00	0.00	0.65	-0.00	-0.00	0.00	0.51	0.51
RbSnCl3	True	Pna2 ₁	R3c	†	-0.00	-0.00	0.01	0.00	0.00	0.65	0.55	0.55	-0.00
RbSnF3	True	Cm	P12 ₁ /c1	***	0.05	0.02	0.03	-0.00	0.00	0.00	0.00	-0.00	0.00
RbSnI3	True	Pnma	Pnma		0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.58	0.58
SrHfO3	True	Pnma	Pnma		0.00	0.00	0.00	0.26	0.00	0.00	0.00	0.27	0.27
SrHfS3	True	Pnma	Pnma		0.00	0.00	0.00	0.53	0.00	-0.00	0.00	0.45	0.45
SrHfSe3	True	Pnma	NaN	*	0.00	-0.00	0.00	0.61	-0.00	-0.00	0.00	0.51	0.51
SrTiO3	True	I4/mcm	I4/mcm		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14
SrTiS3	True	Pna2 ₁	P6 ₃ /mmc	***	-0.00	0.00	0.14	0.00	-0.00	0.49	0.36	0.36	-0.00
SrTiSe3	True	Pna2 ₁	NaN	*	0.00	-0.00	0.15	0.00	0.00	0.58	0.42	0.42	-0.00
SrZrO3	True	Pnma	Pnma		0.00	0.00	0.00	0.30	-0.00	-0.00	0.00	0.31	0.31
SrZrS3	True	Pnma	Pnma		0.00	0.00	-0.00	0.55	0.00	0.00	0.00	0.48	0.48
SrZrSe3	True	Pnma	Pnma		-0.00	0.00	0.00	0.63	0.00	0.00	0.00	0.53	0.53

4.2 Foundational models

The training results of the models are presented in Tab. 4.2, with the model considered optimal marked by an asterisk. These are the models used for MD to determine phase transitions. Fig. 4.1 shows the absolute energy error compared to DFT results at 500 K and 1000 K for the optimal models in Tab. 4.2. In Fig. 4.2 and 4.3 the maximal absolute errors to DFT results are presented for forces and stresses, respectively. Included in these figures are the results of the general models MACE-OMAT-0 (supposed to give better results for phonons), MACE-MATPES-r2SCAN-0 (using a similar functional as this project) and MACE-MP-0 [29]. As different functionals were used in the training data of these models, the RMSE of the general models were minimized by finding the optimal energy offset. This was done to ensure fair comparisons between the models.

The MACE loss function is a weighted sum of the errors in energy, forces and stresses. We therefore evaluate the performance on these three metrics independently. Of these three metrics, the models trained in this project show the least improvement to accuracy of forces compared to the other foundational models, performing somewhat better at 500 K and marginally better at 1000 K.

Table 4.2: The training results of the models, as well as which was considered optimal and used in subsequent analysis.

Model name	Seed	Loss	RMSE energy [meV/atom]	RMSE forces [meV/Å]	Opt.
64x300K	4123	0.019	1.04	19.6	
	4124	0.022	1.64	18.7	
	4125	0.017	0.94	17.7	
	4126	0.020	0.83	20.3	*
	4127	0.031	1.40	22.1	
64x1000K	4123	0.044	2.24	25.5	
	4124	0.034	1.21	27.3	
	4125	0.010	0.68	14.9	*
	4126	0.022	1.76	18.7	
	4127	0.028	2.03	21.3	
128x1000K	4123	0.035	1.63	23.2	
	4124	0.038	1.68	28.3	
	4125	0.009	0.67	13.8	*
	4126	0.017	1.24	17.0	
	4127	0.025	1.84	19.8	

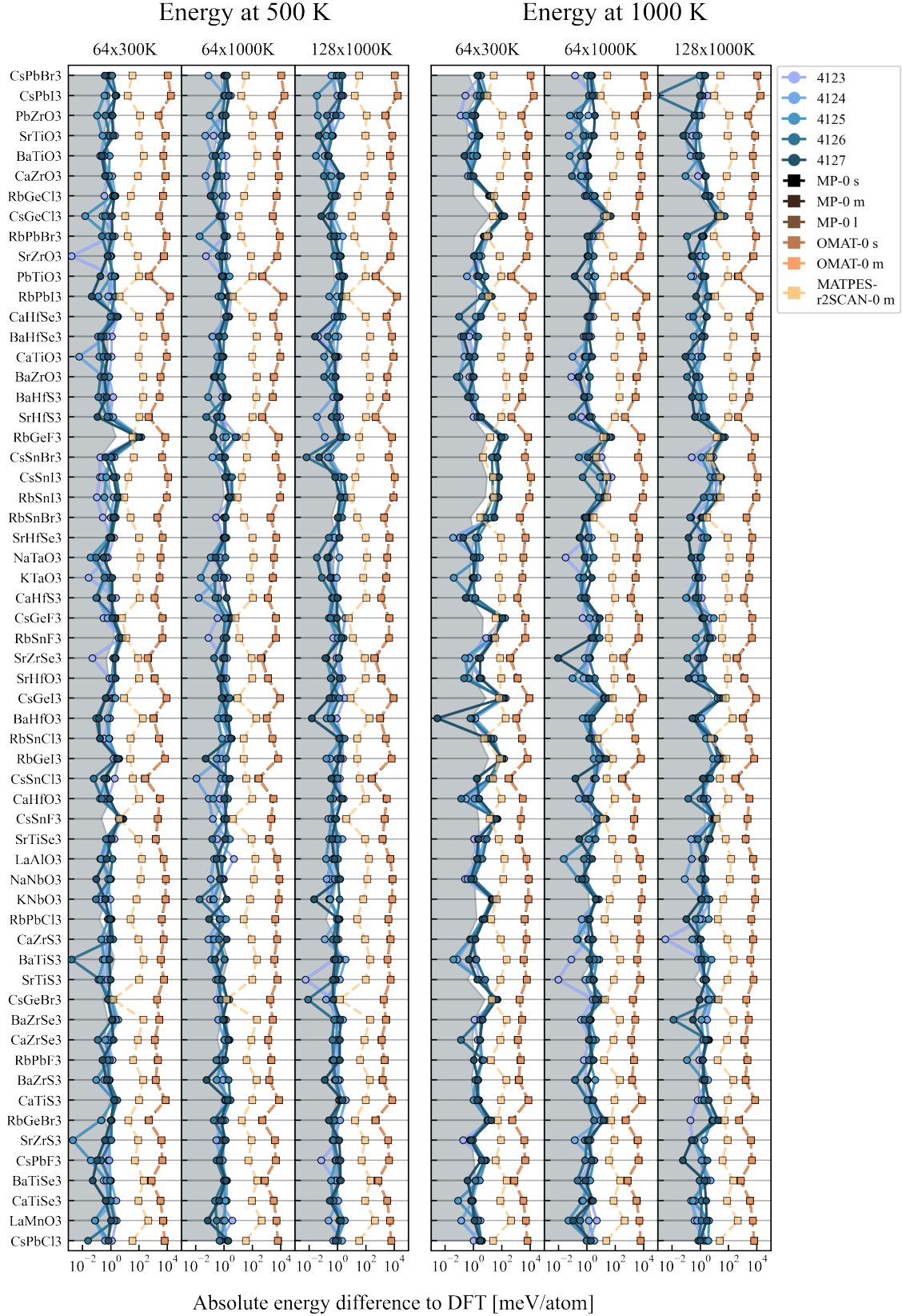


Figure 4.1: The absolute absolute difference from DFT for the models trained in this project and already existing general foundational models (s,m,l denoting small, medium or large models). Shown is the result from MD snapshots at 500 K and 1000 K for all structures. The shaded region shows the estimated model error.

