Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Data-driven magnetic materials inverse design

A thesis submitted for the degree of Doctor of Philosophy
School of Physics Trinity College Dublin

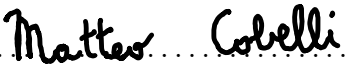
Candidate:
Matteo Cobelli

Supervisor:
Prof. Stefano Sanvito

Year: 2024

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work. I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signature: 

Abstract

Magnetic materials have diverse applications across multiple sectors, ranging from magnetic resonance imaging machines, used to detect diseases, to electric motors, sensors, and wind turbines just to name a few. The demand for novel magnetic materials, tailored for specific applications, is higher than ever. However, the recent advances in technology have not been matched by a comparable rate of material discovery, largely due to the unavoidable low throughput of experimental synthesis. For these reasons, there is a growing need for alternative approaches to material discovery, potentially involving *in-silico* predictions. The modelling capabilities of Density Functional Theory (DFT) make it a promising technique for selecting material prototypes based on computed properties, leading to an inverse-design approach, where the material synthesis is driven by specific application needs. However, the computational cost of DFT is too high to be the only technique adopted for inverse-design purposes. In this work, we present a data-driven approach to the design of magnetic materials. We address various challenges that afflict the material-discovery pipelines, for which we find solutions that leverage recent advancements in artificial intelligence. The result is an end-to-end workflow for materials inverse-design with a strong interdisciplinary nature, borrowing techniques from various domains of machine learning, ranging from statistical modelling to natural language processing (NLP). Specifically, we present several novel methods. Firstly, we introduce an NLP pipeline for the automatic extraction of data from the scientific literature, based on the fine-tuning of transformers-based language models. We then employ a machine-learning-enhanced prototype generation technique to improve the creation of accurate convex hulls for assessing the stability of ternary alloys. Additionally, we have designed a local inversion algorithm for finding the atomic structure associated with a given set of atomic descriptors, that can be coupled with generative models. Finally, we introduce the Jacobi-Legendre potential, a linear machine-learning interatomic potential based on the cluster expansion of the system energy, as well as the spin power spectrum, a set of descriptors of the local chemical environment for the modelling of magnetic materials using machine-learning techniques.

Acknowledgements

"I think there are people that help you become the person you end up being, and you can be grateful for them, even if they were never meant to be in your life forever. I'm glad I knew you too" - Diane Nguyen, BoJack Horseman.

I would like to express my sincere gratitude to my supervisor, Professor Stefano Sanvito, for his invaluable guidance, and to the Irish Research Council for their funding. I enjoyed every single day of my PhD, despite it occurring during particularly challenging times. The reason for this can be found in the amazing people who surrounded me throughout these years. Sharing this journey with Hugo, Laura, Luke, Michelangelo and Mike has been one of the best experiences of my life, contributing significantly to both my professional and personal growth.

I am also incredibly grateful to have met Alessandro, Anais, Anita, Anna, Annie, Akash, Bruno, Eoin, Paddy, Rajarshi, Ümit, Urvesh, Valerio, Willy and the rest of the Computational Spintronics Group.

I would finally like to thank Meabh for always staying by my side and all my family.

List of publications

1. Luke P. J. Gilligan, Matteo Cobelli, Valentin Taufour, and Stefano Sanvito. A rule-free workflow for the automated generation of databases from scientific literature. *npj Computational Materials*, 9(1):222, Dec 2023

2. Michail Minotakis, Hugo Rossignol, Matteo Cobelli, and Stefano Sanvito. Machine-learning surrogate model for accelerating the search of stable ternary alloys. *Phys. Rev. Mater.*, 7:093802, Sep 2023

3. Hugo Rossignol, Michail Minotakis, Matteo Cobelli, and Stefano Sanvito. Machine-learning-assisted construction of ternary convex hull diagrams. *Journal of Chemical Information and Modeling*, Jan 2024

4. Matteo Cobelli, Paddy Cahalane, and Stefano Sanvito. Local inversion of the chemical environment representations. *Phys. Rev. B*, 106:035402, Jul 2022

5. Michelangelo Domina, Urvesh Patil, Matteo Cobelli, and Stefano Sanvito. Cluster expansion constructed over jacobi-legendre polynomials for accurate force fields. *Phys. Rev. B*, 108:094102, Sep 2023

6. Michelangelo Domina, Matteo Cobelli, and Stefano Sanvito. Spectral neighbor representation for vector fields: Machine learning potentials including spin. *Phys. Rev. B*, 105:214439, Jun 2022

Contents

Chapter 1

Introduction

Throughout human history, the discovery of new materials has been closely tied to technological progress. Different eras such as the Iron Age and the Bronze Age were even named after the materials predominantly in use at the time, highlighting the link between human advancement and material science. The introduction of new materials enables the exploration and development of novel tools, which in turn can improve the quality of life and extend life expectancy. This has been evident not only in older times with the processing of metals like iron, bronze, and steel but also in more recent history with the engineering of semiconductors, polymers and alloys. While we have become increasingly efficient in developing new technologies, the discovery of new materials has struggled to keep pace. In fact, most new materials are discovered through labour-intensive, trial-and-error processes in laboratories, relying on a broad base of semi-empirical knowledge. This approach limits the rate at which new discoveries can be made. Currently, the inverse design of a new material [7], namely finding a material with the set of properties needed for a particular application, exists only in the form of incremental optimisation of the stoichiometry or the microstructure of already-known compounds. The vast, combinatorial chemical space of all possible compounds is too large to explore by manually attempting the synthesis of each one of them. Such an endeavour would take more time than the age of the universe. However, if we could predict a compound's properties using first-principles calculations, we could automate the exploration of this "chemical universe."

The chemical properties of a material are ultimately governed by its electronic structure. Unfortunately, simulating multi-electronic systems poses challenges that are insurmountable with conventional computer architectures, as the complexity of the system's electronic wave function grows exponentially with the number of atoms [8]. To circumvent this computational bottleneck, approximations to the Schrödinger equation must be introduced. The most successful of these approaches is unarguably the density functional theory (DFT) [9, 10]. In DFT, the focus shifts from the wave function to the electronic density, simplifying the problem considerably. The scaling of DFT calculations generally follows a cubic trend with respect to the number of

simulated atoms. However, for certain systems, efforts using local basis sets, which rely on the principle of nearsightedness, have succeeded in achieving linear scaling [11]. DFT has enabled high-throughput calculations of *ab-initio* properties for a wide range of compounds. Several initiatives have emerged to compile databases of these calculated DFT properties, offering an ever-growing set of compounds [12, 13, 14]. While these databases are searchable and allow for property-based screening, they are constrained in size and growth rate due to the computational costs of DFT calculations. This limits their immediate applicability in inverse material design. Despite these drawbacks, the rising number of publications, coupled with the increasing computational power and efforts to generate *ab-initio* calculation databases, has enriched the field with data, making it a fertile ground for machine-learning applications.

1.1 Machine learning

Machine learning has been a transformative force in various industries, revolutionizing everything from healthcare and drug discovery to finance and tech. Its impact extends beyond quantitative disciplines like mathematics and physics to even artistic and literary fields, thanks to recent advancements in language and multimodal models [15, 16, 17, 18]. The effectiveness of machine learning is closely tied to our ability to generate data. Over the last two decades, the rise of the Internet and advancements in information technology have made it possible for nearly everyone on Earth to access hardware capable of interfacing with the Internet and generating content. As a result, our proficiency in content creation and sharing has expanded exponentially. Furthermore, improvements in sensor design have enhanced both the availability and quality of sensors, while advances in computer hardware have facilitated a surge in the number and scale of simulations. Collectively, these developments have led to a dramatic increase in available data. This data often exhibits some intrinsic structure.

For example, natural language adheres to specific grammatical rules, while the pixel distribution in an image follows an internal structure that carries meaningful information. If pixels were randomly dispersed, they would convey little to no meaning. One can then think of trying to model these underlying patterns, but given their complexity, the resulting model that describes them must be also sufficiently intricate in order to possess the necessary flexibility necessary to learn them.

Machine-learning models are generally analytic functions between two Euclidean spaces and they depend on a set of parameters known as weights. Both the input values and these weights influence the model's output. In the context of supervised learning, the weights of the model are adjusted over a labelled dataset, also known as the training set, which consists of paired input-output values. The optimisation process, commonly referred to as training, aims to minimize a differentiable function called the loss function. This loss function is usually designed to decrease as the model's outputs

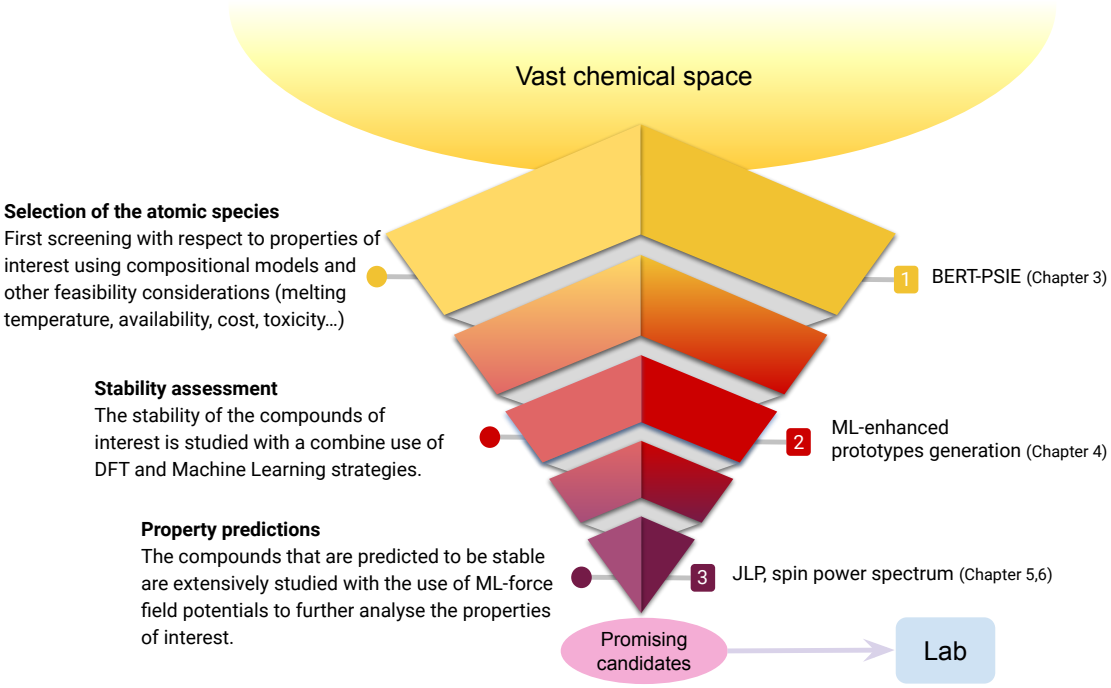


Figure 1.1: The diagram outlines the three stages of an inverse-design workflow: the selection of atomic species, the search for stable compositions along with their associated crystal structures, and finally, property-based screening. Each of these steps presents a variety of challenges, which we address through the development of data-driven techniques.

become close to the expected outputs. Various universal approximation theorems have been established for a range of machine-learning model architectures. These theorems guarantee that well-behaving functions can be approximated to an arbitrary degree of accuracy by commonly used machine-learning models. A key requirement for these theorems is the inclusion of a non-linear activation function, which introduces a non-linear dependency on the model's weights. Although this non-linearity enhances the model's flexibility, it also complicates the training process. Moreover, these models are non-convex functions, meaning that their loss functions have multiple minima. To optimize the weights, an iterative minimisation technique, such as gradient descent, is generally used [19]. Backpropagation offers an efficient strategy for calculating the gradient of the loss function with respect to the model's weights, making the training process computationally viable [20]. As a result, machine learning has emerged as a viable approach for tackling problems plagued by the "curse of dimensionality," which refers to the exponential increase in complexity associated with problems defined in high-dimensional spaces. In this regard, these techniques have the potential to address "hard" problems in material science.

1.2 Data-driven materials inverse design

The aim of this work is to address the challenge of identifying suitable material prototype candidates given a set of properties that are required for a specific application. The suitability of a prototype is determined by its likelihood of successful synthesis and its ability to exhibit the expected properties. Ideally, this inverse-design task should be performed by a model capable of efficiently navigating the vast combinatorial chemical universe. Although recent studies have leveraged the capabilities of generative models [21], there is no consensus yet on the most effective approach. In this work, we focus on an inverse-design workflow based on high-throughput screening. This approach consists of several stages where a pool of potential candidates is progressively refined to a smaller subset of more promising candidates. An example of such an inverse-design workflow is one composed of three distinct steps (see Fig. 1.1). At first, we want to select what elements to include in our search, as well as up to how many species to include in the composition. From a subset of the periodic table of N_{el} elements, we can create $\binom{N_{el}}{2}$ binaries, $\binom{N_{el}}{3}$ ternaries and so on. In order to limit this combinatorial explosion, it is advisable to reduce as much as possible the number of elements N_{el} under consideration. Factors such as availability, cost, toxicity and melting point usually intervene at this stage to remove from the picture certain elements. For each elemental combination, various stoichiometries can then be considered and different crystal structures can be associated with each composition. Recent research indicates that it is possible to infer a wide range of compound properties solely based on composition with reasonable accuracy. This is accomplished using compositional models, machine-learning models that take the chemical composition of a compound as input and return a prediction for one of its properties. These models are trained either on computational data or directly on experimental results. To name some examples, using this approach compositional predictors have been built for the Curie temperature of ferromagnets [22], the superconductors critical temperature [23] and glass-forming ability [24]. Compositional models offer the advantage of being relatively inexpensive to run, making them a powerful tool for initial screening in relation to properties of interest. At this stage of the workflow, they can be used to further refine the pool of elements under consideration. The main drawback associated with compositional models is the lack of structured databases of experimental data.

In order to address this issue, we developed a natural language processing workflow based on the fine-tuning of language models for the automatic extraction of data from the scientific literature. This workflow for the precise scientific information extraction (BERT-PSIE) will be detailed in Chapter 3.

Once the minimal pool of elements to take into consideration has been established we enter the second stage of the inverse-design workflow. This stage involves assessing which compounds and corresponding crystal structures are likely to be stable. The stability criterion for a compound is determined by the Gibbs free energy with respect

to competing phases. The Gibbs free energy comprises two terms, the enthalpy, which is influenced by the internal energy of the system, and the entropy. It has been shown that up to ternary compounds, enthalpy is the main driver of stability while entropy takes over when dealing with a higher number of constituents [25]. As a consequence, the strategy for identifying stable compounds varies depending on the number of constituents under consideration. Up to ternary compounds, a zero-temperature stability criterion is typically used. In this case, the entropy term is ignored and the enthalpy is approximated by the DFT total energy. A convex hull is constructed based on the systems with lower energy and the distance from the convex hull serves as stability criterion. The stability study of quaternary compounds and beyond has to involve an assessment of the entropy term of the Gibbs free energy.

We developed a workflow for an efficient and accurate construction of convex hulls for ternary compounds, leveraging existing data for binary compounds. This machine-learning-enhanced prototype generation workflow will be discussed in detail in Chapter 4. At this point, we avail ourselves of a set of prototypes that are deemed stable, together with their equilibrium structures.

The third and final stage of the inverse-design workflow consists in refining this pool of prototypes by selecting those for which we predict properties that fall in our range of interest. Compositional models can be employed again to further narrow down the list of promising compounds. Additionally, since we now have information on the equilibrium crystal structures, more sophisticated methods can be utilised. Machine-learning force fields (MLFF) can be trained to model specific compositions. The advantage of these data-driven potentials is that they maintain the accuracy of *ab-initio* simulations but at a fraction of the computational cost. These potentials can then be employed to model dynamical properties or defects at scales that are unattainable using conventional DFT methods [26, 27, 28]. In Chapter 5 we introduce the Jacobi-Legendre Potential (JLP) a linear machine-learning potential based on the cluster expansion of the energy of the system. These additional property predictions performed with MLFFs constitute the last stage of the inverse-design workflow. Further stability checks such as verifying the absence of imaginary phonon modes, can be carried out using either DFT or a trained MLFF model. The predicted dynamical properties serve as parameters with respect to which perform further screening. The compounds surviving all stages of the workflow can then be sent to the lab, where their synthesis is attempted. In Fig. 1.1 we report a diagram that outlines all the steps of the described inverse-design workflow, along with the data-driven strategies we developed and that are discussed in this work. The inverse-design workflow presented here is data-driven in the sense that it heavily relies on machine-learning techniques developed to address specific challenges at each one of its stages. This approach has become viable due to the recent surge in the availability of experimental and computational data, as well as advancements in machine-learning technologies.

1.3 Data-driven magnetic materials inverse design

While the data-driven inverse-design workflow discussed in the previous section is general in scope, it can present certain shortcomings for some classes of materials. Our focus here is on a particular class of magnets, specifically ferromagnets. Ferromagnetic materials populate one of the largest classes of magnetic materials. They display long-range alignment of the local atomic magnetic moments in the same direction, leading to the formation of a macroscopic magnetic moment in the absence of an external magnetic field. The critical temperature at which this ordering is destroyed by the thermal fluctuations is called Curie temperature T_C.

Various technologies rely on ferromagnetic materials, from hard-disk storage and magnetic resonance imaging to electric motors. However, only a small fraction of known ferromagnets possess a T_C greater than 600 K, which is generally required for room-temperature applications. Given the technological significance of these materials and the limited range of known compounds, the inverse design of new ferromagnets is particularly important. In this work, we address this challenge by automatically extracting compound-Curie temperature pairs from the scientific literature using BERT-PSIE, a pipeline of fine-tuned language models. We demonstrate that the generated dataset can be used to train compositional models able to screen high-T_C compounds.

Another limitation of the existing inverse-design workflow is its incapability to model magnetic materials using traditional Machine-Learning Force Fields (MLFF). Conventional MLFF models do not account for the magnetic properties of the atoms within the system. As a result, they are unable to correctly model magnetic materials. An inverse-design workflow aimed at magnetic materials would require the generalisation of existing methods to include magnetic properties. In Chapter 6 we propose a solution through the spin power spectrum. This approach allows us to develop models trained on spin-polarised DFT calculations enabling the prediction of a system's magnetic properties. The inclusion of these predictive models into our workflow results in a data-driven approach to the design of magnetic materials.

1.4 Summary

This thesis presents the various protocols that we have developed to address the challenges inherent in the inverse design of magnetic materials. The data-driven nature of these methodologies is a result of the strong interdisciplinary focus of this work. An outline of the relevant methods used is provided in Chapter 2. The following chapters discuss the various steps of the workflow as illustrated in Fig. 1.1. The first stage consists of selecting the pool of elements on which to conduct the search. In Chapter 3 we demonstrate that, by leveraging natural language processing techniques, we can create a pipeline for the automatic extraction of relevant data from the existing scientific literature. This data can then either be queried directly or used to train compositional

models to guide the selection of the elements that we are interested in.

The second stage focuses on identifying which stoichiometric, including the elements of interest, are stable and their corresponding equilibrium crystal structures. Traditional approaches to this problem involve calculating the energy associated with various crystal structure prototypes for a given stoichiometry and constructing a convex hull based on the minimum energy structures found for different stoichiometries. Chapter 4 introduces two data-driven approaches for the generation of atomic structures. The first focuses on ternary structures and relies on the decoration of the stable binary phases of the constituent compounds. The second leverages the use of machine-learning generative methods together with the inversion of the chemical environment descriptors commonly used for MLFFs. Once the stable compounds are isolated the third stage of the workflow aims to predict additional properties of these materials.

Chapter 5 introduces a novel MLFF based on a cluster expansion of the energy of the system, the Jacobi-Legendre potential, which can be used to predict the dynamical properties of the system. Chapter 6 presents an extension of MLFFs to include the spin associated with the magnetic atoms of the system. These potentials can be trained on spin-polarised DFT calculations and used to predict the magnetic properties of a compound. By integrating all these methods together, a data-driven inverse-design workflow for the discovery of novel magnetic materials is established. We have implemented these techniques in a Python library called "Force Fields Machine," which is presented in Chapter 7. Finally, in Chapter 8 we conclude and provide a discussion regarding potential future directions of this research.

Chapter 2

Methods

Before presenting in detail the strategies compounding our data-driven inverse-design workflow, this chapter provides an overview of the methodologies commonly employed in material science to simulate material's properties, as well as the current state of machine-learning techniques applied in this domain.

2.1 Density functional theory

The properties of a material ultimately depend on its chemical composition and on its crystal structure. The knowledge and the ability to solve the laws of physics governing the interactions between atoms would guarantee the ability to simulate and compute all the properties of any material. Although quantum mechanics provides such a set of rules, approximations need to be introduced in order to make this problem computationally treatable, even for simple systems. In the time-independent case and non-relativistic limit, within the Born-Oppenheimer approximation, the total energy of an isolated system of N nuclei and N_e electrons is given by the solutions of the time-independent Schrödinger equation [8]:

$$\hat{H}_e(\{\vec{R}_I\})\Psi_e = E(\{\vec{R}_I\})\Psi_e, \tag{2.1}$$

with,

$$\begin{aligned}\hat{H}_e(\{\vec{R}_I\}) = &\sum_i^{N_e} \frac{-\hbar^2}{2m_e}\nabla^2_{\vec{r}_i} + \frac{1}{2}\sum_{I\neq J}^{N_N} \frac{e^2 Z_I Z_J}{|\vec{R}_I - \vec{R}_J|} - \frac{1}{2}\sum_{i,I}^{N_e,N_N} \frac{e^2 Z_I}{|\vec{r}_i - \vec{R}_I|} + \frac{1}{2}\sum_{i\neq j}^{N_e} \frac{e^2}{|\vec{r}_i - \vec{r}_j|} \\ &= \hat{T}_e + \hat{E}_{NN} + \hat{E}_{Ne} + \hat{V}_{ee}. \end{aligned} \tag{2.2}$$

Here Ψ_e is the electronic wave function of the system, Z_I the atomic number of the nucleus I, m_e is the electron mass and R_I represents the position of the nucleus I. This is an equation in $3N_e$ variables with N_e being the total number of electrons in the system. Within the Born-Oppenheimer approximation, the nuclei are considered as classical point particles. So far we are assuming their position to be fixed, hence

their contribution to the total kinetic energy is zero. From Eq.(2.2) it follows that the Hamiltonian is entirely determined by the position and atomic species of the atoms of the system.

The Hohemberg and Kohn theorem shows that the total energy of the ground state of a system is a unique functional of the electronic charge density and that the density that minimizes such functional is the electronic density of the ground state [8, 9],

$$E = E[\rho] = T[\rho] + E_{Ne}[\rho] + V_{ee}[\rho] + E_{NN}, \tag{2.3}$$

where the electronic single-particle charge density is defined as:

$$\rho(\vec{r}) = N_e \int |\Psi_e(\vec{r}, \vec{r}_2 \ldots, \vec{r}_{N_e})|^2 d\vec{r}_2 \ldots, d\vec{r}_{N_e}. \tag{2.4}$$

Once the electronic charge density of the ground state is known then all the ground-state quantities describing the system can be, in principle, computed. The initial problem is now reduced to the determination of a function in three variables. Unfortunately, the exact functional dependence of the energy from the electronic density is not known and it is necessary to introduce approximations. Kohn and Sham proposed to divide the total energy of the system as follows [10],

$$E[\rho] = T_s[\rho] + E_{Ne}[\rho] + J[\rho] + E_{XC}[\rho] + E_{NN}, \tag{2.5}$$

where T_s is the total kinetic energy of a system of N_e non-interacting electrons chosen such that the electronic charge density of the fundamental state is the same as the initial system. Then, E_{Ne}, J and E_{NN} are the Coulombic contributions of the electron-nucleus interaction, the Hatree term of the electron-electron interaction and nucleus-nucleus interaction, respectively, while E_{XC} is the exchange-correlation energy containing the terms for which an exact dependency from the electronic charge density is not known,

$$E_{XC}[\rho] = T[\rho] - T_s[\rho] + V_{ee}[\rho] - J[\rho]. \tag{2.6}$$

The orbital solutions of the equation relative to the non-interactive system are said Kohn-Sham orbitals,

$$\left[-\frac{\hbar^2}{2m_e} \nabla^2 + V_{\text{eff}}(\vec{r}) \right] \psi_i(\vec{r}) = \varepsilon_i \psi_i(\vec{r}). \tag{2.7}$$

with V_{eff} being an effective potential defined as,

$$V_{\text{eff}}(\vec{r}) = \frac{\delta E_{Ne}[\rho]}{\delta \rho(\vec{r})} + \frac{\delta J[\rho]}{\delta \rho(\vec{r})} + \frac{\delta E_{XC}[\rho]}{\delta \rho(\vec{r})} = v_{\text{ext}}(\vec{r}) + V_H(\vec{r}) + V_{XC}(\vec{r}). \tag{2.8}$$

Here v_{ext} includes the potential generated by the nuclei, V_H is the Hartree potential and it is the solution of the Poisson equation generated by the density ρ. Finally

$V_{XC}(\vec{r})$ is the exchange-correlation potential for which there is no exact expression and it requires some degree of approximation. A solution of the Kohn-Sham equations can be obtained through a self-consistent loop. At convergence, it is possible to estimate the energy of the ground state of the system and the forces acting on the atoms through the Hellmann-Feynman theorem [29]

$$\vec{F}_I = -\frac{\partial E}{\partial \vec{R}_I} = -\langle\Psi|\frac{\partial \hat{H}}{\partial \vec{R}_I}|\Psi\rangle\,, \tag{2.9}$$

while the contribution to the stress tensor $\mathbf{W}$ is given by [30]:

$$\mathbf{W} = \sum_{I=1}^{N} \vec{R}_I \otimes \vec{F}_I. \tag{2.10}$$

Here $\otimes$ is the Cartesian outer product operator. The computation of these quantities allows one to perform relaxation and *ab-initio* molecular dynamics providing access to several properties (e.g. the equilibrium structure, equilibrium volume, phonon dispersion, ...).

2.2 Force fields

Within the Born-Oppenheimer approximation, the nuclei of the atoms are described as classical pointwise particles with a specific atomic mass. Their dynamics is then governed by Newton's laws

$$\vec{F}_I = M_I \frac{d\vec{v}_I}{dt}, \tag{2.11}$$

which relates the force F_I acting on an atom with mass M_I with the variation of its velocity $\vec{v}_I$. For the case of conservative forces, F_I can be expressed as the gradient of a scalar potential U

$$\vec{F}_I(\{\vec{R}_J\}) = \frac{\partial U}{\partial \vec{R}_I}(\{\vec{R}_J\}). \tag{2.12}$$

For an isolated system of N atoms, U is a function of all the atom's positions and their species. The most simple case is the one where U depends only on the pairwise distance between each couple of atoms I and J,

$$U = \frac{1}{2}\sum_{I\neq J} u(|R_I - R_J|), \tag{2.13}$$

in this case, u is called pair potential. In general, the interaction between atoms is insufficiently described as a pair-wise interaction. While the Coulombic interaction is a pairwise interaction, when seen as acting on classical point particles, the quantum mechanical description of the electrons within *ab-initio* calculations makes the Coulombic interaction between atoms non-pairwise. The electrostatic interaction between atoms

deforms their electronic cloud inducing polarisation in a way that is not a simple function of only the pairwise distance between atoms. Given the positions of all the atoms of the system and their velocities at a time t_0 $\{\vec{v}_I(t_0), \vec{R}_I(t_0)\}$ the velocities and positions at a later time t_n can be computed by using iteratively the Verlet integration [31]:

$$\begin{cases} \vec{R}_I(t_1) = \vec{R}_I(t_0) + \vec{v}_I(t_0)\ (t_1 - t_0) \\ \vec{R}_I(t_{n+1}) = 2\vec{R}_I(t_n) - 2\vec{R}_I(t_{n-1})\ (t_{n+1} - t_n) - \frac{1}{M_I}\boldsymbol{\nabla}_I U(\{\vec{R}_J(t_n)\})\ (t_{n+1} - t_n)^2 \end{cases}$$

The resulting molecular dynamic is a trajectory of the system in the microcanonical ensemble. If the potential is a semi-empirical analytical function designed to model a specific class of compounds, the trajectory arising from the solution of these equations of motion is called classical molecular dynamics. The most simple and popular semi-empirical choice for the potential U is the Lennard-Jones potential, a pair potential defined as:

$$u(|R_I - R_J|) = 4\varepsilon \left[\left(\frac{\sigma}{|R_I - R_J|} \right)^{12} - \left(\frac{\sigma}{|R_I - R_J|} \right)^{6} \right]. \tag{2.14}$$

This potential captures the close-range repulsion between atoms and their long-range van-der-Waals attraction. The parameters ε and σ are generally optimised for each specific material considered. The design of these semi-empirical potentials is driven by a trade-off between accuracy and computational complexity. The bottleneck in the integration of the equation of motion is the computation of the gradients of the scalar potential. Semi-empirical potentials are usually simple analytical functions depending on a few parameters that are fitted to best reproduce certain properties (often experimental) of the system under investigation. This fitting procedure does not generally follow a specific strategy, it is application-specific and often driven by physical intuition. The analytical simplicity of these potentials makes them the most computationally efficient methods to perform molecular dynamics allowing the simulation of a large number of atoms and long trajectories. However, this simplicity often prevents their ability to simultaneously capture multiple properties of the system.

An alternative approach consists of computing the forces acting on the atoms of the system from first principles. In the context of DFT, this is facilitated by the Hellmann-Feynman theorem Eq. (2.9). The resulting simulations take the name of *ab-initio* molecular dynamics and differentiate themselves from classical molecular dynamics by the absence of any fitting parameters present in the theory. Any step of the molecular dynamic requires a DFT calculation performed with the updated atomic position to compute the updated forces. The underlying approximation is that the electrons follow adiabatically the movement of the nuclei and the electronic wave-function coincides with the ground-state one at any instant. Having to perform a DFT calculation for each step of the dynamic makes this task particularly computationally intensive. There are strategies to reduce the number of self-consistent cycles needed to

reach convergence by initializing the density with the one obtained at the previous step of the dynamics. This strategy significantly accelerates the convergence for small time steps, nevertheless, *ab-initio* molecular dynamics of thousands of atoms are already at the boundary of what is generally accessible on standard hardware and only possible with codes specifically optimised to achieve linear scaling with respect to the number of atoms. The advantage of *ab-initio* molecular dynamics relies on their ability to accurately describe multiple properties simultaneously, from first principles, without requiring any fitting step. However, the calculation of dynamic properties that require large atomic cells or long trajectories remains locked away from the reach of *ab-initio* simulation.

Thus, the force fields landscape presents a trade-off. On the one hand, classical semi-empirical force fields are computationally cheaper, but limited in their complexity and in the physics that they can describe. On the other hand, *ab-initio* molecular dynamics can describe very complex systems at the cost of being computationally expensive. The ideal force field would have the computational cost of classical molecular dynamics and the complexity and accuracy of *ab-initio* simulations. In the next section, we will discuss how machine-learning force fields aim to achieve both these characteristics.

2.3 Machine-learning force fields

As discussed in the first chapter, machine-learning models aim to efficiently find patterns in the data provided at training time. Once successfully trained the model will be able to predict the corresponding outputs on inputs that it has not seen during the training. This workflow can be directly applied to the design of force fields. For instance, the training set could include different atomic configurations with their corresponding calculated DFT energy. The machine-learning model is then trained to predict the DFT energy given the position and chemical identity of the atoms of the system. The forces acting on each atom can then be predicted by computing the gradient of the model with respect to the input atomic positions. This approach would provide a way to obtain DFT-level results, but at a higher throughput since the computational time required to evaluate the total energy of the system is the one required for a forward pass through the model. This will depend on the model's architecture and on the system under study, but it can be orders of magnitude faster than DFT.

A widely adopted assumption in both machine learning and classical semi-empirical potentials is to be able to express the total energy of the system E as a sum of atomic contributions E_I,

$$E_{Tot} = \sum_{I=0}^{N} E_I. \tag{2.15}$$

Within DFT only the total equilibrium energy of the system is accessible, while the atomic contributions E_I are not well defined. However, this decomposition allows us to

effectively implement the principle of nearsightedness into the model [11] by assuming that each E_I only depends on the atoms within a certain cutoff distance R_{cut} from the atom I. Moreover, targeting the prediction of the atomic energies as opposed to the total energy offers a straightforward way to take into account the fact that the energy is an extensive quantity. This allows one to train over small-size cells and then to scale up to predict the energies of larger systems. When it comes to designing a machine-learning force field multiple choices need to be taken, which ultimately will impact its performance and applicability. Among these, the components that will mostly impact the quality of the force field are the choice of the input features and the model architecture. In the subsequent sections, we will be exploring both these components.

2.3.1 Chemical environment representations

As previously discussed the properties of an isolated system of atoms are completely defined once the position and atomic identity of the atoms are specified. Machine-learning models are functions that map an Euclidean space to another[1]. As such it is necessary to convey the information regarding the system in a way compatible with this framework. Within DFT, the positions of the atoms are naturally expressed in terms of Cartesian coordinates with respect to the simulation cell axis. However, directly using Cartesian coordinates as input for a machine-learning model trained to predict the total energy of the system will surely lead to poor results. The reason for this is ultimately connected with the fact that the total energy of the system is a scalar quantity that is invariant with respect to any global rigid rotation or translation of the system while Cartesian coordinates are not. It is up to the model then to learn the presence of these symmetries. While this is possible in principle, by virtue of the universal approximation theorem in the limit of an infinite amount of training data and a sufficiently large model, it is not feasible in practice. A solution to this problem is to use as input of the model a set of descriptors computed from the Cartesian coordinates which possess the same symmetries of the targeted quantity. The output of the resulting model will consequently inherit all the symmetries implemented in such input descriptors of the chemical environment. For the case of the total energy these symmetries include rotation, translation invariance and invariance with respect to the permutation of any atom of the same species. The design of such invariant features poses a significant challenge. If we trivially consider, for example, the use of angles and inter-atomic distances as input features, then we would have a model that is invariant under rotation and translation of the system. However, the listing of the input should be handled in a way that is invariant under the exchange of two atoms of the same species. A more careful construction is then required to take into account all the desired symmetries. We will now discuss some of the strategies that have been

[1]The problem can be also extended to non-Euclidean spaces [32]

used to implement such invariant descriptors.

Symmetry functions

Symmetry functions were introduced by Behler and Parrinello in their seminal work published in 2007 [33]. Their definition takes inspiration from the definition of pair distribution functions, which describe the distributions of distances between pairs of atoms. The two-body family of symmetry functions generated by the parameters ν and R_s is defined as [33],

$$G_I^{2-\text{body}}(R_{IJ};\eta,R_s)=\sum_{J\neq I}^{all} e^{-\eta(R_{IJ}-R_s)^2} f_c(R_{IJ}), \tag{2.16}$$

where R_{IJ} is the distance between the atoms I and J, while f_c is a cutoff function defined to go to zero smoothly at the cutoff distance R_{cut} and it is null for larger values. A common choice for f_c is

$$f_c(R_{IJ})=\begin{cases}\frac{1}{2}\left[\cos\left(\frac{\pi R_{IJ}}{R_{\text{cut}}}\right)+1\right] & \text{for } R_{IJ}\leq R_{\text{cut}},\\ 0 & \text{for } R_{IJ}>R_{\text{cut}}.\end{cases} \tag{2.17}$$

Here f_c is introduced to enforce locality and to guarantee a smooth variation of the model's output, when atoms move inside or outside the cutoff radius, particularly important during molecular dynamics. Ultimately the specific choice of f_c is arbitrary and it can be regarded as a hyperparameter. The parameters ν and R_{cut} are allowed to assume any real value, in practice, a small finite subset of values is considered. The purpose of the symmetry functions is to introduce an invariant basis set on which the energy of the system can be expanded. Since, as previously pointed out, the total DFT energy cannot be decomposed in terms of only pair contributions, in order to be able to build accurate force fields it is useful to introduce descriptors that incorporate higher-order contributions. The following three-body symmetry functions are commonly used in combination with the two-body ones,

$$\begin{aligned}G_I^{3-\text{body}}(R_{IJ},R_{IK},R_{JK},\theta_{ijk};\eta,\lambda,\zeta)=&\\ 2^{1-\zeta}\sum_{J,K\neq I}^{all}(1+\lambda\cos\theta_{IJK})^{\zeta}e^{-\eta(R_{IJ}^2+R_{IK}^2+R_{JK})}&f_c(R_{IJ})f_c(R_{IK})f_c(R_{JK}).\end{aligned} \tag{2.18}$$

Here the parameters introduced are $\lambda\in\{-1,1\},\eta$ and ζ. Symmetry functions are generally used in combination with multilayer perception neural networks [33]. Further details on this architecture will be discussed in subsequent sections.

Bispectrum components

Another approach used to describe the chemical environment is given by the bispectrum components [34]. The information regarding the local atomic environment surrounding an atom I can be entirely expressed in terms of a neighbour density distribution,

$$\rho_I(\vec{R}) = \delta(\vec{R}) + \sum_J w_J \, f_c(R_{IJ}) \, \delta(\vec{R} - \vec{R}_J), \tag{2.19}$$

where f_c is a cutoff function with the same purpose as the one encountered for the symmetry functions. The $\{w_J\}$'s are weights depending on the species of the atom J. We can now map the point in real space onto the surface of the 3-sphere, in a similar fashion as a Riemann projection. The following polar coordinates can be used for this purpose:

$$\vec{R} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{array}{l} \psi = \arctan\left(y/x\right) \\ \theta = \arctan\left(\frac{\sqrt{x^2+y^2}}{z}\right) \\ \theta_0 = \theta_{max} \, R/R_0 \end{array}, \tag{2.20}$$

where $R_0 > R_{\text{cut}}$ is a parameter of the transformation. By using this transformation we can express the neighbour density distribution as a function on the 3-sphere. The spherical harmonics provide a natural basis for functions defined on the surface of a sphere. Similarly, a natural basis for functions defined onto the 3-sphere is given by the hyper-spherical harmonics $U^l_{m,m'}$,

$$\rho_I = \sum_{l=0,\frac{1}{2},\ldots} \sum_{m,m'=-l}^{l} u^l_{m,m'} U^l_{m,m'}, \tag{2.21}$$

with expansion coefficients $u^l_{m,m'}$ given by the following inner product,

$$u^l_{m,m'} = \langle U^l_{m,m'} | \rho \rangle = U^l_{m,m'}(0,0,0) + \sum_j w_j \, f_c(R_{IJ}) \, U^l_{m,m'}(\theta_0, \theta, \psi). \tag{2.22}$$

The expansion of Eq. (2.21) can be seen as a change of basis. Now the coefficients $u^l_{m,m'}$ contain all the information necessary to describe ρ_I. Unfortunately, these coefficients are not suitable as descriptors, since they are complex-valued and they are not invariant under rotations. However, combinations of $u^l_{m,m'}$ such as the power spectrum possess the desired properties and can be used as input of machine-learning models

$$P_l = \sum_{m,m'=-l}^{l} \left(u^l_{m,m'}\right)^* u^l_{m_1,m'_1}, \tag{2.23}$$

Moreover, it can be proved [34] that the following triple-products of expansion coefficients are real-valued and rotationally invariant,

$$B_{l_1,l_2,l} = \sum_{m_1,m_1'=-l_1}^{l_1} \sum_{m_2,m_2'=-l_2}^{l_2} \sum_{m,m'=-l}^{l} \left(u^l_{m,m'}\right)^* C^{lm}_{l_1 m_1 l_2 m_2} C^{lm'}_{l_1 m_1' l_2 m_2'} u^{l_1}_{m_1,m_1'} u^{l_2}_{m_2,m_2'}, \quad (2.24)$$

where $C^{lm}_{l_1 m_1 l_2 m_2}$ are the Clebsch-Gordan coefficients. This invariant coefficients are called bispectrum components. Once a maximum angular momentum, l_{max}, is specified in order to truncate the infinite summation of Eq. (2.21), the list of different bispectrum components that can be constructed combining the available coefficients $u^l_{m,m'}$ can be used as descriptors for the chemical environment surrounding the atom I. The number of non-zero unique bispectrum components for a given integer l_{max} is given by [35],

$$\frac{(l_{max}+1)(l_{max}+\frac{3}{2})(l_{max}+2)}{3}. \quad (2.25)$$

By definition, the bispectrum components include contributions up to the 4-body order and their symmetries make them suitable to be used as input features for machine-learning force fields.

Incompletness of the chemical environment representations

Atomic structure representations such as the bispectrum components were generally considered to provide a complete basis set for the atomic neighbour density distribution. This changed in 2020 after the work of Pozdnyakov *et al.* [36] in which it was shown that it is possible to construct examples of atomic clusters with the same bispectrum components but different energy. A machine-learning force field will return the same output given the same input, hence, the problem of incompleteness in the representation used exposes the model to a weakness in its ability to discern different chemical environments. The underlying problem is associated with the fact that a n-body descriptor offers a complete basis for an atomic environment consisting of up to n atoms, while it might be under-complete for environments that include more atoms. While this is particularly severe for low body orders, the importance of this aspect becomes less and less significant for high body orders. It is then particularly challenging to construct examples of distinct clusters that share the same descriptors, and the total energy contribution of these higher-order correlations diminishes rapidly. This is also shown by the practical success seen by descriptors such as the bispectrum components or the symmetry functions. Nevertheless, being able to systematically address the completeness of the representation is a valuable feature. In this regard the symmetry functions can be expanded by including higher body order functions similar to what was done with the introduction of the 3-body symmetry functions along with the 2-body ones. The limit associated with this approach is that there is no general and systematic set of rules to design symmetry functions of a given body order beyond

three. Each case needs to be individually crafted. At the same time, the bispectrum components do not offer any straightforward way to include body orders beyond the 4th given the definition of bispectrum.

The first instance of a complete representation was actually provided by Shapeev already in 2016 with the development of Moment Tensor Potentials (MTP) [37]. The MTP definition is related to the inertia tensors of the atomic environment. New MTP terms which include higher body orders can be systematically added. More recently, Drautz generalised the construction of descriptors similar to power spectrum and bispectrum components to an arbitrary body order all within the framework of cluster expansion of the total energy of the system,

$$E_{Tot} = E^{(1)} + E^{(2)} + E^{(3)} + \dots . \tag{2.26}$$

The resulting descriptors called Atomic Cluster Expansion (ACE) have seen a wealth of applications [38, 39, 40, 41, 42]. Following the success of the ACE potential other instances of atomic descriptors that are based on the cluster expansion of the total energy of the system have been introduced. Notable among these are the proper orthogonal descriptors [43] and the Jacobi-Legendre potential (JLP) which we developed, and present in Chapter 5 in greater detail.

2.3.2 Models architecture

In the previous paragraph, we have presented an overview of the possible choice of invariant descriptors that have been designed to be suitable input features for machine-learning models. Starting from Eq. (2.15) we now want to specify the functional relation, $\mathcal{F}$, that connects the input representation $\{\mathcal{B}_I(\{\vec{R}_J\})\}$, to the atomic energy contributions:

$$E_I = \mathcal{F}\left(\left\{\mathcal{B}_I\left(\left\{\vec{R}_J\right\}\right)\right\}, \{w\}\right). \tag{2.27}$$

where, w, represents the set of the model's learning parameters or weights, whose value is optimised during the training. Once the learning parameters, $\{\beta\}$, are defined the inter-atomic potential is fully determined. Given a training set of N_{train} configurations, whose energies are computed with a reliable, but computational-intensive method (e.g. DFT), the optimal values of the coefficients, $\{w\}$, are the ones that minimize the loss function,

$$\ell = \sum_{s=1}^{N_{train}} \Bigg[(E^{ML(s)} - E^{DFT(s)})^2 + \gamma_F \sum_{I=1}^{N_s} \sum_{i\in\{x,y,z\}} (F_{I,i}^{ML(s)} - F_{I,i}^{DFT(s)})^2 + \gamma_W \sum_{i,j\in\{x,y,z\}} (W_{ij}^{ML(s)} - W_{ij}^{DFT(s)})^2 \Bigg], \tag{2.28}$$

where γ_F and γ_W are coupling parameters that control the relative weight given to the forces and to the stress tensor in the optimisation process. These parameters allow us to account for differences in relative magnitude and for the fact that associated with a single atomic configuration there are $3N$ forces components, 6 independent stress tensor components and one total energy. The specific functional relation to use in Eq. (2.27) is again a matter of choice dictated by the type of problem and the amount of data available in the training set. In this section, we will present an overview of the prevalent architectures commonly employed for machine-learning force fields.