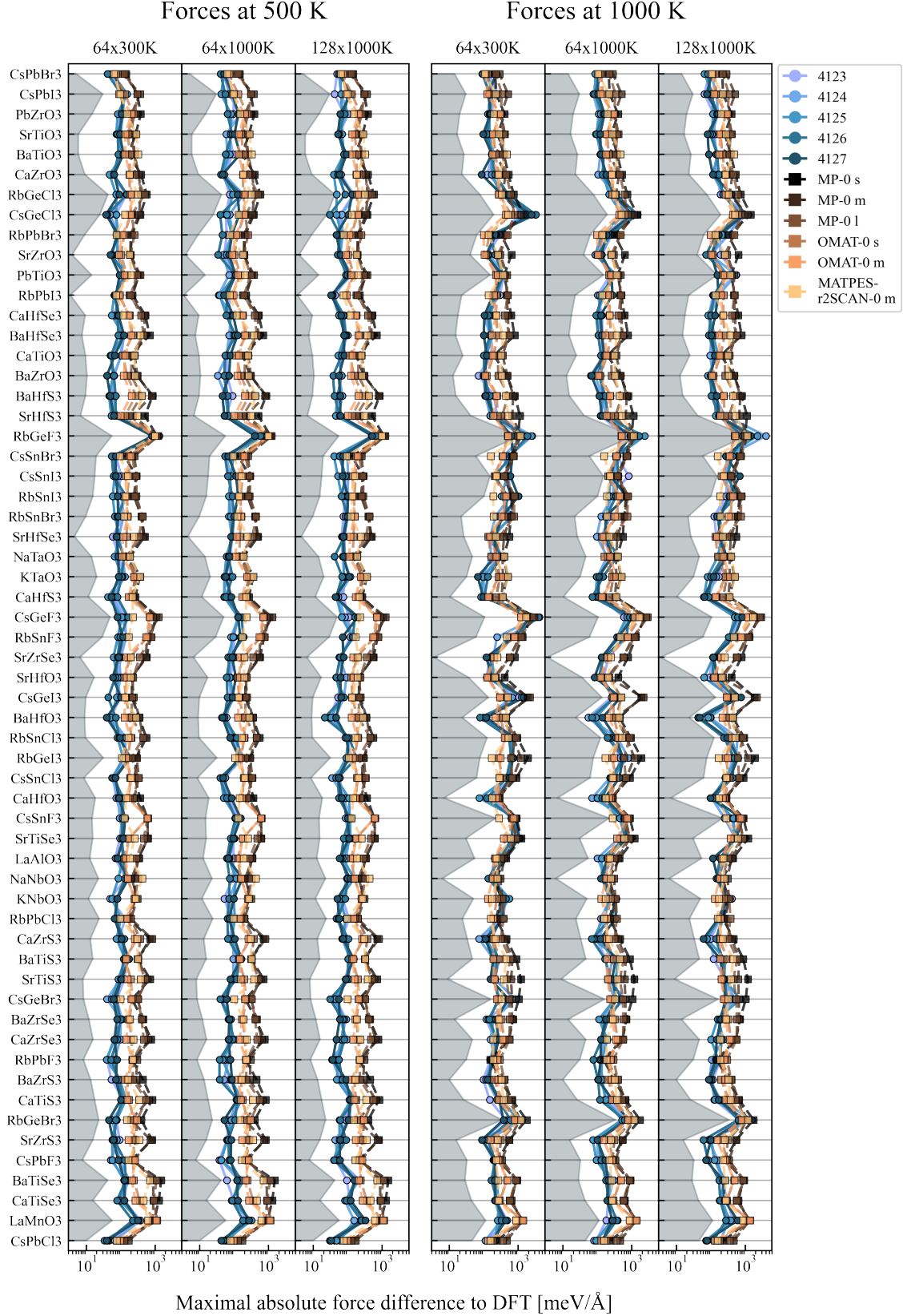


Figure 4.2: The maximal absolute force difference from DFT for the models trained in this project and already existing general foundational models (s,m,l denoting small, medium or large models). The model seeds are named in the legend. Shown is the result from MD snapshots at 500 K and 1000 K for all structures. The shaded region shows the estimated model error.

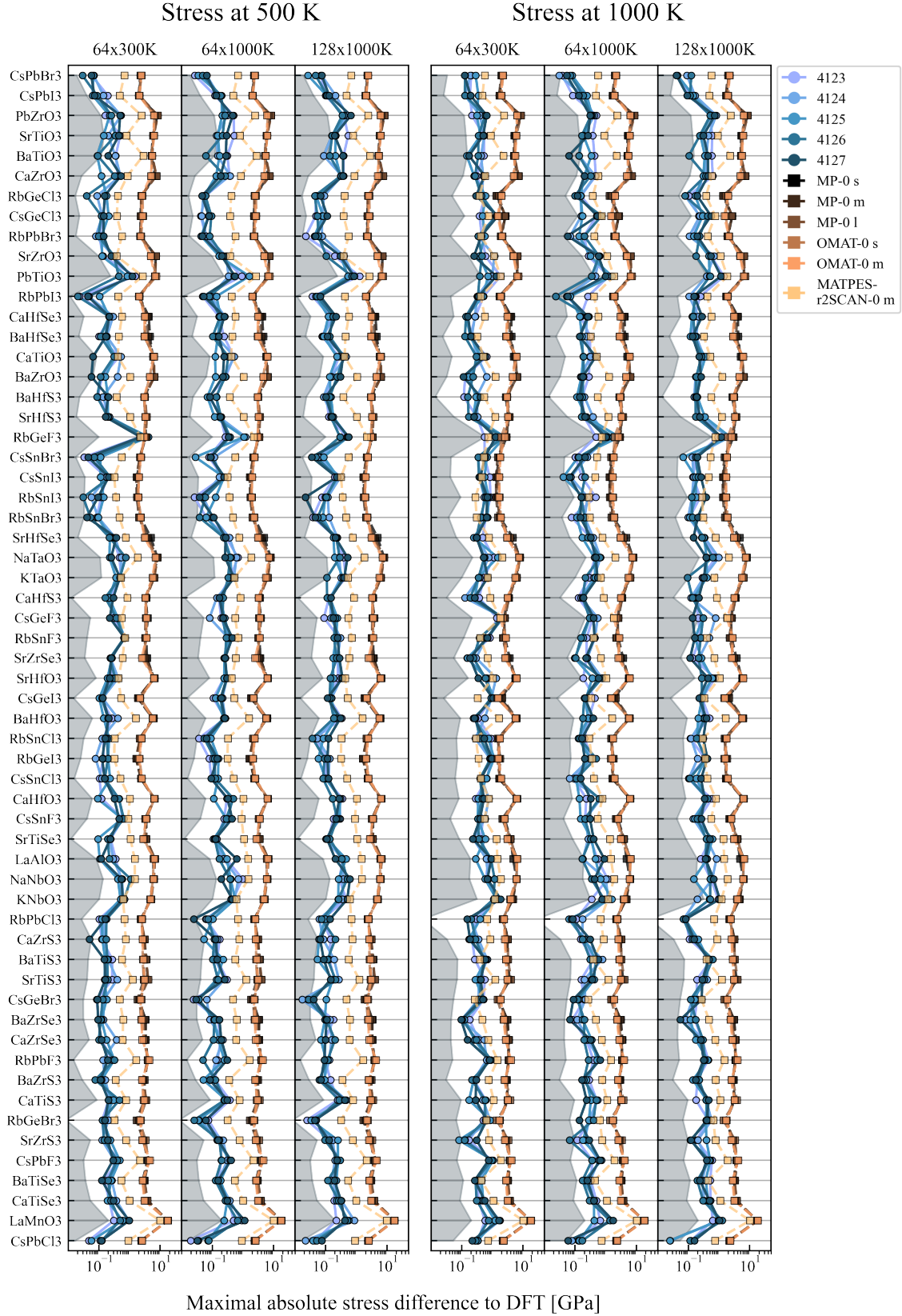


Figure 4.3: The maximal absolute stress difference from DFT for the models trained in this project and already existing general foundational models (s,m,l denoting small, medium or large models). The model seeds are named in the legend. Shown is the result from MD snapshots at 500 K and 1000 K for all structures. The shaded region shows the estimated model error.

4.2.1 Model 64x300K

From Fig. 4.1 and 4.3 it can be seen that model 64x300K generally performs better than MACE-MATPES-r2SCAN-0, and always performs better than the MACE-MP0 and MACE-OMAT-0 models. MACE-MATPES-r2SCAN-0 performs better for some structures, most notably for multiple halides at 1000 K. Model 64x300K performs significantly worse at 1000 K compared to 500 K, especially when considering the forces in Fig. 4.2. Note also the significantly larger estimated model errors. Moreover, it seems to struggle with predicting stresses, which can be seen in the parity plot in Fig. 4.4, where compositions with Ba and Ti perform the worst.