Linear models

The simplest choice of model architecture is provided by linear models. In this case $\mathcal{F}$ is a linear function with respect to the learning parameters [44],

$$E_I = w_0^{(k_I)} + \sum_{j=1}^{M} w_j^{(k_I)} \, g_j \left(\mathcal{B}_{I,j} \right), \tag{2.29}$$

where g_j are scalar non-linear functions and we identify with $\mathcal{B}_{I,j}$ the j-th of M descriptor of the chemical environment surrounding the atom I. The learning parameters depend only on the chemical identity of I. This ensures that atoms of the same species will share weights, so that the total energy of the model remains invariant with respect to the exchange of identical atoms. As shown in the previous paragraph, the invariant atomic descriptors are designed as highly non-linear functions of the Cartesian coordinates of the system. As such, we can assume, without loss of generality, that any non-linear function g_j present in Eq. (2.29) is taken into account in the definition of the descriptors $\mathcal{B}_{I,j}$. We can then rewrite Eq. (2.29) as

$$E_I = w_0^{(k_I)} + \sum_{j=1}^{M} w_j^{(k_I)} \mathcal{B}_{I,j}. \tag{2.30}$$

The total energy of a system with K distinct chemical species is then given by

$$E_{Tot} = \sum_{I}^{N} E_I = \sum_{k=1}^{K} \left(w_0^{(k)} N_k + \sum_{j=1}^{M} w_j^{(k)} \sum_{I|k_I=k} \mathcal{B}_{I,j} \right), \tag{2.31}$$

where N_k is the number of atoms of chemical species k. By computing the gradient of the total energy, we can then obtain an expression for the forces,

$$\vec{F}_I = -\frac{\partial E_{Tot}}{\partial \vec{R}_I} = -\sum_{k=1}^{K} \sum_{j=1}^{M} w_j^{(k)} \frac{\partial}{\partial \vec{R}_I} \left(\sum_{J|k_J=k} \mathcal{B}_{J,j} \right). \tag{2.32}$$

Finally, we can compute the stress tensor components by using Eq. (2.10),

$$\mathbf{W} = \sum_{I=1}^{N} \vec{R}_I \otimes \vec{F}_I = -\sum_{k=1}^{K}\sum_{j=1}^{M}\sum_{I=1}^{N} w_j^{(k)}\, \vec{R}_I \otimes \frac{\partial}{\partial \vec{R}_I}\left(\sum_{J|k_J=k} \mathcal{B}_{J,j}\right). \tag{2.33}$$

We can then write the least-square loss reported in Eq. (2.34) for the linear case, which reads

$$\ell = \left(\sum_{k=1}^{K} A^{(k)} \cdot \boldsymbol{w}^{(k)} - \mathbf{y}\right)^2, \tag{2.34}$$

with:

$$A^{(k)} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ N_k^{(s)} & \sum_{I|k_I=k} \mathcal{B}_{I,1}^{(s)}) & \dots & \sum_{I|k_I=k} \mathcal{B}_{I,M}^{(s)} \\ 0 & \frac{\partial}{\partial x_I}\left(\sum_{J|k_J=k} \mathcal{B}_{J,1}^{(s)}\right) & \dots & \frac{\partial}{\partial x_I}\left(\sum_{J|k_J=k} \mathcal{B}_{J,M}^{(s)}\right) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \sum_{I=1}^{N} x_I \frac{\partial}{\partial x_I}\left(\sum_{J|k_J=k} \mathcal{B}_{J,0}^{(s)}\right) & \dots & \sum_{I=1}^{N} x_I \frac{\partial}{\partial x_I}\left(\sum_{J|k_J=k} \mathcal{B}_{J,M}^{(s)}\right) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

and

$$\boldsymbol{w}^{(k)} = \begin{bmatrix} w_0^{(k)} \\ w_1^{(k)} \\ \vdots \\ w_M^{(k)} \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} \vdots \\ E^{(s)} \\ \gamma_F F_x^{(s)} \\ \vdots \\ \gamma_W W_{x,x}^{(s)} \\ \vdots \end{bmatrix}. \tag{2.35}$$

The superscript $^{(s)}$ we identifies quantities relative to the s-th atomic configuration in the training set. We can make the matrix form of Eq. (2.34) even more explicit by concatenating the terms relative to the different chemical species and incorporating the summation over k into the matrix multiplication:

$$\ell = \left(\mathbf{A} \cdot \boldsymbol{w} - \mathbf{y}\right)^2, \tag{2.36}$$

where $\mathbf{A} = [A^{(1)}, \dots, A^{(K)}]$ and $\boldsymbol{w}^T = [w^{(1)}, \dots, w^{(K)}]^T$. As a result, the least-square loss for a linear model is a convex function of the learning parameters, meaning that any minimum that exists will coincide with the global minimum. The extreme points of the loss are found by setting the gradient to zero, namely

$$\frac{\partial L}{\partial \boldsymbol{w}} = 2(\mathbf{A} \cdot \boldsymbol{w} - \mathbf{y}) = 0 \rightarrow \mathbf{A} \cdot \boldsymbol{w} = \mathbf{y}, \tag{2.37}$$

This forms an overdetermined system of linear equations in $\boldsymbol{w}$. Solving for $\boldsymbol{w}$ we obtain,

$$\boldsymbol{w} = \mathbf{A}^{-1} \cdot \mathbf{y}, \tag{2.38}$$

where we identify with $\mathbf{A}^{-1}$ the pseudoinverse of $\mathbf{A}$. The fact that we can obtain a closed-form solution for the optimal values of $\boldsymbol{w}$ is a valuable property of linear models. It is common to add a regularisation term to the loss to cure the likely ill-condition of the overdetermined system of equations set by Eq. (2.37) and prevent overfitting,

$$\ell = \left(\mathbf{A} \cdot \boldsymbol{w} - \mathbf{y}\right)^2 + \alpha\, \boldsymbol{w}^T \boldsymbol{w}. \tag{2.39}$$

This regularisation term coincides with the L2 norm of the learning parameters, α controls the regularisation strength and it is a hyperparameter of the model. A regression performed with this loss takes the name of Ridge regression. The regularisation term limits the possibility of finding optimal values of the learning parameters, which are too large in size, a behaviour that is generally associated with overfitting. The optimal values of the learning parameters for a Ridge regression can again be obtained in close form,

$$\boldsymbol{w} = \left(\mathbf{A} + \alpha \mathbf{I}\right)^{-1} \cdot \mathbf{y}. \tag{2.40}$$

There is a wealth of examples of linear models for machine-learning force fields in the literature such as SNAP [45], MTP [37], ACE [38], and our JLP [5]. Each one of these differs in the choice of invariant descriptors $\mathcal{B}$ used.

Gaussian process regression

A possible limit of the invariant linear models discussed so far is that they rely on predicting the atomic energy contribution through a linear combination of hand-crafted invariant descriptors $\boldsymbol{\mathcal{B}}_I$. These descriptors are defined and computed independently from the data available in the training set. One might argue that there are potential patterns in the data that can be used by the model to improve upon the invariant descriptors designed so far. Gaussian process regression represents in some ways a generalisation of the Ridge regression in which now the features of the model are designed with the aid of the training set. We can rewrite Eq. (2.29) in a more compact form by introducing the convention $g_0 : (\mathcal{B}_{I,j}) = 1$

$$E_I(\boldsymbol{\mathcal{B}}) = \sum_{j=0}^{M} w_j^{(k_I)}\, g_j\left(\mathcal{B}_{I,j}\right). \tag{2.41}$$

The weights w_j determined during the fitting are the expansion coefficient of the atomic energy over the basis functions g_j. We can treat the estimation of the weights coefficients as a probabilistic process by associating them with a prior probability distribution [46]. If we assume that the prior probability of each weight w_j is Gaussian

with zero mean and standard deviation σ_w this induces a probabilistic treatment of the estimators $E_I(\boldsymbol{\mathcal{B}})$. Within this view, we can then compute the correlation of the atomic energy with respect to two chemical environments:

$$\begin{aligned}\langle E_I(\boldsymbol{\mathcal{B}}), E_I(\boldsymbol{\mathcal{B}'})\rangle &= \int \mathrm{d}\mathbf{w}\, P(\mathbf{w}) \sum_{j=0}^{M} w_j^{(k_I)}\, g_j\left(\mathcal{B}_{I,j}\right) \sum_{j'=0}^{M} w_{j'}^{(k_I)}\, g_{j'}\left(\mathcal{B}'_{I,j'}\right) \\ &= \sum_{j,j'=0}^{M} g_j\left(\mathcal{B}_{I,j}\right) g_{j'}\left(\mathcal{B}'_{I,j'}\right) \int \mathrm{d}\mathbf{w}\, P(\mathbf{w}) w_j^{(k_I)} w_{j'}^{(k_I)} \end{aligned} \quad (2.42)$$

given the assumption that $P(\mathbf{w}) \propto \mathrm{Normal}(\mathbf{w}, \mathbf{0}, \sigma_w I)$ the integral in this equation is proportional to $\sigma_w^2 \delta j j'$ resulting in

$$\langle E_I(\boldsymbol{\mathcal{B}}), E_I(\boldsymbol{\mathcal{B}'})\rangle = C(\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{B}'}) = \sigma_w^2 \sum_{j=0}^{M} g_j\left(\mathcal{B}_{I,j}\right) g_j\left(\mathcal{B}'_{I,j}\right), \quad (2.43)$$

The function C takes the name of kernel function and it can be understood as providing a similarity measure between two atomic chemical environments [47]. Moreover, when a measurement is performed we can assume that a random Gaussian noise ε with zero mean and variance σ^2 is added

$$E_I^{(i)} = E_I^{(i)}(\boldsymbol{\mathcal{B}}) + \varepsilon \quad (2.44)$$

This variable results from the sum of two normally distributed random variables, hence, its probability distribution will also be normal. The joint distribution of a set of observations $\mathbf{E}_I = (E_I^{(1)}, \ldots, E_I^{(M)})$ (which will coincide with our training set) is therefore a multivariate Gaussian with mean $\boldsymbol{\mu}_I$ and covariance $C_M + \sigma^2 I$ where the matrix

$$C_M = \begin{bmatrix} C(\boldsymbol{B}_I^{(1)}, \boldsymbol{B}_I^{(1)}) & \ldots & C(\boldsymbol{B}_I^{(1)}, \boldsymbol{B}_I^{(M)}) \\ \vdots & \ddots & \vdots \\ C(\boldsymbol{B}_I^{(M)}, \boldsymbol{B}_I^{(1)}) & \ldots & C(\boldsymbol{B}_I^{(M)}, \boldsymbol{B}_I^{(M)}) \end{bmatrix}, \quad (2.45)$$

contains the kernel function values between each pair of environments in the set of observations $\mathbf{B}_I = (\boldsymbol{B}_I^{(1)}, \ldots, \boldsymbol{B}_I^{(M)})$ associated to the atomic energies $\mathbf{E}_I$. Without loss of generality, we can assume that the mean $\boldsymbol{\mu}_I$ is zero. If this is not the case such a mean value can be estimated from the set of observations and subtracted. With these assumptions given these previous M observations, the Bayes conditional probability associated with a new observation, $E_I^{(M+1)}$ in correspondence of a new chemical environment $\boldsymbol{B}_I^{(M+1)}$ is

$$P(E_I^{(M+1)} | \mathbf{E}_I) = \frac{P(E_I^{(1)}, \ldots, E_I^{(M)}, E_I^{(M+1)})}{P(E_I^{(1)}, \ldots, E_I^{(M)})} \quad (2.46)$$

which is again a normal distribution, given our assumptions

$$P(E_I^{(M+1)}|\mathbf{E}_I) =\propto \exp\left(-\frac{1}{2}\left[\mathbf{E}_I\, E_I^{(M+1)}\right](C_{M+1}+\sigma I)^{-1}\begin{bmatrix}\mathbf{E}_I\\ E_I^{(M+1)}\end{bmatrix}\right) \tag{2.47}$$

with

$$C_{M+1} = \begin{bmatrix} C_M & \mathbf{c}_M \\ \mathbf{c}_M^T & \kappa \end{bmatrix}, \tag{2.48}$$

where $\mathbf{c}_M^T = (C(\boldsymbol{B}_I^{(1)}, \boldsymbol{B}_I^{(M+1)}), \ldots, C(\boldsymbol{B}_I^{(M)}, \boldsymbol{B}_I^{(M+1)}))^T$ and $\kappa = C(\boldsymbol{B}_I^{(M+1)}, \boldsymbol{B}_I^{(M+1)})$. To predict the mean value of prediction at a new point it is not required to invert C_{M+1}. From Eq. (2.47) we can infer the mean and variance of the distribution, which, after some algebraic manipulation [48], turn out to be

$$E_I^{(M+1)} = \mathbf{c}_M^T(C_M+\sigma^2 I)^{-1}\mathbf{E}_I + \mu_I, \tag{2.49}$$

$$\mathrm{var}(E_I^{(M+1)}) = \kappa - \mathbf{c}_M^T(C_M+\sigma^2 I)^{-1}\mathbf{c}_M. \tag{2.50}$$

Eq. 2.49 can be used for predictions given a training set of observations $(\mathbf{B}_I, \mathbf{E}_I)$ and Eq. 2.50 provides us with an estimation of the error on the prediction of the model. The model is completely determined by the calculation of $(C_M+\sigma^2 I)^{-1}\mathbf{E}_I$ and takes the name of Gaussian Process Regression (GPR) [48]. Notably Eq. (2.49) and (2.50) do not require the knowledge of the basis functions g_j used in Eq. (2.41) if the specific form of the kernel is given. In practice, *ab-initio* calculations provide the total energy of the system and not access to the single atomic contributions. We can adapt Eq. (2.49) for the fitting of the total energy of the system. Let us consider a training set containing M configurations of a system of N atoms of K different species and their corresponding total energy. We can group the local atomic environment with respect to the chemical identity of the central atom. Each group will then contain M_k local environments associated with the same species. The GPR total energy prediction is then given by

$$\begin{aligned} E_{Tot}^{(M+1)} &= \sum_{I=1}^{N} E_I^{(M+1)} \\ &= \sum_{k=1}^{K}\left(\mu^{(k)}N_k + \sum_{m=1}^{M_k}\left[(C_{M_k}+\sigma^2 I)^{-1}\mathbf{E}_{k_I}\right]_m \sum_{I|k_I=k} C(\boldsymbol{B}^{(m,k)}, \boldsymbol{B}_I^{(M+1)})\right), \end{aligned} \tag{2.51}$$

here we made μ_I depend only on the chemical identity of I to impose on the model invariance with respect to the exchange of identical particles. By comparing Eq. (2.51) with Eq. (2.31) and (2.40) is apparent that a Gaussian Process Regression is equivalent to a Ridge regression which uses as descriptors the kernel similarity with respect to the environments in the training set. The number of learning parameters of the model scales with the size of the training set. Among the most successful machine-learning potentials based on Gaussian process regression are the Gaussian approximation potential (GAP)

[49] and the Smooth Overlap of Atomic Positions (SOAP) [34]. These methods differ with respect to the kernel used, for example, the kernel similarity used by GAP has the form,

$$C(\mathbf{B}, \mathbf{B}') = \exp\left(-\sum_j \frac{(B_j - B'_j)^2}{2\sigma_j^2}\right), \tag{2.52}$$

where $\mathbf{B}$ and $\mathbf{B}'$ are the bispectrum component relative to two different environments.

Neural networks

Another model architecture, that also aims at improving the invariant handcrafted descriptors by leveraging the patterns in the data seen during training, is the neural network. Feedforward fully connected neural networks, also called multi-layer perceptron (MLP) address this problem in a general and flexible way. This is done by refining through multiple updates the initial input features passed to the model. Starting with the set of hand-crafted descriptors:

$$\mathbf{h}_I^{(0)} = \boldsymbol{\mathcal{B}}_I, \tag{2.53}$$

We define the following update rule:

$$\mathbf{h}_I^{(l+1)} = \boldsymbol{\phi}^{(l,k_I)}(\mathbf{h}_I^{(l)}) = \sigma^{(l)}(\mathbf{w}_0^{(l,k_I)} + \mathcal{W}^{(l,k_I)} \cdot \mathbf{h}_I^{(l)}), \tag{2.54}$$

with σ non-linear function called activation function which is here intended to be applied element-wise to each element of its argument. This update rule, together with the initial condition of Eq. (2.53), allows one to compute $\mathbf{h}_I^{(l+1)}$ for any l given the weights $(\{\mathbf{w}_0\}, \{\mathcal{W}\})$. The weight matrices $W^{(l,K_I)}$ have dimensions $n_{l+1} \times n_l$ where n_l takes the name of number of nodes of the layer l. The total number of times this update is repeated is called the number of hidden layers of the network (L). the vectors $\mathbf{h}_I^{(l)}$ of size n_l are called embeddings. The total number of learning weights N_w of the MLP is determined by the number of layers and the number of nodes in each layer. Notably, the weights $(\{\mathbf{w}_0\}, \{\mathcal{W}\})$ depend only on the species of the atom I and not on I itself. Atoms with the same chemical identity share the weights of the network maintaining the model invariant with respect to their exchange. At the same time, if the input features $\mathbf{h}_0$ are roto-translational invariant functions of the Cartesian coordinates of the system then the whole model will be invariant with respect to such transformations. For this reason, all the invariant atomic environment representations described in the previous paragraph are an optimal choice as input features of MLPs. The total energy of the system is then given by:

$$E_{Tot} = \sum_I^N E_I = \sum_I^N h_I^L = \sum_I^N \boldsymbol{\phi}^{(L,k_I)} \circ (\cdots \circ (\boldsymbol{\phi}^{(1,k_I)}(\boldsymbol{\mathcal{B}}_I))). \tag{2.55}$$

We can then infer the forces from the gradient of the energy. Since the MLP is defined as multiple compositions of differentiable non-linear update functions $\{\boldsymbol{\phi}\}$ we can use the chain rule to compute these partial derivatives:

$$F_I = -\sum_J^N \frac{\partial}{\partial \vec{R}_I} \boldsymbol{\phi}^{(L,k_J)} \circ (\cdots \circ (\boldsymbol{\phi}^{(1,k_J)}(\boldsymbol{\mathcal{B}}_J))) = -\sum_J^N \frac{\partial \boldsymbol{\phi}^{(L,k_J)}}{\partial \mathbf{h}_I^{(L-1)}} \cdots \frac{\partial \boldsymbol{\phi}^{(1,k_J)}}{\partial \mathbf{h}_I^{(0)}} \cdot \frac{\partial \boldsymbol{\mathcal{B}}_J}{\partial \vec{R}_I}. \quad (2.56)$$

The last term of the chain represents the gradient of the invariant descriptors with respect to the Cartesian coordinates, for which an analytical form is available. The other terms can be easily computed by means of auto-differentiation algorithms allowing for any change in the number of nodes or layers which can then be treated as hyperparameters of the model. The stress tensor components can finally be computed from Eq. (2.10) and Eq. (2.56). Once again the optimal training weights of the model are ideally chosen as the ones that minimize the loss of Eq. (2.28). Contrary to the linear model case the loss is now a non-convex function of the learning weights, hence, it can have multiple local minima which are not guaranteed to correspond to global minima. Moreover, this time, we cannot find a closed-form solution for the values of the weights for which the gradient of the loss is zero. The minimisation of the loss is then performed iteratively leveraging the differentiability of the loss using gradient-based optimisation strategies such as stochastic gradient descent [50].

Graph models

Graphs represent a natural framework for modelling molecules and crystals. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a collection of nodes $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ between pair of nodes [32]. The natural graph representation for atomic systems arises by matching each atom with a node and identifying the presence of bonds with the edges of the graph. Neural networks can be extended to operate on graphs [51], within this framework they inherit invariance under all graph isomorphism which directly translates, for example, to the invariance with respect to the exchange of atoms of the same species. Within a message passing formalism each layer of a graph neural network updates the hidden embeddings $\mathbf{h}_I^{(l)}$ in two phases. At first the messages $\mathbf{m}_I^{l+1}$ associated with each node are updated by gathering the output signals of the message functions $\boldsymbol{\phi}_e^{(l,k_I)}$ coming from the atoms in the neighbourhood $\mathcal{N}(I)$:

$$\mathbf{m}_I^{l+1} = \sum_{J \in \mathcal{N}(I)} \mathbf{m}_{IJ} = \sum_{J \in \mathcal{N}(I)} \boldsymbol{\phi}_e^{(l,k_I)}(\mathbf{h}_I^{(l)}, \mathbf{h}_J^{(l)}, a_{IJ}), \quad (2.57)$$

where a_{IJ} are called edge features and can be included to provide further information associated with the bond between atoms I and J. Finally, the nodes embedding are updated through the vertex update function $\boldsymbol{\phi}_h^{(l,k_I)}$ using the messages computed in the previous step:

$$\mathbf{h}_I^{(l+1)} = \boldsymbol{\phi}_h^{(l,k_I)}(\mathbf{h}_I^{(l)}, \mathbf{m}_I^{l+1}). \quad (2.58)$$

The total energy of the system is obtained by summing for all nodes of the graph the corresponding scalar hidden embedding associated with the last layer:

$$E_{Tot} = \sum_{I}^{N} E_I = \sum_{I}^{N} h_I^L \tag{2.59}$$

The first instance of a graph neural network interatomic potential was provided by Schnet [52] in 2018. Since then the number of potentials relying on a graph architecture has significantly increased with promising models such as M3GNET [53] and equivariant models such as Tensor Field Networks (TFN) [54], NequIP [55] and Equivariant Graph Neural Networks (EGNN) [56].

Equivariant models

So far we have discussed what are called invariant models, after the computation of the roto-translationally invariant features the whole model is unperturbed by any rotation or translation of the input structure. One might regard this approach as potentially flawed in regard to the fact that immediately building invariants from the input coordinates prevents the learning weights of the model from being exposed to potential information that might be extracted from the data. At the same time, as pointed out previously, giving no guidance to the model and using directly the Cartesian Coordinates as input is asking the model to infer too much from the data and likely leads to poor performance. A somewhat middle ground is provided by the so-called equivariant models which are the focus of this paragraph. Given an abstract group G (e.g. the group of rotations), we say that a function $f : X \to Y$ is equivariant with respect to the group G if $\forall g \in G$ and $T_g : X \to X$ there exists an equivalent transformation $S_g : Y \to Y$ such that:

$$f(T_g(x)) = S_g(f(x)), \tag{2.60}$$

Equivariant models are models that include equivariant transformations inside their architecture. A straightforward example of this is provided by the equivariant graph neural networks (EGNN) [56]. In EGNN additional equivariant embeddings called coordinate embeddings ($\mathbf{x}_I^{(l)}$) are introduced. They are initialised to be the same as the atomic coordinates of the atom I:

$$\mathbf{x}_I^{(0)} = \vec{R}_I \tag{2.61}$$

and updated accordingly to the following equivariant rule [56]:

$$\mathbf{x}_I^{(l+1)} = \mathbf{x}_I^{(l)} + \frac{1}{(N-1)} \sum_{J \neq I} \left(\mathbf{x}_I^{(l)} - \mathbf{x}_J^{(l)} \right) \phi_x^{(l,k_I)}(\mathbf{m}_{IJ}), \tag{2.62}$$

Name	Descriptors	Architecture	Equivariant	Year	Ref.
BP	Symmetry functions	MLP	✗	2007	[33]
GAP	Bispectrum	GPR	✗	2009	[49]
SOAP	Power spectrum	GPR	✗	2013	[34]
SNAP	Bispectrum	Linear	✗	2015	[45]
MTP	Moment tensors	Linear	✗	2016	[37]
ANI	Symmetry functions	MLP	✗	2017	[58]
GDML	Global kernel representation	GPR	✗	2017	[59]
q-SNAP	Quadratic bispectrum	Linear	✗	2018	[35]
Schnet	Radial basis functions	Graph	✗	2018	[52]
sGDML	Global kernel representation	GPR	✗	2018	[60]
TFN	Radial basis functions	Graph	✓	2018	[54]
DeepMD	Local system of reference	MLP	✗	2018	[61]
ACE	Bispectrum generalisation	Linear	✗	2019	[38]
MEGNET	Two body representation	Graph	✗	2019	[62]
PACE	Bispectrum generalisation	Linear	✗	2021	[39]
PaiNN	Radial basis functions	Graph	✓	2021	[63]
SpinConv	Local system of reference	Graph	✗	2021	[64]
JLP	Jacobi-Legendre polynomials	Linear	✗	2022	[5]
EDDP	Symmetry functions	MLP	✗	2022	[65]
NequIP	Radial basis functions	Graph	✓	2022	[55]
EGNN	Radial basis functions	Graph	✓	2022	[56]
M3GNET	Three body representation	Graph	✗	2022	[53]
POD	Proper orthogonal descriptors	Linear	✗	2023	[43]

Table 2.1: List of machine-learning force fields that have been developed in recent years. The number of models based on graph architectures has seen an increase.

where the scalar function $\phi_x^{(l,k_I)}$ are introduced to convert the messages $\mathbf{m}_{IJ}$ between the nodes I and J into scalars. The message update rule of Eq. (2.57) is then changed to depend on the coordinate embeddings:

$$\mathbf{m}_I^{l+1} = \sum_{J \neq I} \mathbf{m}_{IJ}^{l+1} = \sum_{J \neq I} \boldsymbol{\phi}_e^{(l,k_I)}(\mathbf{h}_I^{(l)}, \mathbf{h}_J^{(l)}, \|\mathbf{x}_I^{(l)} - \mathbf{x}_J^{(l)}\|^2, a_{IJ}), \tag{2.63}$$

while the hidden embeddings update is kept the same as Eq. (2.58). Models such as TFN [54], NequIP [55] and EGNN [56] are interatomic potentials that target an invariant quantity, the total energy of the system, while containing equivariant layer transformations. Conversely, other models such as Equi-SNAP [57] directly target tensorial quantities. Recently, equivariant models are gaining popularity and showing state-of-the-art performances while being data efficient [54, 55, 56].

2.3.3 Which model is the best?

Table 2.1 contains a list of most of the machine-learning force field potentials that appeared in the literature in recent years. This should provide a feeling of the "Cambrian explosion" that has affected this area of research. Recent models tend to focus

more on graph architectures and the presence of equivariant models has increased with time. There is a certain delay between the design of new potentials and their adoption in novel applications. This delay correlates mostly with their ease of implementation. The mathematical intensive nature of roto-translational invariant descriptors of the chemical environment has made their implementation particularly challenging. For years the availability of these descriptors was confined to the code base developed by the research groups that originally designed them, therefore separated from the mainstream molecular dynamics software. The trained model needs to interface with a scalable molecular dynamic code such as LAMMPS [66] to reach widespread usage, adding further overload on the implementation side.

Ultimately what determines the success of a particular model is a combination of accuracy, performance and accessibility. On top of this, different descriptors can be used along different model architectures. An example of this is the bispectrum components which have been used along with Gaussian process regression, linear models and multilayer perceptron [49, 45, 67]. This increases the "phase space" of possible model design making it particularly challenging to perform quantitative performance comparisons and quantitatively establish a winner, if one exists at all. Once the code base is laid out, to construct a new model instance the user has the responsibility to construct a training set and perform the training.

A new paradigm has recently emerged with the MegNet models [62, 53] which are trained over a large portion of the materials project a database of DFT calculations [12]. These models can directly be used to perform relaxation with remarkable robustness on a given system without further training. As is often the case in computer science, the "best" MLFF models may simply be the ones that win the "hardware lottery" [68], thereby making the most effective use of the available hardware.

2.4 The limits of *ab-initio* simulations

While Density Functional Theory has provided a way to lower the computational cost associated with *ab-initio* simulations and machine-learning interatomic potentials have further boosted the computational efficiency of this theory, there are yet properties of interest for material design that remain inaccessible through simulation. This limitation can arise for various reasons which may differ on a case-by-case basis. In the following, we will provide some examples, namely the estimation of the Curie temperature for the ferromagnetic/paramagnetic transition and the estimation of the fundamental band-gap.

Curie Temperature

The simplest way to model the ferromagnetic transition consists of associating a local magnetic moment to each magnetic atom of the system and parameterising its energy

with a semi-classical Heisenberg Hamiltonian. The transition temperature is then extracted by the magnetisation curve as a function of the temperature, obtained via Monte Carlo sampling. A more detailed discussion of this problem will be presented in Chapter 6. The direct use of DFT for the estimation of T_C is made prohibitive by its computational cost and the need for large enough simulation supercells. These large supercells are required for the correct sampling of the spin-configurations phase space, which become disordered above the Curie temperature. It is then necessary to map the DFT calculations over a lattice Hamiltonian and then perform spin-lattice dynamics on this surrogate model. This approach can lead to T_C estimations largely different from the experimental values [69].

Band-gap

The Kohn-Sham theory can be generalised to account for the partial occupation of the Kohn-Sham orbitals [8]. Within this picture the Kohn-Sham orbital energies ε_i acquire a formal meaning in terms of the Janak theorem [70]

$$\frac{\partial E}{\partial n_i} = \varepsilon_i, \tag{2.64}$$

where n_i is the fractional occupancy of the Kohn-Sham orbital ψ_i at convergence. We can use this theorem to compute the electron affinity of the system

$$A(N) = E(N) - E(N+1) = -\int_0^1 \varepsilon_{\mathrm{LUMO}}(N+n)\,\mathrm{d}n, \tag{2.65}$$

here $\varepsilon_{\mathrm{LUMO}}$ is the Kohn-Sham energy of the lowest unoccupied orbital. Similarly, the ionisation potential is given by

$$I(N) = E(N-1) - E(N) = -\int_0^1 \varepsilon_{\mathrm{HOMO}}(N+n)\,\mathrm{d}n, \tag{2.66}$$

here $\varepsilon_{\mathrm{HOMO}}$ is the Kohn-Sham energy of the highest occupied orbital. In an infinite crystal, the addition or removal of an electron will produce an infinitesimal change in the ground-state density. The fundamental band-gap of the system can then be predicted as [71]

$$E_{gap} = I(N) - A(N) = \varepsilon_{\mathrm{LUMO}}(N) - \varepsilon_{\mathrm{HOMO}}(N). \tag{2.67}$$

Following this argument, it would be possible to calculate exactly the band-gap of a system via DFT using the Kohn-Sham theory. What is observed in practice is that the band-gap predictions done with DFT tend to significantly underestimate the experimental band-gap value. The reason for this inconsistency is due to the fact that the aforementioned argument overlooks the possibility that the Kohn-Sham potential might have a derivative discontinuity by a constant C in correspondence

with an integer number of electrons. Repeating the previous argument, allowing for a derivative discontinuity of the exchange potential leads to:

$$E_{gap} = \varepsilon_{\mathrm{LUMO}}(N) - \varepsilon_{\mathrm{HOMO}}(N) + C. \tag{2.68}$$

The true fundamental gap of the system is given by the Kohn-Sham band structure gap plus an unknown constant which is generally positive [71]. The presence of the derivative discontinuity explains the observed systematic underestimation of the band-gap by DFT with respect to the experimental band-gap. Ways to address this issue include the self-interaction correction and the use of hybrid potentials which mix the traditional DFT exchange-correlation potential with the exact Hartree-Fock exchange [71].

2.5 Machine learning on experimental data

In the previous section, we presented examples of properties for which no reliable and transferable method exists that would allow their high throughput simulation from first principles. However, due to the practical significance of these properties, it is desirable to find alternative methods for their prediction.

For example, for the ferromagnetic phase of a material to be stable and exhibit the properties of interest, its Curie temperature must be significantly higher than the operating temperature of the specific application under consideration. Otherwise, the material would lose its ferromagnetic properties during working conditions.

While these properties offer a significant challenge to simulations, the same is not always true for experimental measurements. A vast literature of experimentally measured band-gaps and Curie temperatures is available and one could think to use this data to train models that can directly predict the experimental properties of a given compound [22, 72]. This section is dedicated to this problem which is closely linked to the field of natural language processing by both the design of a suitable representation of the chemical compounds and the problem of automatically extracting the experimental data from the scientific literature.

2.5.1 Compositional descriptors

So far we have explored the use of machine-learning techniques to model interatomic potentials. In this context, the models are generally trained over a limited chemical space as the focus is on modelling the small energy variations with respect to atomic deformation. The only exception to this among all the models reported in Tab. 2.1 are MEGNET [62] and M3GNET [53], whose training set includes compounds that cover almost all the periodic table. However, this comes at the cost of accuracy when compared with a model trained on a specific compound.

When it comes to building machine-learning models trained to predict experimental

quantities, the ability to discern different crystal structures and small atomic displacements becomes somewhat less important for a number of reasons. When measuring a property of a compound, the measurement is often not coupled with a technique able to discern the crystal structure of the sample. As a consequence, the exact experimental crystal information can be unknown. Thermodynamical properties are insensitive to the fine details of the actual deformation from equilibrium associated with the atoms of the sample as they are the result of a thermodynamic average. For this reason, the use of descriptors designed for interatomic potentials might result in a model with poor performance as they are too sensitive to any variation of the atomic environment and contain an excessive amount of information which is ultimately not useful for this problem. Finally, experimental data are naturally based on compounds that can be stably synthesised as opposed to properties calculated through simulations which can also refer to compounds far from stability. This creates an implicit bias between the property and the crystal structure associated with a given chemical composition. Indeed, if, for example, we are considering the experimental band-gap of sodium chlorine, this will almost surely refer to a face-centered cubic crystal. All these considerations have led at first to disregard any specific structural information. We call compositional models such models that only rely on the information contained in the chemical formula of a compound to perform their prediction.

Once again the problem of finding the best way to represent this information in a form suitable for machine-learning models is a nontrivial one and a rich range of strategies has arisen. In the following, we will discuss three strategies that cover most of the cases found in the literature.

One-hot encoding

The simplest approach to representing a chemical formula consists of building a vector with as many elements as there are in the periodic tables. In this vector, all entries are set to zero except for those corresponding to the elements present in the compound, which are set to one. This convention to design features is called one-hot encoding and has seen large use in the machine-learning community to represent categorical quantities. However, the direct use of one-hot encodings to represent chemical formulas does not convey any information with regard to the relative concentration of each element, for this reason, a common choice is to fill the vector with fractional concentrations $\{x\}$ instead of the values one:

$$\mathbf{v} = (x_H, x_{He}, x_{Li}, \dots)\,. \tag{2.69}$$

There are some drawbacks to this approach, the resulting feature vectors tend to be sparse as most of the elements of the vector will be filled with zeroes. Moreover, no external knowledge about the elements present in the compound is introduced in the feature to help the model's inference.

Property based descriptors

It can be useful to introduce our prior knowledge associated with the different chemical species that appear in the periodic table in the features used by the compositional model. One possible approach consists of using the fractional weighted mean and standard deviation of a set of elemental properties which can be, for example, the atomic number, number of valence electrons, period, group, electronegativity, etc ... [24]. For a given property P we can then construct the following two features namely the fractional weighted average:

$$\langle P \rangle = \sum_i x_i \, P_i, \tag{2.70}$$

and the fractional weighted average deviation:

$$\Delta P = \sum_i x_i \, |P_i - \langle P \rangle|. \tag{2.71}$$

The number of features generated in this way can be increased by using other types of statistics and by considering additional elemental properties.

Literature based descriptors

This concept of informing the compositional features with the known properties of the elements can be brought to the extreme and use a machine-learning model that has been exposed to a large portion of the material science scientific literature to design such descriptors. The first instance of this approach has been provided by the mat2vec model [73] which gives us the opportunity to introduce the discussion about the use of natural language processing techniques in material science. This topic constitutes a significant part of this thesis and its fundamentals are covered in Section 2.6.

Compositional models architecures

Once a suitable representation for the chemical composition is chosen, a particular architecture needs to be selected to complete the model design. This choice extends to linear models, Gaussian process regression, neural networks, and even transformers [74]. However, given the challenge of generating a manually curated dataset of experimental data the training set of compositional models tends to contain a number of entries in the order of thousands. This limited size constrains the use of large deep-learning models.

Random forest algorithms are particularly robust in this data range and are often the best-performing model architecture [22]. A random forest classifier is an ensemble model that builds multiple binary decision trees on different subsets of the dataset. A decision tree is a non-parametric model consistent in a sequence of decision rules learned during training. In a binary tree, each decision rule has a binary outcome. There are two types of nodes constituting a decision tree namely decision nodes and

leaf nodes. Given an input described by n features $\mathbf{x} = (x_1, \ldots, x_n)^T$, in correspondence of a decision node m a rule-based decision is performed considering a particular feature j and a threshold t_m. The decision can be expressed in terms of an if-else statement and its result determines the next node to consider:

if $x_j \leq t_m$ **then**
| Go to left node;
else
| Go to right node;
end

The process is then repeated until a leaf node is reached. To each leaf node, there is an associated value $\bar{y}_m$ which corresponds to the output of the decision tree in correspondence with the input $\mathbf{x}$. What feature j and what threshold t_m to use for each decision is determined at training time together with the output values $\bar{y}_m$ associated with each leaf node.

Given a training set of N descriptors-target pairs $T = \{(\mathbf{x}, y)\}$ the regression is performed by a recursive partition of the feature space dictated by decision nodes whose parameters are determined with a greedy approach. The first partition is determined by the split (j, t_m) that minimises the mean square error loss

$$\mathrm{MSE}_{split}(j, t_m) = \frac{N_{left}}{N}\mathrm{MSE}(T_{left}) + \frac{N_{right}}{N}\mathrm{MSE}(T_{right}), \tag{2.72}$$

Where $T_{left} = \{(\mathbf{x}, y) \in T | x_j \leq t_m\}$ and $T_{right} = T/T_{left}$ are the subsets of size N_{left} and N_{right} respectively, resulting from the split of the training set. The MSE on a subsets $T_m \subseteq T$ of size N_m is defined as

$$\mathrm{MSE}(T_m) = \frac{1}{N_m} \sum_{\{i|(\mathbf{x}_i, y_i) \in T_m\}} (y_i - \bar{y}_m)^2 , \tag{2.73}$$

with

$$\bar{y}_m = \frac{1}{N_m} \sum_{\{i|(\mathbf{x}_i, y_i) \in T_m\}} (y_i) . \tag{2.74}$$

After the optimal split that minimises Eq. (2.72) is found, the process is reiterated on each of the child partitions T_{left} and T_{right}. The growth of the tree continues until a halting condition is met, for example, when a maximum depth of the tree is reached or when the partition of the original training set at that node contains only one element. The meeting of one of the halting conditions leads to the creation of a leaf node with $\bar{y}_m$ determined by Eq. (2.74). The training is complete after each branch of the tree contains a leaf node.

Generally, decision trees are prone to over-fitting as they tend to partition the training set too finely. To limit this effect pruning is often performed after the training [75]. Another strategy to overcome the tendency of over-fitting consists of averaging the output of multiple trees trained on different partitions of the training set and with

random feature selection [76]. The resulting ensemble model takes the name of random forest.

Achievements of compositional models

Compositional models have seen a significant amount of success being able to produce models able to predict the Curie temperature of ferromagnets [22], the critical temperature of superconductors [77], and other experimental properties [24]. This is at first surprising, for example, the Curie temperature (T_C) of a ferromagnet modelled by a Heisenberg Hamiltonian, is governed by the functional form of the exchange parameter which in turn heavily depends on the crystal structure of the system [78]. However, the training of the compositional model is performed over experimental measurement of the Tc implying that these temperatures were measured in correspondence of a stable crystal structure of the compound. This, as already mentioned, creates an indirect link between a composition and its equilibrium crystal structure, a relationship that the model can potentially learn.

At the same time, the choice of only considering the chemical formula of a compound to infer its properties comes obviously with some limitations. The most explicit of which is related to the case of isomorphism where a given compound shows multiple stable phases with different crystal structures and hence different properties. However, of the compounds that display isomorphism, this would constitute a problem only for the cases in which the property variation over the different phases is larger than the uncertainty of the model. In practice, the ability of compositional models to work on a vast range of chemical compositions and their reliance on only the stoichiometry makes these models an invaluable tool for performing an initial coarse screening of the chemical space. Moreover, sometimes they are the only available option as the exact crystal structure of a sample undergoing a measurement is not always known.

2.6 Language models in material science

As discussed in the previous section databases of experimental measurements can be a valuable resource for material science. Due to the low throughput and cost of manual curation, efforts have been made to automate this information extraction process. For a long time, the state of the art in processing scientific text was held by rule-based methods which rely on the formulaic syntax of the scientific language to extract information by matching sentences with the use of hand-made grammar rules defined through regular expressions. The most popular system used for this task is ChemDataExtractor [79]. The main drawback of these approaches is that their performance is ultimately determined by the definition of these rules and on how much the processed text adheres to them.

Given the large quantity of data in natural language form generated and shared ev-

ery day on the internet, numerous data-driven solutions to this problem have recently emerged. One strategy consists of designing and training a model on a portion of this data to automatically learn a text representation suitable for machine learning. This approach was initiated by Google Research with the release of the wor2vec models in 2013 [80]. Once again we are faced with a representation problem, in this case how to achieve a Euclidian representation of natural language. A model that provides such representation is called a language model. The first language models would assign to each word seen during training a representation that is independent of the context from which that word appeared. These types of models are said to provide a static word representation since once the model is trained it will output always the same set of values for a given input word. More recently the use of recurrent neural networks (RNN) has made it possible to design language models that offer contextual word representations. In this approach, the output embeddings generated by the model for a given word are a function of the other words present in the sentence where it appears. Such contextual models allow us to distinguish words that have different meanings in different contexts and provide significantly better performance in downstream tasks. The architecture of these models settled after the introduction of the transformer architecture in 2017 [81]. This architecture has allowed the design of contextual language models that significantly outperformed their previous iterations.

Finally, in the last few years, after the release of GPT-2 [82] in 2018 and more so after GPT-3 [83] in 2020, the focus of the field has shifted towards generative language models. Generative language models are trained to predict the most likely word following a sequence of text. This trend is due to the observation that increasing the size of these models and exposing them during training to a significant portion of the internet would give rise to a set of emergent capabilities. These capabilities are an indirect result of the training objective and consist of what is called few-shot learning. Since the model is trained to complete any piece of text in the most likely way possible, it is possible to "program" the completion performed by providing a small set of examples of the wanted outcome. The model, by trying to complete the document in the most likely way, will then often produce the wanted output. The importance of this paradigm relies upon the fact that these models can now be adapted to perform a vast range of tasks by showing them just a handful of examples.

This way of designing the input prompt to a generative language model in order for it to complete it in the wanted way is called prompting engineering. Given the nature of these models designing the right prompt can be particularly challenging and small differences can result in significantly different outputs. To address this problem recent iterations of language models are "instructed". An instructed version of a language model is a generative language model whose weights have been further optimised by means of reinforcement learning based on a score model trained on human feedback [84]. These models, while less powerful than the base models from which they

are derived, are significantly more intuitive to prompt by non-experts and resulted in products accessible to the general public such as ChatGPT [85]. The following is a brief overview of the implementation details related to the main concepts qualitatively discussed in this paragraph.

2.6.1 Static word representation

As we already faced multiple times, finding a suitable description of a problem in order to apply the known machine-learning techniques is always a nontrivial one. Ideally, in this case, we want to find a map that connects each word in the English dictionary with a vector in the Euclidean space $\mathbb{R}^d$. The resulting vector constitutes a representation of the word and sometimes its interchangeably called embedding. In order for a model built on such representation to be able to draw information related to the underlying structure of the language, as vectors in the Euclidean space, certain geometric properties should be retained. In particular, the embedding of words relative to unrelated concepts should be orthogonal with each other, while the vectorial representations of synonyms and related words should point in similar directions. If this structure persists then the similarity between two words results proportional to the dot product of their representation. Such representation can be obtained in several ways, here we discuss the skip-gram language model [80].

Initially, a vocabulary is constructed containing all the words appearing at least once or above a certain number of times in the corpus considered. The size of the vocabulary $N_{voc.}$ will depend on the corpus chosen and on the language considered. These models tend to be trained over a large portion of high-quality text extracted from the internet (Wikipedia, BooksCorpus, etc...) resulting in vocabulary sizes in the range of $10^5 - 10^7$ terms [86]. The input representation to the model $\mathbf{b}_{w_I}$ of a word w is provided by a one-hot encoding with respect to the dictionary, a vector with the same size as the vocabulary and with all elements set to zero except to the position corresponding to w. During training, given the sentence $w_1, w_2, \ldots, w_S$, for each one of its words, the model tries to predict the C words that are most likely to appear before and after it. This translates into maximising the average log probability:

$$\frac{1}{S}\sum_{s=1}^{S}\sum_{-C\leq i\leq C, i\neq 0}\log\left(p(w_{s+i}|w_s)\right), \tag{2.75}$$

where the probabilities $p(w_{s+i}|w_s)$ are defined using the softmax function:

$$p(w_{s+i}|w_s) = \frac{\exp\left(\mathbf{b}'_{w_O}\cdot\mathbf{b}_{w_I}\right)}{\sum_{w=1}^{N_{voc.}}\exp\left(\mathbf{b}'_{w}\cdot\mathbf{b}_{w_I}\right)}, \tag{2.76}$$

where $\mathbf{b}_w$ is the one-hot encoding input vector representation of the word w while $\mathbf{b}'_w$ is the learned output representation of w of the skip-gram model. The use of Eq. (2.76)

is impractical due to the large size of the vocabulary, hence more computationally efficient approximations are required instead [86]. Once trained the skip-gram model will produce a continuous vector representation for each word in the dictionary. Such representation is said static since the model will generate the same embedding for a given word regardless of the particular context in which the word appears.

Other examples of popular language model architectures for static word representations are the continuous bag of words (CBOW) and GloVe [80, 87]. As a side effect of the training process, the static word representations generated by these neural network models preserve many linguistic regularities and patterns [80] which are made explicit in the geometric relations between these vectors. For example:

$$Paris = \max_{w \in Vocab.} \left(\mathbf{b}'_w \cdot \left(\mathbf{b}'_{Madrid} - \mathbf{b}'_{Spain} + \mathbf{b}'_{France} \right) \right). \tag{2.77}$$

This property allows the use of the model to resolve analogy pairs such as "Man" is to "Woman" as "King" is to "x" where the prediction of "x" is computed as:

$$\text{“}x\text{”} = \max_{w \in Vocab.} \left(\mathbf{b}'_w \cdot \left(\mathbf{b}'_{Woman} - \mathbf{b}'_{Men} + \mathbf{b}'_{King} \right) \right), \tag{2.78}$$

and would result in the word "Queen". In 2019, Tshitoyan *et al.* [73] trained a skip-gram model, which they called mat2vec, specifically on a corpus focusing on material science. They then showed that the resulting model is able to resolve material-related analogies, for example, the analogy "NiFe" is to "ferromagnetic" as "IrMn" is to "x" is solved by mat2vec for x="antiferromagnetic". Furthermore, it was shown that using the mat2vec embeddings as input representation for machine-learning models results in competitive compositional models opening a new venue for the design of compositional descriptors informed by the scientific literature. Mat2vec embeddings are also used in CrabNet one of the state-of-the-art architectures for compositional models [74].

2.6.2 Contextual word representation

The main limit of static word representation is associated with the fact that they are indeed static. The embedding associated with the word "I" in a given sentence would be the same regardless of whether this word refers to a pronoun or the chemical symbol of iodine. Models able to generate a different vector representation of a word depending on its context are called contextual. The first examples of models able to provide contextual word embeddings were based on recursive architectures such as recurrent neural networks (RNN) [20] and Long Short-Term Memory networks (LSTM) [88]. RNNs are based on applying a neural network iteratively on each element of a sequence. At every iteration, the network takes as input the current element of the sequence and the output from the previous iteration. A key aspect of this implementation is weight sharing. The network uses the same parameters in each recursion allowing the information from the previous steps to be maintained and making the architecture

able to process input sequences of arbitrary size. At the same time, weight sharing can make the training particularly challenging as the gradients of the loss tend to either explode or vanish over long sequences [19]. Moreover, RNNs are afflicted by information loss during the processing of long sequences as the influence of early input is lost over successive iterations of the network. These iterative models are inherently non-parallelisable making their training computationally expensive. These limits are addressed by the transformer architecture we discuss in the rest of this section.

The transformer block

The transformer block introduced in 2017 [81] was designed to present an alternative to RNN in the creation of contextual language models. The main drawback of recurrent models is their sequential nature which limits the possibility of fully exploiting the parallelisation potential provided by GPUs. In this regard, one could say that the main reason behind the success of the transformer is related to the fact that they won the hardware lottery [68] as they are the architecture that makes better use of the available hardware. A testament to the quality of the choices made in its initial design is that, despite the extensive research performed on the transformer ever since its release, its more recent iterations maintain fundamentally the same components. The most important element of the transformer block is the multi-head attention.

The attention mechanism is used as a way to remove the necessity of a recurrent model. To make the discussion more clear we can consider the sentence "The cat is on the table". Let us assume that we have access to an initial static word representation of dimension d_b for each word in the sentence: $B^T = (\mathbf{b}_{The}, \mathbf{b}_{cat}, \mathbf{b}_{is}, \mathbf{b}_{on}, \mathbf{b}_{the}, \mathbf{b}_{table})^T$. In its most simple form, the self-attention mechanism acting on this sentence would produce a contextual word representation for the word "cat" in this sentence by performing the following operation:

$$\mathbf{a}_{cat} = \text{Sum}\left[\text{softmax}\left(\frac{1}{\sqrt{d_b}}\begin{pmatrix}\mathbf{b}_{cat}\cdot\mathbf{b}_{The}\\ \mathbf{b}_{cat}\cdot\mathbf{b}_{cat}\\ \mathbf{b}_{cat}\cdot\mathbf{b}_{is}\\ \mathbf{b}_{cat}\cdot\mathbf{b}_{on}\\ \mathbf{b}_{cat}\cdot\mathbf{b}_{the}\\ \mathbf{b}_{cat}\cdot\mathbf{b}_{table}\end{pmatrix}\right)\odot\begin{pmatrix}\mathbf{b}_{The}\\ \mathbf{b}_{cat}\\ \mathbf{b}_{is}\\ \mathbf{b}_{on}\\ \mathbf{b}_{the}\\ \mathbf{b}_{table}\end{pmatrix}\right],$$

which we can rewrite in a more compact form as:

$$\text{Attention}\,(\text{“}cat\text{”}, B, B) = \text{softmax}\left(\frac{B\mathbf{b}_{cat}}{\sqrt{d_b}}\right)B. \tag{2.79}$$

The dot product operation $\mathbf{b}_{cat}B^T$ is proportional to the cosine similarity between the vector representation of the word "cat" and the other words in the sentence. The softmax operation selects in a differentiable way the pairs for which the overlap between

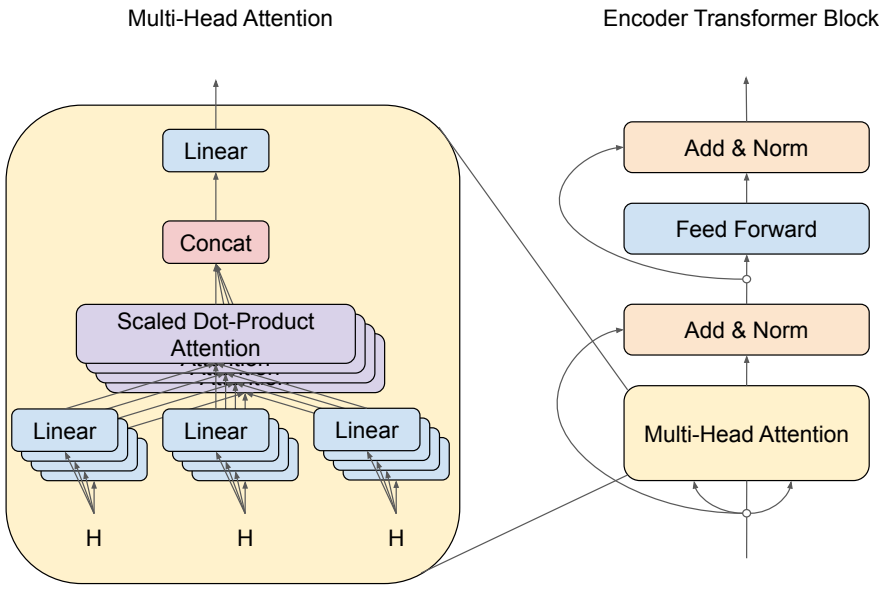


Figure 2.1: Diagrams showing the multi-head attention mechanism on the left and transformer block on the right. These schemes are based on [81]. See the text for more details.

the two vectors is maximum. Finally, the embeddings of each word in the sentence are scaled by the corresponding outputs of the softmax. The resulting embeddings weighted sum generates a contextual representation of the word "cat" within the considered sentence. This operation can then be repeated with respect to the other words in the sentence.

The attention mechanism in itself does not contain free parameters that can be optimised during the training process. In practice, a linear transformation, learned at training time, is performed separately on each argument before entering the attention mechanism:

$$\begin{aligned}\text{Attention}\,(Q,K,V) &= \text{Attention}\,(W_Q B, W_K B, W_V B)\\ &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (2.80)\end{aligned}$$

The arguments Q, K and V are called query, key and values respectively. The attention considered in this example is called scaled dot-product attention [81] other examples of attention functions can be found in the literature, and an extensive effort has been put in order to optimise the attention mechanism as it scales quadratically with the number of words (tokens) contained in the input [89]. In a transformer block the queries, keys and values are projected h times using different learned linear transformations resulting in what is called multi-head attention. Each head can be run in parallel and their outputs are then concatenated and passed through a linear transformation (see Fig.

2.1). To complete the transformer block, the output of the multi-head attention is added to its input and normalised. This results in a residual connection and improves the training of the model [90]. Finally, a fully connected feed-forward neural network transformation is performed with the same weights on each element, passed through another residual connection and normalised. This set of operations is named encoder transformer block (see Fig. 2.1). Its output will result in a set of contextual embeddings $B'^T = (\mathbf{b}'_{The}, \mathbf{b}'_{cat}, \mathbf{b}'_{is}, \mathbf{b}'_{on}, \mathbf{b}'_{the}, \mathbf{b}'_{table})^T$ which can be subsequently used as input of another transformer block.

In practice transformer models do not work in terms of words, they instead provide a representation for each input token. A token is the unit of text processed by this architecture and while it could coincide with words it has often been found to be more efficient to use subwords as tokens. The tokenisation process splits a sentence or a piece of text into its constituent tokens, the function governing this transformation is called tokenizer. The choice of the particular tokenizer used can strongly impact the final performance of the model.

Popular language models such as BERT and GPT consist of stacks of transformer blocks opportunely trained in a self-supervised setting. The training of these models over a large portion of the text scraped from the internet takes the name of pre-training and it is particularly computationally expensive. Pre-trained models can then be adapted to secondary tasks called downstream tasks. In the next two sections, we will focus on discussing the most widely used transformer models and emphasize their similarities and differences together with their use cases.

2.6.3 Masked language models

Masked language models represent a category of language models whose training objective consists of predicting masked tokens in a given sentence. The first and most popular transformer-based masked language model is BERT (Bidirectional Encoder Representations from Transformers) [91]. $\text{BERT}_{\text{BASE}}$ ($\text{BERT}_{\text{LARGE}}$) architecture consists of a stack of 12 (24) encoder transformers blocks for a total of 110M (340M) of learning parameters trained over a masked language modelling objective and a next sentence classification task. The training is performed over the BooksCorpus a large collection of free novel books [92] and English Wikipedia.

The masked language modelling objective consists of giving the model input sentences for which some percentage of the input tokens are replaced by a mask token. The model is then asked to predict such missing tokens and the weights are optimised with respect to this classification task. The bidirectional nature of the encoder transformer block results in the generation of a bidirectional contextual representation in which the output embedding of each token will depend on all the other tokens in the sentence. The resulting BERT model is then further trained for a next sentence prediction task. During this task, the model is presented with two sentences S_A and S_B where 50% of

the times S_B is the actual sentence in the corpus that follows S_A and 50% of the times S_B is a random sentence sampled from the corpus. The model is trained to classify if S_B is the correct sentence that should follow S_A or not. The reasoning behind including this objective is to teach the model an understanding of the relationship between sentences. Newer iterations of masked language models such as RoBERTa [93] tend to not include the next sentence prediction task and base the entire pre-training on only the masked language modelling objective.

Once trained, these models can be fine-tuned to perform different downstream tasks which we will discuss in more detail in Chapter 3. Training BERT on the entire English Wikipedia exposes it to a variety of topics, including chemistry. However, the lack of training on a tailored corpus involving papers regarding condensed matter physics is likely to negatively impact the models' performance on downstream tasks in this domain. Since, when it comes to material science applications, the corpus used to train BERT does not include an adequate representation of the scientific language and content found in materials science papers. Domain-specific pre-training can heavily improve the performance of downstream tasks related to that specific domain [94, 95, 96, 97, 98]. For example, MatSciBERT is a model based on the BERT architecture specifically pre-trained on material science publications. This model was created by continuing the pre-train of SciBERT [98], a BERT model trained from scratch on a corpus of 1.14 million papers randomly sampled from Semantic Scholar. The pre-training continuation that led to MatSciBERT was conducted on a new corpus of 150,000 papers, sourced from the Elsevier Science Direct Database. These papers were selected over various subdomains within material science, including bulk metallic glasses, inorganic glasses and ceramics, cement and concrete, as well as alloys. The pre-training process followed the same training protocol as RoBERTa [93]. In terms of performance on downstream tasks specific to material science, MatSciBERT outperforms both SciBERT and the original BERT model, highlighting the value of domain-specific pre-trained models in this field.

2.6.4 Generative Language models

The other main class of language models is the one of generative language models. The language modelling task is in this case addressed as an unsupervised distribution estimation from a set of sentences constituting the training set. Each sentence is a sequence of words $(w_1, w_2, \ldots, w_n)$ of variable length. Due to the presence of intrinsic patterns resulting from the grammar rules of the language used, the style and the context, the joint probability associated with the occurrence of a sentence can be factorised as a product of conditional probabilities of the next word given the previous ones [99]

$$p(\text{sentence}) = \prod_{i=1}^{n} p(w_i | w_1, \ldots w_{i-1}). \tag{2.81}$$

The unsupervised training of the model is then performed optimising the trainable parameters of the model to maximise the likelihood

$$L(\text{sentence}) = \sum_{i=1}^{n} \log(p(w_i|w_1, \dots w_{i-1})). \tag{2.82}$$

The most successful model architecture to perform this task has been the transformer [100, 82, 83]. For example, the Generative Pretrained Transformer models (GPT) are constituted by a sequence of decoder transformer blocks. Decoder transformer blocks differ with respect to their encoder counterpart for the presence of a causal masking of the attention. This masking imposes that the attention associated with a token is computed only including the previous tokens. The causal masking simulates the directional way in which we read text and decouples the predictions in correspondence of a token from the successive tokens. This allows one to simultaneously provide as many training examples as the number of tokens in a given sentence, making the training more efficient than for masked language models. This efficiency allows for training larger and larger models which also display better and better capabilities.

Instances of models with a number of trainable parameters of the order of 10-100 million are becoming more common [83, 101, 102, 103], as such these models are denominated Large Language Models (LLMs). For a given sequence of words, a trained LLM will produce a next-word prediction as a probability distribution over the dictionary produced by a softmax layer applied over the transformer output embeddings. The predicted word (token) is determined via random sampling over this distribution. In practice, this procedure would bias the next word prediction towards the statistically most used words leading to poor results. To solve this issue, the distribution is generally flattened out via a parametric rescaling controlled by a parameter named "temperature" guaranteeing more variability in the output. This output probability distribution can be conditioned by the appropriate input text.

Depending on the used input prompt, the model can be led to perform different downstream tasks that are not directly related to the unsupervised training objective [82, 83]. It has been shown [83] that by including in the prompt some examples of the addressed downstream task the performance of the model significantly increases. The setting where some example solutions of the task are provided in the prompt, generally between one and five, is called few-shot. If no examples are provided we name the setting zero-shot. The output of the LLM is significantly dependent on the prompt. While there are some heuristic rules for designing prompts that will give the best results on a task, how to design the optimal prompts is still an open problem. An approach that tries to address this issue is the creation of an instructed model via Reinforcement Learning from Human Feedback (RLHF) [84]. Within this scheme, multiple generated outputs are collected for a given prompt. Human annotators are then asked to rank the outputted completions from best to worst. The process is re-

peated over multiple prompts creating a labeled dataset of prompt completions and associated scores. This data is then used to train a score model which assigns a score given a prompt and a completion. This reward model is then used to fine-tune the initial LLM via reinforcement learning. The model resulting from this process is called an instructed LLM. Instructed LLMs are far easier to prompt than base LLM and they can be aligned to generate outputs that are in line with the user needs. A popular example of instructed LLM is ChatGPT.

2.7 Summary

In this chapter, we discussed the main methodologies behind our proposed data-driven inverse-design workflow. It is apparent that this task is extensively interdisciplinary in nature. Trying to use all the available data, which comes from high-throughput *ab-initio* simulations and experimental measurements, requires techniques from various branches of machine learning including deep learning and natural language processing. Most of these techniques rely on supervised training which requires a dataset of gold labels on which the learning is performed. These gold labels are generally computed with *ab-initio* methods of which DFT is the most popular choice. A model can only be as good as the data that are fed into it and as such particular attention should be paid in the cases where DFT is unreliable. In the following chapters, we will provide some use cases in the domain of material science for the techniques introduced so far.

Chapter 3

Composition selection

The content presented in this chapter is based on Ref. [1]. This work has been carried on with equal contribution by Luke Gilligan and the author of this thesis.

The first step towards the inverse design of a new material requires us to decide how many and which elements our prototype compounds should contain. Starting with all the elements contained in the periodic table, we generally exclude any radioactive isotopes and noble gases. The specific selection of elements will then depend on the specific target application. For example, it might be of interest to discard certain elements due to their toxicity, cost or availability as is often the case with rare earth elements. Additional criteria might arise from manufacturing considerations. For instance, in the case of alloys having constituents with similar melting points can simplify the synthesis process. Conversely, if one of the constituents has a significantly different melting point with respect to the others the realisation of that alloy could present a challenge. Once these criteria are established, we can define a pool of elements ($\mathcal{E} \subset$ PeriodicTable) that meets all the requirements.

At this stage of the inverse-design workflow, we only know the potential elements that might compose our target compound, we do not know its stoichiometry and nevertheless its crystal structure. What we might know are additional requirements demanded by the target application. For example, if the goal is to design a ferromagnet, we might specify that its Curie temperature must exceed a certain threshold. Similarly, if our goal is the design of a semiconductor we might want its band-gap to fall within a specific range to meet specific optical or electronic requirements.

Compositional models represent an invaluable tool for providing an initial rough estimation of the properties that are fundamental for the material that we are looking for. In the previous chapter, we detailed the theory behind compositional models, however, we have not yet discussed the origin of the data on which we can train these models. Depending on the target property that we want to model, it might or it might not be feasible to train over publicly available theoretical datasets of *ab-initio* calculations [13, 104, 12, 14]. If this property is not featured in such databases, or if

its calculation is known to be either highly inaccurate or computationally prohibitive, the only option left is to rely on available experimental data. While some successful examples validate the viability of this approach [22, 23, 105], the manual curation of an experimental database is labour-intensive and costly to scale.

As a consequence, current experimental databases are not comprehensive. They often list only a limited set of properties for each compound such as the crystal [106, 107, 108] or magnetic structure [109]. Furthermore, they update slowly and, given the effort needed to construct them, they are generally proprietary in nature with the exception of some open-access initiatives [106, 109]. The existing landscape of experimental datasets is fragmented, and most of the known experimental results remain confined to unstructured scientific literature.

Given the large volume of literature regularly published, manual curation is impractical. To illustrate the scale of the problem, consider that there are 231 journals listed in the "materials science and engineering" category of the Clarivate Master Journal List (`https://mjl.clarivate.com/home`). With an average of around 1,000 articles published annually per journal, this amounts to approximately a quarter of a million articles containing materials information published each year. Clearly, experimental databases of such magnitude cannot be manually curated. This suggests that automation is the only feasible approach to tackle this challenge. Currently, the state-of-the-art method for automating the data-extraction process from the scientific literature is ChemDataExtractor [79].

ChemDataExtractor

ChemDataExtractor employs a hybrid approach, combining conditional random fields [110] and a dictionary-based recognizer for entity recognition. The process of information extraction relies on manually defined grammar rules to link identified entities. Therefore, the performance of a new model depends on the user's ability to define these rules adequately. Furthermore, because natural language can describe different quantities in structurally diverse ways, each new extraction task necessitates the creation of unique grammatical and syntactic rules by the user. This makes the process labour-intensive and restricts the widespread application of such methods across a broad array of properties. To automate the expansion of user-defined grammar rules, the Snowball algorithm is occasionally integrated with ChemDataExtractor [111, 112]. This algorithm is designed to identify relationships within the corpus automatically through probabilistic analysis of their occurrences.

Language models

As discussed in the previous chapter, in recent years, significant progress has been made in constructing tools that improve our ability to automatise information extraction from natural language. Research in this domain has long focused on creating textual rep-

resentations suitable for advanced, context-aware natural language processing (NLP). Traditional NLP tasks have often relied on simple representations such as one-hot encoding of vocabulary dictionaries, bag of words models, or statistical weightings such as term frequency-inverse document frequency (TF-IDF) [113]. These representations have proven effective for tasks such as sentiment analysis or simple classification tasks, especially when dealing with large-scale texts. However, for more accurate processing of individual terms or sentences, more sophisticated methods are necessary.

Static word embeddings such as the ones provided by skip-gram models naturally follow as viable candidates for NLP applications [80, 87, 73]. transformer networks elaborate further on these ideas and are the current state of the art in contextual natural language representations [81].

At the heart of transformer networks is the use of self-attention, which captures the syntactic interdependencies between words. This allows these networks to excel at parsing the context in which a term appears in a sentence. As described in Chapter 2, one of the most popular transformer-based models for NLP is the Bidirectional Encoder Representations from Transformers (BERT), a model comprising a sequence of encoder transformer blocks with 110 million tunable parameters [91]. Since its introduction in 2018, BERT has rapidly become a cornerstone in many NLP applications, achieving state-of-the-art performance across various benchmarks [114, 115]. The contextual word representation generated by BERT can serve as input for machine-learning models in downstream tasks such as sentimental analysis, named entity recognition or relationship classification. To specialise the pre-trained BERT model over these tasks, the original weights are adjusted in a process that takes the name of fine-tuning.

Recently, transformer-based generative large language models have gained a significant amount of attention after the release of ChatGPT (`https://chat.openai.com/`). While these models show promising performances in general language modelling tasks, they present some significant drawbacks that can limit their usability. First, their large scale imposes substantial hardware requirements, both for inference and fine-tuning. Private companies such as OpenAI (`https://openai.com/`) provide services that offer access to their models through API. However, the usage fees can become prohibitive for extensive information-extraction workflows that can require a large number of prompts to be sent to these models. Despite these limitations, recent studies have successfully developed data-extraction workflows for material science applications using large generative language models, either through fine-tuning [116, 117] or iterative prompting [118]. In this chapter, we present a rule-free workflow for automatically extracting information from scientific literature. The workflow developed is based on the fine-tuned BERT models. Towards the end of the chapter, we will discuss potential ways to expand this work to take advantage of the capabilities of LLMs.

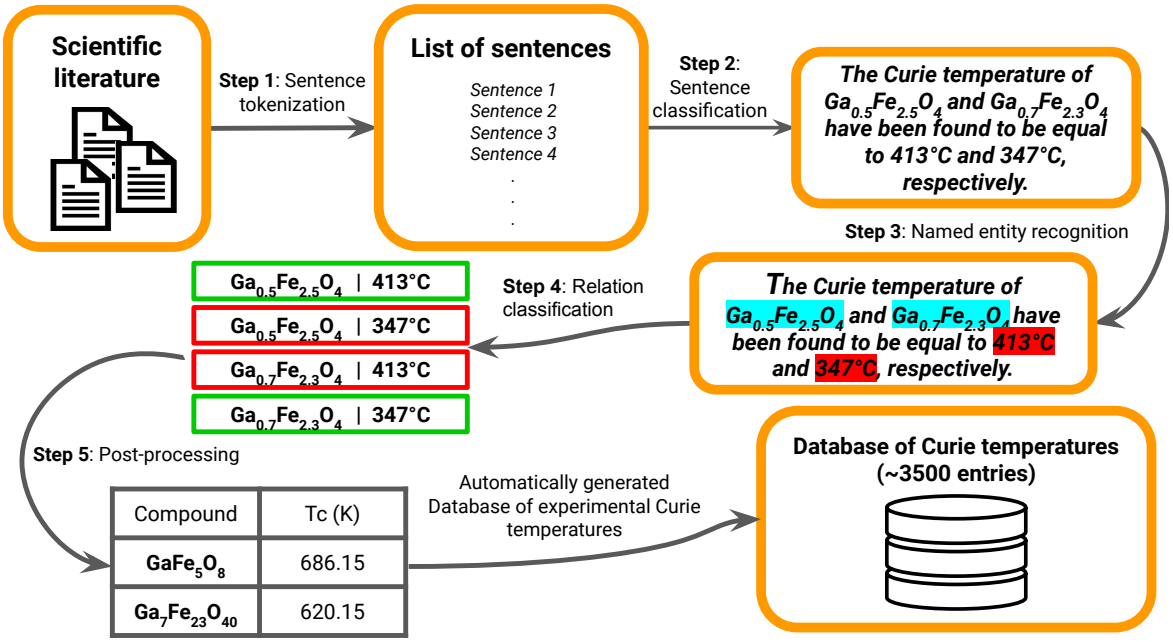


Figure 3.1: Scheme of the BERT-PSIE pipeline applied to the automated extraction of compound-Curie temperature pairs from the scientific literature. The workflow is based on the combined use of a collection of BERT models fine-tuned for different downstream tasks such as sentence classification, named entity recognition and relation classification. See text for more details.

3.1 Automatic extraction of experimental data

The contextual awareness of the BERT representation allows the grammatical and syntactic rules necessary to perform information extraction from natural language to be learnt during fine-tuning by the transformer model from a small sample of labelled text. The text labelling step replaces the need for designing grammar rules and eliminates the requirement for prior expertise in natural language processing or coding to implement new extraction pipelines. We developed a novel self-contained literature-to-structured-properties database pipeline, which we named BERT Precise Scientific Information Extractor (BERT-PSIE). The automatic information extraction is carried out through a sequence of BERT models, each fine-tuned for specific downstream tasks. By taking advantage of the transfer-learning capabilities of BERT models, it is possible to achieve good performance on these tasks with a relatively small training set of labelled text. This workflow can be adapted to the extraction of any binary-related information and can potentially be extended to more complex forms of relations. We present in Fig. 3.1 a scheme of the entire pipeline. The full corpus of papers, on which the extraction is performed, is assembled using a keyword-based search with the Crossref REST API (`https://api.crossref.org/`). A classifier isolates relevant sentences from the downloaded corpus and a Named Entity Recognition (NER) module extracts material-property relations from sentences containing single, unambiguous relations. Sentences containing a single entity only, either the compound or the property, are

discarded. Subsequently, a second module performs relation classification for sentences featuring multiple mentions of compounds and/or material properties. The extracted material-property relations are then compiled into a database. We will proceed now to discuss each of these downstream tasks employed in BERT-PSIE in greater detail.

Relevant Sentences Classifier

The first challenge addressed by our workflow is to locate the information of interest in the text of the papers downloaded via the Crossref REST API. We achieve this goal by fine-tuning BERT to classify relevant sentences. Specifically, we build a binary classifier on top of the BERT embeddings. The simplest model for binary classification is the logistic regression, according to which the probability of a sentence $\mathcal{S}$ of being relevant is given by:

$$p(\text{relevant}|\mathcal{S}) = \frac{1}{1 + e^{-w_0 + \mathbf{w}\cdot\mathbf{b}_{[\text{CLS}]}}}, \tag{3.1}$$

where the parameters $\{w\}$ are learned during the fine-tuning and $\mathbf{b}_{CLS}$ corresponds to the output BERT embedding of the [CLS] token. The [CLS] token is a special symbol added in front of any input by the BERT tokenizer. Its purpose is to be used for classification tasks as its corresponding output embedding will depend on all the other words in the input sentence given the bidirectionality of the attention mechanism. Moreover, the length of this output embedding is fixed and does not depend on the length of the sentence. Alternative strategies could involve using the mean or the sum of the output embeddings of all the words in the sentence. During fine-tuning the values of the weights $\{w\}$ and the parameters of BERT are optimised using stochastic gradient descend in order to minimise the cross-entropy loss over a set of M manually labelled examples $\{\mathcal{S}_i, y_i\}$:

$$\ell = -\sum_{i=1}^{M} \left[y_i \log p(\text{relevant}|\mathcal{S}_i) + (1 - y_i) \log \left(1 - p(\text{relevant}|\mathcal{S}_i)\right) \right], \tag{3.2}$$

where y_i is the label annotated for the sentence $\mathcal{S}_i$, which is chosen to be 1 if the sentence is deemed relevant and 0 if not. This decision is somewhat subjective given the variable forms in which a given information can be delivered in natural language. During the manual labelling, guidelines need to be provided in order to make the label choice as robust as possible. The details of such guidelines will vary case by case, depending on the quantity that represents the extraction target. However, the general underlying principle that we chose is to deem as relevant the sentences that we would like to be further processed by our pipeline and as irrelevant otherwise.

After the BERT model is fine-tuned to reproduce the labels provided in the training set, we now avail of a model that can be used to filter out the vast majority of the sentences in the corpus. Following this step, we are left with sentences that the model predicts to be relevant for the extraction task.

Named Entity Recognition

The sentences deemed relevant are then processed by a second BERT model that has been fine-tuned to perform NER. Unlike the previous classification task, which operated on a sentence level, it's possible to train a BERT model for token-level classification. In this case, the output embedding of each token in the input sentence is processed by a multiclass classifier to identify its corresponding entity. These entity categories are user-defined. In this work, we consider three entity classes, namely $CHEM$, $TEMP$, and GAP, which are associated with the mention of a chemical compound, Curie temperature and band-gap respectively. Each classification task also includes an empty class, denoted as O, which is reserved for tokens that do not belong to any of the other entity classes. The multiclass classification for C entity types is performed using a softmax layer applied to the output embeddings of BERT. The predicted probability that the j-th input token belongings to the i-th class is given by:

$$p(\text{class}_i|\text{tok}_j) = \frac{\exp\left(\left(W\mathbf{b}_{\text{Tok}_j}\right)_i\right)}{\sum_{k=1}^{C}\exp\left(\left(W\mathbf{b}_{\text{Tok}_j}\right)_k\right)}, \tag{3.3}$$

where $\mathbf{b}_{\text{Tok}_j}$ is the N_h-dimensional BERT output embedding corresponding to the j-th token and W is a $C\times N_h$ matrix containing the learnable parameters of the classification layer. The fine-tuning is then performed over a small dataset of manually labelled examples with the objective of minimising the following log loss:

$$\ell = -\sum_{i=1}^{M}\log\left(p(\text{class}_{y_j}|\text{tok}_j)\right), \tag{3.4}$$

where class_{y_j} is the manually labeled class of the j-th token.

Relationship Extraction

The final stage in our workflow for automatic data extraction involves establishing the presence of relations between mentions of chemical compounds and their associated properties (either Curie temperature or band-gap) in all the sentences classified to be likely to contain such information.

For the sentences where the NER model predicts a single mention of a chemical compound and a single mention of Curie temperature, we assume that these two quantities are related and we add them to the database. However, if for example, a sentence contains multiple mentions of chemical compounds and/or several Curie temperatures, the compound-temperature association will become ambiguous. This kind of ambiguity is not uncommon in scientific literature. For instance, one might find sentences such as "the Curie temperature of Fe and Co are 1043 K and 1394 K, respectively". While easy for a human reader to understand, these semantically ambiguous statements can appear in several forms and present a challenge that cannot always be resolved through simple

heuristic rules. To address this issue, we frame the problem as a relation classification task following the approach of Soares *et al.* [119]. We developed a BERT architecture with the aim to classify whether a pair of entities in a sentence is related by the "has a T_C of" ("has a band-gap of") relation. This is achieved by finetuning a BERT model for binary sentence classifications, using modified inputs that contain special entity markers to highlight the tokens of interest. For example, from a sentence containing two chemical compounds and two Curie temperatures (band-gaps), we generate four sentences. In each, a different pair of entities is surrounded by entity markers. We use the markers $[E1_{\text{start}}]$, $[E1_{\text{end}}]$ to identify mentions of chemical compounds and $[E2_{\text{start}}]$ and $[E2_{\text{end}}]$ to identify mentions pf the Curie temperature (or band-gap).

For each sentence, we consider all possible pairs of compound-T_C (or compound-band-gap) mentions, as identified by the NER model, one by one. The entity markers are added at the beginning and at the end of each entity mention. Thus, by taking as an example the sentence, "The Curie temperature of $Ga_{0.5}Fe_{2.5}O_4$ and $Ga_{0.7}Fe_{2.3}O_4$ have been found to be equal to 413 °C and 347 °C, respectively" (see Fig. 3.1), we construct the following four relationship associations:

1. "The Curie temperature of $[E1_{\text{start}}]$ $Ga_{0.5}Fe_{2.5}O_4$ $[E1_{\text{end}}]$ and $Ga_{0.7}Fe_{2.3}O_4$ have been found to be equal to $[E2_{\text{start}}]$ 413 °C $[E2_{\text{end}}]$ and 347 °C, respectively."

2. "The Curie temperature of $[E1_{\text{start}}]$ $Ga_{0.5}Fe_{2.5}O_4$ $[E1_{\text{end}}]$ and $Ga_{0.7}Fe_{2.3}O_4$ have been found to be equal to 413 °C and $[E2_{\text{start}}]$ 347 °C $[E2_{\text{end}}]$, respectively."

3. "The Curie temperature of $Ga_{0.5}Fe_{2.5}O_4$ and $[E1_{\text{start}}]$ $Ga_{0.7}Fe_{2.3}O_4$ $[E1_{\text{end}}]$ have been found to be equal to $[E2_{\text{start}}]$ 413 °C $[E2_{\text{end}}]$ and 347 °C, respectively."

4. "The Curie temperature of $Ga_{0.5}Fe_{2.5}O_4$ and $[E1_{\text{start}}]$ $Ga_{0.7}Fe_{2.3}O_4$ $[E1_{\text{end}}]$ have been found to be equal to 413 °C and $[E2_{\text{start}}]$ 347 °C $[E2_{\text{end}}]$, respectively."

The relationship classifier model is fine-tuned on a set of sentences with marked entity pairs which have been manually labelled for a binary classification task. During the manual labelling, we deem a sentence positive if a relationship is present between the two marked entity mentions, and as negative if such a relationship is not present.

This concludes the collection of BERT models trained for different downstream tasks creating the BERT-PSIE rule-free pipeline for the automatic extraction of data from text.

3.1.1 Fine-tuning

In this work, we focus on extracting information about the Curie temperature of ferromagnets and the electronic band-gap of semiconductors/insulators. To construct our scientific literature corpus, we use the Crossref REST API to perform a keyword search across all literature published by Elsevier. This search yields metadata which we filter

to ensure the availability of the full-text version of the paper for data mining. This metadata includes both abstracts and download links to the full-text articles.

For the extraction of Curie temperature data, the strategy used to build the training set required for the fine-tuning of the different BERT models starts from the collection and the manual labelling of 800 abstracts containing the term "Curie temperature". We split these abstracts into sentences using the Natural Language Toolkit [120] (NLTK) sentence tokenizer. We then label the sentences which reference a Curie temperature as relevant and the ones that do not as irrelevant (step 1 in Fig. 3.1). This step yielded a database of about 4,000 sentences of which 189 have been labelled as relevant. The labelled dataset is used to fine-tune the relevant sentence classifier BERT model. Subsequently, we manually labelled the entities present in the abstracts and in 200 relevant sentences extracted from the corpus of the papers whose abstract was used in the previous step. This combined corpus is then used to fine-tune a BERT model for Named Entity Recognition (NER-BERT).

In the case of the electronic band-gap extraction, we adopt a slightly different strategy for collecting the required training data. We download the arXiv metadata from the Kaggle dataset [121] and compile an initial corpus of 1,000 abstracts by searching for the terms "band gap", "bandgap" or "band-gap". Besides this difference, the rest of the workflow mirrors the one used for the T_{C}. The manual labelling stage yields, in this case, a dataset of 672 sentences of which 171 are deemed relevant.

To fine-tune the relation classifier module, we sample 100 sentences predicted to contain multiple entity mentions by the NER-BERT model. For each sentence, all the possible pairs of compound-T_{C} (compound-band-gap) mentions are considered one by one with the added entity markers. Every constructed sentence is then labelled as positive or negative if a relation between the pair of highlighted entities is present or not.

In this work, we use the MatSciBERT [95] weights as starting point for the fine-tuning. This particular BERT model has been exposed during pre-training to a large corpus of material science literature making it the optimal choice for the task considered here. Each dataset used for fine-tuning is divided into train, validation and test sets. The models' parameters optimisation is performed solely on the training set using stochastic gradient descent. The Hyper-parameters of the models have been optimised over the validation set. We observe that the models' performances are consistent over a range of values for all hyperparameters. For all models, we use a training batch size of 32 and the learning rates of $5 \cdot 10^{-5}$, $5 \cdot 10^{-5}$ and $2 \cdot 10^{-6}$ for the classifier, NER, and relation models respectively. We apply gradient clipping with a cut-off of 8 for all models and implement an early stopping based on the validation set loss to prevent overfitting.

To preliminary assess the performance of the models after fine-tuning, we carry out a performance evaluation on the test set. The performance of the models is assessed

at this stage by standards performance metrics used for classification tasks such as precision, recall and F1-score. To maintain consistency between the testing of the three BERT modules we adopt the following convention. For the relevance classifier module, a positive instance corresponds to a relevant sentence while a negative instance corresponds to an irrelevant sentence. Within the NER module, for a given class, a positive instance corresponds to an instance of that class while a negative instance corresponds to an instance of any other class. Finally, for the case of the relation classifier, a positive instance indicates the presence of a relationship between the entities highlighted by the entity markers, while a negative instance corresponds to the case where such a relationship is not present. Within these conventions, any prediction in these classification tasks yields one of four possible outcomes, which can be grouped as follows:

1. True positive (TP) correct positive predictions.

2. False positive (TN) correct negative predictions.

3. False positive (FP) incorrect positive predictions.

4. False negative (FN) incorrect negative predictions.

The precision (P) performance metric then quantifies how often the model is correct when it returns a positive prediction

$$P = \frac{TP}{TP + FP}, \tag{3.5}$$

while the recall (R) performance metric quantifies the rate of correct positive predictions with respect to the total of positive instances

$$R = \frac{TP}{TP + FN}. \tag{3.6}$$

A well-performing classifier should display both high precision and high recall. To condense these two metrics into a single quantity, their harmonic mean is computed, known as F_1 score:

$$F_1 = 2\frac{P \cdot R}{P + R}. \tag{3.7}$$

We report in Table 3.1 the test set performance metrics of the different fine-tuned BERT models for the Curie temperature extraction case. The sentence-level relevancy classifier metrics are presented in the upper row of Table 3.1. Both precision, P, and recall, R exceed 0.8, indicating good performance on the test data.

During the construction of the training dataset, particular attention was paid to ensure that it was as representative as possible of the literature. Nonetheless, syntactic similarities in the construction used to report a temperature value are unavoidable,

Model	Entity	P	R	F_1	Support	TrS	TeS
Classifier		0.83	0.80	0.81		3941	801
NER	Chem	0.92	0.86	0.89	754	1,769	168
	T_C	0.97	0.81	0.88	42		
Relation		0.72	0.64	0.68		200	50

Table 3.1: Performance of the three modules composing BERT-PSIE for the Curie temperature extraction: the sentence-level relevancy classifier, the NER and the relation classifier. Results are presented for the test sets. Here we report: precision, P, recall, R, and F_1 score. The size of the test (TeS) and training (TrS) sets are also given (number of sentences used). For the case of NER, we report results for both chemical entities (Chem) and T_C, as well as the support.

introducing noise in the extracted data. For instance, if we consider the sentence "Barium titanate ($BaTiO_3$) is a ferroelectric with a Curie temperature of 120 °C". In this case, "Curie temperature" refers to a paraelectric-ferroelectric transition and not to ferromagnetism. However the syntactic structure closely resembles those describing the magnetic T_C and despite the contextual capabilities of the BERT embeddings, the model might struggle to resolve these two cases. Moreover, further ambiguities can also be found in constructions like "The melting temperature of a compound marks the solid-liquid phase transition. This critical temperature for Fe is 1,538 °C". Given the fact that the classification is performed at the sentence level, the content of "This critical temperature for Fe is 1,538 °C" is processed independently from that of the preceding sentence, leading to potential misclassification. These limitations are inherited from working at the sentence level and further work is necessary to effectively address these issues. To mitigate these drawbacks, we focus our analysis on scientific texts taken from the field of magnetism.

The second row of Table 3.1 shows the performances of the named-entity-recognition step in our automated extraction pipeline. The precision, recall and F_1 score associated with the classified entities, namely compound and Curie temperature, are all consistently high. This indicates a good ability of the BERT model in performing token classification, allowing us to identify mentions of compounds and Curie temperatures in sentences selected as relevant by the sentence-level relevancy classifier. Additionally, the contextual awareness of BERT-based language models allows them to discriminate similar entities based on the syntactic and grammatical context in which they appear. However, this context-awareness has limitations, especially in distinguishing between different types of critical temperatures related to phase transitions.

Finally, the last row of Table 3.1 summarizes the key evaluation metrics for the BERT relationship-extraction model. Extracting relationships has proved to be the most challenging task in our pipeline, largely due to the vast quantity of potential combinations of words in different syntactic structures. This complexity also poses a significant challenge in the construction of grammar rules for rule-based extraction methods such as ChemDataExtractor. Despite the fact that this module presents the

lowest scores, the model still exhibits reasonably good performance. Therefore, it can be used to associate the correct compound-property pairs, thus improving the quality of our final database.

A similar analysis has been performed for models fine-tuned for the extraction of band-gap values. The performance metrics of each of the modules in our pipeline for the band-gap extraction are summarised in Table 3.2.

Model	Entity	P	R	F_1	Support	TrS	TeS
Classifier		0.95	1.00	0.97		404	134
NER	Chem	0.80	0.96	0.87	1166	4000	1000
	Band-gap	0.78	0.97	0.87	119		
Relation		0.88	0.88	0.88		300	80

Table 3.2: Performance of the three modules composing BERT-PSIE for the band-gap extraction: the sentence-level relevancy classifier, the NER and the relation classifier. Results are presented for the test sets. Here we report: precision, P, recall, R, and F_1 score. The size of the test (TeS) and training (TrS) sets are also given (number of sentences used). For the case of NER, we report results for both chemical entities (Chem) and band-gap, as well as the support.

The first row of Table 3.2 summarises the test set performance of the relevancy classifier module. Once again we observe significantly good performance on this task, with a perfect recall and a slightly lower precision at 0.95. These metrics surpass those found for the Curie temperature, indicating an almost perfect capability in distinguishing between sentences that do or do not contain information about a compound's band-gap. The superior performance of this classifier can be attributed to a reduction in ambiguity in the reporting of band-gaps. This contrasts with the case observed in temperature reporting, which, as previously pointed out, often features similarities in the syntactic structures used for reporting different types of critical temperatures.