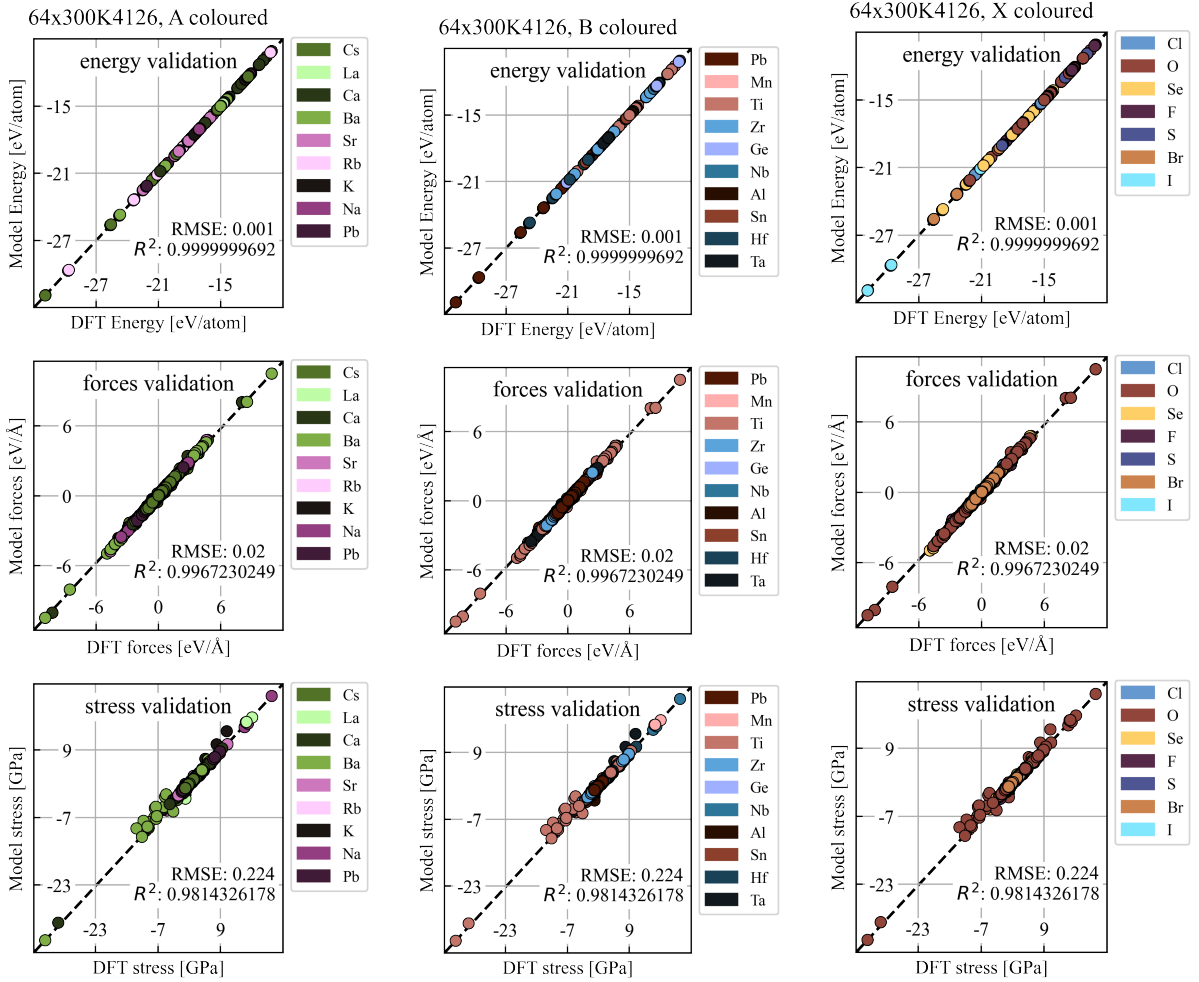


Figure 4.4: Parity plots from model 64x300K for the energy, forces and stresses on the validation data sets. From left to right, the points are coloured according to A-cation, B-cation and X-anion species.

4.2.2 Model 64x1000K

Because of the shortcomings of model 64x300K, a new model with data at 500 K and 1000 K was trained. In addition, the second stage stress weights were changed from 1 to 100. This model, 64x1000K, has higher accuracy for energy, force and stress, as seen in Fig. 4.1, 4.2 and 4.3. The largest improvement are seen in the halide energies at 1000 K. The model seems to be able to predict stresses more accurately according to Fig. 4.5 and the error compared to DFT results, while the estimated model uncertainty is largely unaffected.

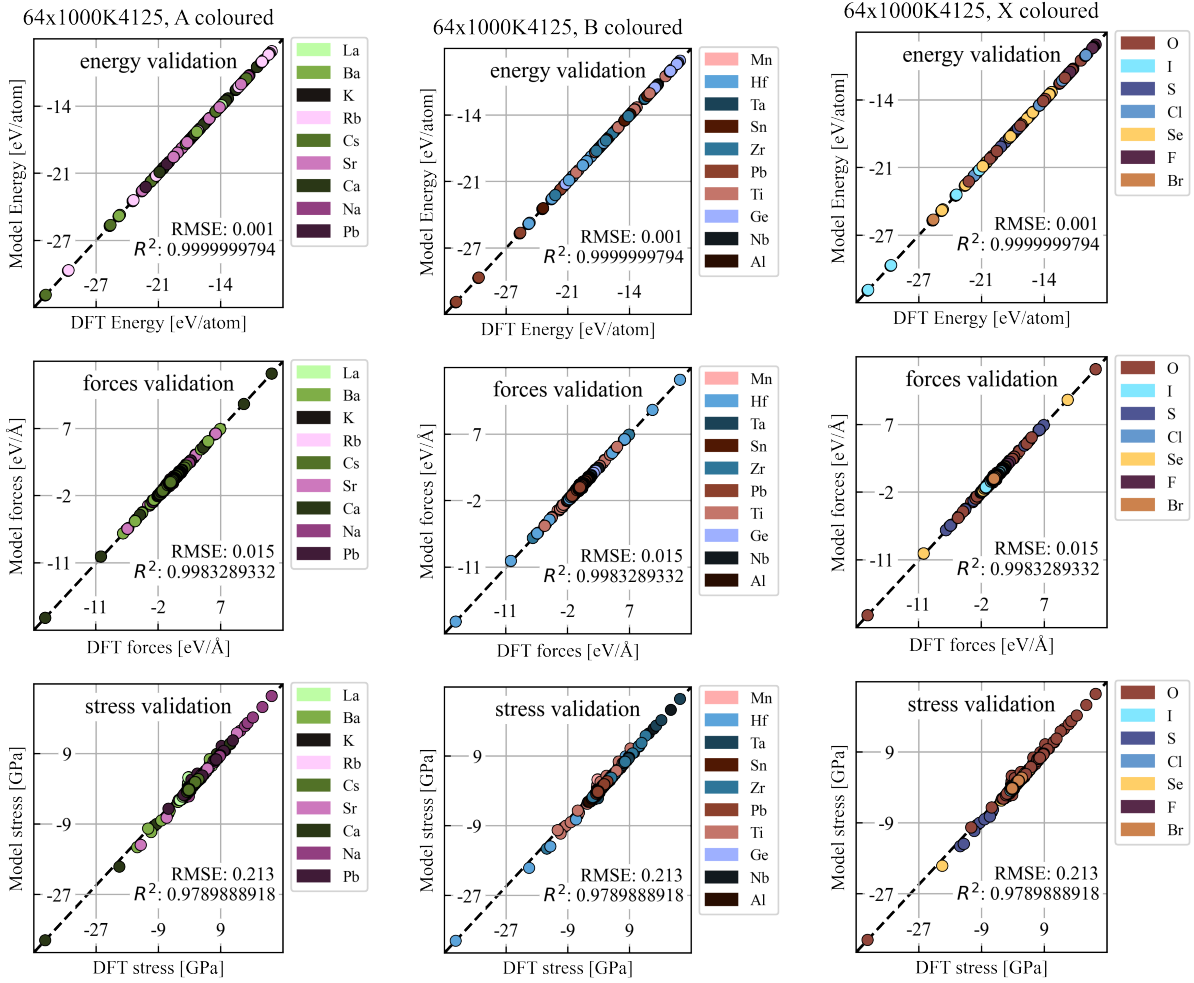


Figure 4.5: Parity plots from model 64x1000K for the energy, forces and stresses on the validation data sets. From left to right, the points are coloured according to A-cation, B-cation and X-anion species.

4.2.3 Model 128x1000K

The previous model shows that a small addition of data improves the model significantly for some compositions. Another path is to change the model parameters, such as increasing the number of hidden irreps. This is the motivation for model 128x1000K, which is identical to model 64x1000K except for this increased size. Fig. 4.1, 4.2 and 4.3 shows that it is able to reduce errors somewhat in certain compositions, but does seem to have a large effect. In general the errors are reduced, but the difference might not motivate the increased computational cost of the larger model. The parity plots in Fig. 4.6 show similar results to the 64x1000K model.

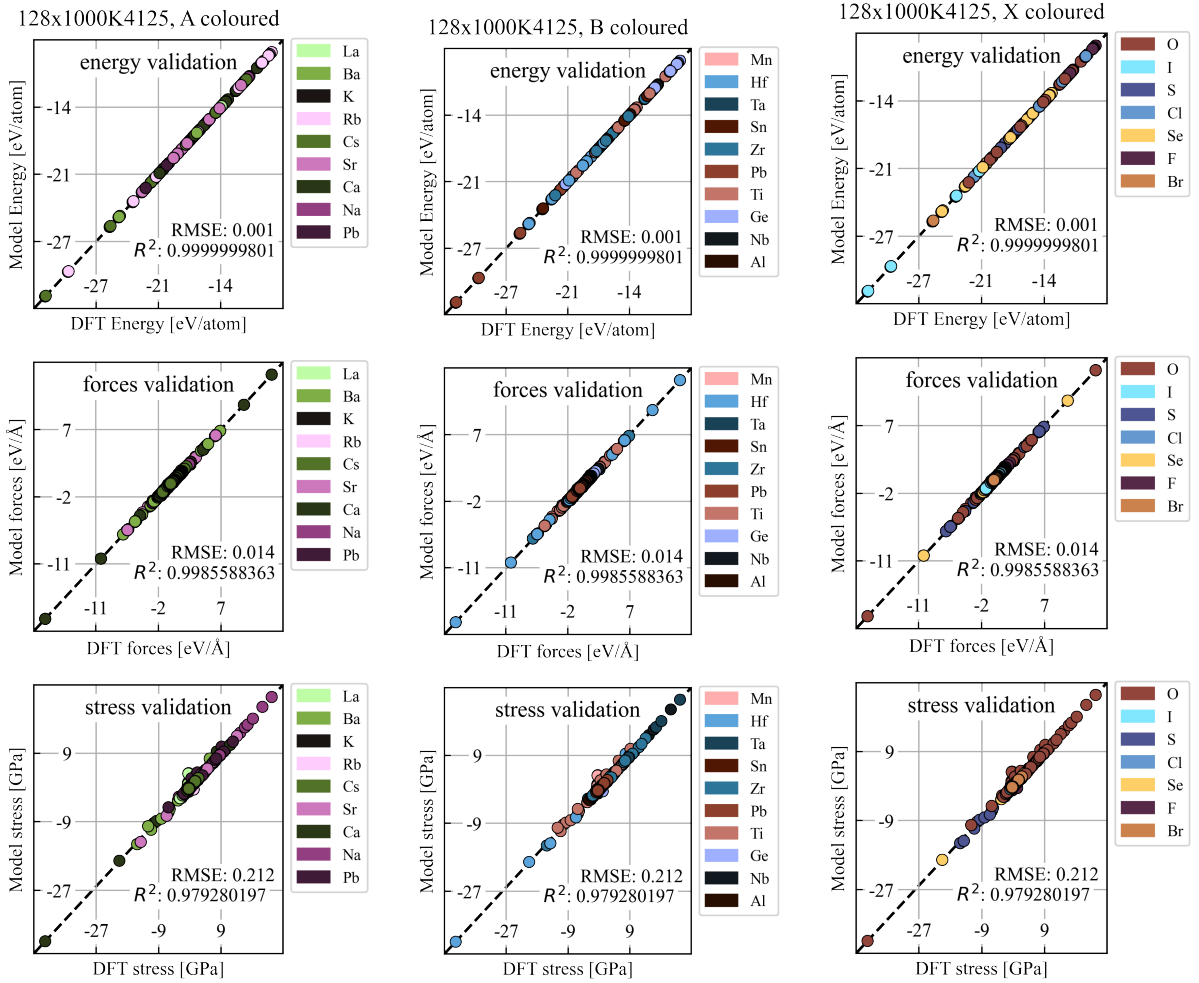


Figure 4.6: Parity plots from model 128x1000K for the energy, forces and stresses on the validation data sets. From left to right, the points are coloured according to A-cation, B-cation and X-anion species.