Similarly, the NER module's performance remains consistently good, with a high precision, recall and F_1 score.

Finally, we find that the relationship-extraction step for the band-gap extraction case significantly outperforms that for the Curie temperature extraction. We attribute this once again to the more standardised way in which band-gaps appear to be reported in the scientific literature. Sentences reporting band-gap measurements tend to have structures generally more formulaic than the ones reporting Curie temperatures. This hypothesis is further reinforced by the improved performance in the final extraction when using a relationship association between compounds and band-gaps based on their order of appearance in the sentence, as will be further discussed in the following sections.

3.1.2 Comparison with rule-based methods

To directly compare the performance of our BERT-PSIE pipeline with the state-of-the-art rule-based methods, we adopted an approach similar to those described in [112, 122]. We manually annotated 200 unique abstracts for the case of compound-Curie temperature extraction and separately other 200 abstracts for compound-band-gap extraction. These abstracts were sourced, once again, from the arXiv dataset [121] and they were selected to guarantee that there was no overlap with the abstracts used for the fine-tuning or for the validation set of the models.

These abstracts were screened by performing a keyword search. For the Curie temperature case, the keyword used was "Curie", and we excluded abstracts containing the term "Weiss". Similarly, the band-gap corpus for this test was constructed using a keyword search selecting only the abstracts containing any of the terms "band gap", "band-gap" or "bandgap", alongside the term "eV". Random abstracts were then sampled from these pools to create test sets of 200 abstracts for each task. This selection strategy aimed to maximize the number of positive extraction targets (support).

Both BERT-PSIE and ChemDataExtractor were run on these test sets. A record in the extracted database was considered true positive only if all entities in the target compound-property pair were present and matched the manual annotation. The number of true positives, false positives and false negatives were manually counted for the extraction tasks. This allowed the calculation of the precision, recall and F_1 score for each model.

We used the same ChemDataExtractor model for the extraction of Curie temperature as the one provided by the rule-based pipeline of Ref. [112]. It was not possible to include the snowball model of this extraction pipeline since the model was not readily available. However, for the assessment of the band gap extraction performance with ChemDataExtractor, the full hybrid extraction method, which includes the Snowball model, was utilised as provided by Ref. [122].

The results of this direct comparison between BERT-PSIE and ChemDataExtractor are reported in Table 3.3.

As it can be observed from the data reported in Table 3.3, the precision of both BERT-PSIE and ChemDataExtractor extractions remains consistent across the different properties extracted. Remarkably the BERT-PSIE workflow slightly outperforms the hybrid ChemDataExtractor model on the band gap extraction task.

As previously discussed the relationship extraction module of the BERT-PSIE workflow represents the most challenging component to train. In order to isolate the contribution of this module and estimate the amount of noise that it might introduce, we also examine the case where we only consider the extraction from sentences for which the NER module predicts only a single compound and property mention. Doing so we skip the use of the relationship classifier module in this case and we directly associate the two mentions as an extraction. We refer to this subset as 'Single mentions'. For this

Model	P	R	F_1
Curie Temperature			
ChemData.	0.67	0.49	0.56
Single mentions	0.82	0.20	0.32
BERT-PSIE	0.67	0.31	0.42
BERT-PSIE + ChemData.	0.64	0.64	0.64
Band Gap			
ChemData.	0.68	0.55	0.61
Single mentions	0.78	0.23	0.35
BERT-PSIE	0.70	0.40	0.51
BERT-PSIE + ChemData.	0.63	0.72	0.67

Table 3.3: Direct comparison between the extraction carried by BERT-PSIE and the rule-based ChemDataExtractor on the same test corpus of 200 annotated abstracts for both the T_C and band gap. The precision, recall and F_1 score are presented for BERT-PSIE (single mentions only and the full pipeline), ChemDataExtractor and the combination of the two methods. The manually annotated datasets have a support of 45 entries for T_C and 109 entries for the band gap.

case, we observe a significant increase in the extraction precision. However, this gain in precision is offset by a significant reduction in recall, ultimately causing a reduction of the F_1 score.

The BERT-PSIE extraction pipeline tends to be more selective in the data extracted. We can attribute this to the fact that any potential contextual ambiguity when reporting a property value is more likely to result in a failed extraction in a context-based system than a rules-based one. As a consequence, we observe a lower recall than the one found for the extraction performed by the ChemDataExtractor rules-based and hybrid pipelines. Furthermore, one could claim that recall is less important than precision when it comes to the assessment of the reliability or utility of the resulting extracted data. To explore this argument, in the following sections, we will introduce novel metrics for the assessment of the usefulness of the extracted data, that simulate real-world use-cases of these data.

As a last direct comparison, for both extraction targets, we combined the extracted data from both the BERT-PSIE and ChemDataExtractor models. This combination led to a marked increase in recall for the merged dataset compared to either method alone implying that the rule-based and BERT-based pipelines each have unique strengths in extraction, as the distinct increase in recall is due to a limited overlap between the extractions performed by the two methods. The reduction in precision relative to either of the two pipelines for this particular test is caused by the fact that the true positive extractions that overlap for the two pipelines are not double counted and any incorrectly extracted value is added to the combined dataset. This leads to an increase in false positives relative to the true positives, thereby reducing the overall precision of the combined dataset with respect to either of the constituent datasets.

3.2 Evaluating the quality of the extracted data

The metrics discussed so far provide insights into the performance of each module within our workflow, as well as into BERT-PSIE's overall ability to accurately extract the target quantity. In performing these tests we followed the general practice found in the literature when introducing a new extraction pipeline [112, 122, 79, 116, 96]. However, we believe that these metrics present some limitations, due to the limited size of the test sets on which they are calculated and the fact that they might not correlate with the actual real-world usage of the data extracted.

To address these concerns and further evaluate how different design choices impact our extraction workflow, we decided to focus on the extraction of the Curie temperature of ferromagnetic materials and on the band gap of semiconductor/isolator materials. We chose these two properties due to the availability of two corresponding, manually curated databases of experimental measurements, which we can use as expected values during comparisons with the automatically extracted data produced by BERT-PSIE. Furthermore, as we discussed in Chapter 2, it is particularly challenging to compute these two properties reliably with DFT.

We populated the two corpora on which we performed the Curie temperature (band gap) extractions performing a keyword search using the Crossref API searching for papers containing instances of the term "magnetic" ("electronic"). This yielded a database of approximately 180,000 (77,000) full-text URLs of papers potentially containing a mention of a Curie temperature (band gap) value. The full text of these papers was then automatically downloaded and parsed into a list of sentences generated with the NLTK sentence tokenizer. For the Curie temperature extraction, we also include a corpus of relevant PDF documents converted into plain text using PDFminer [123].

The compiled corpus is then processed by the BERT classifier modules of our pipeline which select the sentences likely to contain the property of interest. This resulted in roughly 55,000 sentences predicted to contain mentions of Curie temperature and around 126,000 sentences relevant to the band gap extraction task. The relevant sentences are then processed by the NER and the relation classifier modules, to identify the entities and establish their relationships.

Finally, the compound-property pairs returned by the last stage of BERT-PSIE are then post-processed. We standardise all temperature units to Kelvin and all band gap units to electron volts. All chemical formulas are normalised to have integer coefficients (e.g. $Ga_{0.5}Fe_{2.5}O_4$ becomes $GaFe_5O_8$). Carrying out these steps results in the creation of two databases of 3,518 and 2,090 unique compound-property pairs for Curie temperature and band gap respectively.

Curie temperature

We begin our examination of the extracted data with the case of Curie temperature.

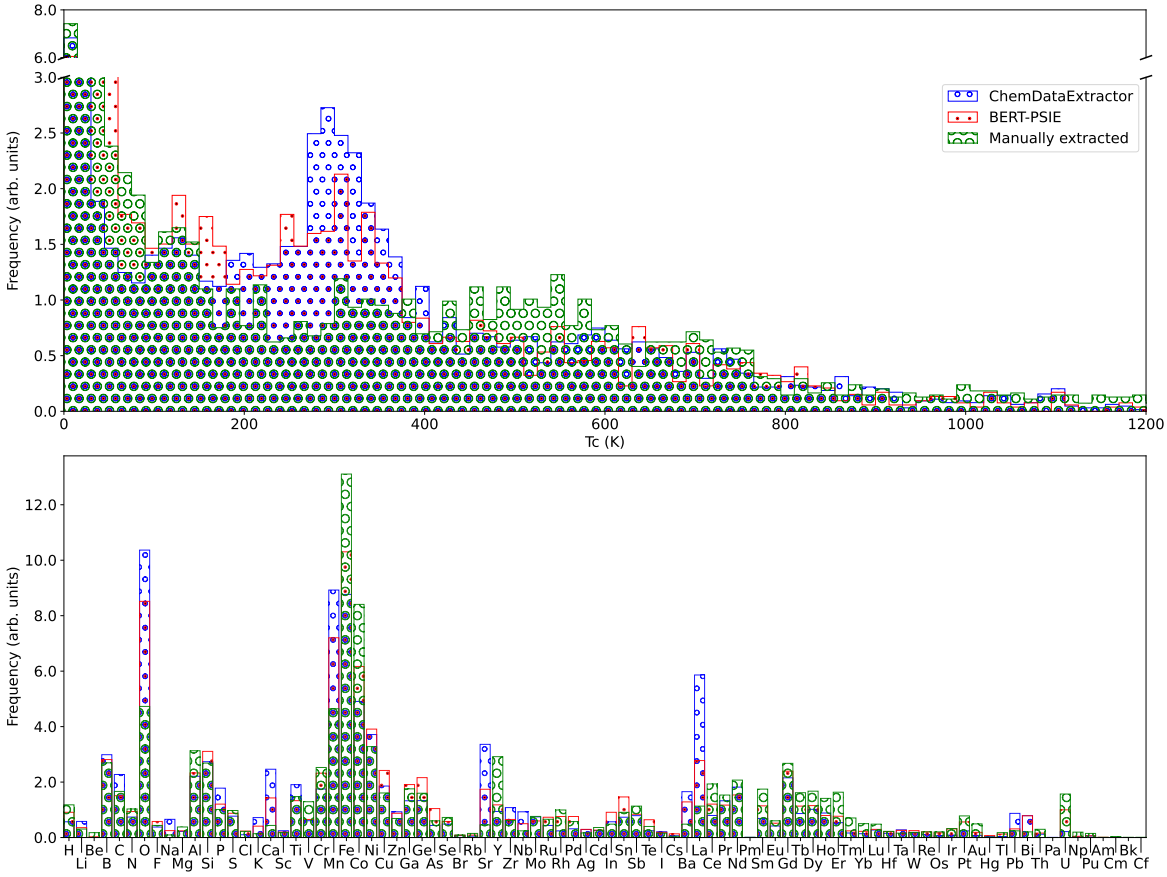


Figure 3.2: Comparison between the data present in the three different databases considered: BERT-PSIE *(red)*, ChemDataExtractor *(blue)* and the manually-extracted database of Ref. [22] *(green)*. Top panel: Normalised distribution of the Curie temperatures extracted. A peak is visible in the distribution around 300 K in both the automatically generated databases, which is not present in the manually extracted one. Bottom panel: Relative elemental abundance across the compounds present in each database. Although there is general agreement among the three databases, additional peaks are observed for various elements in the case of automatically extracted data, which are not present in the manually curated dataset. The most pronounced of these differences is in the relative abundance of Mn- and O-containing compounds. Note that the automatically extracted datasets and the manually curated ones are sourced from different corpora.

Ultimately the value of a database stems from the reliability and comprehensiveness of the data that it contains, as well as its potential use in secondary tasks such as the training of machine-learning models. Conducting such an assessment is typically made challenging due to the lack of manually curated data, which are time-consuming to collect.

In this case, however, by choosing the Curie temperature as the extraction target we avail of a manually curated database compiled from various sources. We primarily utilise the database of Nelson *et al.* [22]. This dataset has been created by aggregating the *AtomWork* database [124], *Springer Materials* [125], the *Handbook of Magnetic Materials* [126] and the book *Magnetism and Magnetic Materials*. [78]. Furthermore,

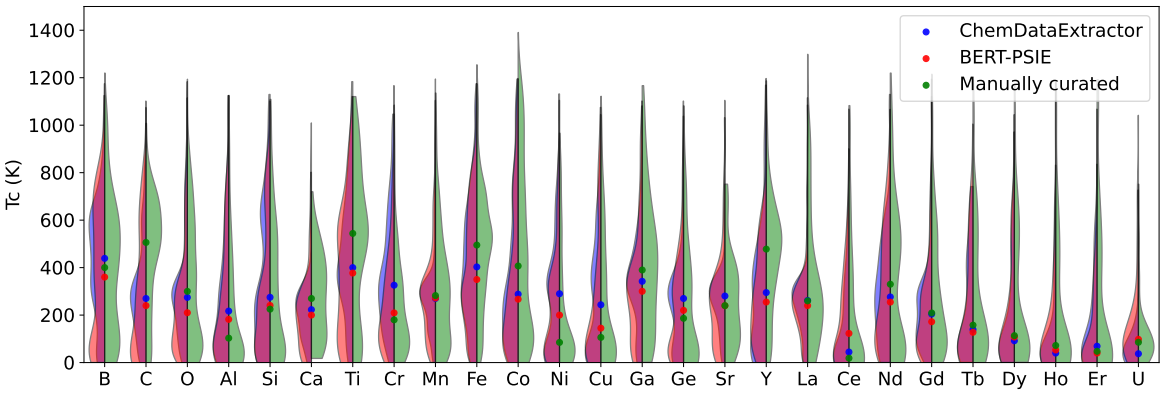


Figure 3.3: Violin plots comparing the T_C distribution of the compounds containing specific elements in the dataset automatically generated with BERT-PSIE *(red)* and ChemDataExtractor *(blue)*, and in the manually curated ground truth *(green)*. Only the most common elements appearing in the datasets are displayed here. The dots show the median of each distribution.

we augment this database with the T_C values from a dataset manually aggregated by Byland *et al.* [127], which is mainly focused on, although not limited to, Co-containing compounds. Consequently, this combined database is considered to be our "ground truth" for the Curie temperature extraction task and amounts to 3,638 unique ferromagnetic compounds and their associated Curie temperatures.

The choice of the Curie temperature as the extraction target allows us also to compare the extraction with the one performed by the ChemDataExtractor rule-based pipeline combined with a semi-supervised snowball algorithm as reported in [112]. Both automatically extracted databases are built from a rather similar corpus since they rely on the download of papers via the Crossref API. These corpora primarily include relatively recent articles, and the automatic extraction is performed solely from the text. Interestingly, despite the similarity in their respective source corpora, there is little overlap between BERT-PSIE and ChemDataExtractor databases, each containing several thousand data points, they share only 694 compounds.

In contrast, Nelson *et al.*'s database is largely built on data presented in tables and incorporates a significant amount of "historical" information, including results published as far back as the 1950s. The size of the overlap between the automated and manual datasets is 687 for BERT-PSIE vs. manually-curated and 595 for ChemDataExtractor vs. manually-curated. Overall the three datasets (BERT-PSIE, ChemDataExtractor and the manually curated one) share only 262 compound entries. For the purpose of our analysis, we take the median Curie temperature value for compounds for which multiple entries have been extracted.

We observe that properties extractions relative to elemental compounds (e.g. Fe, Co, Ni, Gd, etc.) tend to be unreliable and subjected to large variance. We attribute the source of these errors to the inherent challenge that the NER module faces in differentiating between an elemental compound and an element serving as a dopant in

an otherwise non-magnetic material (e.g. bulk Mn vs. Mn-doped GaAs). As dopants can appear in a multitude of concentrations and in a large variety of hosts, erroneous assignments may result in a large spread in the distribution of the temperatures collated. With this exception, the distributions of Curie temperatures across the different databases are in very good agreement with each other, as can be seen in the top panel of Fig. 3.2.

The agreement between our automatically extracted dataset and the one constructed with ChemDataExtractor is notably close. However, both exhibit a peak in the distribution at around room temperature, a feature absent from the manually curated dataset. This discrepancy could be attributed to different reasons, the more recent literature could present a bias towards critical temperatures close to 300 K, or during the automatic extraction the temperature values close to the room temperature are erroneously extracted and attributed to the T_{C} with a high frequency. Supporting the latter hypothesis, it is the fact that mentions of room temperature feature heavily in sentences containing the target information, even if the room temperature is not a mention of T_{C}. For instance, consider the sentence: "The magnetisation curve at 300 K was obtained and the Curie temperature was determined by TGA under a magnetic field, yielding a Curie temperature of 1043 K for Fe.". Despite these differences, the three Curie-temperature distributions present strong similarities indicating that our automated extraction technique has adequately captured the relative abundance of high- and low-temperature ferromagnetic materials without necessitating the definition of complex grammar rules.

Further insights into the extracted data can be obtained by looking at the relative elemental abundance across the unique compounds present in the database (the frequency at which a particular element appears in the database). This is shown in the lower panel of Fig. 3.2, again for all three datasets. As expected the largest abundances are observed in correspondence with the magnetic transition metals, some of the rare earths and oxygen. This feature is shared by all databases and corresponds to the actual elemental distribution among magnets as reflected by our manually curated dataset. Remarkably, the automatically compiled databases seem to overestimate the presence of Mn and O, and that of di- and tri-valent alkali metals (Ca, Ba, Sr and La). This overestimation with respect to the manually extracted dataset is more pronounced for the ChemDataExtractor data than for the ones obtained with our BERT-PSIE pipeline.

We attribute these variations in the element distributions to differences in the data sources used for the extractions which differ between manually and automatically curated datasets. Specifically, the most recent literature used in our extraction and in that performed by ChemDataExtractor, include numerous entries related to Ca-, Ba-, Sr- and La-containing perovskites (e.g. manganites). The impact of the original data source on the final dataset is further validated when comparing the T_{C} distributions of compounds containing the 25 most common elements, as presented in Fig. 3.3. Overall,

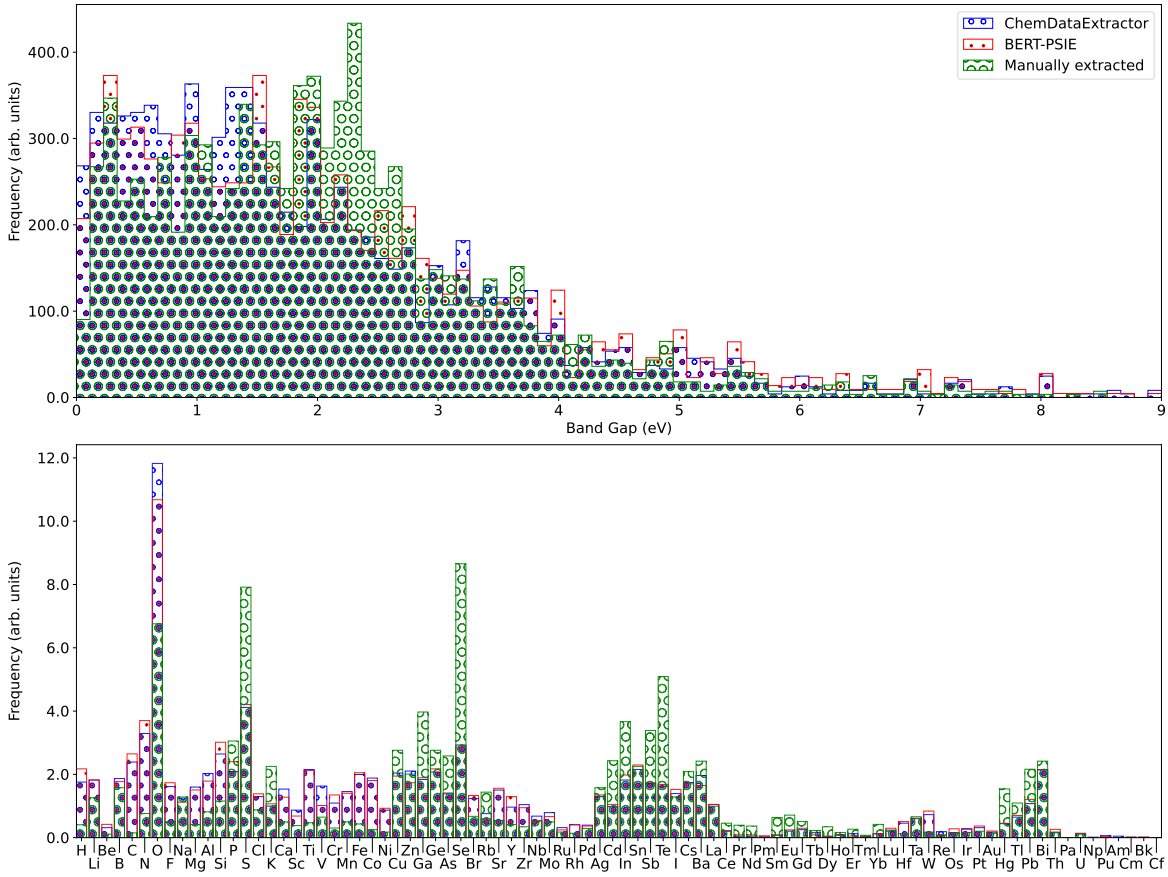


Figure 3.4: Comparison between the content of the different band-gap databases: BERT-PSIE *(red)*, ChemDataExtractor *(blue)* and the manually-extracted database of Ref. [72] *(green)*. Top panel: Normalised distribution of the band gaps extracted. Bottom panel: Relative elemental abundance across the compounds present in a database. Note that the automatically extracted datasets and the manually curated ones are sourced from different corpora.

there is good agreement between the distributions of the two automatically extracted datasets, both of which contain entries extracted from similar sources. BERT-PSIE generally reproduces a distribution similar to the manually extracted values, although for certain elements there are evident discrepancies. This could indicate a historical shift in research focus between the sources used for the manual extraction and those used for the automatic extraction tasks.

Band gap

A similar comparison is also performed on the distribution of extracted band gaps, which we report in Fig. 3.4. The manually curated dataset used in this case comes from [72]. As with the case of Curie temperatures, we find a close similarity between the values distributions in our automatically generated database and the one obtained from ChemDataExtractor. However, both exhibit some level of discrepancy with the manually curated one. To gain insight into the origin of such discrepancies we examine

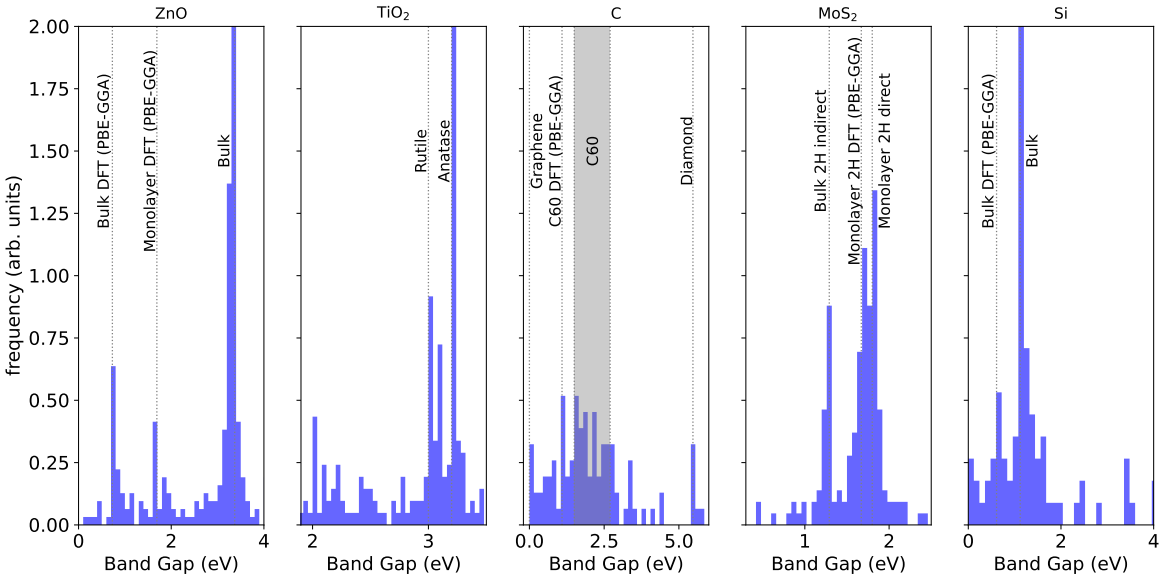


Figure 3.5: Band gap values distribution for the five most common chemical formulas found in the BERT-PSIE-extracted band gap database. The histograms report the relative abundance, while dashed lines indicate gap energies corresponding to specific experimental measurements or theoretical calculations (see text for more details).

the BERT-PSIE-extracted band-gap distributions of the five most common chemical formulas in the database, namely ZnO, TiO_2, C, MoS_2 and Si. These distributions are reported in Fig. 3.5. Notably, while there is a spread of band-gap values for all five compounds, these are not uniformly distributed. Instead, the band-gap frequencies exhibit a pronounced peak structure, with the presence of multiple high-frequency values. This variation can be attributed to the different methods employed to obtain the band gap of a material (experimental optical, experimental transport, theory, etc.), as well as to different polytypes, structures or doped compounds. As detailed in Chapter 2, DFT tends to underestimate the experimental band gap.

Going into more detail, let us consider first the case of ZnO (leftmost panel of Fig. 3.5). Within its distribution, we identify three distinct peaks, which can be readily associated with the experimental bulk band gap (3.37 eV [128]), the DFT-calculated band gap for bulk ZnO (0.73 eV [128] for PBE-GGA) and the DFT-calculated band gap for monolayer ZnO (1.69 eV [129] again PBE-GGA), respectively.

A similar structure is encountered for Si (rightmost panel of Fig. 3.5). Here, the two principal peaks correspond to the experimental bulk indirect gap (1.1 eV [130]) and the one obtained from DFT simulations (0.61 eV [13], PBE-GGA).

In contrast, the peaks in the distribution for TiO_2 have an experimental origin. The two predominant peaks correspond to experimental gaps of two distinct polymorphs, namely anatase (3.2 eV) and rutile (3.0 eV) [131].

Finally, MoS_2 and C are those displaying more complexity. For MoS_2, three dominant peaks are discernible. In fact, alongside the experimental bulk indirect band gap

of 1.29 eV [132], multiple mentions in the literature refer to the experimental band gap of the monolayer form of MoS_2 (1.8 eV [133]) and the DFT estimate of the same (1.67 eV [134], PBE-GGA). Carbon, in contrast, presents a unique distribution profile due to the large variety of possible polymorphs. In fact, two clear peaks are visible which are related to semimetal graphene [135] and bulk diamond structure (5.47 eV [136]), respectively. Then, there is a uniformly distributed region, which is characterised by band gap values associated with carbon buckminsterfullerenes, C60. This extends over the 1.5-2.7 eV range, and has a clear peak at the DFT value of 1.09 eV (PBE-GGA) [137].

To better assess the quality of the extracted databases we define two tests designed around real-world use cases which will be discussed in the following sections.

3.2.1 The query assessment

Given a database, a common use case involves performing queries to retrieve data from it. A desirable property of the database is then to return correct data for a given query. We can quantify the quality of the retrieved data by comparing it to the ones returned by the manually curated reference dataset for the same query. We name this test "query test". In order to make the comparison between our database and the ChemDataExtractor-generated one not dependent on the particular class of compounds extracted, we only compare entries that are shared by all the datasets. The metrics that we utilise to quantitatively assess the agreement between the extracted data and the reference data are the coefficient of determination (R^2), the mean absolute error (MAE) and the root mean square error (RMSE). The R^2 can assume values between zero and one, it offers a measure of the correlation between the retrieved values y_i and the expected ones y_i^*. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i^*)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{3.8}$$

where the sum is extended over all the n compounds included in the test and $\bar{y}$ is the mean value of the extracted data considered. The MAE and RMSE provide a measure of the error between the retrieved data and the expected values. They are defined as follows,

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^*|, \tag{3.9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^*)^2}. \tag{3.10}$$

From their definition, it can be deduced that the RMSE tends to be more susceptible to the presence of outliers than the MAE.

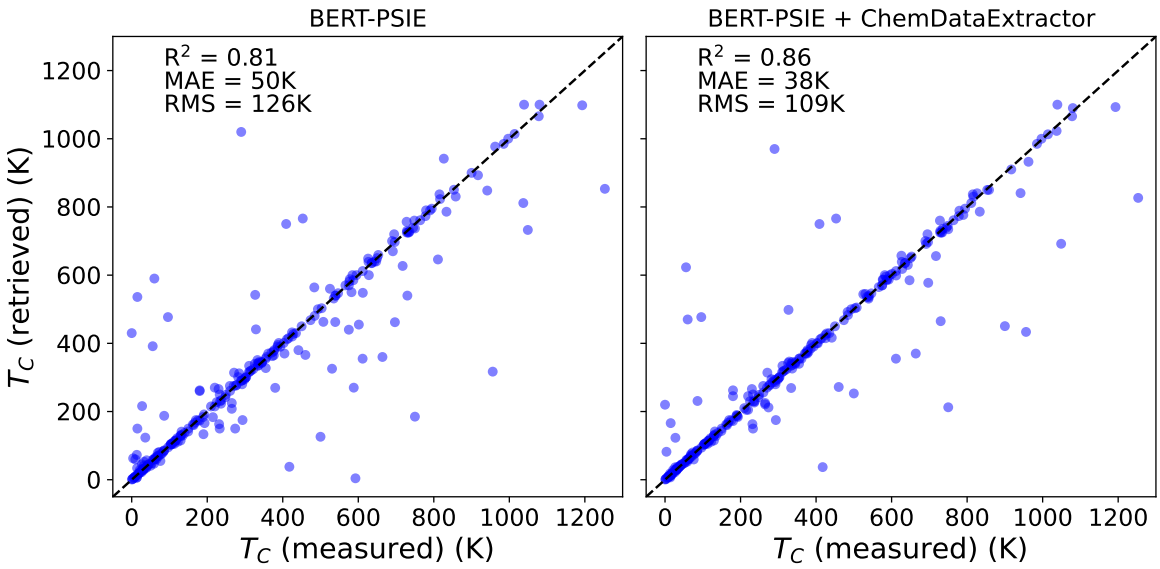


Figure 3.6: Comparison between the T_C queried in the dataset automatically generated by BERT-PSIE and the values contained in the manually curated dataset (left panel). The comparison is performed over the 262 compounds that are shared by all datasets examined in this work. The median value is returned whenever multiple T_C values are collected for a given compound. The same comparison is performed on the dataset resulting by combining the one generated by BERT-PSIE and the one generated by ChemDataExtractor (right panel).

Curie temperature

The query test results for the BERT-PSIE extracted T_C databases are reported in Fig. 3.6, while the performance metrics for the different datasets are summarised in Table 3.4. As discussed in the previous section, the most challenging step in our extraction workflow is the relation-classification step. In order to evaluate the impact of this module on the overall final result a variety of additional extraction strategies have been attempted and compared against the manually curated datasets which serve as ground truth. The first of these relation extraction strategies involves, as previously, utilising only the "Single mentions" results extracted from sentences containing only a single mention of a compound and a single mention of a Curie temperature value. In this case, we are assuming that the two entity mentions are related to each other, thus completely eliminating the need for any relation-assignment step ("Single mention" in Table 3.4). The second strategy imposes a rule that associates compound/value pairs based on the order in which they appeared in the text ("Order of appearance" in Table 3.4). Finally, we have taken every possible combination of compound/value pairs, in order to compare our results with random associations ("All combinations" in Table 3.4). This choice corresponds to a relation-classifier model that always outputs a positive classification. The results in Table 3.4 are complemented by those obtained with our constructed BERT relationship classifier ("BERT-PSIE"), with the data extracted by ChemDataExtractor and by aggregating these last two datasets ("ChemDataExtractor

	# Entries	Query		
		R^2	MAE (K)	RMSE (K)
ChemData.	4,289	0.78	**48**	137
Single mentions	1,858	0.77	51	139
Order of appearance	2,682	0.77	51	141
All combinations	4,308	**0.81**	52	127
BERT-PSIE	3,518	**0.81**	50	**126**
BERT-PSIE + ChemData.	7,052	0.86	38	109

Table 3.4: Query test performance comparison between the different T_C datasets against the manually curated one from Ref. [22, 127]. Together with the BERT-PSIE and ChemDataExtract databases we also consider different BERT-assembled datasets obtained by using different relation-classification strategies (see details in the text). The query benchmark is done over the 262 compounds that are shared by all the datasets. Values for the best-performing datasets are in bold.

+ BERT-PSIE").

As it can be observed in Table 3.4, all of the datasets generated with our rule-free pipeline have metrics comparable to those of ChemDataExtractor. Notably, the one constructed with BERT-PSIE appears to be the best performing on almost all the query-test metrics. In particular, BERT-PSIE returns the best R^2 coefficient of 0.81 and RMSE of 126 K. Interestingly, the MAE for BERT-PSIE dataset is slightly larger than that obtained with ChemDataExtractor. This indicates that while BERT-PSIE matches the accuracy of ChemDataExtractor, producing datasets similar to the manually curated one, it is slightly less prone to display large outliers.

In Fig. 3.6, we present the parity plot associated with the query test for the data extracted with BERT-PSIE. The entries are either on the parity line, representing an exact extraction, or away from it, indicating erroneous extractions, without any particular correlation with the actual T_C value. The superior performance of BERT-PSIE over other BERT-based models using different relation-classifier strategies demonstrates that the inclusion of a context-aware strategy for extracting compound-value pairs from literature is advantageous. However, this improvement is not substantial, as the metrics are rather close to those obtained by considering all possible compound-value relations pairs ('all combinations'). Indeed, more sophisticated methods to establish the correct compound-property associations might help in producing better-automated datasets, a direction that we will explore more later in this chapter with the use of LLMs.

Given the limited overlap between our BERT-PSIE dataset and the one generated by ChemDataExtractor, consisting of only 694 compounds, we have combined the two to form an additional dataset. This unified database contains 7,052 distinct entries and performs best on all the metrics evaluated for the query test (see the last line of Table 3.4 and the right plot in Fig. 3.6. The improvement in performance is likely attributable to the significantly larger size of the dataset (approximately double the

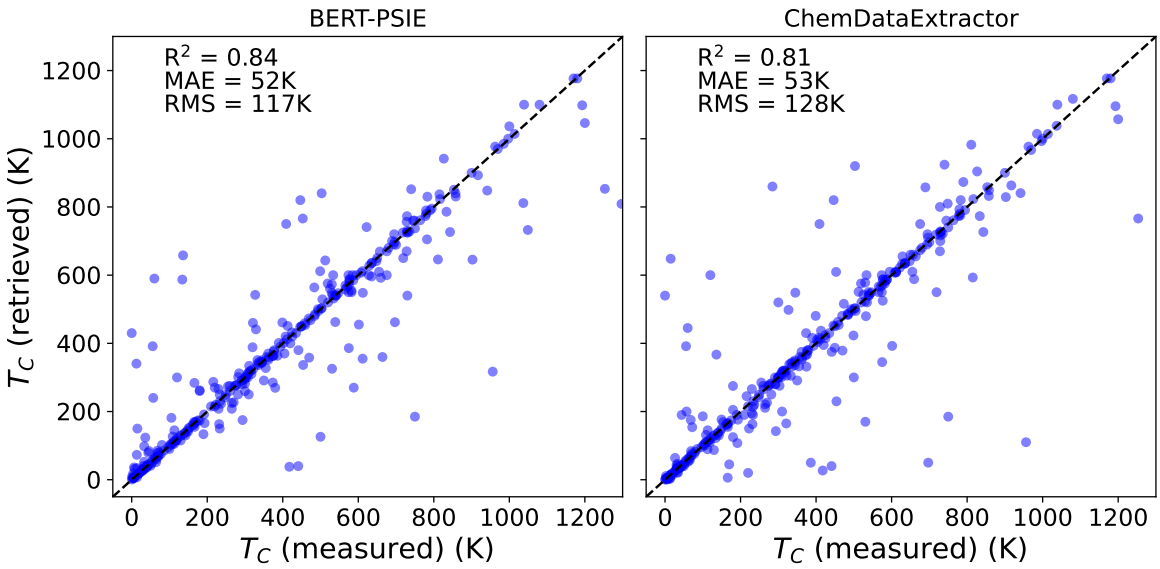


Figure 3.7: Comparison between the T_C queried in the dataset automatically generated by BERT-PSIE and the values contained in the manually curated dataset (left panel). The same comparison is performed on the dataset generated by running ChemDataExtractor on the sentences deemed relevant by our BERT classifier (right panel). The comparison is performed over the 322 compounds that are shared by both datasets.

original two) and the corresponding reduction of the noise present in the median values.

Up to this point, we have compared our database of Curie temperatures, generated with BERT-PSIE, with a database generated with ChemDataExtractor made available by [112]. Nonetheless, the two extractions were executed on different corpora. While overlap is certainly present, since both studies make use of the Elsevier API, this difference can impact the quantitative validity of our comparison. To mitigate such ambiguity, we have run ChemDataExtractor on the sentences deemed relevant by our classifier model. This comparison serves as a benchmark of the performance of our workflow when we replace the NER model and the relation classifier with an extraction purely based on grammar rules, such as the one provided by ChemDataExtractor. The results of the query test on the overlapping entries contained in these two datasets are depicted in Fig. 3.7. Once again, the two automatically generated datasets appear to perform similarly. This result further reinforces the conclusion that BERT-PSIE is able to generate databases of similar quality to those produced by rule-based methods without necessitating the explicit construction of grammar rules.

Band gap

To further validate the performance of BERT-PSIE, we conducted a similar study focusing on the extraction of compounds and their associated band gap. For the manually curated test set in this instance, a database of band gaps from reference [72] is utilised. We compared our results with those of the hybrid ChemDataExtractor model

from Dong *et al.* [122], which was run on the same corpus. The original dataset from this paper is not used as a direct comparison between the models as the workflow implemented in Ref. [122] also processes tables separately, a step not considered by BERT-PSIE. The results of the comparison between the two methods on the same corpus are detailed in Table 3.5. In this case, the BERT-PSIE pipeline outperforms the hybrid ChemDataExtractor method by every metric, while extracting a very similar number of unique compound-band gap relationships. Remarkably, when dealing

	# Entries	Query		
		R^2	MAE (eV)	RMSE (eV)
ChemData.	2185	0.54	0.78	1.34
Single mentions	1,246	0.65	0.67	1.17
Order of appearance	1819	**0.67**	**0.64**	**1.13**
All combinations	2581	0.63	0.71	1.21
BERT-PSIE	2021	0.64	0.67	1.19

Table 3.5: Query test band gap performance comparison between the different datasets against the manually curated one from Ref. [72]. Together with the databases constructed using BERT-PSIE and ChemDataExtractor, we also consider different BERT-assembled datasets obtained by using different relation-classification strategies (see details in the text). The query benchmark is done over the 231 compounds that are shared by all the datasets. Values for the best-performing datasets are in bold.

with sentences containing multiple mentions, the most effective strategy to resolve relations appears to be the order of appearance, which outperforms all other methods. This is in contrast with the degradation in performance observed for the case of Curie temperature, Table 3.4. This disparity could be attributed to an inherent difference in how these two quantities are reported in natural language. It then appears that the reporting of the band gaps is far more procedural than the reporting of the Curie temperatures, thus suggesting that the use of a more sophisticated method of establishing the correct associations between compounds and properties introduces a source of noise. While this result is evidently property-dependent, it is also important to note that the difference in performance between the different relationship extraction methods is marginal.

Finally, we present the parity plot associated with the query test on the data extracted by BERT-PSIE in Fig. 3.8. The results are similar to those observed for the Curie temperature, although in this case, the data are more scattered with respect to the parity line. This larger variance can be associated with the spread in the distribution of various band-gap instances as discussed before (see Fig. 3.5), as well as the noise introduced by the values coming from DFT calculations.

3.2.2 Suitability for machine learning

Another use case for a dataset of experimental compound-properties pairs is to generate a machine-learning predictor of the property given the compound through supervised

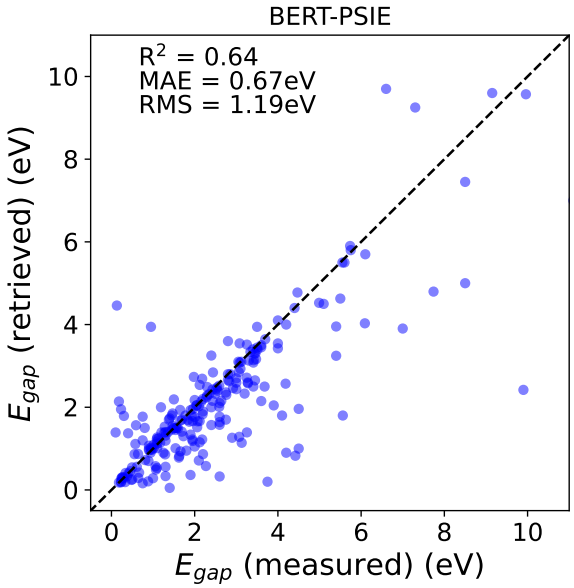


Figure 3.8: Comparison between the band gaps queried in the dataset automatically generated by BERT-PSIE and the values contained in the manually curated dataset from Ref. [72].

training. We already discussed the success of compositional models trained on experimental data and emphasised their importance in an inverse-design setting. The major shortcoming associated with these models is the scarce availability of data on which to perform the training. Here we want to answer the question, Is it possible to train compositional models on automatically generated datasets? And how well would they compare with models trained on manually curated data? With this goal in mind, we introduced a "Suitability for machine learning test" which consists of training on each automatically generated dataset a random forest (RF) model that takes as input compositional features, as presented in Chapter 2. We have chosen the same input features for all the RF models trained, since in all the cases considered we have not observed any improvement when adding more features. We then compare the predictions of the models on a set of compounds that are not present in the training set with the values extracted manually from the literature. For the sake of consistency, the evaluation of this test is done using predictions on compounds that are not present in any of the automatically generated datasets involved in the comparison. The performance metrics used for this test are once again the R^2, MAE and $RMSE$, this time defined with respect to the RF model's predictions on unseen compounds.

Curie temperature

In the case of the Curie temperature extraction task, we consider predictions over 2,623 compounds for which we avail of a manually extracted T_C that does not appear in any of the datasets automatically extracted. For compounds with multiple values of extracted T_C, we use once again the median value of the collated results, following the

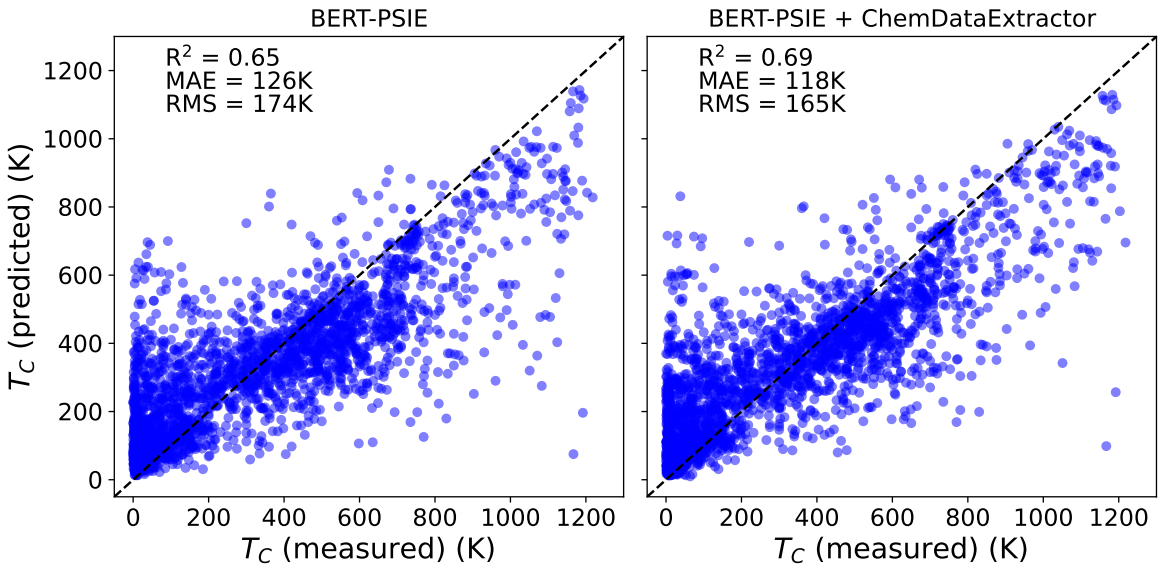


Figure 3.9: Parity plot (predicted T_C vs manually extracted T_C) for the best RF compositional model constructed on the BERT-PSIE dataset (left panel) and on the combined BERT-PSIE and ChemDataExtractor dataset (right panel). The test set consists of the 2,623 compounds that are not present in any of the automatically generated datasets considered in this work, but for which we have a T_C manually extracted from the scientific literature.

same procedure introduced in Ref. [22]. We have also tested other summary statistics, such as the mean and the mode, without finding any significant difference in the results.

The outcomes of this test are reported in Table 3.6, which clearly illustrates that BERT-based extraction workflows perform comparably to the established rule-based method. In particular, the full workflow, BERT-PSIE, has an R^2 identical to that obtained by ChemDataExtract, with a better RMSE but worse MAE, similar to what was observed for the query test. Remarkably, we observe that incorporating entries extracted during the relations-classification step does not improve the predictor's performance. In fact, using only the "Single mentions" datasets we are able to train models with a better R^2 value of 0.66 and an RMSE of 174 K over the test set, whereas BERT-PSIE results in a slightly degraded R^2 at 0.65 and an identical RMSE, although it slightly improves the MAE (by about 2 K). This can possibly be attributed to the additional noise introduced to the database by including entries from sentences with multiple entity mentions. As a consequence, even though the model is trained over a significantly larger dataset, no significant improvement is observed.

The parity plot associated with the RF model trained on the full BERT-PSIE dataset is presented in Fig. 3.9. In general, the T_C trends are captured, however, the model's predictions are significantly worse than the ones of the model presented in Ref. [22]. The model of this paper, trained on manually curated data, reports an MAE of 57 K roughly a factor of two smaller than the 126 K obtained using the data extracted with BERT-PSIE. Part of this discrepancy can be attributed to noise in the

	# Entries	RF predictions		
		R^2	MAE (K)	RMSE (K)
ChemData.	4,289	0.65	**123**	176
Single mentions	1,858	**0.66**	128	**174**
Order of appearance	2,682	0.65	126	176
All combinations	4,308	0.61	134	184
BERT-PSIE	3,518	0.65	126	**174**
BERT-PSIE + ChemData.	7,052	0.69	118	165

Table 3.6: RF T_C predictor performance comparison between the different datasets against the manually curated one from Ref. [22, 127]. Together with the BERT-PSIE and ChemDataExtract databases we also consider different BERT-assembled datasets obtained by using different relation-classification strategies (see details in the text). The RF predictions are done over 2,623 compounds that are not present in any of the automatically collated datasets. Values for the best-performing datasets are in bold.

data, for instance to the likely presence in the BERT-PSIE dataset of critical temperature associated with antiferromagnetic. Moreover, the data used in Ref. [22] underwent extensive curation post-collection. For example, additional data on paramagnets was included to improve the predictions on low-T_C materials, and data corresponding to different concentrations of metallic alloys was selectively excluded in order to balance better the chemical distribution. None of these post-processing steps were undertaken in this study, as our primary objective is to assess the intrinsic quality of the automatically generated dataset.

We also report, in the last line of Table 3.6, the performance of an RF model trained over the combined BERT-PSIE and ChemDataExtractor datasets. Once again, this combined database performs best on all the metrics evaluated in each test. The parity plot associated with this test is located in the right-side plot of Fig. 3.9. The considerably larger number of compounds allows for a better sampling of the chemical space, resulting in more accurate predictions. As it stands, this combined dataset represents the best database available for ferromagnetic T_C, automatically extracted from scientific literature according to the tests designed here. This implies that the quality of automatically extracted databases can improve significantly with an increase in the number of diverse sources on which the extraction is performed. Moreover, as also suggested by the results in Table 3.1.2, this reinforces the possibility that a combination of rules-based and rule-free methods may represent the best-performing strategy for automated extraction.

Finally, we report in Fig. 3.10 the parity plot associated with the RF predictions test comparing the data extracted by BERT-PSIE and ChemDataExtractor when ChemDataExtractor is run on the same sentences deemed relevant by our relevant sentences BERT classifier, further confirming the similarity in performance between these two extraction methods.

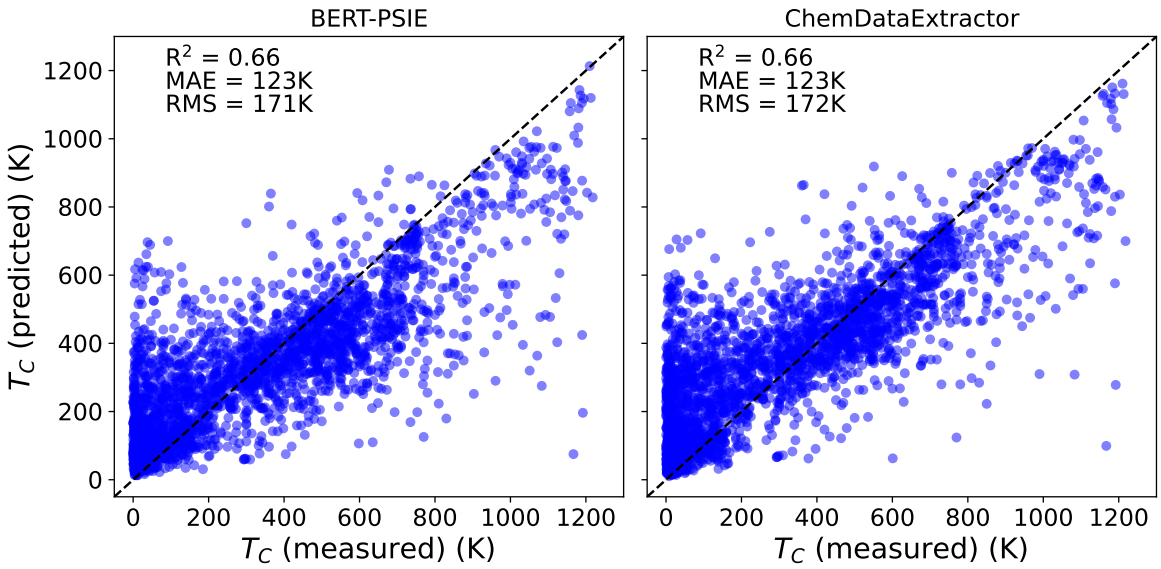


Figure 3.10: Parity plot (predicted T_C vs manually extracted T_C) for the best RF compositional model constructed on the BERT-PSIE dataset (left panel) and on the dataset generated running ChemDataExtractor on the sentences deemed relevant by our BERT classifier (right panel). The test set consists of the 2,885 compounds that are not present in any of the two datasets, but for which we have a T_C manually extracted from the scientific literature.

	# Entries	RF predictions		
		R^2	MAE (eV)	RMSE (eV)
ChemData.	2185	0.59	**0.62**	0.87
Single mentions	1,246	0.61	**0.62**	0.85
Order of appearance	1819	**0.62**	0.63	**0.84**
All combinations	2581	0.60	0.63	0.86
BERT-PSIE	2090	0.61	**0.62**	0.85

Table 3.7: RF predictions performance comparison between the different band gaps datasets against the manually curated one from Ref. [72]. Together with the databases constructed using BERT-PSIE and ChemDataExtractor, we also consider different BERT-assembled datasets obtained by using different relation-classification strategies (see details in the text). The RF predictions are done over 2046 compounds that are not present in any of the automatically collated datasets. Values for the best-performing datasets are in bold.

Band gap

In Table 3.7 we present the RF prediction metrics for the band gap case. Similarly as observed for the query test the BERT-PSIE pipeline outperforms or matches the performance of the hybrid ChemDataExtractor method across all metrics. The parity plot for the RF model prediction for band-gap is shown in Fig. 3.11. The RF model presents a slightly lower R^2 than that constructed for the T_C, but benchmarks similarly with models that can be constructed on manually curated data. In fact, we obtain an MAE of 0.62 eV, to be compared with the value reported on MatBench [138] of

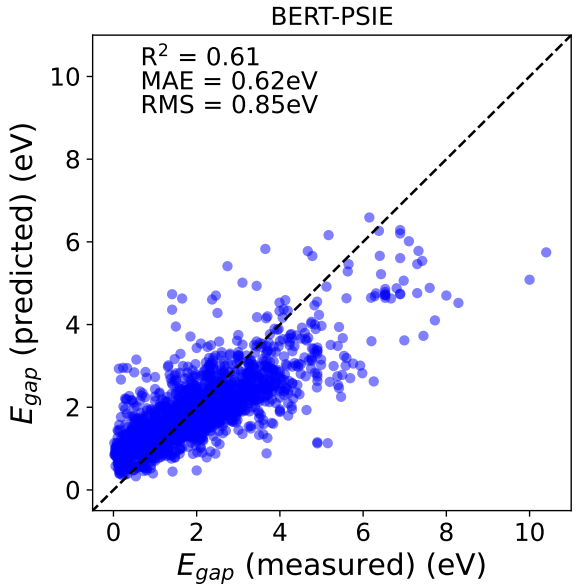


Figure 3.11: Parity plot for the best RF band gap compositional model constructed on the BERT-PSIE dataset. The test set consists of the 2046 compounds that are not present in the dataset but for which we have a band gap from the manually curated corpus.

0.33 eV, for the best-performing model trained on the same dataset. Once again the model trained over manually curated data presents a MAE roughly a factor of 2 lower than the model trained over data extracted automatically.

3.2.3 Screening for inverse design

To simulate the usage of these compositional models trained on our automatically generated dataset, within an inverse materials design workflow, we test their ability to screen unseen compounds with respect to a certain T_C threshold. As previously mentioned, typical magnets employed as part of some room-temperature technology (*e.g.* data storage, electrical motors) require a T_C of the order of 600 K. For this reason, being able to screen potential magnetic compounds according to their predicted T_C is of significant technological relevance. This task is particularly challenging for the T_C since, as discussed in Chapter 2, there is no reliable, high-throughput approach for its *in-silico* prediction.

Using the RF model trained on the automatically generated dataset, we predicted whether magnets have a critical temperature exceeding 300 K, 600 K and 900 K, respectively. We then tested this predictor on compounds that are present in our manually curated dataset, but that do not appear in the one generated by BERT-PSIE. The results of this screening, alongside the distribution of the expected T_C of these compounds, can be found in Fig. 3.12. The shaded blue area of Fig. 3.12 displays the distribution of values predicted to have a T_C greater than the dashed line, representing the screening temperatures of 300 K, 600 K and 900 K respectively. While

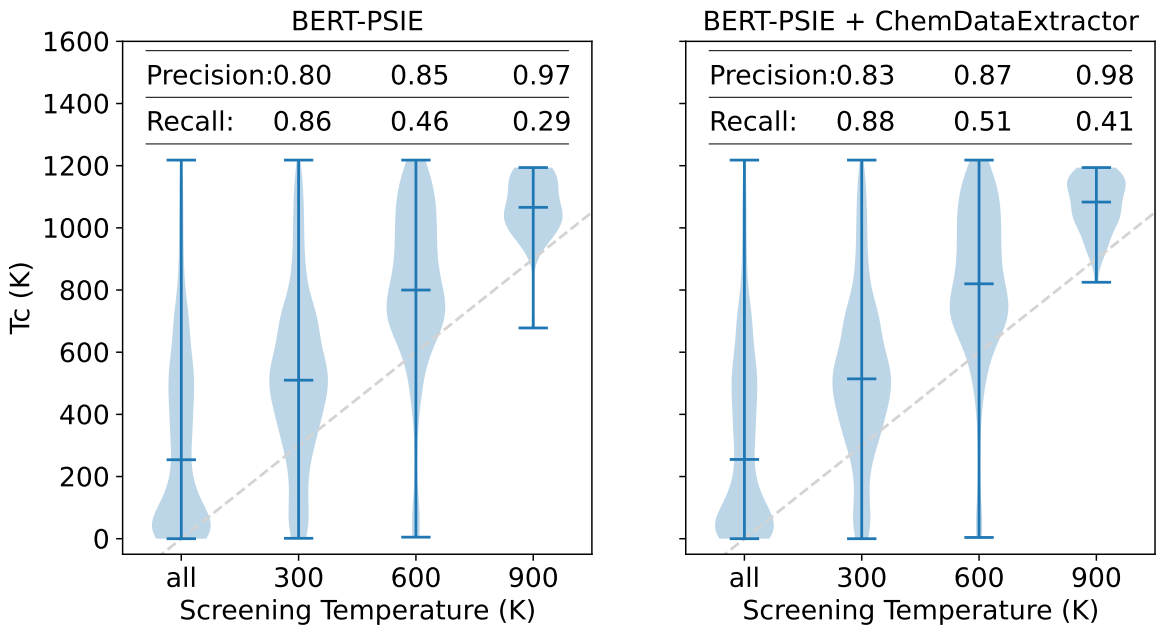


Figure 3.12: Violin plots showing the T_C distributions of the compounds screened using an RF model trained on the BERT-PSIE data and compared with the manually extracted values (left panel). The dashed line is the parity line highlighting how the median of the screened distribution increases as the screening threshold increases. Despite a low recall, the precision is high enough to select compounds likely to have a T_C higher than a given threshold. The screening is done on compounds not present in the training set of the RF. The same test is performed by training an RF model on the combination of the BERT-PSIE and ChemDataExtractor datasets (right panel).

the recall of this screening process is relatively low, the high precision biases the initial distribution into sets with higher and higher T_C, thereby demonstrating the utility of the extracted database in screening for compounds with T_C above a desired threshold. The low recall indicates that certain compounds with T_C above the desired temperature will not be predicted to belong in the set of compounds exceeding that temperature. However, due to the high accuracy of the model, the compounds passing the screening consistently show a T_C above the desired threshold. The use of the better-performing RF model trained over the combined BERT-PSIE and ChemDataExtractor datasets leads to an increase in the recall (left plot Fig. 3.12). This demonstrates that the value of the automatically generated dataset can be improved by increasing its size and by refining the extraction method used.

3.3 Using LLMs

The workflow that we developed relies on the fine-tuning of bidirectional language models such as BERT. Recently, autoregressive generative model models have seen a surge in popularity due to their performance as multitask learners. These models are based on decoder transformer blocks, which apply causal masking when computing

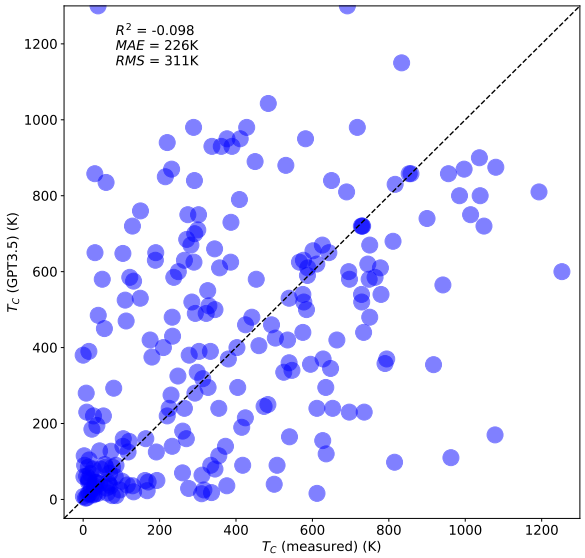


Figure 3.13: Curie temperatures returned by GPT3.5 in response to the no-context zero-shot prompt discussed in the text. The list of chemical compounds provided consisted of the one for which a manually extracted T_C was available. This test shows that the LLM prompt does not retain a prior knowledge of the T_C for a diverse set of compounds.

the attention between the inputs, allowing for more efficient training. As a result, we have seen the creation of LLMs with a number of learnable parameters up to three orders of magnitudes larger than BERT. These models have shown zero-shot learning capabilities [83], namely, they can achieve surprisingly good performances at tasks that they have not been directly trained for. In this section, we explore the integration of LLMs in our extraction pipeline. Recent examples of the usage of LLMs for data extraction from the material science literature rely on the fine-tuning of LLMs in order to generate a JSON file containing the structured data extracted from the paragraph or abstract passed as input [116, 117]. While new strategies to make the fine-tuning of LLMs more efficient have emerged [139], performing a full extraction by just relying on LLMs is unfeasible and probably a waste of resources, given the large amount of text that is required to process (∼100,000 full-text papers). In this section, we explore strategies that minimally modify our workflow for the inclusion of LLMs and observe how this impacts the quality of the data extracted. For this study, we use GPT3.5, an instructed model based on the GPT3 series provided by OpenAI, accessible via API.

3.3.1 No-context zero-shot predictions

Before performing an extraction from scientific papers we want to assess how much the model already knows about the problem. While we do not know the training set used to train GPT3.5, as it has not been disclosed, we can conservatively assume that it has been trained over a snapshot of the entire internet available at the time of training. As such GPT3.5 should have already been exposed to most of the text on

which we are performing the extraction. If we consider the task of extracting the Curie temperature, we want here to assess the ability of the model to provide a T_C value given a chemical formula. In order to make the model perform this task we exploit the instructed nature of GPT3.5. This model has been optimised via reinforcement learning from human feedback to act as a chatbot. From a design point of view, the expected way to interact with this model is by asking him tasks as, if it were a human. Note that slight modifications of the prompt can have a substantial impact on the final answer. In fact, finding what are the features of an optimal prompt is an ongoing research question. We have tried the following prompt which was complemented by a list of compounds, whose curie temperature was available in our manually curated dataset.

> I am going to provide you with a list of chemical compounds and you will generate a list containing the Curie temperature associated with each compound in a JSON object. From now on, you will answer by providing just the requested JSON object and no further information.

In this prompt we are not providing the model with any context, as such in order to answer correctly it needs to have gained such a knowledge during training. This is not the intended use of GPT3.5 however it creates a baseline that we can use to compare more sophisticated strategies. We observe that the answer of the model is accurate regarding most common ferromagnetic elements such as iron and cobalt. However, as can be seen in Fig. 3.13, where we report the comparison between the values returned by GPT3.5 with the manually extracted ones, the model in general shows no prior knowledge for this task.

3.3.2 Contextualised zero-shot predictions

Providing no-context from where to extract the required information results in the model hallucinating the answer, that is, it will provide an answer highly different from the expected value. This is not based on specific domain knowledge, but it is simply "made up". In our study of BERT-PSIE, we have found that the most poorly performing module is the relationship extraction step. In the task of band-gap extraction assigning the relation between compound and property based on the order of appearance led to better performance than training a relation classifier model. In order to explore alternative ways to address this relationship extraction problem, we modify this last step of our workflow to include the use of an instructed LLM. We maintain the use of the relevancy classifier and NER model to screen the bulk of text from the corpus and limit as much as possible the amount of text processed by the LLM in order to save computational resources. We take the sentences classified as relevant and for which the NER model finds an instance of at least one mention of a

chemical compound and at least one mention of a property mention. Each one of these sentences then provides the context (⟨CONTEXT⟩) given in the prompt from which the model is asked to perform the extraction. The following prompt is then sent for each chemical entity ⟨CHEM⟩ found in the sentence

---

You are a material science expert.
You will use the text delimited by triple quotes to answer the question: What is the ⟨PROPERTY⟩ of ⟨CHEM⟩?
If the information requested is not contained in the text write "None"
Answer with a JSON object with a single key named ⟨PROPERTY⟩
Text: """ ⟨CONTEXT⟩ """
Q: What is the ⟨PROPERTY⟩ of ⟨CHEM⟩?
A:

---

where ⟨PROPERTY⟩ can refer to either Curie temperature or band-gap. By using the OpenAI GPT3.5 API, we can programmatically prompt the model customising each prompt by replacing the terms within the angled brackets ⟨⟩. This strategy, allows us to leverage the capabilities of the LLM to establish the presence of a relation between compound and property and correctly extract these two quantities. We applied this approach to process all the sentences for which the NER model identified the occurrence of at least one chemical entity and one property. This effectively replaces the relationship-extraction module of the BERT-PSIE workflow with this GPT3.5 zero-shot prompting procedure.

In Table 3.8, we report the query and suitability for machine learning test results performed on the datasets generated in this way for T_C and band-gap. These generated datasets have improved query performance compared to those extracted using the BERT-PSIE method, thereby demonstrating GPT-3.5's ability to resolve relational dependencies. Another advantage of this procedure is that it eliminates the need for fine-tuning a relationship classifier model.

However, the overall improvements achieved by the addition of LLMs into the workflow are somewhat limited. The suitability of the extracted data to support the training of compositional models sees an improvement with the T_C extraction case, but a deterioration for the band-gap extraction. For the band-gap extraction task, the best strategy to assign the presence of a relation between entities remains the order of appearance, remarking a difference in the reporting of this quantity with respect to the Curie temperature.

This test suggests that the major bottleneck on the generated database quality is mainly related to the narrow context provided when working at a sentence level, rather than shortcomings in the relationship extraction step. Although LLMs can handle significantly larger text inputs than BERT, the computational cost (or API

access cost) associated with the increase of input tokens number limits the scalability of extraction workflows relying on processing large amounts of text. Therefore, any strategy that tries to expand the context size on which the information extraction is performed needs to be carefully designed.

	#Entries	Query			RF predictions		
		R^2	MAE	RMSE	R^2	MAE	RMSE
GPT3.5 (T_C)	2462	0.86	37 K	108 K	0.65	122 K	174 K
GPT3.5 (Gap)	1942	0.69	0.65 eV	1.09 eV	0.58	0.62 eV	0.88 eV

Table 3.8: Performance comparison between the different automatically extracted datasets against the manually curated ones. The left-hand side of the table refers to the query test, while the right-hand side refers to the RF predictor. The extraction is here performed by replacing the last stage of the BERT-PSIE pipeline with a prompting strategy to GPT3.5 (See text for more details).

3.4 Summary

In this chapter, we have presented a workflow for the automatic extraction of structured data from unstructured scientific literature. By relying on the fine-tuning of language models, there is minimal implementation effort required from the final user, and there is little to no need for familiarity with complex grammar-rule definitions and natural language processing. The software implementation of the workflow will be discussed in greater detail in Chapter 7.

All the BERT models utilized in this work were fine-tuned on a single Nvidia A100 GPU, accessed through Google Colab, with each model requiring less than 30 minutes for fine-tuning. Overall, the most significant time-consuming task in adapting BERT-PSIE to a new extraction target is the manual labelling required to generate the training data used for fine-tuning. In order to streamline this process, we developed a graphical user interface (see Chapter 7), which enables the generation of a new extraction workflow in about one week.

We here applied the extraction workflow to two use cases, showing its ability to generate a database of ferromagnetic Curie temperatures and electronic band-gaps comparable to the ones generated using ChemDataExtractor, the state-of-the-art rule-based method for data mining from the scientific literature. Unique to this work, we have carefully benchmarked the constructed databases against manually curated reference ones, using two newly proposed tests that try to reproduce the real-world use cases of the generated data, namely the query test and the suitability to machine learning test. The query test assesses the quality of the data retrieved through the automatic extraction, while the suitability to machine learning test evaluates the correctness of the predictions made by a machine-learning model trained on the extracted data, when applied to unseen data. These two tests offer insights into the impact of the design

choices made during the creation of the workflow and suggest avenues for improvement.

We have also tested how BERT-PSIE would fit within our proposed data-driven inverse-design strategy, by benchmarking the ability of a model trained on the extracted data to perform property-based screening on unseen compounds. We observed that the model was able to bias the Curie temperature of the screened compounds towards temperatures above a set screening threshold.

Finally, we have explored possible ways of integrating LLMs into the extraction workflow highlighting possible directions for future development.

Chapter 4

Atomic structures generation

The content presented in this chapter is based on Refs. [2, 3, 4]. The author of this thesis proposed the original iteration of the machine-learning-assisted workflow for the creation of ternary alloy convex hulls and provided software support. Michail Minotakis and Hugo Rossignol improved upon the original workflow and conducted the study. The thesis author developed and implemented the algorithm for the local inversion of chemical environment representations.

While compositional models have shown surprisingly good performances, the fact that they ignore the crystal structure limits how far the utility of these methods can be pushed in an inverse-design setting. After the first stage of our data-driven inverse-design workflow, we have selected N_{el} elements over which to conduct our search of prototypes. The second stage must focus on establishing, which compounds made up with these elements are likely to be stable. In this context, stability is defined in relation to the constituent elements of the compound. In order to evaluate a compound's stability, we need a methodology to assess the likelihood that it will decompose into its constituents. The thermodynamic stability of a multi-component system, characterised by the atomic concentrations $\{x_i\}$, at fixed pressure p and temperature T, is determined by the minimum of the Gibbs free energy G per atom [140],

$$G(p, T, x_i) = H - TS, \tag{4.1}$$

where H is the enthalpy and S is the entropy of the system. For a fixed stoichiometry at zero temperature, the only variable left in Eq. (4.1) is the crystal structure of the compound. The structure that minimises the enthalpy is the equilibrium configuration of that compound. Achieving this condition is not sufficient for the compound's stability as it has to compete against different phases and the possibility of separating into its constituents. In this regard, a compound will be stable, if its formation Gibbs energy is negative with respect to all possible starting points that can be created based on its stoichiometry $\{x_i\}$. The entropy term of Eq. (4.1) usually becomes the main driver of stability for compounds with a number of constituents larger than

four [25]. For compounds with up to three elements, it is often a reasonable approximation to ignore the entropy term, which is challenging to compute, and consider only the zero-temperature case. Within this approximation, it becomes feasible to estimate the phase diagram of ternary compounds via high-throughput DFT calculations a task that has been carried on by different projects [13, 14, 12]. The quality of a phase diagram is tied to the ability to find the minimum energy structure of each composition examined. Stable phases are represented by compounds that lie on the convex hull of the enthalpy versus stoichiometry diagram. Given the computational cost of DFT, an effective strategy for generating atomic structures is essential for both feasibility and reliability. In this chapter, we discuss two data-driven approaches that we have developed to accomplish this task. One approach focuses on utilizing existing data on binary compounds to construct the phase diagram of ternary compounds, while the other leverages machine-learning generative models.

4.1 ML accelerated ternary phase diagrams

The standard approach adopted by AFLOW for the construction of a ternary convex hull involves performing DFT calculations on a set of prototype crystal structures available for a given stoichiometry. This pool of prototypes is selected from a fixed database of naturally occurring crystal structures and takes the name of dictionary method [141, 142, 143]. The reliability and accuracy of the resulting convex hull will depend on the size of this pool and on the appropriateness of the prototype structures within it. A larger set of prototypes increases the likelihood of identifying the lowest-energy structure, provided that the chosen structures are physically sound. However, the computational cost of DFT calculations limits the number of possible energy estimations that can be performed for each stoichiometry. A natural solution to this problem would be to use MLFF to perform the energy predictions at a fraction of the computational cost of DFT. For this strategy to work a trained model is required. The standard procedure for generating training sets for MLFF involves carrying out DFT calculations on a representative sample of the phase space of the chemical compounds under consideration. However, the cost to generate the training set would likely offset the efficiency gain obtained from using the trained model. For this reason, we explored alternatives to this approach that would not require any additional DFT calculation for the training of the model. Due to the rapid growth of the chemical space with the increase in the number of constituents, the space of binary compounds is significantly better sampled than that of ternary compounds. A significant portion of stable binary compounds are already known and a substantial amount of DFT calculations for these are available on open databases such as AFLOW [13], Materials Project [12] and OQMD [14]. One can then wonder how well an MLFF model trained over binary compounds would perform on the energy prediction of ternary ones.

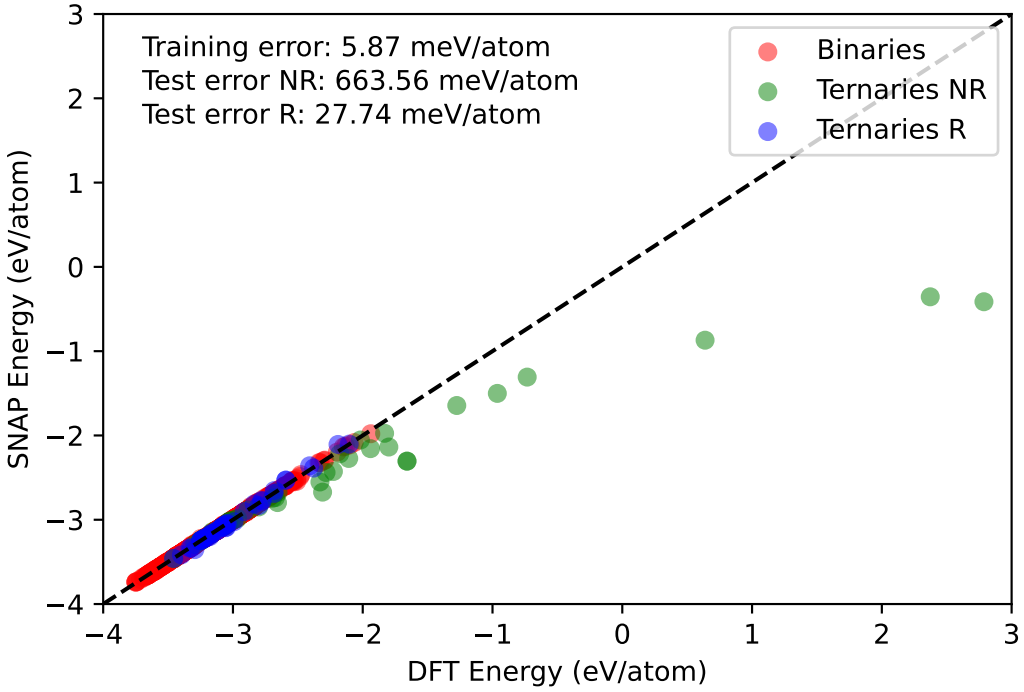


Figure 4.1: Parity plot relative to the predictions of a SNAP model trained on the DFT energies of 715 binary structures comprising Cu, Ag, and Au, downloaded from the AFLOW library. Two different test sets are considered: a dataset of 42 non-relaxed ternary structures generated using the AFLOW dictionary method (NR) and their corresponding relaxed structures (R). The model exhibits poor performance in predicting the energy of the unrelaxed ternary prototypes.

4.1.1 Training over the binaries to predict the ternaries

In order to evaluate the feasibility of using a model trained on binary compounds for energy prediction in ternary compounds, we have focused for simplicity on three noble metals: Cu, Ag, and Au. We have downloaded all available DFT calculations for binaries containing these elements from the AFLOW database. Specifically, these correspond to 261, 191, and 263 structures for Ag-Au, Cu-Ag, and Cu-Au, respectively. Although our ultimate goal is to utilize pre-existing DFT results, we chose to re-perform all DFT calculations in VASP [144, 145, 146], following the AFLOW standard [147], for the sake of proof of concept and to remove possible fluctuations due to inconsistency. We have trained a SNAP model [45] over these 715 binary structures and then generated 42 novel ternary configurations using the AFLOW dictionary method [141, 142, 143]. Since the lattice of these ternary prototypes is not optimised a DFT relaxation was performed on each one of them. We then used these structures to create two test sets of ternary compounds: one containing the initial non-relaxed structures (NR) and another containing the relaxed structures (R). We report in Fig. (4.1) the parity plot relative to the SNAP predictions over these two datasets. The model performs

significantly better on the relaxed prototypes than on the unrelaxed ones. Specifically, it shows an MAE of 663.56 meV/atom on the non-relaxed prototypes, while achieving a significantly lower MAE of 27.74 meV/atom when predicting the energies of the relaxed ternary structures. This indicates that a model trained on binary phases is able to reliably predict the energy of associated ternaries only when their structures are close to equilibrium. However, the model is generally unable to perform relaxation when the starting point is far from the equilibrium. To provide an insight into why this is the case, we have performed a Principal Component Analysis (PCA) on the descriptors representing the chemical environments found in the training set and the ones coming from a DFT relaxation trajectory of a ternary structure that the SNAP model was unable to properly relax. We report in Fig. (4.2) the plot of the two principal components relative to environments in which the central atom is Ag. This plot highlights how the relaxation starts in a region distinct from the environments found in the training set, while the relaxed structure falls well within a region very similar to the environments found in the training. This suggests that the model's poor performance on non-relaxed configurations is likely due to an inadequate sampling of similar chemical environments in the training set. In contrast, the relaxed structures tend to exhibit environments similar to those found in the binary compounds, for which the model has higher predictive accuracy.

4.1.2 Building ternary prototypes from the binary structures

As observed, MLFF models perform better on structures that share chemical environments with the ones found in the training set. We can exploit this fact by constructing ternary prototypes starting from the associated binaries structures near the convex hull. These binary structures are stripped of their chemical identity to create a set of parent prototypes. For each specified stoichiometry, a supercell is built from each parent prototype and is then decorated in a manner compatible with the given composition. The Enulib code is used to enumerate all unique site occupations efficiently through group theory methods [148, 149, 150]. We then trained five SNAP models over the dataset of Cu, Ag, and Au binary compounds, each with individually optimised weight hyperparameters through ten-fold cross-validation. The binary compounds dataset was randomly split in a train and test set ten times with an 80/20 ratio and a model was trained and tested over each split. This strategy was adopted to break the weight symmetry of the models. If the same weights are used for all chemical species then the bispectrum components will be the same for all decorated structures that share the same parent prototypes. The decision to train multiple SNAP models is a consequence of the observed limitations of these models trained on the binary compounds in performing relaxation of the associated ternary structure. Using structures inherited from the binary prototypes helps us ensure that the chemical environments are similar to those found in the training set. Moreover, for each decorated prototype a relax-

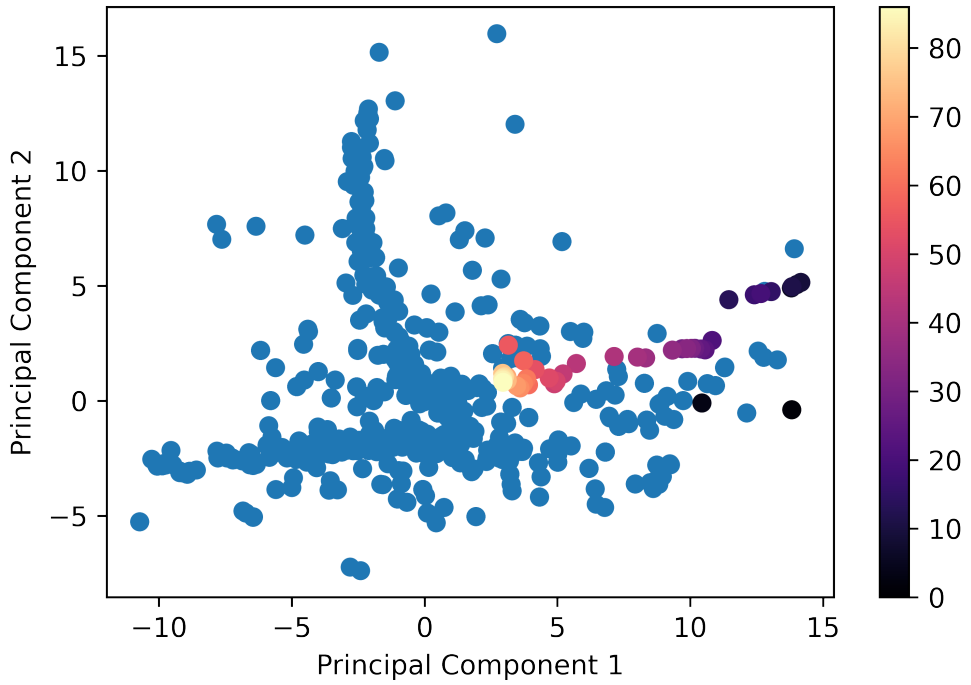


Figure 4.2: Plot of the first two principal components from a PCA performed on the bispectrum components associated with the Ag environments found in the binary structures training set (blue circles). The coloured circles represent the Ag environments of a ternary prototype undergoing DFT relaxation. The colour code indicates the relaxation step index. The DFT relaxation starts in a region poorly sampled by the training set but ends in a region with a high density of similar environments.

ation is performed with each of the five SNAP models leading to five different relaxed structures. The mean and standard deviation of the energy predicted by the other four SNAP models not involved in the relaxation is then calculated for each one of these five relaxed structures. The structure with the lowest associated standard deviation is retained, while the rest are discarded. This process is repeated for all the decorated prototypes and their energies are ranked from lowest to highest. For each explored stoichiometry we select 15 structures with the lowest estimated energy and perform DFT relaxation on them. These relaxed structures are added to the phase diagram and the process is repeated for different stoichiometries. The entire workflow is summarised in Fig. 4.3. In summary, the SNAP models trained on available binary compound data are used to relax and rank decorated ternary prototypes generated from the lowest energy binary structures. The 15 structures with the lowest predicted energy are then deemed "promising" and undergo DFT relaxation. The DFT-relaxed structure with the lowest energy is then added to the phase diagram. The procedure is reiterated for different stoichiometries and the convex hull is constructed. The assumptions on which this workflow is based are:

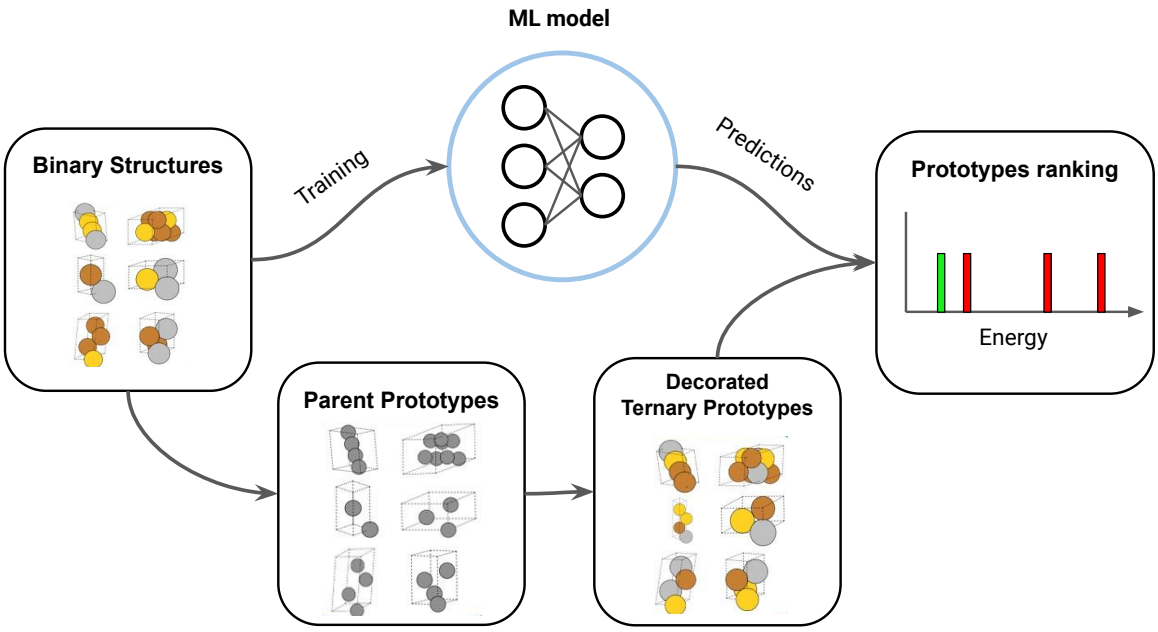


Figure 4.3: Diagram of the implemented workflow for the machine-learning accelerated construction of ternary convex hull diagrams. A pool of prototypes is created from the low-energy structures of the associated binary compounds available on AFLOW. For a given composition, ternary prototypes are generated by decorating supercells of these binary parent structures. Machine-learning models trained on the binary compounds are then used to relax and screen these ternary prototypes. DFT relaxation is subsequently performed on the structures with the lowest predicted energy. The process is repeated for different stoichiometries, and a convex hull diagram is constructed by combining the lowest DFT-energy structures found for each composition.

1. The equilibrium ternary compound structures do not differ significantly from the ones of their associated binary counterparts.

2. MLFF trained on the binary compounds can be used effectively to rank the ternary compounds, especially if they encounter environments similar to the ones in the training set.

In Fig. 4.4 we report the formation enthalpy ΔH_f and the distance from the convex hull for the lowest energy Cu-Ag-Au configurations predicted by our workflow compared with the ones found in AFLOW. Remarkably, our workflow identifies lower energy structures than the ones present in AFLOW for all the cases that we considered. Furthermore, the intermetallic compound $Cu_1Ag_1Au_2$ is predicted to be stable, a result that is compatible with the experimentally observed formation of solid solutions in the gold-rich region of the phase diagram [151]. The ability of our workflow to outperform the AFLOW dictionary method resides in the fact that by leveraging the computational efficiency of MLFF models we can explore a larger pool of prototypes than what would be conventionally possible with DFT. Additionally, by focusing our search on structures derived from the most stable configurations found in associated binary systems,

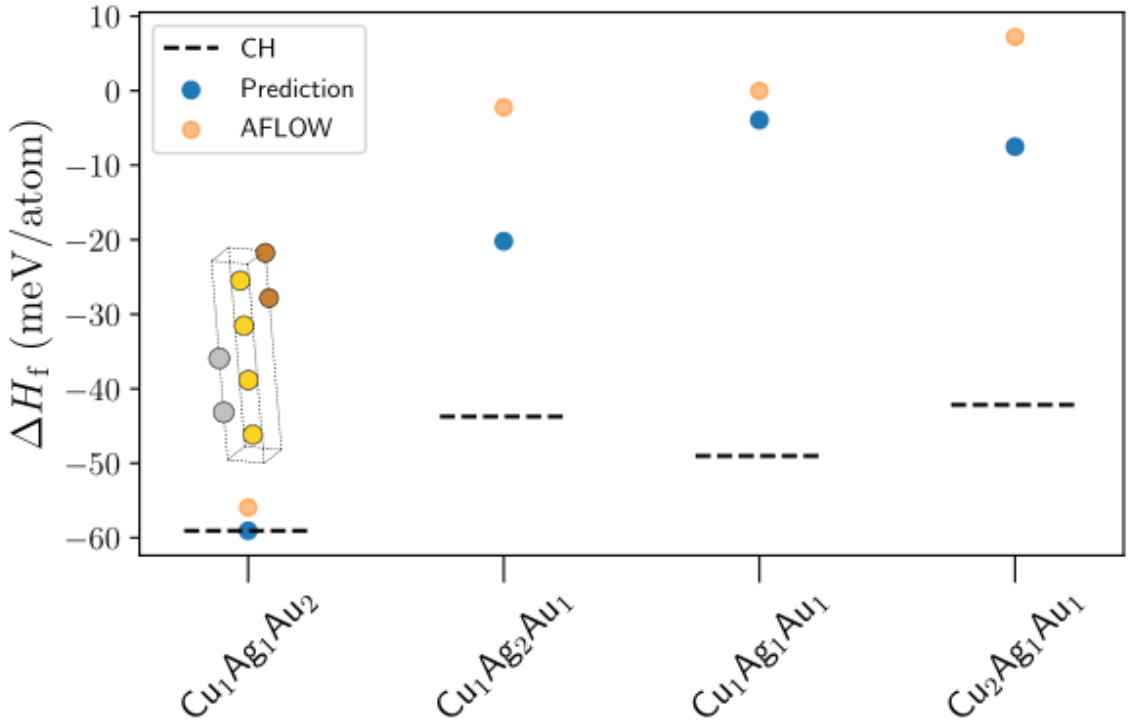


Figure 4.4: Structures with the lowest enthalpy of formation found for four different compositions using our data-driven workflow (blue points). Our results are compared with structures having the lowest formation enthalpy found in AFLOW for these stoichiometries (orange points). The dashed line (CH) marks the position of the tie-plane on the convex hull. The unit cell of the stable structure found for $Cu_1Ag_1Au_2$ is also reported.

we bias our search toward sensible candidates. The reconstruction of the Cu-Ag-Au phase diagram using our workflow involved the screening of around 300,000 candidate prototypes. Performing DFT relaxations over the configurations with the lowest predicted energy ensures that the final convex hull is constructed with DFT-level accuracy. The main limitation of this technique comes from situations where the ternary equilibrium structure is significantly different from the ones of the associated binaries. While this is generally not an issue for metallic alloys, such as the ones considered so far, it could become problematic when considering ternaries including a more diverse set of chemical species. For such cases, generating an accurate convex hull may require broadening the search to include structures generated by different criteria, such as dictionary methods or data-driven generative techniques. The remainder of this chapter will focus on exploring this second approach.