4.3 Phases

To compare with experimental results the structural phases of BaTiO_3 , BaZrO_3 , BaZrS_3 , CsGeBr_3 , CsPbI_3 , NaNbO_3 and SrTiO_3 were determined. The 64x1000K model was chosen for this, as it displays similar performance to 128x1000K, while being less computationally costly due to the smaller number of parameters. For each structure all non-zero 120 high symmetry modes are shown. Please use Fig. 2.1, 2.2, and 2.3 to understand how the modes affect the structure.

4.3.1 BaTiO_3

Here are the results for BaTiO_3 , where Fig. 4.7 shows the lattice parameters and non-zero mode amplitudes for MD simulations with the 64x1000K model. The structures start off with Ti and O displacements, which subsequently go to zero between 100 K and 300 K. In Fig. 4.8 these results are compared with experimental results [44]. The model agrees well with experiment for low temperatures, but drifts as the temperature increases. The final phase transition to the cubic phase is off by 100 K.

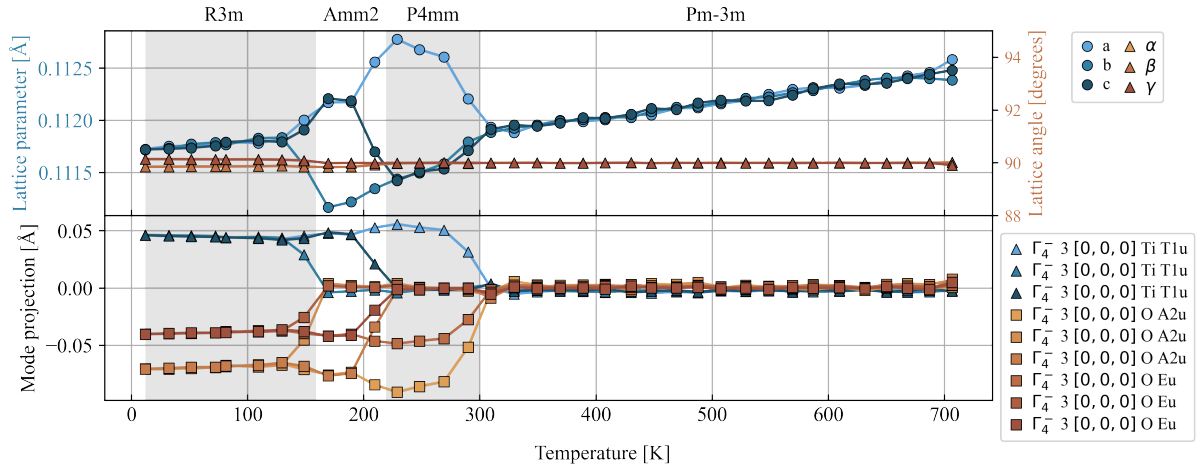


Figure 4.7: Phase diagram for BaTiO_3 from 0 K to 700 K. The top plot shows the cell lengths and angles, while the bottom plot shows all non-zero mode amplitudes. Four distinct phases are distinguished, corresponding to space groups $R3m$, $Amm2$, $P4mm$ and $Pm\bar{3}m$.

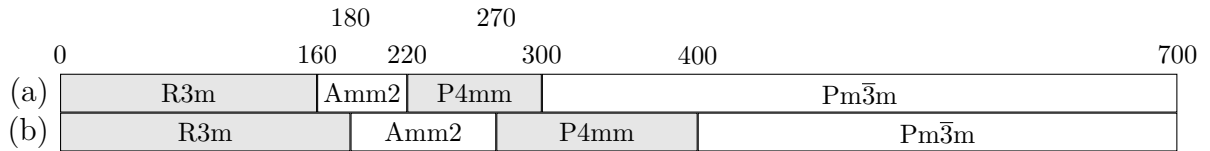


Figure 4.8: The phase transitions of BaTiO_3 from the 64x1000K model (a) compared to experimental results (b) [44]. Shows good initial agreement to experiments, but drifts at higher temperature ending in a 100 K difference at the transition to the cubic phase.

4.3.2 BaZrO₃

The lattice parameters during the MD simulation are shown in Fig. 4.9. The structure starts in the highest symmetry state with space group $\text{Pm}\bar{3}m$, and does not transition during the entire heating run. The modes are thus also all zero. This agrees with experimental results [45], as shown in Fig.4.10.

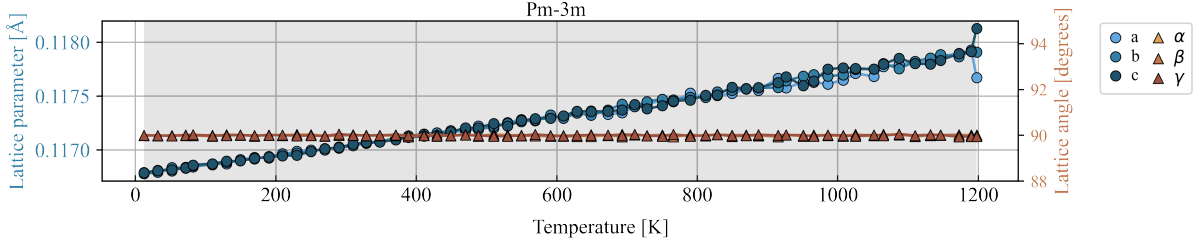


Figure 4.9: Phase diagram of BaZrS₃, showing the lattice parameters change as a function of temperature. The structure is cubic through the whole process, and all mode projections are zero.

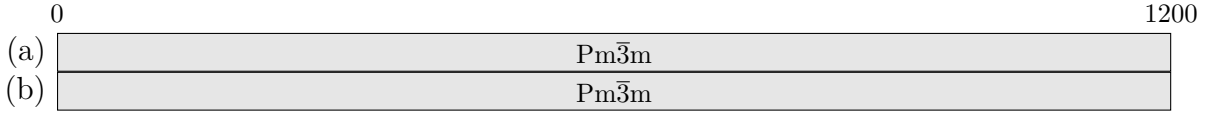


Figure 4.10: Shows the complete agreement on the single phase structure BaZrO₃ between this project (a) and experimental results (b) [45]

4.3.3 BaZrS₃

During heating of BaZrS₃, multiple mode projections are non-zero, as can be seen in Fig. 4.11. At 0 K the out-of-phase and in-phase octahedral tilting is active, along with anti-polar shifts (X modes) of the Ba-cations. At the transition around 650 K, all these go to zero, except the out-of-phase tilting which increases in magnitude. This transition agrees well with previous studies [46], as can be seen in Fig. 4.12. The model fails to capture the transition to the cubic phase at 900 K. The MD simulation was extended to 1200 K in an attempt to find it, but the MD simulation became unstable above 1000 K, possibly due to melting.

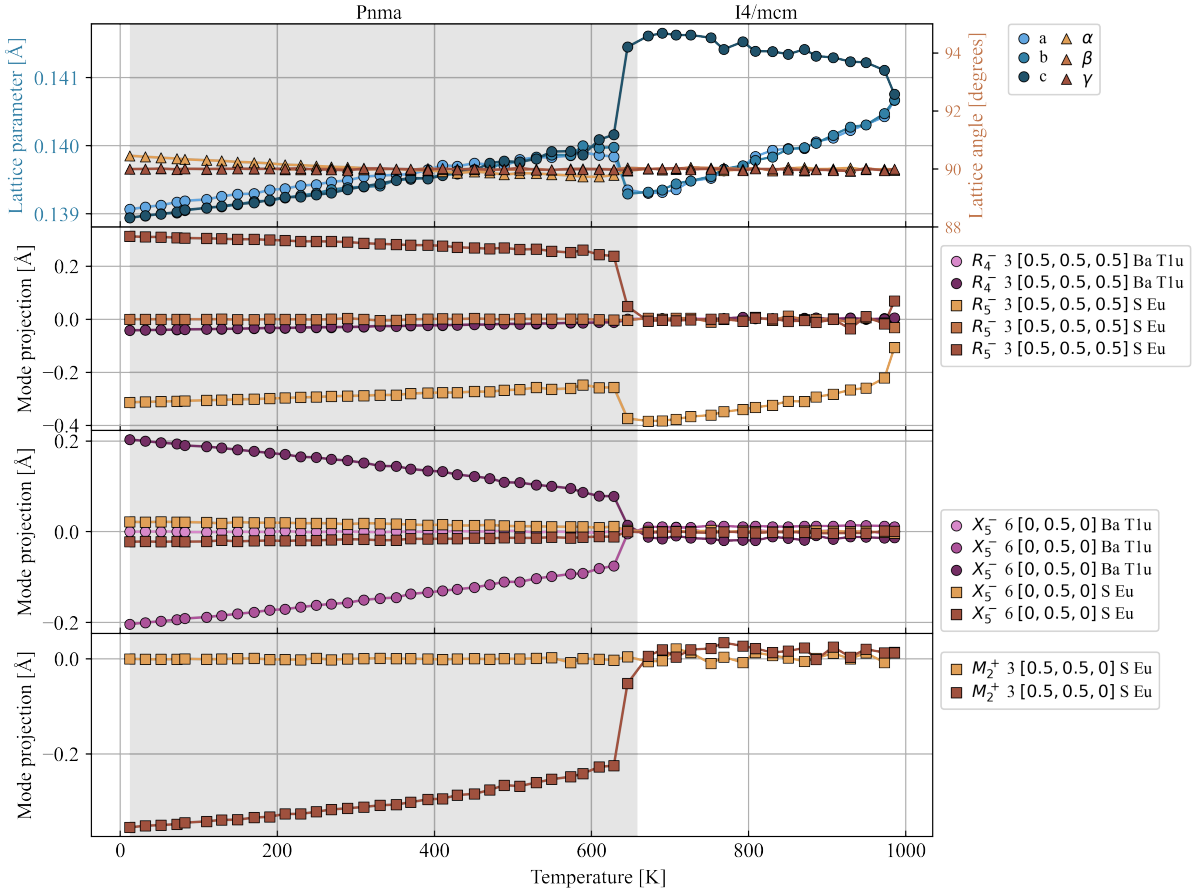


Figure 4.11: Phase diagram of BaZrS₃ showing multiple active out-of-phase, in-phase and anti-polar modes being active below 650 K, with only the out-of-phase mode remaining after the transition.

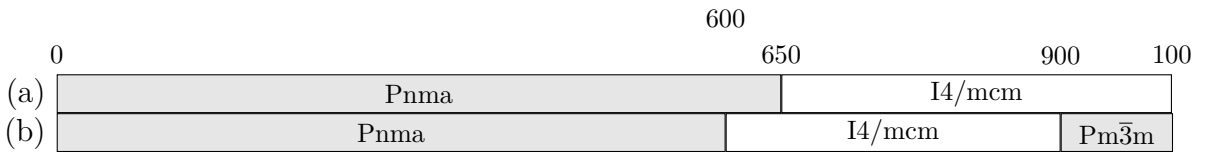


Figure 4.12: Comparison between 64x1000K (a) and previous computational results (b) [46] show good agreement for the first phase transition, but 64x1000K fails to find the second transition at 900 K.

4.3.4 CsGeBr₃

The heating of CsGeBr₃ shown in Fig. 4.13 displays one phase transition from the trigonal R3m to the cubic Pm $\bar{3}$ m symmetry. This phase agrees with experiments, but the transition temperature differs by about 80 K, as seen in Fig. 4.14

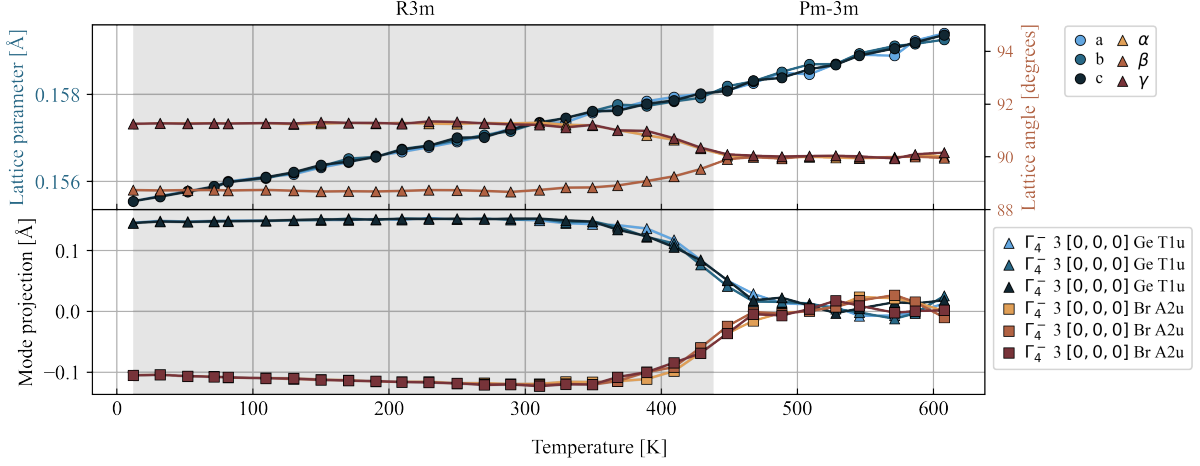


Figure 4.13: Phase diagram of CsGeBr₃, showing one transition at around 430 K, where the displacement of Ge goes to zero.

	0	430	510	600
(a)	R3m		Pm $\bar{3}$ m	
(b)	R3m			Pm $\bar{3}$ m

Figure 4.14: The phases determined in this project (a) agree well with the experimentally determined phases (b) [47], but the transition temperature is shifted by about 80 K

4.3.5 CsPbI₃

Up until 500 K, CsPbI₃ has active in-phase, out-of-phase and anti-polar modes, as shown in Fig. 4.15. Above 610 K they all go to zero and arrive at the cubic symmetry. The symmetry of the region between 500 K and 600 K is harder to determine. There seems to be some in-phase octahedral tilting still present, indicating an intermittent symmetry. If this is considered as a separate phase is dependent on where the transition is drawn and on the tolerances used in determining the symmetry. Shifting the transition to around 560 K, one can argue that the mode goes to zero at the same temperature as the others, and that the remaining mode amplitude is simply noise. This is further complicated by the comparison to experimental results in Fig. 4.16, where the intermittent phase is present during cooling, but not during heating [48][49].

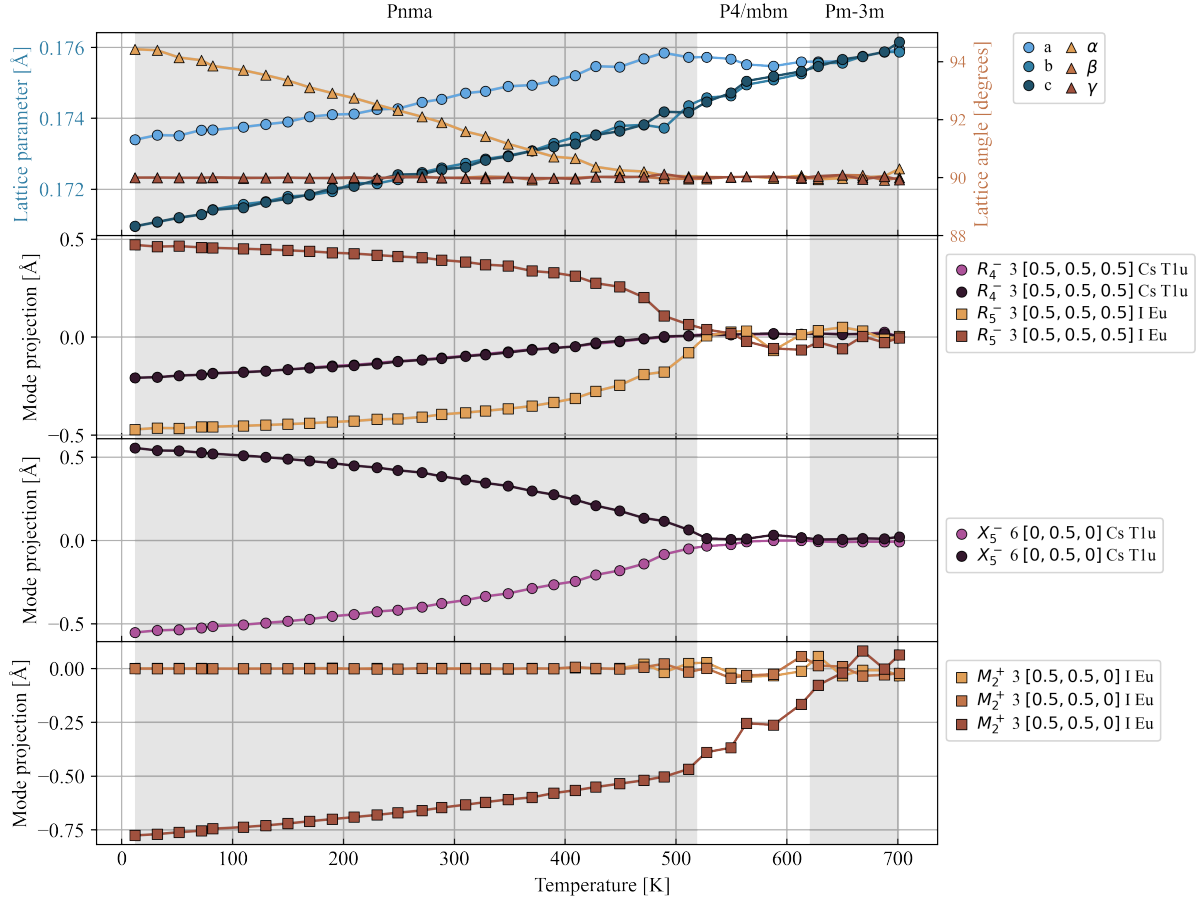


Figure 4.15: Phase diagram of CsPbI_3 , showing two or three phases depending on the precise drawing of the transition lines and the tolerances used.