4.2 ML generative models

In the previous section, we introduced a prototype generation strategy, whose efficacy is based on the assumption that the equilibrium ternary compound structures do not differ

significantly from the ones of their binary constituents. While this assumption often holds true for metallic alloys, it is not universally applicable. An alternative route for the generation of promising stable crystal structures involves using generative machine-learning techniques to learn the underlying distribution of discovered stable crystals and use these models to generate new structures similarly distributed. Among the most popular generative techniques, there are the Variational Autoencoder (VAE) [152] and the Generative Adversarial Networks (GAN) [153].

A variational autoencoder is composed of two neural networks, an encoder and a decoder. The encoder projects the input features into a lower dimensional latent space while the decoder has the aim to reconstruct the data from this latent space. The variational formulation of autoencoders introduces a probabilistic interpretation of the latent space and converts its sampling into a stochastic process [154].

In contrast, GAN models feature two networks: a generator and a discriminator. The task of the discriminator is to distinguish between the "real" data sourced from experiments or *ab-initio* simulations and "fake" data produced by the generator. During training, the generator is optimised to create "fake" configurations that would fool the discriminator, while the discriminator is trained to recognize if a certain configuration is "real" or "fake". The training stops when the discriminator is incapable of distinguishing between the "real" configurations and the "fake" ones.

As discussed in Chapter 2, it is not feasible to rely on the simple use of the Cartesian coordinates to describe an atomic structure to a machine-learning model. Introducing an inductive bias in the descriptors by making them roto-translational invariant is one of the most established strategies when it comes to representing the atomic chemical environment for machine learning purposes [33, 34, 37, 38].

The structural descriptors used for the construction of MLFF are not required to be interpretable. In general, this requirement is not considered and the focus in their design is directed toward the correct implementation of the relevant symmetries, as they are what ultimately determine the performance of the model. This means that one can associate a given structure with a set of invariant descriptors, but never has to answer the inverse question, namely which structure is associated with a given set of descriptors. The same is not true for generative methods [155]. In this case, the feature used by either a VAE or a GAN to describe an atomic structure should still be roto-translational invariant. This is because a crystal structure maintains its identity under translations and rotations. However, the output should also be mappable to an interpretable structure, for instance, the chemical identity and the Cartesian coordinates of the atoms forming a molecule. As such, the molecular representation used in generative models should also be invertible.

Possible solutions to this problem include representations that distinguish between different structures based on the concept of chemical bonds, such as the SMILES encoding for organic molecules [156, 157] or general graphs encoding [158, 159]. These

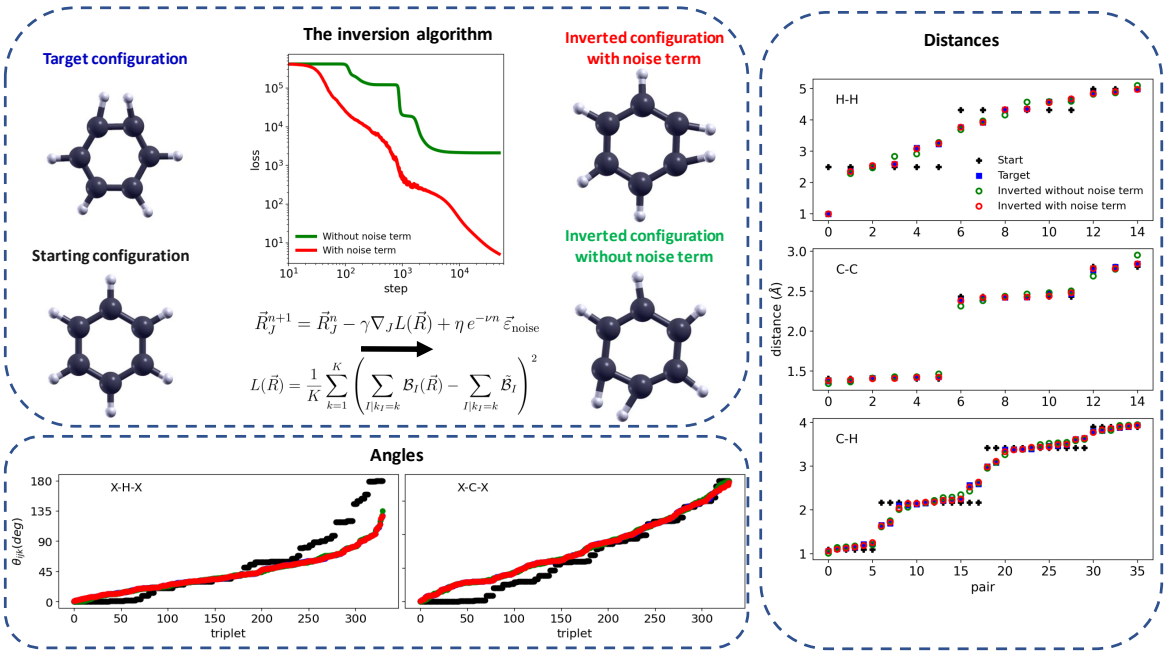


Figure 4.5: Scheme associated with an example of inversion of the bispectrum components starting from a relaxed benzene structure. The target is a deformed benzene molecule chosen for the sake of visualisation. A comparison of the collection of distances and planar angles from all the atomic pairs and triplets in the molecule is shown. After $5 \cdot 10^4$ iterations of the gradient descent algorithm, the inverted configuration (red) closely resembles the target one (blue). We also show in green the result of the inversion process in the absence of the noise term in the update rule of Eq. (4.3), see text for details.

methods, by construction, are capable of capturing the general structure of a chemical entity, but they cannot distinguish between different deformations of the same molecule. For instance, all the configurations encountered by a given molecule over a molecular dynamic trajectory will share the same encoding. As a solution, one can construct representations based on fractional coordinates with respect to a unit cell, which is then able to distinguish between distortions of the same system. These, however, lack rotational and translational invariance and heavily rely on data augmentation to incorporate the fundamental symmetries in the model [160].

Alternatively, the problem can be constrained to a very specific family of structures and discretise the possible atomic positions. This makes the inversion from the model representation to the Cartesian coordinates more easily achieved [161, 162]. However, this strategy lacks universality.

Recent efforts have been made to define invertible descriptors [163] and graph equivariant autoencoders present a promising approach to tackle this challenge [56].

The ideal solution would be to develop a general algorithm that can invert the structural invariant descriptors used in MLFF back into a Cartesian representation. We have developed an algorithm to address this problem, which will be the focus of the following section.

4.2.1 Inversion of the chemical environment representations

The work presented in this section is a first step towards a general inversion strategy of the invariant descriptors of the local chemical environments. In particular, we have built a scheme to locally invert any representation based on many-body local descriptors of the chemical environment, with the goal of making them available to be used in generative models. The idea is to frame the problem as a structural optimisation having as a target an unknown structure, expressed in terms of the many-body representation. This does not represent a general inversion scheme, meaning that we are not able to find a general one-to-one relation between a many-body and the Cartesian representation. Nevertheless, it allows one to find unknown target structures of known molecules. For example, it can determine the Cartesian coordinates of a given molecule with an unknown distortion.

Most MLFFs assume that the total energy of a molecule/solid can be expressed as a sum of atomic contributions each dependent on the local environment of the associated atom of the system. For instance, the total energy of a molecule made of N atoms can then be written as the sum of atomic contributions as seen in Eq. (2.15). Each atomic contribution is then a function of the atomic environment, $\mathcal{B}_I$ as shown in Eq. (2.27). The specific choice of the descriptors significantly impacts the performance of the MLFF. As such, it is often necessary to construct the $\mathcal{B}_I$'s so as to satisfy the symmetries of the quantity that one wants to predict. In the case of the total energy, the structural descriptors are designed to be invariant with respect to roto-translations, and permutation of atoms of the same species. The best-performing local descriptors are often many-body in nature [164] and their transformation from the Cartesian coordinates is not globally invertible.

Here, we show that the local inversion of this transformation can be achieved through an iterative optimisation of an initial atomic configuration, by means of a gradient descent algorithm. The core concept, illustrated in Fig. 4.5, consists of optimising a molecular structure until its descriptors representation matches a given set of target descriptors (for instance, obtained from a generative model). Therefore, given a set of target descriptors, $\{\tilde{\mathcal{B}}_I\}$, and the Cartesian coordinates of a starting configuration, $\{\vec{R}_I\}$, our algorithm updates the atoms' positions until the associated structural descriptors of the molecule, $\{\mathcal{B}_I(\vec{R})\}$, coincide with $\{\tilde{\mathcal{B}}_I\}$ within a numeric tolerance. In order to quantify the distance between the target descriptors and the optimised ones, we introduce the following loss function,

$$L(\vec{R}) = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{I|k_I=k}\mathcal{B}_I(\vec{R}) - \sum_{I|k_I=k}\tilde{\mathcal{B}}_I\right)^2 , \tag{4.2}$$

where K represents the number of distinct chemical species present in the system. Accordingly, the external sum runs over the possible species, while the internal one

runs over the atoms belonging to a given species. The form of $L(\vec{R})$ has been chosen to be invariant under the permutation of atoms of the same species. By using a gradient descent algorithm [44] it is then possible to update the coordinates of the initial configuration to minimise $L(\vec{R})$. At the n-th iteration the $(n+1)$-th update of the Cartesian coordinates $\{\vec{R}_I\}$ is given by,

$$\vec{R}_J^{n+1} = \vec{R}_J^n - \gamma \nabla_J L(\vec{R}) + \eta\, e^{-\nu n}\, \vec{\varepsilon}_{\text{noise}}, \tag{4.3}$$

with:

$$\nabla_J L(\vec{R}) = \frac{2}{K} \sum_{k=1}^{K} \left[\left(\sum_{I|k_I=k} \mathcal{B}_I(\vec{R}) - \sum_{I|k_I=k} \tilde{\mathcal{B}}_I \right) \cdot \sum_{I'|k_{I'}=k} \nabla_J \mathcal{B}_{I'}(\vec{R}) \right]. \tag{4.4}$$

In this context, γ is the learning rate and $\vec{\varepsilon}_{\text{noise}}$ is a vector whose components are random numbers sampled from a uniform distribution between -1 and 1 at each gradient-descend iteration. This noise term introduced in Eq. (4.3) has the purpose of breaking the symmetry at configurations, where the gradients of the descriptors of the local environment tend to vanish and, in general, it is found to make the inversion process more robust. The coefficient η determines the coupling strength of this term, while ν controls its exponential decay with the iteration number. Since the relation between the Cartesian coordinates and the atomic descriptors is globally invertible, the loss function of Eq. (4.2) has multiple global minima. For example, for a given set of rotationally invariant descriptors, every rotation of the target configuration will correspond to a global minimum of the loss with $L = 0$. As a consequence, this is a non-convex optimisation problem. The region of the possible coordinates explored during the optimisation process is constrained by the initial configuration selected. This guarantees the possibility of local inversion of the relation between the Cartesian coordinates and the descriptors.

We also want to remark that the same argument applies when the descriptors are incomplete [36], as discussed in Chapter 2. The incompleteness of the descriptors implies that it is possible to find two distinct atomic structures, which are mapped to the same set of descriptors. This, in turn, results in an increase in the number of global minima of the loss $L(\vec{R})$. Consequently, the configuration reached at convergence will depend on the initial configuration, which then needs to be cleverly chosen.

Results

We now demonstrate the validity of our presented inversion method by inverting the relation between Cartesian coordinates and descriptors for a selected sample of molecules. Specifically, we choose as structural descriptors the bispectrum components [34] presented in Chapter 2 and defined in Eq. (2.24). In Fig. 4.5 we illustrate an example of inversion of the bispectrum components of a deformed benzene molecule. The start-

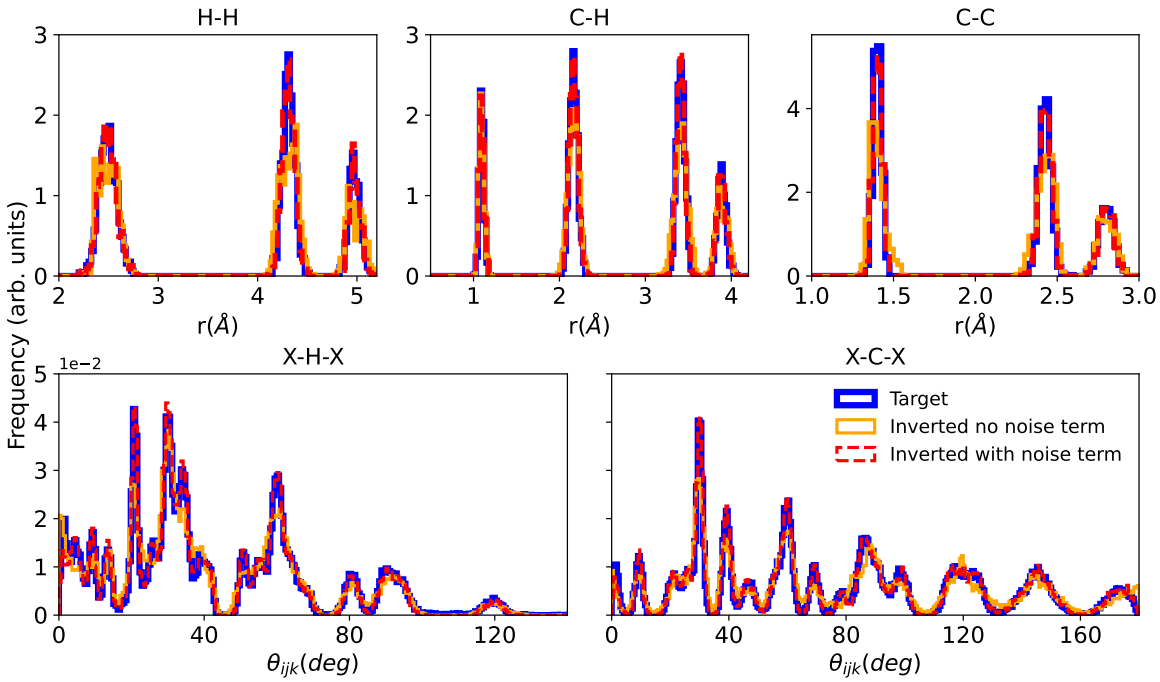


Figure 4.6: Comparison of the partial pair distributions and of the angular distributions between the target molecules and the ones resulting from the inversion for a sample of 120 benzene molecules.

ing configuration in this case is an optimised benzene molecule at equilibrium. The carbon-carbon and carbon-hydrogen distances in this molecule are 1.40 Å and 1.09 Å, respectively.

The structure associated with the target descriptors is found by iteratively updating the atomic positions with the goal of minimising the loss of Eq. (4.2). In order to quantitatively evaluate the quality of the inversion, we compare the atom-pair distances between the molecule before and after the optimisation procedure against the target. A similar comparison is then repeated with the angles formed by all possible atoms triplets in the molecule. We observe that $5 \cdot 10^4$ iterations are enough to reconstruct a molecule that closely resembles the target one.

Additionally, in Fig. 4.5 we also show the results obtained with an inversion in the absence of the noise term in the update rule of Eq. (4.3). In this second case, the final configuration reaches a local minimum of $L(\mathbf{r})$, with all atoms remaining in the planar arrangement of the initial configuration, despite the target atoms being located out of plane. This occurrence is due to the fact that the gradients of the bispectrum components in the out-of-plane direction of a planar configuration are zero. Notably, this occurrence is not restricted to the case of planar molecules, but it appears at configurational high-symmetry points. It is a feature of all local invariant descriptors of the chemical environments induced by the symmetries imposed on them. However, in our context, this feature limits the configurational space that our inversion algorithm is able to explore. In order to address this issue, we have included the noise term in

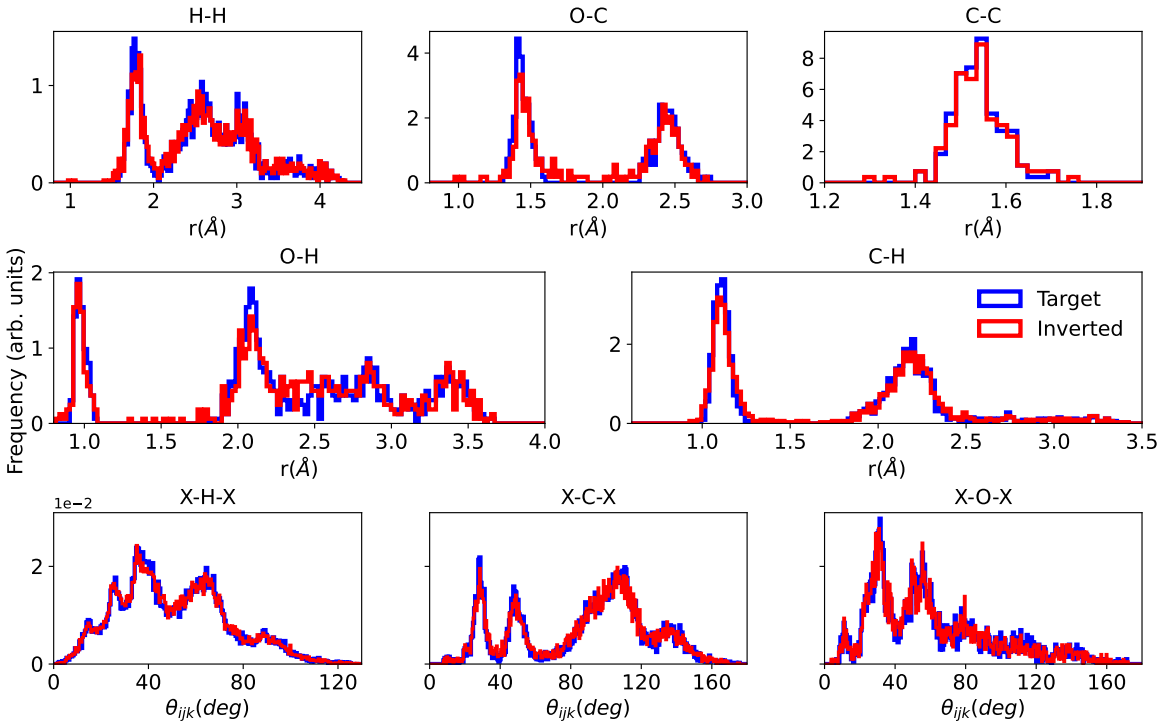


Figure 4.7: Comparison of the partial pair distributions and of the angular distributions between the target molecules and the ones resulting from the inversion for a sample of 120 ethanol molecules.

the update rule with the purpose of breaking all the possible symmetries of the initial configuration, thus improving the outcome of the inversion procedure.

For a more systematic study, we use the bispectrum components of a sample of molecules extracted from the MD-17 benchmark dataset [59] as target descriptors. This dataset comprises *ab-initio* molecular dynamics trajectories of simple organic molecules at 500 K. As starting configuration of the inversion process, we utilise the first relaxed geometry of the molecular dynamics trajectories. The inversion algorithm is then used to reconstruct the Cartesian coordinates associated with 120 sets of bispectrum components sampled from each trajectory. The parameters used for the gradient descent algorithm are $\gamma = 4 \times 10^{-8}\text{Å}^2, \eta = 1 \times 10^{-2}\text{Å}$ and $\nu = 1 \times 10^{-3}$, while the bispectrum components have been computed with: $l_{\max} = 4$ and $r_c = 6.0\,\text{Å}$.

In Fig. 4.6 we present the partial pair-distance distributions and angular distributions of the Cartesian coordinates of reconstructed benzene molecules, comparing to the target configurations sampled from the trajectory. Each inversion is carried for 5×10^4 iterations. It is evident that our inversion procedure can generate configurations, which closely reproduce the structural distributions of the targets. Importantly, the omission of the noise term in the update rule of Eq. (4.3) results in a deterioration of the inversion performance, consistently with our previous discussion. Benzene represents an optimal choice for the application of our inversion procedure since atoms of the same species are all equivalent and overall the molecule is fairly rigid.

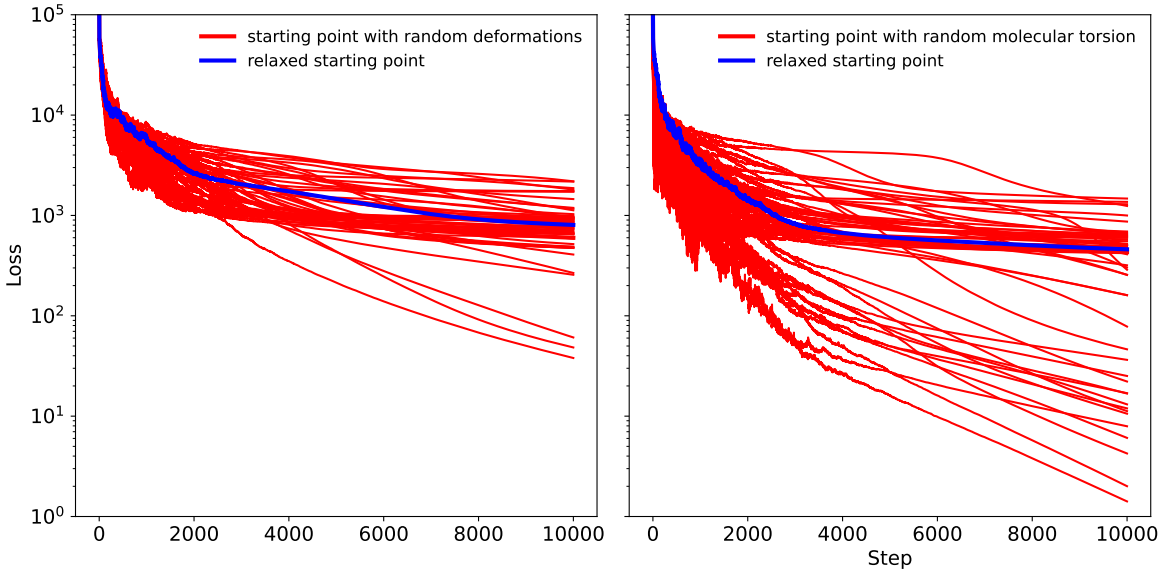


Figure 4.8: Example of the convergence path of a distorted ethanol molecule, where the optimisation procedure is initiated from different starting configurations, see text for details

Ethanol, however, presents a more challenging test. Each carbon atom in ethanol is surrounded by a different local chemical environment and both the C-O and C-C bonds are mobile, thus allowing for molecule torsions. Inversion results for this case are reported in Fig. 4.7 for a sample of 120 ethanol molecules. Here we compare again the partial pair and the angular distributions between the inverted configurations and the targets. In this instance as well, the distributions associated with the inverted configurations are in good agreement with the targets. In particular, the width of the peaks of the partial pair distributions is very similar. However, in this second example, we encounter some configurations that demonstrate to be more problematic to invert. This introduces a general deterioration of the performance of the inversion procedure, evidenced by the different peak heights in the two distributions. These less-converged configurations can be identified by the loss alone, since during the optimisation $L(\vec{R})$ stops improving and stabilises at a value relatively higher than the zero value expected for ideal convergence, indicating the presence of a local minimum.

The configurations explored by the inversion algorithm heavily depend on the choice of the starting configuration. Multiple restarts of the inversion, using different initial configurations, can potentially lead to improvements in the overall performance of the algorithm. This is demonstrated here, again for the ethanol molecule, where we compared two approaches to generate starting configurations. In the first case, we distort the relaxed configuration, previously used as a starting point, by applying random uniform displacements in the atomic positions in the range $[-0.1, 0.1]$ Å. In the second case, two random independent rotations on the C-O and C-C bond respectively, on the relaxed ethanol configuration.

We show in Fig. 4.8 the convergence curves (loss function vs. iteration number) of the inversion of the descriptors of a distorted ethanol molecule, where different curves correspond to 60 different starting configurations generated in the two ways described. After 10^3 steps the majority of the inversions halt at configurations with a loss between 10^3 and 10^4 relative to the target configuration and the optimisation seems to have reached a plateau, indicating that the optimisation is trapped in a local minimum. However, for some initial conditions, the loss continues to decrease reaching significantly small values. This is true in particular for the case where the initial condition has been sampled among different molecular torsions. This example demonstrates how the knowledge of the deformation modes of the system under consideration can guide the choice of the initial conditions, leading to an improvement of the inversion procedure and faster convergence.

Lastly, we examine the inversion process on the remaining molecules within the MD-17 dataset. Once again we use the same optimised structure taken from the first step of the trajectory for all the molecules considered. We report the inversion results in Fig. 4.9 where we compare the total pair-distributions of the generated structures with respect to the target ones. An inspection of the loss evolution with the iteration number for Malonaldehyde, Salicylic acid and Aspirin shows that several configurations reach a local minimum of the loss function, similar to what was observed with Ethanol. This results in only a partial optimisation of the molecule and leads to an incomplete ability to reproduce the target distributions. In contrast, for Uracil, Toluene and Naphthalene almost all configurations are converged as confirmed by the remarkable similarity between the two distributions.

All the results presented here are for finite molecules, leading one to question whether the same algorithm can be applied to solids, either crystalline or amorphous. In general, the pair distribution function of a molecule tends to be more sharp and sparse than that of a solid. This distinction translates into the fact that constructing an MLFF for molecules is typically less data-hungry than for solids, with the resulting potential often being more stable. These characteristics are likely to extend to the inversion problem, which in general will be numerically more complex for solids than for molecules. For these reasons, in the context of solids, we anticipate that inversion optimization will be more nuanced and may necessitate a broader exploration of initial conditions.

The inversion algorithm introduced has been here applied only to the bispectrum components. However, its implementation is universal and can be combined with any local descriptors, such as the power spectrum [34], symmetry functions [33], ACE [38] or JLP [5]. We have showcased the effectiveness of this method across various simple molecules and demonstrated that the algorithm can efficiently overcome symmetry-induced local minima by introducing a noise term in the iteration updating rule. An improved convergence can be achieved by performing the simulations over a multiple

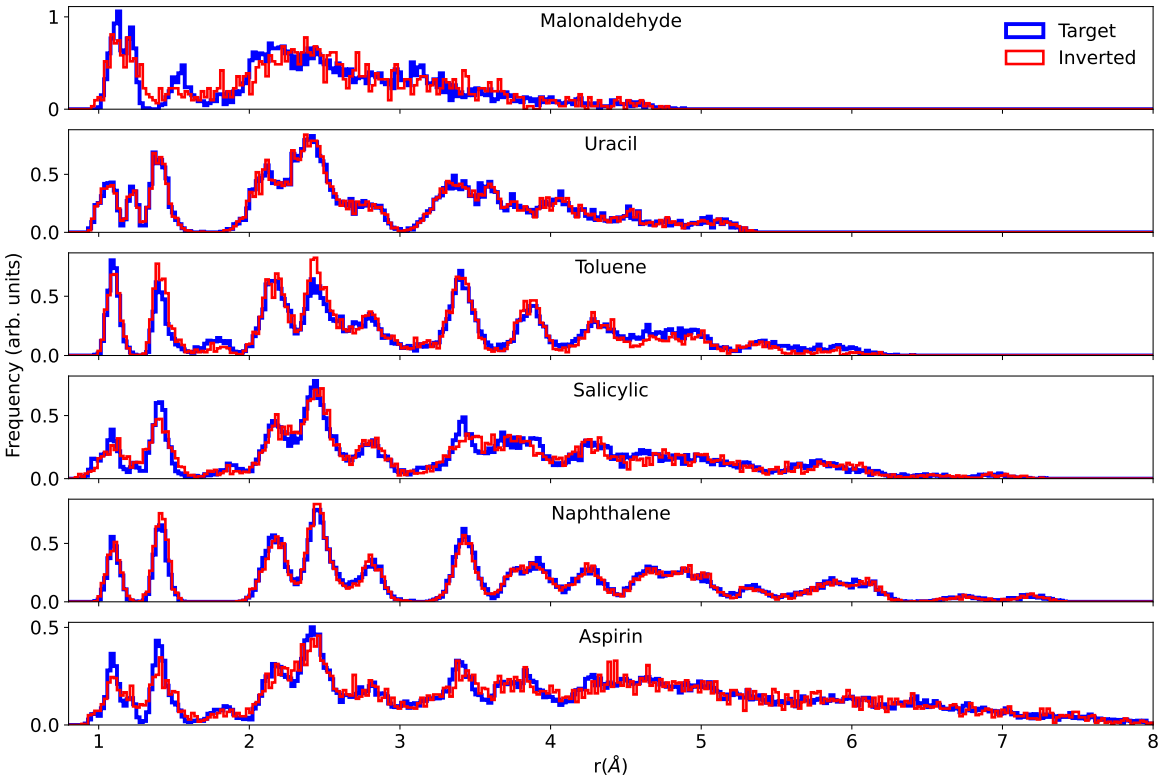


Figure 4.9: Comparison of the total pair distribution between the target molecules and the ones resulting from the inversion for a range of molecules contained in the MD17 dataset.

set of initial conditions. The highly non-convex nature of this optimisation problem might benefit the use of derivative-free optimisation algorithms, however, we reserve the exploration of this direction for future work.

4.3 Summary

In this chapter, we have addressed the problem of determining the stable compositions and associated structures derived from a given pool of elements. This challenge is ultimately connected to the one of finding a sensitive strategy for the generation of crystal structure prototypes. The data-driven solutions proposed in this work comprise the generation of ternary prototypes using the low-energy binary crystal structures as a starting point and the use of generative machine-learning models to reproduce the distribution of the known stable crystals.

We were able to develop a workflow that outperforms the AFLOW strategy in constructing the ternary convex hull for Cu-Ag-Au alloys. Furthermore, we proposed an algorithm for the local inversion of the descriptors of the chemical environments, which are commonly used as features for MLFF models. The proposed inversion algorithm is a first step in the construction of generative models designed to utilise such invariant descriptors as this method allows us to find the Cartesian coordinates of a structure

compatible with a given set of these descriptors.

Once a compound is predicted to be stable it is passed to the third stage of our proposed data-driven inverse-design workflow, which focuses on a more fine-grained property-screening.

Chapter 5

Property predictions

The content presented in this chapter is based on Ref. [5]. The author of this thesis contributed to the software implementation and testing of the Jacobi-Legendre Potential, together with Michelangelo Domina and Urvesh Patil.

At this stage of our inverse-design pipeline, we have selected the chemical species to be included in our final proposed prototype compounds, as well as the most promising stable stoichiometries and their associated crystal structures. The screened compounds can now be further tested by computing their dynamical properties using machine-learning force fields trained on datasets tailored to model these compounds with the same accuracy as first principle methods. As discussed in Chapter 2 there is a rich literature on machine-learning potentials each one suitable within its specific use case. In this chapter, we present the Jacobi-Legendre potential (JLP), a set of roto-translational invariant descriptors based on a cluster expansion of the energy of the system which displays some similarities with the Atomic Cluster Expansion descriptors [38].

5.1 The Jacobi-Legendre potential

We assume that the atomic energy contributions to the total energy of the system can be expressed as a multi-body expansion

$$E_I = E_I^{(1)} + E_I^{(2)} + E_I^{(3)} + \dots, \tag{5.1}$$

where each n-body contribution $E_I^{(n)}$ depends only on n-atoms clusters including the atom I. For example, $E_I^{(1)}$ represents a constant shift in the energy and only depends on the chemical identity of I. $E_I^{(2)}$ will be a function of atoms pairs including I, while $E_I^{(3)}$ considers atoms triplets including I etc.... From this assumption and from Eq. (2.15) the total energy of the system can also be decomposed in a multi-body cluster expansion:

$$E_{Tot} = E^{(1)} + E^{(2)} + E^{(3)} + \dots, \tag{5.2}$$

where each n-body term is given by

$$E^{(n)} = \sum_{I=1}^{N} E_I^{(n)}, \tag{5.3}$$

It is generally assumed that the atomic contributions $E_I^{(n)}$ depend only on the atoms within a certain cutoff. This assumption is based on the consideration that the interaction between two atoms diminishes as their distance increases. It also ensures linear scaling with respect to the number of atoms in the system for the machine-learning potential. Ever since the successful generalisation of the coupling scheme from the power spectrum (a 3-body representation) and the bispectrum (a 4-body representation) [49, 34] to include higher-body orders as first showcased in the ACE potential [38], every new potential derived from this concept shares the same fundamental assumptions (energy's many-body expansion and locality). Their primary distinction lies in the construction of their basis functions or in the introduction of entirely novel basis sets [165]. The JLPs follow this trend. They're structured upon a specific selection of basis functions (both radial and angular), ensuring their comprehensiveness and enabling a direct correlation between the components of a JLP and its ACE counterpart. Specifically, we choose the Jacobi polynomials for the radial basis and the Legendre polynomials for the angular dimension. The Jacobi polynomials with real argument [166],

$$P_n^{(\alpha,\beta)}(x) = \frac{1}{2^n} \sum_{j=0}^{n} \binom{n+\alpha}{j} \binom{n+\beta}{n-j} (x-1)^{n-j} (x+1)^j \,. \tag{5.4}$$

are parametrised by two real parameters, α and β. Each pair of parameter values is associated with a complete set of orthogonal polynomials in the interval $[-1, 1]$. Two notable examples are the Legendre polynomials ($\alpha = \beta = 0$) and the Chebyshev polynomials of the second kind ($\alpha = \beta = \frac{1}{2}$). Treating α and β as hyper-parameters enables optimisation of the radial basis set, eliminating the need for manually choosing the best polynomial basis.

Two body contribution

The two-body contribution $E_I^{(2)}$ is a function of pair-wise interactions including the atom I in the total energy of the system. By imposing the roto-translational invariance on this term we force it to depend only on the distance and atomic identity of the atoms involved. We can then decompose these terms into pair potentials $v^{(2)}_{Z_I Z_J}(R_{JI})$ describing the interaction of the dimer:

$$E_I^{(2)} = \sum_{J \neq I} v^{(2)}_{Z_I Z_J}(R_{IJ}) \,. \tag{5.5}$$

These two-body potentials are defined to be invariant with respect to the exchange of I and J, namely $v^{(2)}_{Z_I Z_J} = v^{(2)}_{Z_J Z_I}$. The following is the proposed JLP expansion of the two-body potential:

$$v^{(2)}_{Z_I Z_J}(R_{JI}) = \sum_{n=1}^{n_{\max}} a_n^{Z_I Z_J} \widetilde{P}_n^{(\alpha,\beta)} \left(\cos\left(\pi \frac{R_{IJ}}{R_{\text{cut}}} \right) \right) , \tag{5.6}$$

where the sum is truncated up to a maximum degree $n_{\max}$ in order to handle only a finite amount of terms that will be used for the linear model. The expansion coefficients $a_n^{Z_I Z_J}$ are in this context learnable parameters that are optimised at training time. The polynomials $\widetilde{P}_n^{(\alpha,\beta)}$ are defined in terms of the Jacobi polynomials as:

$$\widetilde{P}_n^{(\alpha,\beta)}(x) = P_n^{(\alpha,\beta)}(x) - P_n^{(\alpha,\beta)}(-1) \quad \text{for } -1 \leq x \leq 1 \, , \tag{5.7}$$

and they are now used as the invariant descriptors of the local chemical environments of the model. Their rotational invariance follows immediately by the fact that they only depend on relative atomic distances between pairs of atoms of the system. The definition in Eq. 5.7 guarantees that they vanish smoothly at the cutoff R_{cut}, which is fundamental to guarantee continuity in the forces during relaxation or molecular dynamics. These descriptors of the chemical environment present four hyper-parameters, α, β, r_{cut} and $n_{\max}$, with α, β real numbers greater than -1. In practice, we observe that choosing $\alpha = \beta$ leads to well-performing models while reducing the total number of hyperparameters required to be optimised. The expansion coefficients $a_n^{Z_j Z_i}$ should inherit the same symmetries imposed on the potential, in particular, they should be symmetric under the exchange of the atomic species, which translates to the condition $a_n^{Z_j Z_i} = a_n^{Z_i Z_j}$. Care needs to be taken when implementing these descriptors so that this symmetry is enforced while fitting the model.

Three body contribution

The three-body contribution $E_I^{(3)}$ is a function of the interactions between triplets of atoms that include I. In this case, the enforcement of a roto-translational symmetry imposes each three-body potential term to be a function of the relative distances between the atoms of the triplet and of the angles that they define. To reconstruct the three-body cluster, we only need one angle from the triangle formed by the triplet, provided that the distances are known. Therefore, we can express the three-body contribution associated with the atom I as:

$$E_I^{(3)} = \sum_{JK} v^{(3)}_{Z_I Z_J Z_K}(R_{JI}, R_{KI}, \hat{R}_{JI} \cdot \hat{R}_{KI}) \, , \tag{5.8}$$

where the scalar product $\hat{R}_{JI} \cdot \hat{R}_{KI}$ encodes the angle formed by the three atoms with respect to I. The definition of $v^{(3)}$ should be independent of the order of the two

non-central atoms

$$v^{(3)}_{Z_I Z_J Z_K}(R_{JI}, R_{KI}, \hat{R}_{JI} \cdot \hat{R}_{KI}) = v^{(3)}_{Z_I Z_K Z_J}(R_{KI}, R_{JI}, \hat{R}_{KI} \cdot \hat{R}_{JI}) \,, \tag{5.9}$$

The JLP expansion of the three-body potential is chosen as

$$v^{(3)}_{Z_I Z_J Z_K}(R_{JI}, R_{KI}, \hat{R}_{JI} \cdot \hat{R}_{KI}) = \sum_{n_1,n_2=2}^{n_{\max}} \sum_{l=0}^{l_{\max}} a^{Z_I Z_J Z_K}_{n_1 n_2 l} \, \overline{P}^{(\alpha,\beta)}_{n_1 JI} \overline{P}^{(\alpha,\beta)}_{n_2 KI} P_l^{IKJ} \,, \tag{5.10}$$

where $P_l^{IKJ} = P_l(\hat{R}_{JI} \cdot \hat{R}_{KI})$ is the Legendre polynomial of degree l [166]. The polynomials

$$\overline{P}^{(\alpha,\beta)}_{n_1 JI} = \overline{P}^{(\alpha,\beta)}_{n_1 JI} \left(\cos\left(\pi \frac{R_{IJ}}{R_{\text{cut}}} \right) \right) \,, \tag{5.11}$$

based on the Jacobi polynomials, are defined so that they vanish smoothly at the origin and at the cutoff distance.

$$\overline{P}^{(\alpha,\beta)}_n(x) = \widetilde{P}^{(\alpha,\beta)}_n(x) - \frac{\widetilde{P}^{(\alpha,\beta)}_n(1)}{\widetilde{P}^{(\alpha,\beta)}_1(1)} \widetilde{P}^{(\alpha,\beta)}_1(x) \,. \tag{5.12}$$

By imposing this constraint, the three-body contribution vanishes when one of the neighbour atoms approaches the central atom. has a consequence, the repulsive short-distance interaction between atoms can be modelled exclusively by the two-body term of our expansion. This facilitates the introduction of an inductive bias in the model by selecting the potentials that display a repulsive pair potential at a short range.

Since the two distances and the angle appearing in Eq. (5.10) are independent of each other they can be factorised and treated individually, the two distances using Jacobi polynomials and the angular part using Legendre polynomials. In principle, one could use Jacobi polynomials for the angular part as well, at the cost of introducing more hyperparameters to the problem. The choice of the Legendre polynomials, which are a special case of the Jacobi polynomials, stems from their relationship with the spherical harmonics which are commonly used to expand the angular part of functions in spherical coordinates [166, 34, 38]:

$$P_l(\hat{R}_1 \cdot \hat{R}_2) = \frac{4\pi}{2l+1} \sum_{m=-l}^{l} (-1)^m Y_l^m(\hat{R}_1) Y_l^{-m}(\hat{R}_2) \,. \tag{5.13}$$

Four body contribution

Similarly, the four-body contribution $E_I^{(4)}$ is a function of the interactions between 4-body clusters of atoms that include I. The enforcement of a roto-translational symmetry imposes each four-body potential term to be a function of the relative distances between the atoms, as well as of the three angles with respect to I formed by the three

3-atoms sub-clusters constituting the quadruplet.

$$E_I^{(4)} = \sum_{JKP} v_{IJKP}^{(4)}(R_{JI}, R_{KI}, R_{PI}, s_{IJK}, s_{IPK}, s_{IJP}) \,, \tag{5.14}$$

with $s_{IJK} = \hat{R}_{JI} \cdot \hat{R}_{KI}$. The JLP expression of $v^{(4)}$ is then built in a similar fashion to Eq. (5.10) associating Jacobi polynomials to each relative distance and Legendre polynomials to each angle

$$\begin{aligned} &v_{IJKP}^{(4)}(R_{JI}, R_{KI}, R_{PI}, s_{IJK}, s_{IPK}, s_{IJP}) = \\ &= \sum_{n_1 n_2 n_3 = 2}^{n_{max}} \sum_{l_1 l_2 l_3 = 0}^{l_{max}} a_{n_1 n_2 n_3, l_1 l_2 l_3}^{Z_I Z_J Z_K Z_P} \overline{P}_{n_1 JI}^{(\alpha,\beta)} \overline{P}_{n_2 KI}^{(\alpha,\beta)} \overline{P}_{n_3 PI}^{(\alpha,\beta)} P_{l_1}^{IJK} P_{l_2}^{IJP} P_{l_3}^{IKP} \,. \end{aligned} \tag{5.15}$$

As in the previous cases Eq. (5.15) contains identical terms which need to be imposed to be equal to guarantee that the final model displays the correct permutation symmetries, namely that:

$$\begin{aligned} &a_{n_1 n_2 n_3, l_1 l_2 l_3}^{Z_I Z_J Z_K Z_P} = a_{n_2 n_1 n_3, l_1 l_3 l_2}^{Z_I Z_K Z_J Z_P} = a_{n_3 n_2 n_1, l_3 l_2 l_1}^{Z_I Z_P Z_K Z_J} = \\ &= a_{n_1 n_3 n_2, l_2 l_1 l_3}^{Z_I Z_J Z_P Z_K} = a_{n_2 n_3 n_1, l_3 l_1 l_2}^{Z_I Z_K Z_P Z_J} = a_{n_3 n_1 n_2, l_2 l_3 l_1}^{Z_I Z_P Z_J Z_K} \,. \end{aligned} \tag{5.16}$$

Higher body contribution

The procedure outlined so far for the construction of the potentials can be extended at any body order. To reflect the invariance when permuting the order of any non-central atoms within the cluster, symmetries must be imposed on the expansion coefficients. These constraints are associated with the symmetry of the Jacobi-Legendre expansion, particularly in terms of the permutation of relative distances and angles within the cluster.