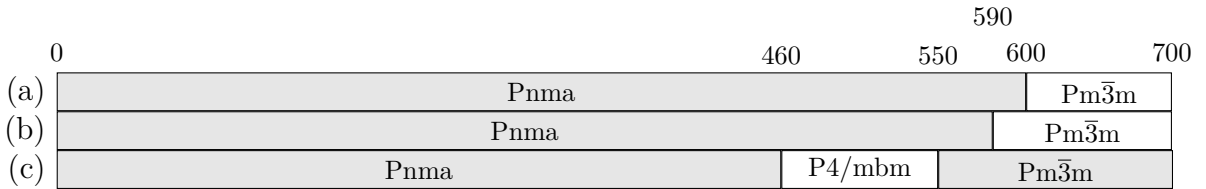


Figure 4.16: Comparison of the phases determined by the 64x1000K model (a), experimental heating runs (b) [49][48] and experimental cooling runs (c) [49][48]. The intermittent phase can be argued to be or not to be present in the heating performed by the foundational model.

4.3.6 SrTiO₃

SrTiO₃ shows one phase transition at low temperatures (around 100 K), which the model is able to recognize. Fig. 4.17 shows that this transition takes place around 140 K, which differs by about 40 K from experiments [50], as shown in Fig. 4.18

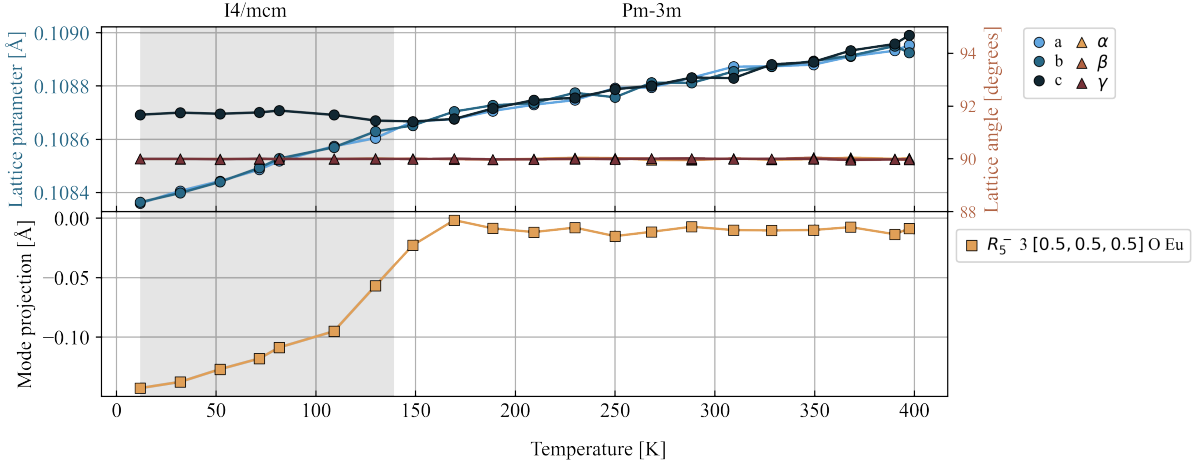


Figure 4.17: Phase diagram of SrTiO₃ showing the single phase transition below room temperature.

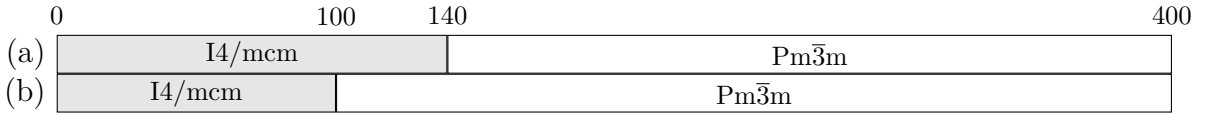


Figure 4.18: Comparisons between the 64x1000K model (a) and experimental results (b) [50] show agreement on the phases with about 40 K difference.

4.3.7 NaNbO₃

NaNbO₃ is known to have many complex phase transitions. The model is able to recognize that some phase transitions take place, as shown in Fig. 4.19. Six phases seem to be distinguishable by visual inspection. All except the first one are dubious. The transitions were set by hand, unlike previous transitions (except CsPbI₃) that were determined by finding peaks in the derivative of the mode amplitudes. These transition points were mostly based on the in-phase tilting distortions, and verified by the anti-polar displacements as well as kinks in the out-of-phase mode amplitudes. The noisy data makes it hard to determine the symmetries, and as such all uncertain phases have been set as P1, the lowest possible symmetry state. One clear phase transition appears at ~620 K, where the structure transitions away from the R3c symmetry. The complexity of the phases are apparent from experimental results [51], and it is clear that the model cannot determine the many phases.

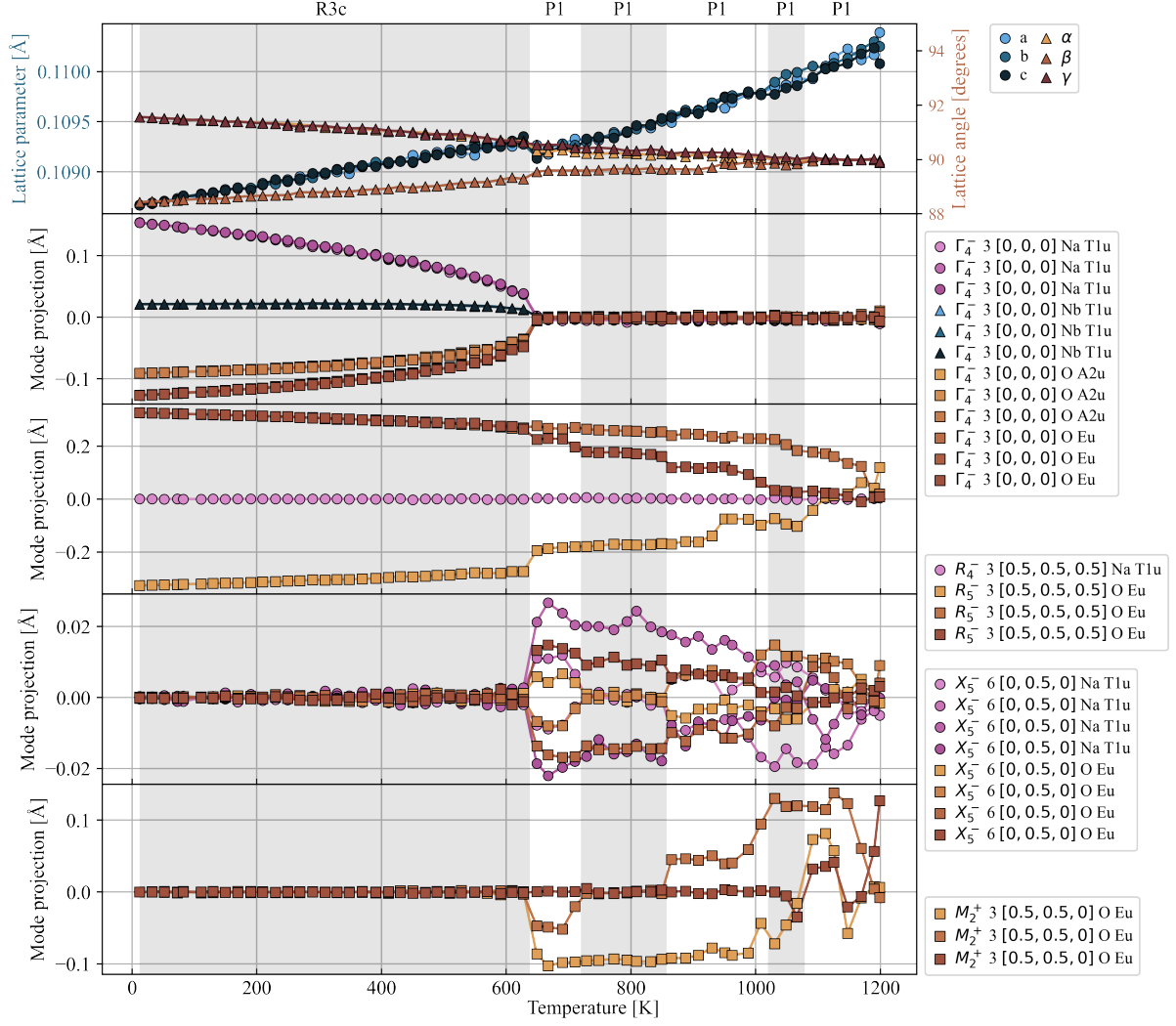


Figure 4.19: Phase diagram of NaNbO_3 showing one clear phase transition at around 620 K and complex behaviour above this, where some phases might be distinguishable.

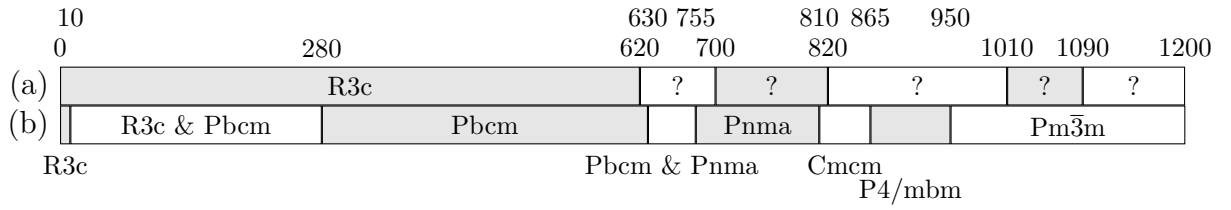


Figure 4.20: Comparison between the 64x1000K model (a) and experimental results (b) [51] for NaNbO_3 . The model is unable to capture the complex dynamics.

4.4 Improvements

There are multiple methods that can be employed to improve on the models. It is always possible to add more data, specifically targeting structures where the model is uncertain. The choices of such structures may be guided by the uncertainty estimation in Fig. 4.1, 4.2 and 4.3, shown as the shaded region. We see that the error generally follows the trend of the absolute difference to DFT data, which indicates that the estimated error captures actual errors well. Adding larger structures would also improve upon the model, as this would decrease the impact of finite size effects present in the MD trajectories. This is a heavy computational burden, and 320 atoms was deemed the upper atom count limit for this project. It is possible to add structures such as $4 \times 2 \times 2$ and $4 \times 4 \times 2$ as a compromise to decrease computational costs.

One can also employ correlation and information entropy based methods to maximize the diversity of the training set, before submitting structures to DFT [52]. This can decrease computational costs as only structures which contribute maximally to the training set are evaluated. The same amount of computation would then create a more diverse training set, which can improve model accuracy.

It is also possible to tweak the model parameters. Increasing the size of the model to 256 hidden irreps. is one alternative, but this will also increase training and evaluation costs. As the step from 64 to 128 only showed significant improvement for certain outlying structures, it can be assumed that the 256 hidden irreps. would not generally increase accuracy by much either. There are, of course, multiple other model parameters that can be tweaked.

Another option is to fine-tune the already existing general foundational models by iterating the training with perovskite structures. This has the benefit of using the general models' vast space of compositions while maintaining focus on accurate results for perovskites. This would also increase the transferability of the model, as elements not included in the perovskite data would still be represented in the total dataset.

To see improved phase diagrams, slower MD-heating can be employed, which would give sharper transition points and might help to account for the phases of NaNbO_3 .

4.5 Future work

With this foundational model in place, there is much work that can be built on top of it. One option is to include more perovskite compositions in the training data. It is also possible to create foundational models for perovskite derivatives. For example a double perovskite foundational model, or an organic-inorganic perovskite foundational model. The package pfox is built to be modular, with all perovskite commonality built into a parent class, and where specific perovskite derivatives can be built as child classes. Throughout this project, the standard perovskite class has been implemented.

The model can be used to generate data much in the same way that DFT was used in this project. The MACE model can quickly output large sets of accurate data, which can be

used to train models specialized for specific materials. By using the NEP architecture, the required computational resources needed to perform MD are reduced, and more in-depth analysis can therefore be carried out. Examples include determining a two-dimensional phase diagram, where both pressure and temperature are allowed to vary, or running heating and cooling to determine effects of hysteresis.

Finally, using these models a high-throughput study can be performed determining the properties of many perovskites, and adding these results to the database. With such a database in place, an inverse design model could potentially be trained, which takes in desired properties and returns the most probable candidate.

5

Conclusion

This project set out to develop a foundational machine-learning model for perovskites. The first step entailed generating structures and determining the ground states using DFT calculations. The ground states generally agree well with previous results (see Tab. 4.1), with a few errors that cannot be explained by small energy differences or limitations of the project. As such, the data was considered accurate enough to carry on with the project.

MD simulations were run to get approximate structures at finite temperatures, which were then passed to DFT for accurate predictions. The auxiliary goal of creating a perovskite database with accurate DFT calculations was completed, and it is available on GitLab.

MACE models were then trained on this database, where the three most promising, named 64x300K, 64x1000K and 128x1000K, show more accurate results than the already existing foundational MACE models. The conclusion was therefore drawn that models trained in this project are improvements upon these, and can be said to produce more accurate results for perovskites.

Because of the balance between speed and accuracy, the 64x1000K model was chosen to be used in comparisons to experimental and previous computational results. The metric used was the structural phase transitions of seven perovskites. The model was able to predict all phases correctly within 100 K, except for BaZrS_3 and NaNbO_3 . For the former, the model fails to find the cubic phase at about 900 K, and in the latter it fails to account for the multiple fast phase transitions above 600 K, although it is able to predict that some complex dynamics do take place.

The goal of creating a foundational model that is fast and accurate, which can serve as the starting point and guide for subsequent research, has thus been accomplished. There are multiple ways to improve and iterate upon this model, including considering more data and chemical compositions, exploring fine-tuning alternatives, constructing foundational models for other perovskite derivatives, and distilling the model to faster architectures allowing more exhaustive analyses of materials.

Bibliography

- [1] Quinten A Akkerman and Liberato Manna. “What defines a halide perovskite?” In: *ACS Energy Letters* 5.2 (2020), pp. 604–610. DOI: 10.1021/acsenergylett.0c00039.
- [2] T He et al. “Superconductivity in the non-oxide perovskite MgCNi_3 ”. In: *Nature* 411.6833 (2001), pp. 54–56. DOI: 10.1038/35075014.
- [3] R E J Cava et al. “Bulk superconductivity at 91 K in single-phase oxygen-deficient perovskite $\text{Ba}_2\text{YCu}_3\text{O}_{9-\delta}$ ”. In: *Physical Review Letters* 58.16 (1987), p. 1676. DOI: 10.1103/PhysRevLett.58.1676.
- [4] Yaoguang Rong et al. “Challenges for commercializing perovskite solar cells”. In: *Science* 361.6408 (2018), eaat8235. DOI: 10.1126/science.aat8235.
- [5] Kebin Lin et al. “Perovskite light-emitting diodes with external quantum efficiency exceeding 20 per cent”. In: *Nature* 562.7726 (2018), pp. 245–248. DOI: 10.1038/s41586-018-0575-3.
- [6] David S Sholl and Janice A Steckel. *Density functional theory: a practical introduction*. John Wiley & Sons, 2009. DOI: 10.1002/9780470447710.
- [7] *Mindat.org, Perovskite*. Accessed 23-04-2026. URL: <https://www.mindat.org/min-3166.html#autoanchor26>.
- [8] Harold T Stokes et al. “Group-theoretical analysis of octahedral tilting in ferroelectric perovskites”. In: *Acta Crystallographica Section B: Structural Science* 58.6 (2002), pp. 934–938. DOI: 10.1107/S0108768102015756.
- [9] G H Kwei et al. “Structures of the ferroelectric phases of barium titanate”. In: *The Journal of Physical Chemistry* 97.10 (1993), pp. 2368–2377. DOI: 10.1021/j100112a043.
- [10] *Best Research-Cell Efficiency Chart*. Accessed 23-04-2026. URL: <https://www.nrlr.gov/pv/cell-efficiency>.
- [11] Ralf Drautz. “Atomic cluster expansion for accurate and transferable interatomic potentials”. In: *Physical Review B* 99.1 (2019), p. 014104. DOI: 10.1103/PhysRevB.99.014104.
- [12] Ilyes Batatia et al. “MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022. DOI: 10.48550/arXiv.2206.07697. arXiv: 2206.07697.
- [13] Zheyong Fan et al. “Neuroevolution machine learning potentials: Combining high accuracy and low cost in atomistic simulations and application to heat transport”. In: *Physical Review B* 104.10 (2021), p. 104309. DOI: 10.1103/PhysRevB.104.104309.
- [14] Wentao Zhang et al. “Constructing machine learning interatomic potentials with minimum amount of ab initio data”. In: *npj Computational Materials* 12 (2026), p. 174. DOI: 10.1038/s41524-026-02023-y.

- [15] Philip Hofmann. *Solid state physics: an introduction*. John Wiley & Sons, 2022.
- [16] Richard M Martin. *Electronic structure: basic theory and practical methods*. Cambridge University Press, 2020. DOI: 10.1017/9781108555586.
- [17] Wolfgang Nolting and William D Brewer. *Fundamentals of many-body physics*. Vol. 3. Springer, 2009. DOI: 10.1007/978-3-540-71931-1.
- [18] VASP. *Category: Exchange-correlation functionals*. 2026.
- [19] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.
- [20] John R Durbin. *Modern algebra: An introduction*. John Wiley & Sons, 2008.
- [21] Jean-Pierre Serre et al. *Linear representations of finite groups*. Vol. 42. Springer, 1977. DOI: 10.1007/978-1-4684-9458-7.
- [22] Anthony M Glazer. “The classification of tilted octahedra in perovskites”. In: *Acta Crystallographica Section B: Structural Science* 28.11 (1972), pp. 3384–3392. DOI: 10.1107/S0567740872007976.
- [23] Michael Glazer, Gerald Burns, and Alexander N Glazer. *Space groups for solid state scientists*. Elsevier, 2012.
- [24] G Venkataraman. “Soft modes and structural phase transitions”. In: *Bulletin of Materials Science* 1.3 (1979), pp. 129–170. DOI: 10.1007/BF02743964.
- [25] Michael Tinkham. *Group theory and quantum mechanics*. Courier Corporation, 2003.
- [26] D. M. Hatch H. T. Stokes and B. J. Campbell. *ISODISTORT, ISOTROPY Software suite*. URL: iso.byu.edu.
- [27] Justin Gilmer et al. “Neural message passing for quantum chemistry”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1263–1272. DOI: 10.48550/arXiv.1704.01212. arXiv: 1704.01212.
- [28] Ask Hjorth Larsen et al. “The atomic simulation environment—a Python library for working with atoms”. In: *Journal of Physics: Condensed Matter* 29.27 (2017), p. 273002. DOI: 10.1088/1361-648X/aa680e.
- [29] Ilyes Batatia et al. “A foundation model for atomistic materials chemistry”. In: *The Journal of Chemical Physics* 163.18 (2025), p. 184110. DOI: 10.1063/5.0297006.
- [30] Georg Kresse and Jürgen Hafner. “Ab initio molecular dynamics for liquid metals”. In: *Physical Review B* 47.1 (1993), p. 558. DOI: 10.1103/PhysRevB.47.558.
- [31] 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: 10.1016/0927-0256(96)00008-0.
- [32] Georg Kresse and Jürgen Furthmüller. “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”. In: *Physical Review B* 54.16 (1996), p. 11169. DOI: 10.1103/PhysRevB.54.11169.
- [33] VASP. *Category: Van der Waals functionals*. 2026.
- [34] James W Furness et al. “Accurate and numerically efficient r2SCAN meta-generalized gradient approximation”. In: *The Journal of Physical Chemistry Letters* 11.19 (2020), pp. 8208–8215. DOI: 10.1021/acs.jpclett.0c02405.
- [35] Albert P Bartók and Jonathan R Yates. “Regularized SCAN functional”. In: *The Journal of Chemical Physics* 150.16 (2019), p. 161101. DOI: 10.1063/1.5094646.
- [36] Jianwei Sun, Adrienn Ruzsinszky, and John P Perdew. “Strongly constrained and appropriately normed semilocal density functional”. In: *Physical Review Letters* 115.3 (2015), p. 036402. DOI: 10.1103/PhysRevLett.115.036402.