5.1.1 JLP for Carbon

To demonstrate the capabilities and use cases of the JLP, we present the construction of an interatomic potential for carbon trained and tested on DFT calculations performed in [27] used to train a Gaussian Approximation Potential (GAP) [49]. The dataset includes various phases of carbon, spanning from crystalline structures like graphene, graphite, and diamond, to surfaces and amorphous phases. For the fitting process, we excluded all carbon dimers as well as any structures with absolute maximum force components exceeding 30 eV/Å. In total, 37 structures were removed: 30 carbon dimers used in the two-body GAP fitting and 7 other structures that failed to meet the maximum force criteria. The remaining $4,043$ structures were divided into a training set of $2,830$ and a test set of $1,213$ configurations. We utilised the DFT energy, forces, and virial stress of each configuration as training targets of the linear model as described in Eq. (2.36). A discussion regarding the derivatives required to

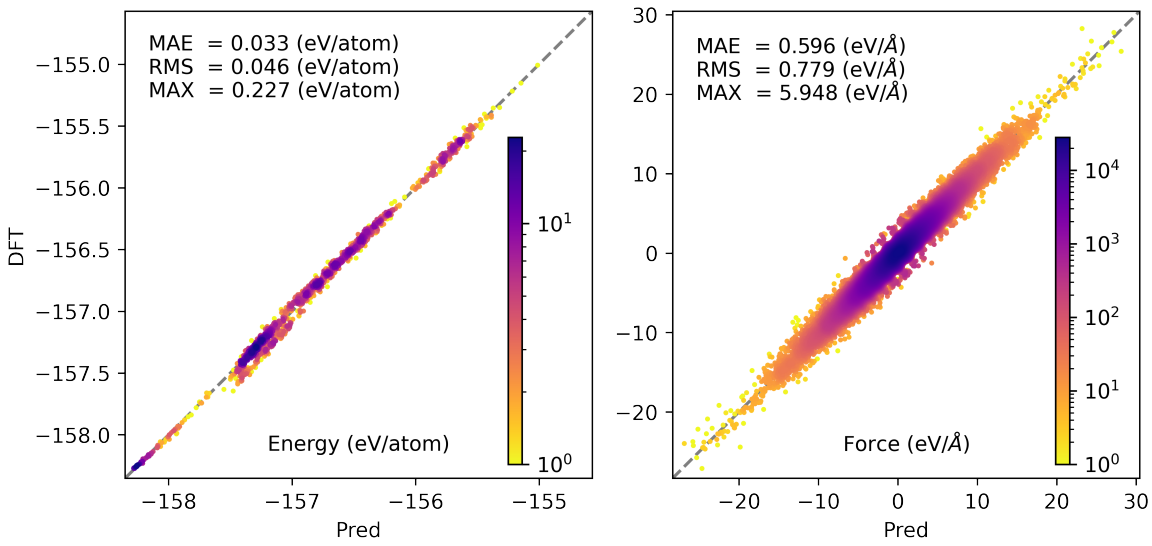


Figure 5.1: Parity plots computed over the test set for the JLP energies (left) and forces (right) predictions. The Mean Absolute Errors (MAEs) and Root Mean Square Error (RMSE) are reported for each plot, alongside with the error on the worst prediction. The colour code indicates the data density (number of points).

compute the forces and the stress from a JLP model is reported in Appendix A. All

	Two Body	Three Body	Four Body
$n_{\max}$	10	6	4
$l_{\max}$	–	5	3
R_{cut} (Å)	3.7	3.7	3.7
$\alpha = \beta$	1	1	1
num. of features	10	90	364

Table 5.1: hyperparameters used for the JLP trained on the carbon dataset from Ref. [27]. In order to reduce the number of hyperparameters, we fix α and β to be equal. The resulting linear model has 465 learning parameters.

the hyperparameters of the model have been optimised on the cross-validation performance over the training set. The values chosen for this problem are reported in Tab. 5.1. The coupling of the forces γ_F and the stress γ_W in the loss function, are chosen to be 0.5 and 0.075 respectively. In Fig. 5.1 and Fig. 5.2 we report the parity plots relative to the predictions over the training set. The model reaches a root mean square error (RMSE) of 43.9 meV/atom, 0.779 eV/Å and 6.62 eV over the energy, forces and stress predictions. The structures deviating the most from the energy-parity plot are all associated with amorphous carbon. These structures are particularly challenging since the dataset contains amorphous configurations at various densities displaying a large variation of coordination numbers. The predicted components of the virial stress appear to be in remarkable agreement with the ones computed with DFT. As it can be seen from the energy parity plot in Fig. 5.1, the trained carbon JLP is able to describe, on an equal footing, both the physics of crystalline carbon around equilibrium

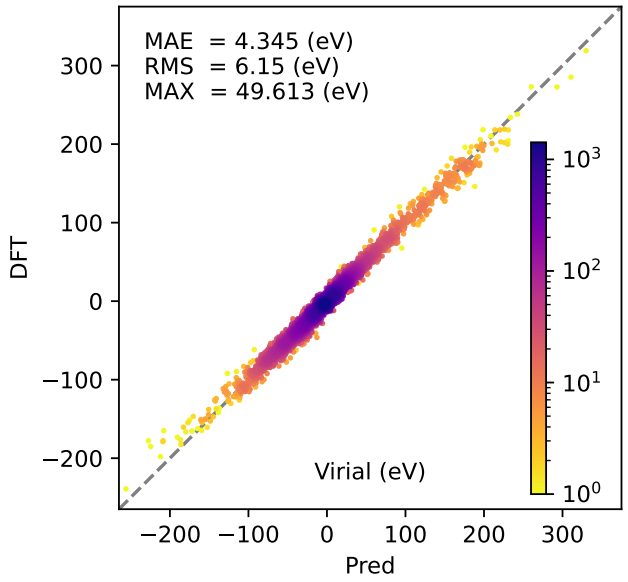


Figure 5.2: Parity plots computed over the test set for the JLP virial stress predictions. The Mean Absolute Errors (MAEs) and Root Mean Square Error (RMSE) are reported, alongside with the maximum error registered among the predictions. The colour code indicates the data density (number of points).

(low-energy points) and the liquid and amorphous structures (high-energy points).

Reproducing the correct short-range behaviour of the interatomic potential can be challenging for machine-learning models. The reason for this is due to the fact that generally, the training set does not contain configurations in which the atoms are significantly close to each other. Since such configurations would be unphysical and hard to converge with DFT. As a consequence of this, the predictions of the model over configurations containing atoms at very short distances are performed in an extrapolation regime and they can be unreliable. It is a possibility that the short-range behaviour of the potential, learned by the model, turns out to be attractive, limiting the model's ability to perform molecular dynamics. There are two strategies commonly used to alleviate this problem. The inclusion of compressed structures can improve the sampling of the repulsion behaviour of the potential, however, there is a limit to what it is the maximum compression which is feasible to explore imposed by the ability to converge the DFT calculation. Another strategy consists of adding a repulsive term to the potential, this is commonly done with SNAP models [45]. With JLP the short-range behaviour is controlled by the two-body contribution due to the constraint imposed in the higher order terms for the radial part polynomials to vanish at the origin. We can reconstruct the two-body potential by means of Eq. (5.6) and analyze its behaviour at the origin, we can then select the hyperparameters of the models relative to the two-body term to naturally display a short-range repulsive behaviour. This approach results in the introduction of an inductive bias in our model driven by our prior physical knowledge of the problem. We report in Fig. (5.3) the two-body reconstruction relative to the carbon JLP showing a strong short-distance

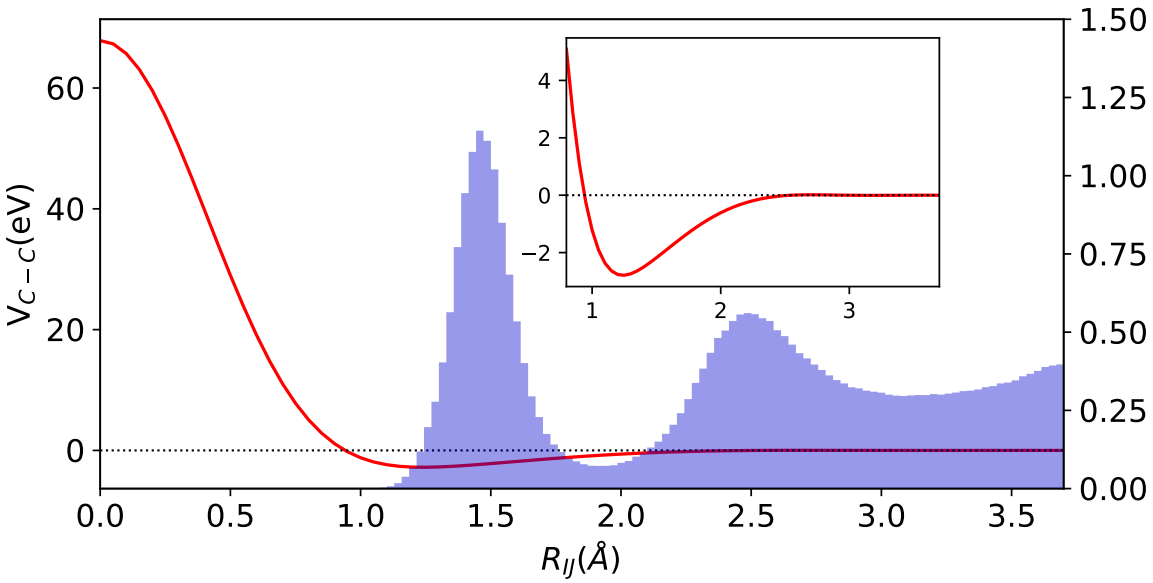


Figure 5.3: Reconstruction of the 2B potential from Eq. (5.6) (red curve). The insert shows a magnification around the minimum, while the histogram reports the pair-distance distribution of the entire dataset. Qualitatively, the potential shows a strong repulsive behaviour for small distances and a minimum, which is consistent with the position of the first peak in the pair-distances distribution.

repulsive behaviour. Moreover, the potential displays a shallow minimum in proximity to the first peak in the radial distribution function computed over the training set (blue shadow). Once the force field has been fitted it can be used to predict the dynamical properties of the system via molecular dynamics. As the case examined here, there is also the possibility of simultaneously modelling multiple phases of the system, and similar examples of this can be found in the literature [26].

In order to show one of the properties that can be computed using the carbon JLP we calculate the phonon dispersion curves (see Appendix B) for diamond and graphene. Firstly we relax the unit cell of diamond and graphene using the trained JLP potential then we compute the phonon dispersion via finite difference using the phonon3py package [167, 168]. The results obtained are compared in Fig. 5.4 with a reference phonon dispersion for crystalline diamond taken from the materials project [12] and for graphene taken from the phonon website[1]. These reference calculations have been performed using density functional perturbation theory using the ABINIT code [169]. This test is particularly challenging, since the training dataset has an energy spread of several eV/atom, while the energy differences computed in the finite-difference scheme used here are a few meV/atom from the equilibrium energy. Overall, we observe good agreement between the JLP-computed phonon bands and the DFT reference ones, for both the acoustic and optical branches. The largest disagreement is

[1] `https://henriquemiranda.github.io/phononwebsite/phonon.html?json=http://henriquemiranda.github.io/phononwebsite/localdb/graphene/data.json`

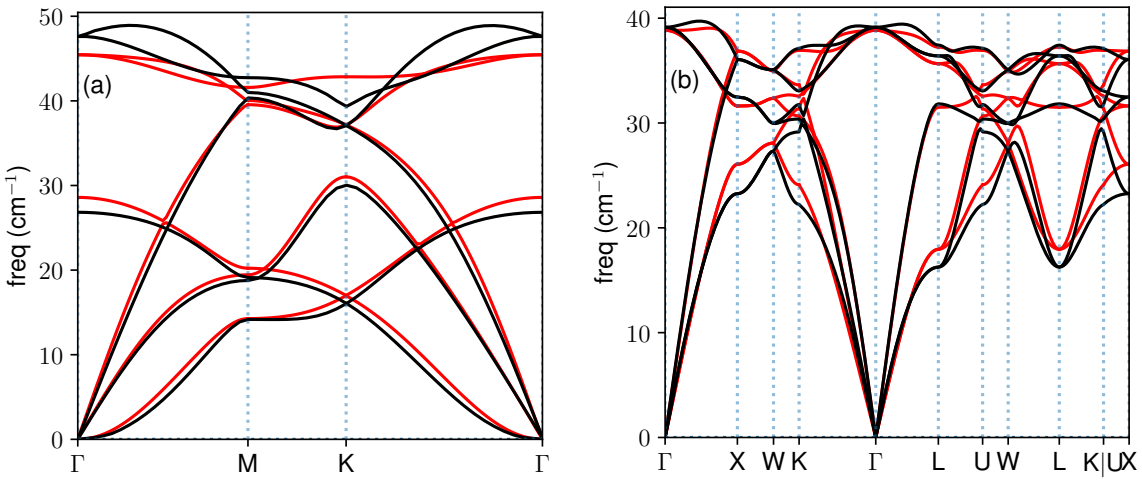


Figure 5.4: Phonon spectra for (a) graphene and (b) diamond computed with the optimised JLP described in the text (red lines). The reference DFT calculations (black lines) have been obtained with density functional perturbation theory as implemented in the ABINIT code.

generally found for the optical branches and it is of the order of 2 cm^{-1} (see, for instance, the graphene bands at around 45 cm^{-1}). These differences can be explained by the fact that the DFT dataset used for the training was obtained with the CASTEP code [170] and the phonon via finite differences, while our DFT reference has been generated with ABINIT [169] and density functional perturbation theory, hence a perfect match between the two dispersion it is not expected. Additional differences can also be ascribed to the different pseudopotentials used and to details in the DFT implementation. The good agreement between the phonon dispersion curves highlights the ability of this potential to be used for the study of the dynamical stability of the candidate prototypes within an inverse-design workflow.

5.2 Summary

In this chapter, we explored the final stage of our proposed inverse-design pipeline which focuses on further computing properties of the compounds that the previous stage predicts to be stable. We introduced the Jacobi-Legendre potential, a linear interatomic potential that uses as features invariant descriptors of the local chemical environments based on the Jacobi and Legendre polynomials. As an example of the capabilities of the JLP and of MLFF in general, we trained a model for carbon trained over crystal configurations as well as amorphous ones. Depending on the application-specific target, different properties predictions will become the priority at this stage of the inverse design. MLFF allow the calculation of properties that can be inaccessible with conventional DFT. This is made possible by the efficiency and scalability of the model architectures once trained. Up to this point, we have not considered the modelling of the magnetic properties of the system, which are of primary importance for

the design of magnetic materials. Conventional local chemical environment descriptors used in MLFF do not include any information relative to the magnetic state of the system. As such, they can reliably model magnetic materials only in cases where only the lowest energy magnetic state is explored and this is strongly coupled with the atomic configuration. If this is the case, the model would be in principle able to infer the magnetic configuration of the system from the local chemical environment of the atoms. However, whenever the modelling of excited magnetic configurations is required, as would be the case for computing the Curie temperature, better integration of the magnetic degrees of freedom into the model is required.

Chapter 6

Magnetic property predictions

The content presented in this chapter expands on Ref. [6] to provide an example application of the spin power spectrum. Michelangelo Domina conceptualised the spin power spectrum. Ümit Daglum performed the spin spiral DFT calculations. The author of this thesis provided the software implementation of the spin power spectrum and conducted the study.

In the previous chapter, we discussed how machine-learning techniques can accelerate the calculation of materials properties from first principles. In this chapter, we want to focus specifically on the modelling of magnetic materials' properties. Magnetic properties arise from the presence of an atomic magnetic moment attributed to the atoms of the system. The motion of the electrons around the nucleus combined with their spin can lead to a net non-zero magnetic moment associated with the atom, in the case of atoms with unfilled electron shells such as transition metals or rare-earth elements. The properties of materials containing such elements are strongly influenced by the dynamics of these atomic spins. These dynamics can generate long-range ordering, resulting in macroscopic effects such as ferromagnetism and antiferromagnetism [78]. For these materials, the total energy is not only a function of the atomic positions, as for the cases discussed so far, but it also depends on the specific orientation of the atomic magnetic moments (See Fig. 6.1).

6.1 Force fields with spin

For a system of N magnetic atoms, the total potential energy is now a function of both the atomic positions $\{\vec{R}\}$ and of the atomic magnetic moments $\{\vec{S}\}$. Time-reversal symmetry imposes that the Hamiltonian of the system should not depend on the choice of the quantisation axis. This constrain propagates to the total energy of the system

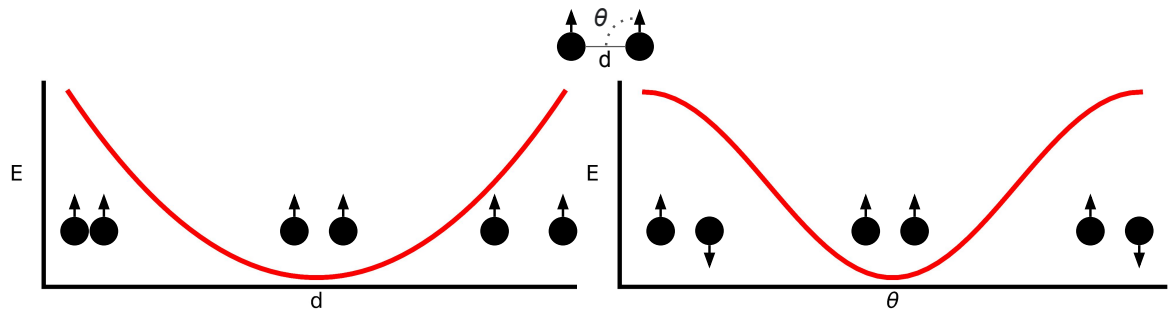


Figure 6.1: In a semi-classical view of a magnetic material we associate a magnetic moment vector to each magnetic atom of the system. The potential energy (red line) depends on both the position of the atoms (left panel) and also on the orientation of these atomic spins (right panel). In general, this dependence is coupled.

imposing the following form [171]

$$E_{Tot}(\{\vec{R}\},\{\vec{S}\}) = E(\{\vec{R}\}) + \sum_{I=1}^{N} E^{(1)}(\{\vec{R}\})\vec{S}_I^2 + \sum_{I=1}^{N} E^{(2)}(\{\vec{R}\})(\vec{S}_I^2)^2 + \sum_{I\neq J} E^{(3)}(\{\vec{R}\})\vec{S}_I \cdot \vec{S}_I + \ldots \quad (6.1)$$

In general, this expansion will contain an infinite number of even powers of magnetic moments and their scalar products. Longitudinal fluctuations can be neglected in some ferromagnetic materials such as iron. In this case, by considering only the leading term of the expansion, the total potential energy of the system can be approximated by the Heisenberg Hamiltonian,

$$E_{Tot} = E(\{\vec{R}\}) - \frac{1}{2}\sum_{I\neq J} \mathcal{J}_{IJ}(\{\vec{R}\})\, \vec{s}_I \cdot \vec{s}_I, \quad (6.2)$$

where $\vec{s}_I$ are unit vectors with the direction of the atomic magnetic moments and the magnitude is included in the exchange coefficients $\mathcal{J}_{IJ}$, which are a function of the positions of the atoms I and J. Within a semi-classical approximation, we treat each atomic spin as a classical vector moment and not as a quantum angular momentum operator. The magnetisation of the system is then given by the average net magnetic moment of all the atoms,

$$\vec{\mathcal{M}}(\{\vec{S}\}) = \frac{1}{N}\sum_{I=1}^{N} \vec{S}_I. \quad (6.3)$$

In the absence of an external magnetic field, the magnetisation of the system will depend on the temperature and on the strength of the magnetic contribution coefficients, $E^{(1)}$, of Eq. (6.1). At zero temperature for a system modelled by the Heisenberg

Hamiltonian, the ground-state spin configuration will depend on the sign of the exchange coefficients. If $\mathcal{J}_{IJ}$ is positive for all $I-J$ pairs a configuration with all the spin in the same direction is favoured, resulting in a ferromagnetic state. At finite temperature, ferromagnetism is only observed if the ensemble average of the mean magnetic moment of the system is non-zero,

$$\langle\vec{\mathcal{M}}\rangle(T) = \frac{1}{N}\int \vec{\mathcal{M}}(\{\vec{S}\})\exp\left(-\frac{E_{Tot}(\{\vec{R}\},\{\vec{S}\})}{k_B T}\right)\mathrm{d}\Omega(\{\vec{R}\},\{\vec{S}\}), \tag{6.4}$$

where the integral is extended over the phase space of all atomic and spin configurations accessible to the system. In practice computing the integral of Eq. (6.5) is unfeasible and it is necessary to approximate it. One approach involves taking the average over a spin-lattice dynamic trajectory, where the spins are allowed to evolve accordingly to their equations of motion, in a fashion similar to molecular dynamics [172]. Another approach consists of performing the configuration averages obtained via Metropolis importance sampling Monte Carlo [173].

6.1.1 Metropolis Monte Carlo

The probability distribution of the spin-lattice configurations accessible to the system is given by,

$$p(\{\vec{R}\},\{\vec{S}\}) = \frac{\exp\left(-\frac{E_{Tot}(\{\vec{R}\},\{\vec{S}\})}{k_B T}\right)}{\mathcal{Z}}. \tag{6.5}$$

Here $\mathcal{Z}$ indicates the partition function, which ensures the normalisation of the probability distribution. The challenge of efficient sampling from $p(\{\vec{R}\},\{\vec{S}\})$ is given by the fact that the majority of the configurations that can be obtained by randomly assigning values to the atomic positions and spin components would have a high total energy end consequently an extremely low probability to occur at any given practical temperature. In order to be able to approximate the integral of Eq. (6.5) it is necessary to find a sampling strategy that correctly captures the leading terms of the integral. Starting from a given initial state of the system we can discretize its evolution by performing at discrete intervals an update of its state. The corresponding sequence of states takes the name of Markov chain. If the system is in a state n, the probability of transitioning to another accessible state m will take the name of transition probability, w_{nm}. The rate of change of total probability p_n that at the time t the system is found in the state n is given by [173],

$$\frac{\mathrm{d}p_n}{\mathrm{d}t} = \sum_{m\neq n}\left(w_{mn}p_m - w_{nm}p_n\right). \tag{6.6}$$

At equilibrium, this derivative is zero and a detailed balance condition is reached,

$$w_{nm}p_n = w_{mn}p_m. \tag{6.7}$$

The particular choice of transition strategy of the Markov chain will define the transition probability rate. Any rate that satisfies Eq. (6.7) is a valid choice and, depending on the problem, will impact the convergence of the integral Eq. (6.5). Metropolis *et al.* [174] suggested the following choice

$$w_{nm} = \begin{cases} \exp\left(-\frac{\Delta E}{k_B T}\right) & \text{if } \Delta E > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{6.8}$$

Here $\Delta E = E_n - E_m$ represents the difference in energy between the two states. If we keep the atomic positions fixed and only update the spin degree of freedom, the Metropolis Monte Carlo algorithm will take the following form:

Select an initial configuration;
for $i \leftarrow 1$ **to** N_{steps} **do**
  for $I \leftarrow 1$ **to** N **do**
    Select an atom I;
    Sample uniformly a unit vector on the sphere;
    Update the magnetic moment of the atom I;
    Calculate ΔE *using Eq. (6.2)*;
    if $\Delta E \leq 0$ **then**
      Accept the update in the spin of the atom I;
    else
      Generate a uniform random number $r \in [0, 1]$;
      if $r < \exp\left(-\frac{\Delta E}{k_B T}\right)$ **then**
        Accept the update in the spin of the atom I;
      else
        Reject the update;
      end
    end
  end
end

This algorithm is repeated for a number of steps N_{steps}. We can use this sampling approach to compute the average magnetisation of the system as a function of the temperature, also known as the magnetisation curve. We report in Fig. 6.2 the magnetisation curves obtained using Metropolis Monte Carlo for a Body-Centered Cubic (bcc) iron crystal with periodic boundary conditions using supercells of different sizes. The energy of a particular spin configuration is given by a Heisenberg Hamiltonian with the following exchange used in [172]

$$\mathcal{J}_{IJ}(\vec{R}_{IJ}) = \mathcal{J}_0(1 - R_{IJ}/R_c)^3\,\Theta(R_c - R_{IJ}), \tag{6.9}$$

where $\Theta(R_c - R_{IJ})$ is the Heaviside step function. As in Ref. [172] we use a cutoff of $R_c = 3.75$ Å, which falls between the second and third nearest-neighbor and $J_0 = 904.90$ meV. From Fig. 6.2 we observe that this Hamiltonian is able to describe a ferromagnetic phase

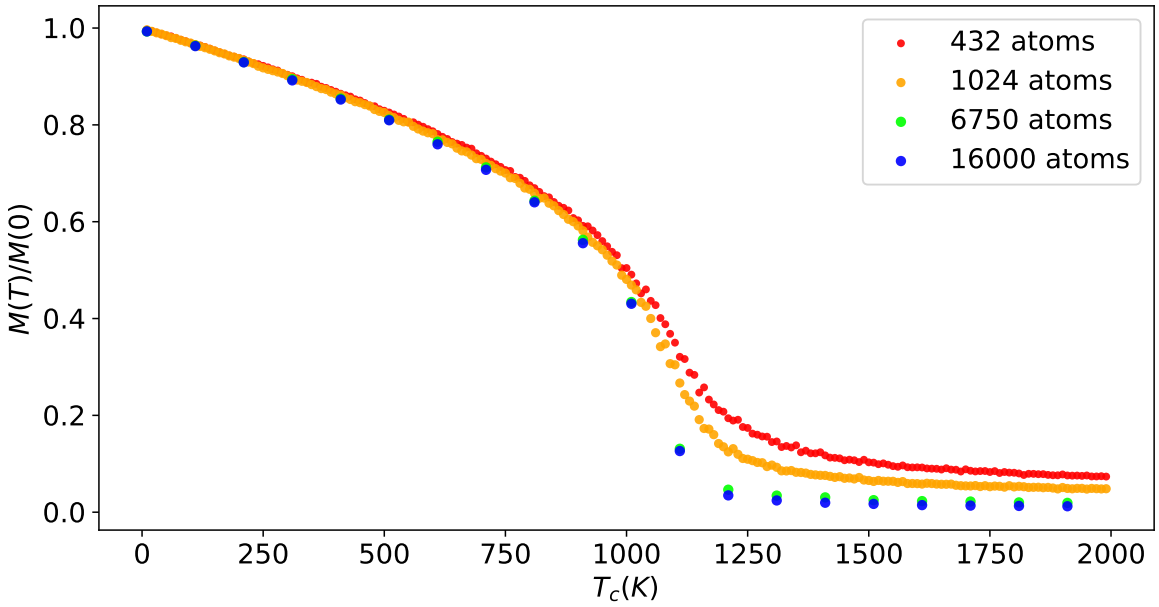


Figure 6.2: Magnetisation curve obtained with Metropolis Monte Carlo for bcc iron supercells of different sizes using periodic boundary conditions. The energy of the system is given by the Heisenberg Hamiltonian parametrisation reported in Ref. [172]. See text for more details.

transition. The steepness of the magnetisation drop in correspondence with the Curie temperature increases with the size of the unit cell.

6.2 Machine-learning force fields with spin

The machine-learning force-field models introduced so far neglect entirely the presence of any magnetic moment associated with the atoms of the systems they describe. As a result, these models will fail to capture the magnetic properties of the system, if they are not strongly connected to its atomic configuration. For example, the descriptors of a ferromagnetic and an antiferromagnetic ground state would be the same unless the different phases are also associated with different structures. For the application of MLFF to the modelling of magnetic materials, it is necessary to find ways to generalise these models to systems that include magnetic atoms.

One initial attempt to address this challenge consisted in generalising the symmetry functions to include information associated with the spin direction for the collinear spin-polarised case [175]. An alternative strategy merges conventional MLFF models with a semi-classical spin Hamiltonian, creating a hybrid model that predicts energies considering both the atomic positions and spin configurations.

Nikolov *et al.* [176] recently adopted this method, augmenting a SNAP model with a classical Heisenberg Hamiltonian to calculate thermodynamic properties, including the Curie temperature of bcc iron. The fitting of the SNAP and the Heisenberg Hamiltonian is iteratively refined through an optimisation strategy. The training set used in

this work includes non-collinear spin-polarised calculations for different crystal phases and for liquid iron. The Heisenberg Hamiltonian part was fitted on spin-spiral calculations, performed at different degrees of lattice compression. For a discussion on DFT spin-spirals calculations, we invite the reader to see Appendix D. The Curie temperature of the system was then calculated using spin-lattice dynamics and determined to be ~ 716 K.

Chapman *et al.* [177] expanded on this strategy by including a neural network term, using symmetry functions, to model the leading $E^{(n)}(\{\vec{R}\})$ coefficients of Eq. (6.1). Once again the training of the model is performed in multiple steps fitting one component at a time while keeping fixed the rest, similar to Nikolov's work. In this case, the predicted Curie temperature for bcc iron was ~ 900 K, whereas the experimental one is 1043 K.

The main challenge associated with developing spin-lattice potentials is related to the fact that the total energy of the system is now a parametric function of $6N$ degrees of freedom, twice as many as the non-magnetic case. This increase in complexity poses difficulties in adequately sampling the phase space when building a training set. Furthermore, the sampling of excited spin configurations is made even more challenging by the fact that at convergence DFT can only predict the ground-state density of the system, which corresponds to the relaxed spin configuration with minimum energy. Constrained DFT or spin spiral calculations are then required to capture excited states of the magnetic system.

Chapman's model neural network architecture requires a substantial amount of data for training. This presents an even more significant drawback for this problem given the elevated computational cost of the spin-polarised calculations required. Nikolov's approach is more data efficient, however, it decouples the SNAP component with the Heisenberg part of the energy predictor, requiring a multi-step fitting which is optimised using the DAKOTA software package [178].

In the section that follows, we present the spin power spectrum, a novel set of invariant descriptors of the local chemical environment that include the atomic magnetisation of the atoms in the system. We will then use these descriptors to train a linear model on bcc iron spin-spiral DFT calculations and estimate its Curie temperature.

6.2.1 Spin power spectrum

We can generalise the neighbour density distribution relative to the I-th atom presented in Eq. (2.19) to include the information associated to the atomic magnetic moment, $\vec{S}_I$, of each atom of the system. By doing so, we can write the following vectorial neighbour density distribution

$$\boldsymbol{\rho}_I(\vec{R}) = \sum_J w_J \, f_c(R_{IJ}) \, \delta(\vec{R} - \vec{R}_J)\vec{S}_J, \tag{6.10}$$

We can express the spin components in terms of spherical versors [179],

$$\vec{S}_J = \sum_{q=0,\pm1} S_{J,q}\,\hat{\mathbf{e}}_q \quad \text{with} \quad \begin{cases} S_{J,\pm1} = \mp\frac{1}{\sqrt{2}}(S_{J,x} \mp iS_{J,y}), \\ S_{J,\pm0} = S_{J,z}, \end{cases} \tag{6.11}$$

with,

$$\hat{\mathbf{e}}_{\pm1} = \mp\frac{1}{\sqrt{2}}(\hat{\mathbf{e}}_x \pm \hat{\mathbf{e}}_y) \quad , \quad \hat{\mathbf{e}}_0 = \hat{\mathbf{e}}_z. \tag{6.12}$$

With this notation and by expanding the spatial part in terms of spherical harmonics, we can rewrite the vectorial neighbour density distribution as,

$$\boldsymbol{\rho}_I(\vec{R}) = \sum_{n=0}^{+\infty}\sum_{l=0}^{n}\sum_{m=-l}^{l}\sum_{q=0,\pm1} c_{nlmq}\,\mathcal{R}_{nl}(R)Y_l^m(\hat{\mathbf{e}}_{\vec{R}})\,\hat{\mathbf{e}}_q, \tag{6.13}$$

with,

$$c_{nlmq} = \sum_J w_I\, f_c(R_{IJ})\,\mathcal{R}_{nl}(R_J)S_{a,q}\,Y_l^{m*}(\hat{\mathbf{e}}_{\vec{R}_J}). \tag{6.14}$$

The functions $\mathcal{R}_{nl}$ constitute a basis for the radial part, as the spherical harmonics are a complete basis only for functions on the unit sphere. The particular choice of the radial functions becomes a hyperparameter of the problem. In this work, we use radial functions based on spherical-Bessel descriptors as introduced in Ref. [180].

Using the Dirac notation we can rewrite this expansion as:

$$|\boldsymbol{\rho}_I\rangle = \sum_{nlmq} c_{nlmq}\,|nlm1q\rangle \ , \tag{6.15}$$

where:

$$\langle\mathbf{R}|nlm\rangle = \mathcal{R}_{nl}(R)Y_l^m(\hat{\mathbf{e}}_{\vec{R}}), \quad \text{and} \quad |1\,q\rangle \equiv \hat{\mathbf{e}}_q\,. \tag{6.16}$$

The angular momentum of $\mathbf{1}$ associated with the spin part of the distribution reflects the vectorial nature of the magnetic problem. We can then change the basis to the one of the combined angular momentum $\mathbf{L}+\mathbf{1}=\boldsymbol{\mathcal{J}}$,

$$|l-1| \le \mathcal{J} \le l+1 \quad \text{and} \quad -\mathcal{J} \le M \le \mathcal{J}\,. \tag{6.17}$$

This change of basis is mediated by the Clebsh-Gordan coefficients [181]:

$$|nlm1q\rangle = \sum_{\mathcal{J}M} C^{\mathcal{J}M}_{lm1\,q}\,|nl1\mathcal{J}M\rangle \ . \tag{6.18}$$

By substituting this expansion back into Eq. (6.15) we obtain,

$$|\boldsymbol{\rho}_I\rangle = \sum_{nl\mathcal{J}M}\left(\sum_{mq} C^{\mathcal{J}M}_{lm1\,q}c_{nlmq}\right)|nl1\mathcal{J}M\rangle \ . \tag{6.19}$$

Similarly to Eq. (2.23) we can combine the expansion coefficients to compute the power spectrum,

$$p_{nl\mathcal{J}} = \sum_{M} \left| \sum_{mq} C^{\mathcal{J}M}_{lm1\,q} c_{nlmq} \right|^2 . \tag{6.20}$$

We call these descriptors Spinpowespectrum. The sum over n of Eq. (6.19) contains an infinite number of terms. In practice, this expansion is truncated up to a certain n_{max} and a finite number of features is considered. Combining Eq. (6.14) with Eq. (6.20) we have the following explicit expression of these coefficients:

$$p_{nl\mathcal{J}} = \sum_{M} \left| \sum_{J} w_I f_c(R_{IJ})\, \mathcal{R}_{nl}(R_J) \sum_{mq} C^{\mathcal{J}M}_{lm1\,q}\, S_{a,q}\, Y_l^{m*}(\hat{\mathbf{e}}_{\vec{R}_J}) \right|^2 . \tag{6.21}$$

In this case, the spin power spectrum is rotationally invariant with respect to any simultaneous rotation of the system and its atomic magnetic moments. We provide proof of the rotational invariance of the spin power spectrum in Appendix C.

6.2.2 Predicting magnetic properties with MLFFs

We can use the spin power spectrum to approximate the total potential energy of the system described by Eq. (6.1). The simplest model architecture that we can pair with these descriptors is a linear model. In this work, we decided to use a Ridge regression (see Eq. (2.39)). This is a linear model that includes a regularisation term in the training loss to prevent overfitting. The resulting model has the following expression,

$$E_{Tot}(\{\vec{R}\},\{\vec{S}\}) = \sum_{k=1}^{K} \left(w_0^{(k)} N_k + \sum_{nl\mathcal{J}} w^{(k)}_{nl\mathcal{J}} \sum_{I|k_I=k} p_{I,nl\mathcal{J}} \right) . \tag{6.22}$$

In Fig. (6.3) we report the magnetisation curve obtained performing Metropolis Monte Carlo over a $3 \times 3 \times 3$ supercell of 54 atoms of bcc iron. The energies were predicted by a linear spin power spectrum model, which was fitted on configurations whose energy was computed using a Heisenberg Hamiltonian with the exchange parametrisation reported in Eq. (6.9) [172]. The model is able to closely reproduce the phase transition after being trained over a sample of 2000 configurations. This shows that, in principle, for systems whose magnetic behaviour can be modelled by a semi-classical Heisenberg Hamiltonian, a model that uses the spin power spectrum as input features is able to describe its phase transition from ferromagnetic to paramagnetic.

In practice, extending this approach to a model trained on first principles spin-polarised calculations presents several challenges. DFT is fundamentally a ground-state theory for the electronic component of the system that determines its magnetic behaviour. As such, at convergence, spin-polarized DFT calculations will find the ground-state electronic density of the system. From this density it is possible to com-

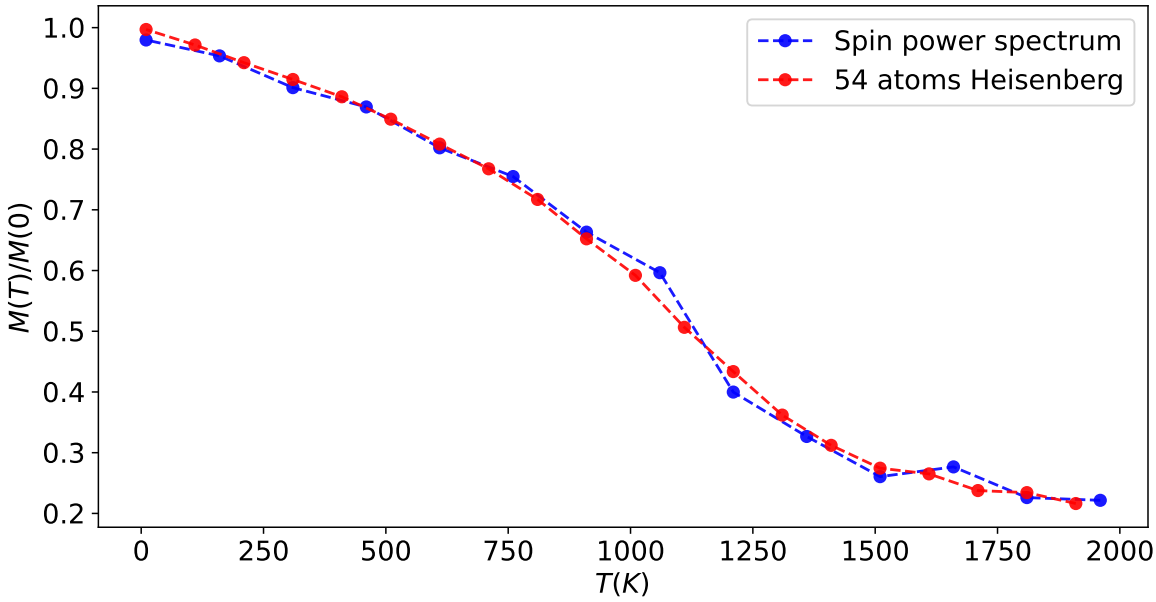


Figure 6.3: Magnetisation curve obtained with Metropolis Monte Carlo for bcc iron (54 atoms supercell). The energy of the system is given by a linear model using the spin power spectrum trained over configurations whose energy was given by a Heisenberg Hamiltonian. The magnetisation obtained with the spin power spectrum model (blue) is compared with the magnetisation obtained with the Heisenberg Hamiltonian used to build the training set (red).

pute the associated atomic magnetic moments, however, these moments will correspond to the lowest energy magnetic configuration of the system. This "relaxation" of the magnetic configuration heavily limits our ability to sample the phase space of a magnetic system as it is required for the training of the model.

One possible strategy to overcome this issue consists of using constrained DFT. By imposing a constraint on the atomic spins' direction the ground-state density associated with that magnetic configuration can be computed. However, constrained DFT is more computationally expensive and hard to converge. Furthermore, to correctly sample different excited spin-configurations large supercells are required. Building a training set of hundreds of configurations of spin-polarised, non-collinear, constrained DFT calculations on large supercells is not practically feasible. This approach also cannot be easily performed on various materials within the framework of high-throughput inverse material.

For all these reasons, we explore here the possibility of predicting the magnetic properties of a system with a machine-learning model trained over a minimal set of spin-polarised calculations. We chose to use spin-spirals spin-polarised calculations for the creation of the dataset. This technique allows us to sample the fundamental magnetic excitations of the system while performing the calculation only on the unit cell. This is made possible by exploiting the periodicity of the excitation as detailed in Appendix D.

We investigate here the possibility of using the spin power spectrum to model the ferromagnetic phase transition of bcc iron using only DFT spin-polarised calculations. In order to sample the excited states of the spin configuration, we employ spin-spirals, as implemented in VASP [144, 145, 146]. For the creation of the training set, we perform 200 spin-spiral calculations for a bcc iron unit cell (2 atoms per unit cell) using only spin-spirals with a period commensurate with the lattice of the system, namely, we consider all possible integer periods from 1 unit cell to 200, and propagating along the z-axis direction. We use a cutoff of 600 eV for the plane-wave basis and PBE functional, together with a $5 \times 5 \times 5$ uniform k-point mesh.

The spin power spectrum is not designed to work directly on spin-spiral configurations, its implementation imposes periodic boundary conditions on the supercell given as input. During training, for each spin spiral configuration, we build a supercell extending the unit cell along the z-direction for a number of times equal to the period of the commensurate spin spiral. The spin components are then rotated around the z-axis to reproduce the entire spiral. Periodic boundary conditions are then applied to this supercell and used as a training example for the linear spin power spectrum model.

Due to this setup, we only include spin spirals with commensurate periods in our training set. This sampling strategy leads to the introduction of a strong bias towards small q-vectors. As a result, testing the performance of the trained models with respect to a conventional train-test split does not provide adequate insight into their actual modelling capabilities. We decided instead, for this preliminary study, to train the model on all 200 spin-spirals calculations and evaluate its performance with respect to the predicted magnetisation curve. We calculate the magnetisation curve for a 54 atoms supercell of bcc iron using Metropolis Monte Carlo importance sampling to approximate the ensemble average at different temperatures. We report in Fig. 6.4 the magnetisation curve obtained starting for a random spin configuration at 2000 K and reducing the temperature of 10 K every $N_{steps} = 200$ iterations of the Metropolis Monte Carlo algorithm. The curve indicates the presence of a ferromagnetic phase transition.

We use spin power spectrum descriptors up to $n_{max} = 3$ corresponding to 23 features. For higher maximum degrees, the trained models do not replicate a ferromagnetic transition. We attribute this fact to the overfitting of the model due to the limited size of the training set.

In order to estimate the Curie temperature of the system associated with the predicted magnetization curve, we fit the magnetisation curve with the Curie-Bloch law which describes the critical behaviour of a fixed lattice system described by a Heisenberg Hamiltonian [182]

$$M(T)/M(0) = \left(1 - \frac{T}{T_C}\right)^{\beta}, \tag{6.23}$$

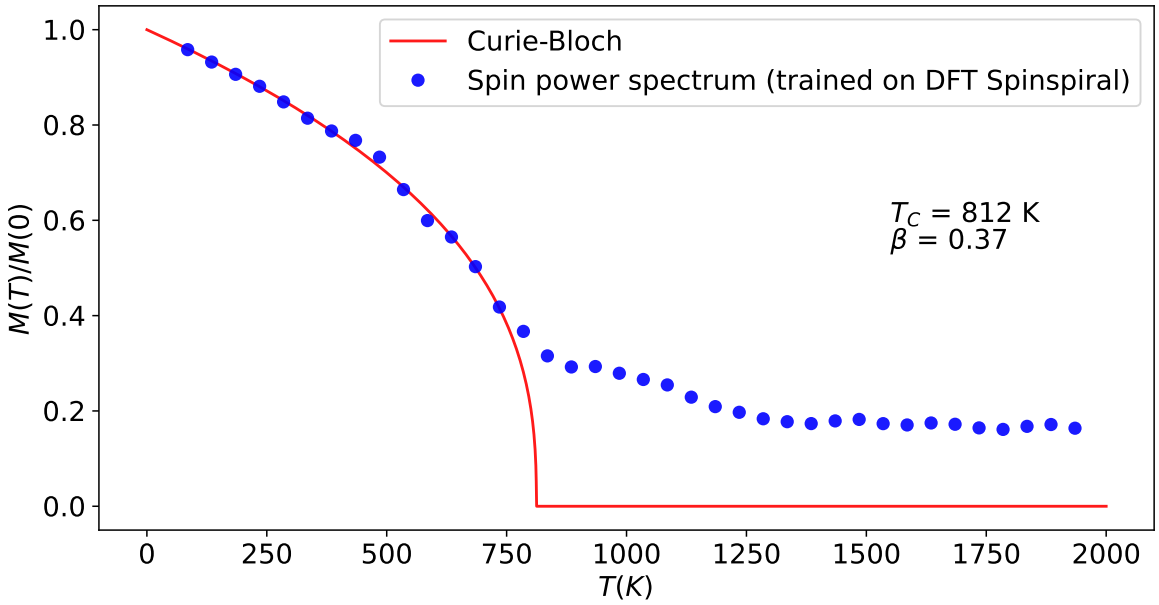


Figure 6.4: Magnetisation curve obtained with Metropolis Monte Carlo for bcc iron (54 atoms supercell). The energy of the system is given by a linear model using the spin power spectrum trained over configurations whose energy was given DFT spin-spiral calculations. The magnetisation obtained with the spin power spectrum model (blue) is fitted with a Curie-Bloch function to determine the Curie temperature of the system from first principles (red).

where the critical exponent β should be equal to $\sim 0.\bar{3}$ for a pure Heisenberg Hamiltonian. The fitting of the magnetisation curve produced by our spin power spectrum model returns a $\beta = 0.37$ not too far from the ideal value. This validates the fact that iron is correctly modelled by a simple Heisenberg Hamiltonian. Due to the limited size of the simulation cell, the curve does not appear particularly steep. As shown in Fig. 6.2 we would expect a sharper magnetisation drop for larger simulation cells. The estimation of the Curie temperature is 812 K significantly lower than the experimental value of 1043 K. This discrepancy might be attributed to an incomplete sampling of the phase space resulting from the limited scope of our training set, which considers only commensurate spin-spirals.