-
- [37] Riccardo Sabatini, Tommaso Gorni, and Stefano De Gironcoli. “Nonlocal van der Waals density functional made simple and efficient”. In: *Physical Review B* 87.4 (2013), p. 041108. DOI: 10.1103/PhysRevB.87.041108.
- [38] Oleg A Vydrov and Troy Van Voorhis. “Nonlocal van der Waals density functional: The simpler the better”. In: *The Journal of Chemical Physics* 133.24 (2010), p. 244103. DOI: 10.1063/1.3521275.
- [39] Jinliang Ning et al. “Workhorse minimally empirical dispersion-corrected density functional with tests for weakly bound systems: r2SCAN+rVV10”. In: *Physical Review B* 106.7 (2022), p. 075422. DOI: 10.1103/PhysRevB.106.075422.
- [40] VASP. *Structure optimization*. 2026.
- [41] A. P. Thompson et al. “LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales”. In: *Computer Physics Communications* 271 (2022), p. 108171. DOI: 10.1016/j.cpc.2021.108171.
- [42] Anders Johansson et al. “LAMMPS-KOKKOS: Performance Portable Molecular Dynamics Across Exascale Architectures”. In: *Proceedings of the SC’25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2025, pp. 1217–1232. DOI: 10.1145/3731599.3767498.
- [43] Anubhav Jain et al. “The Materials Project: A materials genome approach to accelerating materials innovation”. In: *APL Materials* 1.1 (2013), p. 011002. DOI: 10.1063/1.4812323.
- [44] Matias Acosta et al. “BaTiO₃-based piezoelectrics: Fundamentals, current status, and perspectives”. In: *Applied Physics Reviews* 4.4 (2017), p. 041305. DOI: 10.1063/1.4990046.
- [45] Adrien Perrichon et al. “Unraveling the ground-state structure of BaZrO₃ by neutron scattering experiments and first-principles calculations”. In: *Chemistry of Materials* 32.7 (2020), pp. 2824–2835. DOI: 10.1021/acs.chemmater.9b04437.
- [46] Prakriti Kayastha et al. “Octahedral tilt-driven phase transitions in BaZrS₃ chalcogenide perovskite”. In: *The Journal of Physical Chemistry Letters* 16.8 (2025), pp. 2064–2071. DOI: 10.1021/acs.jpcllett.4c03517.
- [47] Gerhard Thiele, Heinz Wilhelm Rotter, and Klaus Dieter Schmidt. “Kristallstrukturen und phasentransformationen von caesiumtrihalogenogermanaten (II) CsGeX₃ (X= Cl, Br, I)”. In: *Zeitschrift für anorganische und allgemeine Chemie* 545.2 (1987), pp. 148–156. DOI: 10.1002/zaac.19875450217.
- [48] Rasmus Lavén et al. “Unraveling the nature of vibrational dynamics in CsPbI₃ by inelastic neutron scattering and molecular dynamics simulations”. In: *The Journal of Physical Chemistry Letters* 16.19 (2025), pp. 4812–4818. DOI: 10.1021/acs.jpcllett.5c00778.
- [49] Arthur Marronnier et al. “Anharmonicity and disorder in the black phases of cesium lead iodide used for stable inorganic perovskite solar cells”. In: *ACS Nano* 12.4 (2018), pp. 3477–3486. DOI: 10.1021/acs.nano.8b00267.
- [50] K A Müller, W Berlinger, and F Waldner. “Characteristic structural phase transition in perovskite-type compounds”. In: *Physical Review Letters* 21.12 (1968), p. 814. DOI: 10.1103/PhysRevLett.21.814.
- [51] S K Mishra et al. “Phase stability and structural temperature dependence in sodium niobate: a high-resolution powder neutron diffraction study”. In: *Physical Review B* 83.13 (2011), p. 134105. DOI: 10.1103/PhysRevB.83.134105.

- [52] Daniel Schwalbe-Koda et al. “Model-free estimation of completeness, uncertainties, and outliers in atomistic machine learning using information theory”. In: *Nature Communications* 16.1 (2025), p. 4014. DOI: 10.1038/s41467-025-59232-0.

A

Settings and parameters

Here the settings and parameters used in DFT (Tab. A.1), ASE MD (Tab. A.2), LAMMPS MD (Tab. A.3) and MACE training (Tab. A.4) are presented.

Table A.1: DFT parameters used in VASP.

Parameter	Value
GENERAL	
METAGGA	R2SCAN
LUSE_VDW	.TRUE.
BPARAM	11.95
CPARAM	0.0093
LASPH	.TRUE.
KSPACING	0.25
KGAMMA	T
ENCUT	520
EDIFF	1e-5
NEMLIN	4
NELM	120
ADDGRID	T
PREC	acc
ALGO	norm
LCHARG	F
LWAVE	F
LMAXMIX	4
KPAR	2
NPAR	4
LPLANE	T
LREAL	F
ISMEAR	0
RELAXATION	
NSW	15
IBRION	1
ISIF	3
EDIFFG	-0.01
SINGLEPOINT	
NSW	0

Table A.2: The parameters used in ASE MD heating.

Parameter	Value
GENERAL	
pressure	0
timestep	2.5
ttime	25
pfactor	3375
t_interval	20
MD300K40	
supercell_matrix	[[2, 0, 0], [0, 2, 0], [0, 0, 2]]
strain_lim	[-0.05, 0.02]
initial_temperature	0.1
final_temperature	300
nsteps	100
MD300K320	
supercell_matrix	[[4, 0, 0], [0, 4, 0], [0, 0, 4]]
strain_lim	[0, 0]
initial_temperature	0.1
final_temperature	300
nsteps	100
MD1000K40 & MD1000K40CV	
supercell_matrix	[[2, 0, 0], [0, 2, 0], [0, 0, 2]]
strain_lim	[-0.05, 0.02]
initial_temperature	10
final_temperature	1000
nsteps	500

Table A.3: The parameters used in LAMMPS MD heating.

Parameter	Value
GENERAL	
units	metal
atom_style	atomic
boundary	p p p
atom_modify	map yes
newton	on
read_data	structure.data
change_box	all triclinic
pair_style	mliap unified model.pt 0
pair_coeff	* * A B X
timestep	0.001
velocity	all create 1 28459 rot yes dist gaussian mom yes
fix	1 all npt temp 1 1 0.1 iso 0.0 0.0 1
fix	2 all momentum 1 linear 1 1 1 angular
run	50000
unfix	1
reset_timestep	0
thermo	100
thermo_style	custom step temp press vol lx ly lz xy xz yz
SrTiO₃	
fix	1 all npt temp 1 400 0.1 tri 0.0 0.0 10
run	200 000
CsGeBr₃	
fix	1 all npt temp 1 600 0.1 tri 0.0 0.0 10
run	300 000
BaTiO₃, CsPbI₃	
fix	1 all npt temp 1 700 0.1 tri 0.0 0.0 10
run	350 000
BaZrO₃, BaZrS₃, NaNbO₃	
fix	1 all npt temp 1 1200 0.1 tri 0.0 0.0 10
run	600 000

Table A.4: The parameters used MACE training.

Parameter	Value
GENERAL	
valid_fraction	0.1
default_dtype	float64
model	Scale Shift MACE
r_max	6.0
num_radial_basis	8
max_ell	3
correlation	3
num_interactions	2
E0	average
max_num_epochs	1600
swa	True
start_swa	800
ema	True
ema_decay	0.99
amsgrad	True
restart_latest	True
swa_energy_weight	10000
swa_forces_weight	550
energy_weight	1
forces_weight	10
stress_weight	10
64x_300K	
hidden_irreps	64x0e
batch_size	100
swa_stress_weight	1
64x_1000K	
hidden_irreps	64x0e
batch_size	100
swa_stress_weight	100
128x_1000K	
hidden_irreps	128x0e
batch_size	50
swa_stress_weight	100