A more thorough study, which also involves different materials is still needed to reinforce the feasibility of the modelling strategy presented here. This will be the focus of our future research. We should also note that the Monte Carlo calculations reported so far are performed with fixed lattice and longitudinal spin components. The Metropolis Monte Carlo algorithm previously discussed can be extended to include atomic displacements and variations of the longitudinal part of the spin. The spin power spectrum is able to discern simultaneously both these variations. In general, the inclusion of other excitation modes will lower the Curie temperature of the system, as new pathways become available to include low symmetry configurations.

6.3 Summary

When implementing an inverse-design strategy for magnetic materials, the ability to accurately model the magnetic properties of the systems is essential. Conventional MLFFs do not account for the magnetic degrees of freedom of the system and as such are unsuitable for this task. In this chapter, we introduced the spin power spectrum, a set of roto-translational invariant descriptors of the chemical environment that include the magnetic moment of the atoms in the system. We have demonstrated the invariant properties of this descriptor and its suitability to create machine-learning models able to predict the magnetic properties.

As a case study to showcase the capabilities of this approach, we have trained a linear model that uses the spin power spectrum as features to predict the energies of DFT spin spiral configurations of bcc iron. We then used the trained model to perform a Monte Carlo simulation to generate the magnetisation curve as a function of the temperature. The resulting curve correctly displays a ferromagnetic transition. We then estimate the Curie temperature to be 812 K by fitting the magnetisation curve.

This promising approach would offer several advantages compared to existing work in the literature. The constructed model integrates simultaneously the configurational and magnetic degrees of freedom. The fit is performed only once across the entire dataset as opposed to the strategies adopted in [176, 177]. Moreover, we utilise a relatively small dataset of DFT spin-spiral calculations, which do not impose a significant computational burden.

All these aspects make our work particularly well-suited for property-based screening within an inverse-design workflow. However, it is important to note that the research presented here is still in its preliminary stages, and further study is required.

Chapter 7

Software

This chapter presents an overview of the software implemented by the author of this thesis. The Force Field Machine Python library has been developed in collaboration with Urvesh Patil.

Given the interdisciplinary nature of this work, we found that the software needed to implement the various methods developed was either fragmented or unavailable. The construction of a working MLFF requires alone codes from different sources. For instance, the training set is generated using DFT, the construction of the model requires the implementation of the descriptors and the machine-learning architecture is generally available on a separate code base. Furthermore, the trained MLFF then needs to be integrated with software for molecular dynamics, relaxation, phonon calculations, and more.

To address this complexity, we have developed a Python library called "Force Fields Machine." This library aims to integrate the various elements needed for the creation and deployment of MLFFs. It includes most of the methods we have developed and described in this thesis.

In addition to this package, we also created a code base in support of the BERT-PSIE workflow for the automatic extraction of data from the scientific literature described in Chapter 3. Given the reliance on the fine-tuning of language models, this extraction workflow offers the advantage of requiring little to no prior knowledge of NLP from the final user. This is in stark contrast with previously established methods such as ChemDataExtractor [79], which relies on the definition of grammar rules via regular expressions. In order to leverage this advantage, the software implementation of BERT-PSIE should reflect this accessibility. We developed a Graphic User Interface (GUI) that facilitates the annotation of the text required for the fine-tuning, together with a series of notebooks that can be run on Google Colab for the fine-tuning and deployment of the language models composing BERT-PSIE. By using these tools, any non-expert user can then easily create a new extraction workflow for the properties of interest.

7.1 Force Fields Machine Python library

The Force Fields Machine (FFM) library is a Python library designed for the creation and use of MLFF. The primary objectives of FFM are to streamline and make more accessible the development and training of MLFF, while providing a certain degree of flexibility. This flexibility is achieved by giving access to different descriptors of the local chemical environments and to different model architectures. Importantly, each descriptor is compatible with every architecture, allowing a wide range of possible combinations. Additionally, we included several quality-of-life features that simplify common operations performed on datasets of atomic configurations. This section will explore the principal Python classes implemented together with their functionalities. We will also provide some examples of scripts for the creation of different MLFF models.

Structure

The *Structure* object represents the datatype associated with a single atomic structure. This class is similar in scope to the *Structure* and *Atoms* classes implemented in PyMatGen [183] and ASE [184], respectively. Along with attributes containing information regarding the atoms *positions*, *cell* and atoms *types* we also introduce attributes relative to the *descriptors* and *neighbourlist*. Associating the *neighbourlist* with the structure allows its reuse in various situations. For example, when changing the hyperparameters of the local descriptors, the *neighbourlist* is updated only if the cutoff radius is increased, thereby improving the performance during the hyperparameters optimisation. To compute the neighbour list, we utilise the KDTree algorithm implementation from the SciPy library [185], which we have adapted to account for periodic boundary conditions. This approach is significantly more computationally efficient than the current ASE implementation.

The methods *from_ase*, *to_ase*, *from_pymatgen* and *to_pymatgen* are implemented to provide a bridge between the FMM *Structure* class with the ASE *Atoms* and PyMatGen *Structure* classes.

Dataset

The *Dataset* object represents a collection of *Structure* objects. The expected use of this class is to store the datasets of crystal structures along with their associated DFT energies, forces and stresses. This data is required for the training, validation and testing of MLFFs. Some design choices have been taken to make common operations performed on datasets easily available to the user. Accessing a *Dataset* object as an array will return a *Dataset* containing only the *Structures* corresponding to the selected indexes. Summing two *Dataset* objects will return the concatenation of the two. Masking operations are supported as with NumPy arrays [186], allowing to select *Structures* satisfying specific conditions. This functionality is achieved through an

implementation of the *__getitem__* method able to handle slices. This also enables the use of common utility functions defined on arrays such as the *train_test_split* function from Sci-Kit learn [187]. Each *Dataset* is equipped with a *properties* attribute, which is expected to store a dictionary containing the targets for the supervised training. To help analyse the dataset content we implemented the *pair_distribution* method to compute the pair distribution function over all the configurations contained in the *Dataset*.

Models

Models objects are all inherited from the *AbstractModel* class. The available model architectures within the FFM library include linear models, Gaussian process regression and feed-forward neural networks. As described in Chapter 2, the Gaussian process regression can be interpreted as a kernel ridge regression, where the features used are based on kernel similarity. Taking advantage of this, we only need to implement two architecture classes namely the *LinearModel* and the *NNModel*. Every model class implements a *fit* method that takes care of the supervised training. Any property saved in the *properties* dictionary of the *structure* objects contained in the *dataset* can be used as a training target. If the target is the DFT energy, as discussed in Chapter 2, it will be likely to also have access to the atomic forces and stress. The arguments *fit_forces* and *fit_stress* allow one to toggle the inclusion of the corresponding terms into the loss function (see Eq. (2.28)). In order to save and load the weights of trained models, we have implemented the *dump* and *load* methods for writing to and reading from a JSON file, respectively.

Descriptors

Descriptors objects are inherited from the *AbstractDescriptor* class and are required to include the implementation of a *compute_descriptors* method, which takes as input a *Structure* and returns its associated descriptors of the local chemical environments together with their gradients. The hyperparameters characterising the descriptor are passed during initialisation. The *compute_descriptors* method is called by the models when needed during both training and inference. Among the descriptors currently implemented in FFM there are the bispectrum components [49], the spin power spectrum, and the JL descriptors. The descriptors objects are not directly passed to the model object, they must first be stored in a *Features* object.

Features

In order to offer a high degree of flexibility, in constructing MLFF models we have adopted a slightly unconventional approach in designing the features class. Each model architecture class implemented in FFM expects as input a *Features* object. The *Features* object is accessed as a list of lists, where each one of its rows is called “level”.

The method *add_descriptor* allows one to add to the current level a descriptor object. Multiple independent descriptors can be added in this way allowing, for example, to build a model that uses as input features two sets of bispectrum components with different cut-off radii. It is also possible to combine in this way any of the descriptors implemented in FFM. The input features passed to the model in such cases consist of the concatenation of the different descriptors added to that level. The method *add_level* adds a level to the *Features* object and moves the current level to the newly created one. Once again, multiple descriptors can be added to the new level by calling the *add_descriptor* method. At the creation of the model architecture objects a different model is initialised for each one of the levels in the *Features*. Each one of these models expects as input the descriptors linked to the *Features* objects for that level. When performing a prediction, the output is then given by the sum of the models for all the levels

$$E_I = \sum_{i=1}^{N_{levels}} \mathcal{F}_i \left(\text{Concat} \left[\mathcal{B}_I^{(i,1)}, \dots \mathcal{B}_I^{(i,N_d^{(i)})} \right] \right), \tag{7.1}$$

where $N_d^{(i)}$ is the number of sets of different descriptors added at the level i of the *Features* object. The training is performed one level at the time by subtracting the output of the models given from previous levels each time.

The design flexibility offered by this approach is substantial, granting easy access to a vast design space. The level structure implemented in the *Features* object allows for a "perturbative" fitting. By storing descriptors of increasing body order at increasing *Features* levels i, the fitting will be firstly performed using only lower body order descriptors and then refined by the predictions corresponding to the models at higher levels, which use higher body order descriptors. As previously mentioned the *add_descriptor* methods allow for the concatenation of different sets of descriptors. For example, the code required to build a linear model that takes as input features the concatenation of two sets of bispectrum components with different cutoff radii is:

```
from forcefieldsmachine.descriptors import Bispectrum
from forcefieldsmachine.models import LinearModel

features = Features()
features.add_feature(Bispectrum(rcut=2.8, twojmax=8))
features.add_feature(Bispectrum(rcut=4.2, twojmax=8))

model = LinearModel(features=features, alpha=1e-5)
```

In the rest of this section, we will provide a more detailed discussion of some of the MLFF models that can be built with this library.

7.1.1 SNAP

A SNAP model consists of a linear model that uses as input features the bispectrum components [45]. In this library, we rely on the extremely efficient LAMMPS implementation of the bispectrum components [66] for their computation. Our *Bispectrum* class calls LAMMPS via its Python wrapper. The argument names of this class reflect the ones required by LAMMPS to calculate the descriptors. This approach makes the implementation more user-friendly, as many of the intricacies involved in using LAMMPS are handled behind the scenes. The user can easily compute the bispectrum components over the configurations contained in a *.xyz* file with a few lines of code, without having to know how to create a LAMMPS input file. The following is the code required to train a SNAP model for benzene using the FFM library:

```
import numpy as np
import matplotlib.pyplot as plt

from forcefieldsmachine import Dataset, Features
from forcefieldsmachine.descriptors import Bispectrum
from forcefieldsmachine.models import LinearModel

# Load the dataset
train = Dataset.fromExtXYZ("./datasets/benzene_train.xyz")
test = Dataset.fromExtXYZ("./datasets/benzene_test.xyz")

# Create the model
features = Features()
features.add_feature(
    Bispectrum(rcut=3.8, twojmax=8, weights={"C": 1, "H": 0.6})
)

snap = LinearModel(features=features,
                   alpha=1e-5,
                   fit_forces=True,
                   fit_stress=True,
                   coupling_forces=1/(3*12),
                   coupling_stress=1/6,
)

# Fit the data
snap.fit(train)

# Plot the predictions
snap.plot_predictions(train, filename="train_snap", per_atom=
   True)
```

```
snap.plot_predictions(test, filename="test_snap", per_atom=True
    )

# save the model
snap.dump(filename="snap_benzene")
```

The train and test set configurations stored in a *.xyz* files are firstly loaded. Subsequently, the model descriptors and architecture are defined and then the training is performed by calling the *fit* function. We implemented some utility functions such as the *plot_predictions* method in the *AbstractModel* class, which creates a parity plot that tests the predictions of the model against the expected values contained in the *Dataset*. Finally, the trained model can be saved in a JSON file for future use.

7.1.2 GAP and SOAP

In order to reproduce architectures based on Gaussian-process regression we can reuse the *LinearModel* implementation by leveraging the kernel Ridge regression interpretation of Eq. (2.49). For this purpose, we have defined a *GPR* descriptor class. When initialised, this class expects as arguments a *kernel* similarity function, a *Descriptor* object and a set of *reference* environments. This design choice enables the use of all kernel functions implemented in Sci-Kit learn [187]. For example, the GAP model is reproduced by combining a Radial Basis Function kernel (*RBF*) with the bispectrum components [49], while the SOAP model can be expressed in terms of the *DotProduct* kernel and the normalised power spectrum [34]. Recent iterations of the Gap potential include two-body and three-body terms, which are concatenated with the Gap kernel (see Eq. (2.67)) to improve the modelling of short-range atomic interactions [27]. It is possible to reproduce this implementation using the *add_descriptor* method:

```
from sklearn.gaussian_process.kernels import RBF
from forcefieldsmachine.descriptors import Bispectrum, GPR,
    TwoBody,

K_twob = GPR(
    kernel=RBF(0.2),
    descriptor=TwoBody(rcut=3.7),
    reference=TwoBody.uniform_grid(
        0, 4.1, step=0.2, types=train["all_types"], rcut=4.1
    ),
)

K_gap = GPR(
    kernel=RBF(0.2),
    descriptor=Bispectrum(rcut=3.7, twojmax=8),
    reference=train
```

```
    ),
)

features = Features()
features.add_feature(K_twob)
features.add_feature(K_gap)

Gap = LinearModel(features=features, alpha=1e-5)
```

7.1.3 Jacobi-Legendre descriptors

The FFM library also includes a Cython [188] implementation of the descriptors based on the Jacobi-Legendre polynomials, required for the construction of a JLP model (see Chapter 5). The JLP is based on a cluster expansion of the energy and we designed the descriptors implementation so that each *JL* object only considers a specific body order. The full n-body expansion is then created by concatenating *JL* object relative to different body orders using the *add_features* method of the *Features* class. The following is an example of the code required for the initialisation of a JLP potential up to the 4-body term for carbon:

```
from forcefieldsmachine.descriptors import JL
from forcefieldsmachine.models import LinearModel

rc = 3.7

features = Features()
# 2-body
features.add_feature(JL(nmax=10,
                        rcut=rc,
                        alpha=1,
                        beta=1,
                        gamma=1.0,
                        nbody=2,
                        type_groups=["C,C"])
)
# 3-body
features.add_feature(JL(nmax=4,
                        lmax=3,
                        rcut=rc,
                        alpha=1,
                        beta=1,
                        gamma=1.0,
                        nbody=3,
```

```
                          type_groups=["C,C,C"])
)
# 4-body
features.add_feature(JL(nmax=3,
                          lmax=3,
                          rcut=rc,
                          alpha=1,
                          beta=1,
                          gamma=1.0,
                          nbody=4,
                          type_groups=["C,C,C,C"])
)


JLP = LinearModel(features=features, alpha=1e-5)
```

The *type_groups* argument expects a list of strings containing as many chemical symbols as the body order specified by *nbody* argument. This argument is intended to control the number of descriptors computed particularly for multi-species systems. Descriptors will only be computed for the sequences specified.

7.1.4 Spin power spectrum

The *SpinPowerspectrum* class features a Cython implementation of the spin power spectrum, as introduced in Chapter 6. To properly compute these descriptors, it is expected that the atomic magnetic moment of each *Structure* is stored in the *properties* attribute under the key *Spins*. A linear model that uses the *spin power spectrum* as input features can be easily constructed with the following code:

```
from forcefieldsmachine.descriptors.spinpowerspectrum import
   SpinPowerspectrum
from forcefieldsmachine.models import LinearModel

features = Features()
features.add_feature(SpinPowerspectrum(rcut=rcut, nmax=nmax))

spinsnap = LinearModel(features=features)
```

7.1.5 Future development

Trained models created using the FFM library can then be used to perform relaxation or molecular dynamics using the methods implemented in *ASE*. This is made possible by the implementation of an *ASE Calculator* compatible with the models defined within our library. Phonon dispersion calculations via finite difference can be carried out with

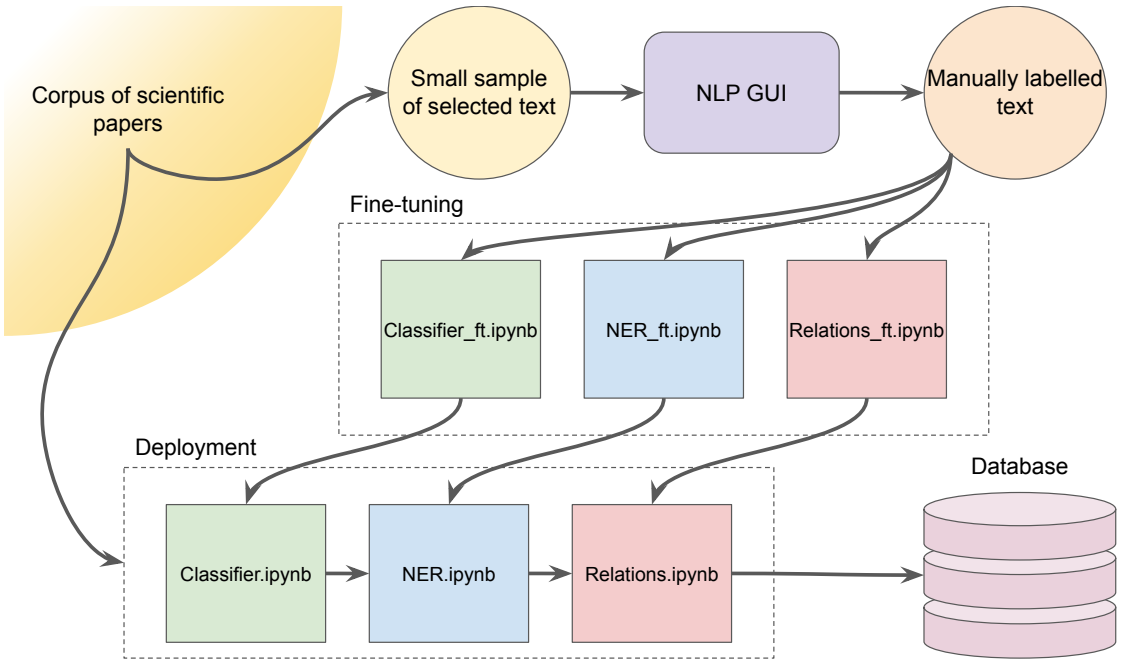


Figure 7.1: Diagram showing the software involved in the construction and use of a BERT-PSIE workflow. We wrote a GUI to streamline the labelling required for the fine-tuning of the models. The finetuning and deployment of the BERT models can be performed via Jupyter notebooks designed to run on Google Colab.

the phonon3py library [167, 168]. Virtually all packages that interface with *ASE* and *PyMatGen* can be easily made compatible with the FFM. It is also possible to create and train compositional models using an implementation of the One-Hot encoding and properties-based descriptors as compositional input features, as discussed in Chapter 2. This implementation is made compatible with most of the regression and classification models available on Sci-Kit learn [187] such as the *RandomForestRegressor*.

The FFM library is still under development. Further work is required before the release of the first stable version. So far this library has been used internally in the group providing a significant contribution to the creation of MLFF used in [5, 2, 3] and of the compositional models used in [1]. Another model architecture implemented is the fully connected feed-forward neural network, for which we relied on PyTorch for the implementation [189]. Future efforts will focus on enlarging the possible design space of the model architectures and on trying to include graph and equivariant models. Additionally, scalability could be improved through a multi-threaded approach for descriptor computation and via the integration of more efficient molecular dynamics codes such as LAMMPS [66].

In training SNAP models it is common, for example, to either add or subtract terms to the DFT total energy. For instance, the Ziegler-Biersack-Littmark (ZBL) empirical potential term is sometimes subtracted from the total energy to take care of the close-range repulsion between atoms [45]. Similarly, to account for van der Walls interactions, external corrections to the energy, which depend on the atomic

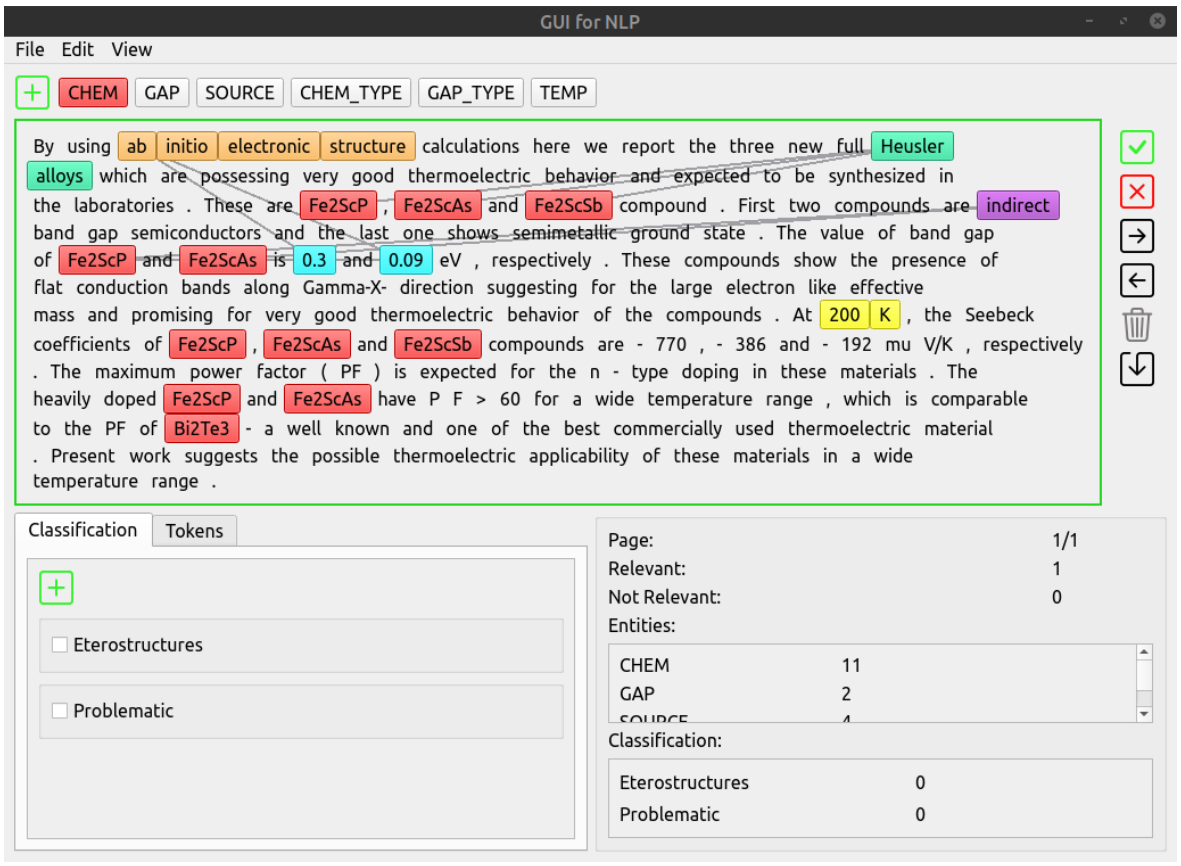


Figure 7.2: Screenshot of the GUI for NLP developed in this work. This software allows to manually label text that can be used for language models fine-tuning. The user can define entities and label the corresponding mentions, define categorical classes and highlight relations between entities.

configuration are added [190]. Future design efforts will focus on organically integrating the possibility of these corrections to the potential.

7.2 BERT-PSIE workflow

In Chapter 3, we presented the BERT-PSIE workflow for automatically extracting data from the scientific literature. One advantage of using language models over techniques reliant on the definition of grammar rules stems from their accessibility and rapidity in being adapted to new extraction targets. The grammar rules required for the extraction are now implicitly defined in the manually labelled text used for the fine-tuning of the models. The creation and use of the workflow do not require any knowledge of NLP from the user as long as the software implementation supports this accessibility. With this goal in mind, we built a graphic user interface (GUI) to facilitate the labelling task. Furthermore, we have implemented the code necessary for fine-tuning and deploying each of the BERT models that are a part of BERT-PSIE. This code is available through Jupyter notebooks [191], which can be run on Google Colab (see Fig. 7.1).

The GUI for NLP is written using Python bindings for QT v5[1] and it streamlines the generation of the labelled data required for all the fine-tuning tasks. Fig. 7.2 shows a screenshot of the GUI taken during the labelling of a scientific abstract. This software allows the user to classify the displayed text for relevancy, define entities and highlight the observed entity mentions in the text. Furthermore, users can annotate relationships between entities, which are visually represented by connecting lines, and define additional categorical classes. The labelling performed using the GUI is saved onto a JSON file. The consistency of this output allows the implementation utility scripts to manipulate it and adapt it to various fine-tuning tasks. We used the GUI output to fine-tune language models for classification, NER and relation classification. Additionally, the JSON output can also be used to fine-tune LLMs following a similar approach as seen in [116, 117].

Prioritising accessibility, as previously mentioned, we made available the code required for the fine-tuning and deployment of each of the BERT-PSIE modules via Jupyter notebooks. Each one of the fine-tuning notebooks expects as input JSON files generated with the GUI, containing labelled text. When executed, these notebooks will save the trained model, which can then be used by the associated deployment scripts. Once all the modules of the BERT-PSIE pipeline have been fine-tuned for a specific extraction task, the deployment scripts can be run. The deployment scripts are designed to be run in sequence and the extracted data will be stored in a CSV file together with the source of the extraction.

7.3 Summary

In this chapter, we explored the software solutions developed for the integration of the various data-driven techniques designed to tackle the challenge of inverse material design. The interdisciplinary nature of this task is reflected in the diverse origin of the software involved. The main driver of our development efforts has been the improvement in user accessibility to the tool required for this task. The FFM Python library aims to collect within a single framework multiple instances of descriptors of the chemical environment and model architectures to facilitate the design and use of MLFF models. Among the methods implemented, in the library feature the JLP, described in Chapter 5, and the spin power spectrum, discussed in Chapter 6. The library also enables the use of the trained models to perform tasks such as relaxation, molecular dynamics, and phonon calculations, relying on external packages. With the same philosophy, we wrote a user-friendly implementation of the BERT-PSIE NLP workflow presented in Chapter 3. Together with the code required for the fine-tuning and deployment of the language models, we created a GUI to streamline the labelling of text used for fine-tuning.

[1] `https://www.qt.io`

Chapter 8

Conclusions and future work

In this work, we introduced a data-driven approach to the inverse design of magnetic materials. Our proposed pipeline largely follows a standard methodology for addressing this challenge. Initially, we narrowed down the chemical space under study to include only the atomic species of interest. We then determine, which compounds are likely to be stable. Finally, further properties of these stable compounds are predicted. Those compounds that meet application-specific requirements are then sent to the lab where their synthesis is attempted.

Conventional approaches present limitations for each stage of this pipeline. For instance, selecting promising candidates based on experimentally measured property values is often impractical. This is primarily due to the lack of structured databases containing experimental properties, making the retrieval of the information needed a demanding task. Moreover, when building an *ab-initio* convex hull to identify stable stoichiometries and associated crystal structures, only a limited number of prototypes can be examined with DFT due to the computational cost of this method. Additionally, computing system properties with solely DFT imposes compromises, such as limiting the size of the simulation cells, making inaccessible the computation of properties that would require large simulation cells.

In order to address these challenges, our work focuses on data-driven solutions that leverage the recent progress in machine learning and the continuous growth of open databases of DFT calculations.

In Chapter 3 we discussed the problem of data extraction from the scientific literature, particularly in the field of material science. The recent advancements in NLP, especially following the introduction of the transformer architecture, have made available new tools to address this task. Our proposed solution, BERT-PSIE, is a pipeline consisting of a series of language models specialised to perform different sub-tasks. BERT-PSIE favourably compares against the state-of-the-art rule-based strategies. Its extraction is driven by the fine-tuning of language models on a small dataset of manually labelled data. The transfer learning capabilities of these models facilitate the adaptation of the workflow to new extraction targets. In order to streamline the data

labelling process we developed a GUI, which we integrated with the rest of our code base as detailed in Chapter 7. This integration provides a user-friendly usage of BERT-PSIE and facilitates the adaptation of the workflow for the extraction of new properties.

While designing BERT-PSIE, we found a lack of benchmarking strategies in the existing literature that would reflect the real-world usage of the data generated by automatic data extraction workflows. Our contribution in this direction consists of the definition of two tests to be performed on the extracted data: the Query Test and the Suitability-to-Machine-Learning Test. The Query Test assesses the quality of the extracted data against a reference database, while the Suitability-to-Machine-Learning Test evaluates the quality of the predictions performed by a machine-learning model trained on the extracted data against a reference dataset. Both these tests require a manually curated reference dataset against which to perform the comparison.

We were able to make use of these tests by focusing our data extraction on properties for which we have available a manually curated dataset of experimental measurements. We then used the insight that these tests provided to guide the design of the workflow and evaluate the impact on the final extraction of any design choice performed. These tests highlight challenges and provide insight into the best way to address them. We also explored the integration of LLMs into our workflow. However, their inclusion resulted in only limited improvements in handling the relationship resolution between entities. This suggests that the major bottleneck in the extraction performance might be due to the limited context provided by working on a sentence level. Overcoming this limitation presents a challenge due to the scaling in memory requirements and computation of the transformer with respect to the input size. Addressing this issue will be the focus of our future efforts.

Moreover, leveraging language models, we focused on extracting data from only the main text of papers, even though scientific data is also presented in tables and figures. Reconstructing the table structure from a pdf file can be challenging, therefore a significant portion of the existing techniques for processing tables are based on optical character recognition. Adapting these techniques to a material science domain would necessitate a domain-specific training to improve the recognition of chemical formulas, which are often misread due to the presence of figures as suffixes. At the same time, extracting data from plots would require image processing techniques paired with NLP to identify the content of the figure from both the text and its caption. A comprehensive pipeline for information extraction from the scientific literature should combine all these techniques to unlock as much of the available data as possible.

In Chapter 4, we confronted the problem of predicting the stability of a given compound. The approach taken to address this task depends on the number of chemical identities involved, as the entropy contribution to the stability becomes more and more important with the increase of the number of species included in the compound. In this work, we focused on systems containing up to three chemical species. In such cases,

the dominant driver of stability is the configurational enthalpy, which is ultimately determined by the atomic structure. Therefore, identifying the most stable phases is subordinated to the exploration of all possible competing configurations. However, the extent of this exploration is bound to be limited by the computational cost associated with the *ab-initio* method used for calculating the formation enthalpy. Given a fixed number of DFT calculations performed to determine the phase diagram of the system, the most accurate convex hull will be obtained when the most stable structures are included in the set of calculations. In this context, we proposed two novel data-driven strategies for the design structures prototypes that are likely to be found to be stable.

One approach introduces a workflow for the generation of atomic structure prototypes for ternary alloys. Within this workflow, all possible ternary decorations of the low-energy associated binary structures are created and screened using MLFF models trained over the binary compounds formation energies that are available on online databases. DFT calculations are then performed on the prototypes selected. We applied this strategy for the construction of the Cu, Ag, Au convex hull finding systematically lower energy structures than the ones found in Aflow leading to the creation of a more accurate convex hull for this ternary alloy. As the results suggest, this workflow is a viable strategy for the construction of accurate convex hulls for ternary alloys. Furthermore, its design has been performed with the goal of not requiring additional DFT calculations for the training of the MLFF used for the screening of the prototypes. Such calculations would be "wasted" as they would not directly be used for the construction of the convex hull. Instead, the proposed workflow relies on data readily available on online databases. Future efforts on this task will involve the study of more challenging systems and the further optimisation of the workflow components. A study of this workflow applied to the Mo-Ta-W ternary compounds can be found in Ref. [3].

The second approach for structure prototype generation that we discuss relies on the use of generative models to learn the distribution of naturally occurring structures and generate novel prototypes based on that knowledge. Here we perform a first step in the direction of implementing such a methodology, we propose the use of local invariant descriptors of the chemical environment. These descriptors can be made suitable for generative models when coupled with a local inversion algorithm, which can reconstruct the structure corresponding to a given set of descriptors. A recent first example of a similar "inversion" strategy applied to the descriptors of a variational autoencoder can be found in Ref. [192]. Future work will focus on integrating this strategy and evaluating its utility in constructing sensible prototypes that can improve the prediction of the convex hulls of ternary alloys.

In Chapter 5, we introduced the Jacobi Legendre potential, a novel linear MLFF based on the cluster expansion of the energy. Using this MLFF, we provided an example of the capabilities of these models by predicting the dynamical properties of carbon.

Future efforts will focus on expanding the features implemented for the JLP to allow the access to active learning techniques that would sustain the automatic creation of training sets. In Chapter 6, we discussed a new approach to incorporate information related to atomic magnetic moments into the design of invariant descriptors for the local environment, the spin power spectrum. Utilising these descriptors, we created a model for bcc iron, which we used to predict its Curie temperature from first principles calculations. The methodology presented here will be extended to other compounds and further explored to incorporate a better study of the coupling between lattice and spin. Both the JLP and spin power spectrum exemplify how MLFF can be used to compute properties that may be otherwise inaccessible through DFT.

Finally, in Chapter 7 we detail the code developed to implement the various techniques introduced here. While the techniques presented have been designed to address challenges within an inverse-design workflow, their range of applicability is wider. For example, the Jacobi Legendre descriptors have been employed in the creation of machine-learning models for the prediction of the electronic charge density [193], showing a promising methodology to accelerate DFT calculations.

All in all, we provided a compelling case for the utility of data-driven strategy in solving complex problems in material science. The inherently interdisciplinary nature of this field has required extensive efforts in creating a code base that would integrate the various software tools demanded for each task. This led to the development of the Field Machine Python library. We foresee the open-access release of this library in the near future. Ultimately, the conclusive test of the validity of the work presented here is its ability to lead to the experimental realisation of novel magnetic materials. As such, future close collaborations with experimental groups, aimed at synthesizing the proposed prototypes, are essential.

Appendices

Appendix A

Computing the gradient of the JLP

In the following, we discuss the derivatives of the Jacobi-Legendre descriptors introduced in Chapter 5 necessary to fit and predict the system's forces and stress tensor. The gradient of the energy is related to the forces through Eq. (2.13) and can be computed by applying the chain rule to the definition of the descriptors.

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}x}\widetilde{P}_n^{(\alpha,\beta)}(\cos(x)) = \frac{\mathrm{d}}{\mathrm{d}x}P_n^{(\alpha,\beta)}(\cos(x)) = \\ &= -\frac{\alpha+\beta+n+1}{2}\sin(x)P_{n-1}^{(\alpha+1,\beta+1)}(\cos(x))\,. \end{aligned} \tag{A.1}$$

The derivatives of the Legendre polynomials follow as a particular case of the ones for the Jacobi polynomials

$$\frac{\mathrm{d}}{\mathrm{d}x}P_l(x) = \frac{\mathrm{d}}{\mathrm{d}x}P_l^{(0,0)}(x) = \frac{l+1}{2}P_{l-1}^{(1,1)}(x)\,. \tag{A.2}$$

Finally the differentiation rule for the polynomials $\overline{P}_n^{(\alpha,\beta)}$ defined in Eq. (5.12) are given by:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}x}\overline{P}_n^{(\alpha,\beta)}(\cos(x)) = \\ &= -\frac{\sin(x)}{2}\Bigg((\alpha+\beta+n+1)P_{n-1}^{(\alpha+1,\beta+1)}(\cos(x)) + \\ &\qquad -(\alpha+\beta+2)\frac{\widetilde{P}_n^{(\alpha,\beta)}(-1)}{\widetilde{P}_1^{(\alpha,\beta)}(-1)}\Bigg)\,. \end{aligned} \tag{A.3}$$

Appendix B

Phonon dispersion

Within the Born-Oppenheimer approximation, the total potential energy of the system is a parametric function of the atomic positions:

$$U = U(\vec{R}_1, \ldots, \vec{R}_N). \tag{B.1}$$

In practice, as discussed in Chapter 2, $U(\{\vec{R}_J\})$ can be the ground-state energy predicted by DFT or the potential energy provided by a classical or machine-learning force field.

Near an equilibrium configuration of the system, where the forces acting on the atoms are zero, we can consider the Taylor expansion to express the energy variation caused by small displacements ($\vec{u}_I = \vec{R}_I - \vec{R}^*_I$) around the equilibrium positions ($\{\vec{R}^*\}$):

$$U = U(\{\vec{R}^*\}) + \frac{1}{2} \sum_{\substack{IJ \\ \alpha\beta}} \frac{\partial^2 U}{\partial u_{I,\alpha} \partial u_{J,\beta}} (\{\vec{R}^*\})\, u_{I,\alpha} u_{J,\beta} + \mathcal{O}(u^3), \tag{B.2}$$

where the indexes α and β indicate the spatial directions. Within what takes the name of harmonic approximation we ignore the terms of the expansion beyond the second order. the expansion coefficients

$$K_{I\alpha,J\beta} = \frac{\partial^2 U}{\partial u_{I\alpha} \partial u_{J\beta}} (\{\vec{R}^*\}), \tag{B.3}$$

take the name of force constants. The symmetries interesting the problem allow to express K as a Fourier expansion over the Brillouin zone of the crystal.

$$K_{I\alpha,J\beta} = \frac{1}{N} \sum_{\vec{k}}{}' K_{\alpha,\beta}(\vec{k}) e^{i\vec{k}\cdot(\vec{R}_I - \vec{R}_J)}, \tag{B.4}$$

for a crystal lattice with a single atom per unit cell the matrix $K_{\alpha,\beta}(q)$ is a 3×3 positive-definite symmetric matrix. It has three real positive eigenvalues for each of the N possible values of $\vec{k}$. Within the framework of the harmonic approximation, the system can be decomposed into a set of independent harmonic oscillators. Their

frequencies ω, as a function of $\vec{k}$, give rise to a phonon dispersion curve, as the ones reported in Fig. 5.4. For a system with only one atom in the unit cell, the only branches present are three acoustic branches, characterised by $\omega \propto k$. In systems with more atoms in their unit cell, optical branches are also present.

Appendix C

Spin power spectrum rotational invariance

In this appendix, we provide proof of the rotation invariance of the spin power spectrum descriptors defined in Eq. (6.21). We can rewrite Eq. (6.19) as,

$$|\boldsymbol{\rho}_I\rangle = \sum_{nl\mathcal{J}M} u_{nl1\mathcal{J}M}\,|nl1\mathcal{J}M\rangle\ , \tag{C.1}$$

with

$$u_{nl1\mathcal{J}M} = \sum_{mq} C^{\mathcal{J}M}_{lm1\,q} c_{nlmq}\ . \tag{C.2}$$

Under a global rotation of the system $\hat{R}$, which also includes a simultaneous rotation of the atomic magnetic moments, the vectorial neighbour density distribution transforms as [181]:

$$\begin{aligned} \hat{R}\,|\boldsymbol{\rho}\rangle &= \sum_{nl\mathcal{J}M} u_{nl\mathcal{J}M}\hat{R}\,|nl1\mathcal{J}M\rangle = \\ &= \sum_{nl\mathcal{J}M} u_{nl\mathcal{J}M} \sum_{M'} D^{\mathcal{J}}_{M'M}(\hat{R})\,|nl1\mathcal{J}M'\rangle\ , \end{aligned} \tag{C.3}$$

where $D^{\mathcal{J}}_{M'M}(\hat{R})$ is the Wigner D-matrix associated with the rotation. From this follows that the spin power spectrum,

$$p_{nl\mathcal{J}} = \sum_{M} u^*_{nl\mathcal{J}M} u_{nl\mathcal{J}M}, \tag{C.4}$$

transforms as:

$$\begin{aligned} \hat{R} : p_{nl\mathcal{J}} \ \rightarrow\ & \sum_{M'M''} u^*_{nl\mathcal{J}M'} u_{nl\mathcal{J}M''} \underbrace{\sum_{M} \left(D^{\mathcal{J}}_{MM'}(\hat{R})\right)^* D^{\mathcal{J}}_{MM''}(\hat{R})}_{=\delta_{M'M''}} \\ &= \sum_{M'} u^*_{nl\mathcal{J}M'} u_{nl\mathcal{J}M'} = p_{nl\mathcal{J}}\ , \end{aligned} \tag{C.5}$$

where we exploited the unitarity of the Wigner D-matrices [181]. This concludes the demonstration of the rotational invariance of the spin power spectrum.

Appendix D

DFT Spinspirals

A spin spiral configuration is characterised by a constant-angle rotation of the atomic magnetic moments along a direction of the crystal $\hat{e}_q$. A specific spin spiral configuration is entirely specified by the propagation vector $\mathbf{q} = 2\pi/\lambda\hat{e}_{\mathbf{q}}$ with λ wavelength of the spiral.

In a spin spiral system, the magnetic moment of an atom α identified by the basis vector $\boldsymbol{\tau}_\alpha$ in the unit cell n from the origin in the z-direction, is given by:

$$\mathbf{m}_{n\alpha} = m_{i\alpha} \begin{pmatrix} \cos(\mathbf{q}\cdot(\mathbf{R}_n + \boldsymbol{\tau}_\alpha) + \phi_\alpha)\sin(\theta_\alpha) \\ \sin(\mathbf{q}\cdot(\mathbf{R}_n + \boldsymbol{\tau}_\alpha) + \phi_\alpha)\sin(\theta_\alpha) \\ \cos\theta_\alpha \end{pmatrix},$$

where θ_α and ϕ_α are the atomic magnetic moment's polar angles. If $\theta_\alpha \neq \pi/2$ a conical spiral structure arises. Spin-spiral configurations well describe low energy excitation modes of ferromagnets and antiferromagnets and can describe the ground state of some materials [194]. Standard spin-polarised DFT approaches struggle to model spin spiral configurations due to the large supercells required to simulate long wavelength spirals. In particular, spin spirals with a period that is incommensurate with respect to the lattice cannot be fully captured within a supercell.

By ignoring the presence of spin-orbit coupling and decoupling the the magnetic configurations of the system from its crystal lattice, it is possible to apply a generalised Bloch theorem which allows us to treat the problem by just considering a unit cell of the system. Without loss of generality, the propagation vector of the spiral is assumed to be along the z-axis.

Given a spin spiral configuration over a perfect infinite lattice, due to the symmetry of the problem, a translation of n unit cell along the z-direction will result in a spin rotation plus a translation of the ground-state wave function of the system:

$$\mathcal{T}_n H(\mathbf{r})\boldsymbol{\psi}(\mathbf{r}) = H(\mathbf{r})U_z(-\mathbf{q}\cdot\mathbf{R}_n)\boldsymbol{\psi}(\mathbf{r}+\mathbf{R}_n), \tag{D.1}$$

where the matrix $U(\phi)$ is the spin-1/2 rotation matrix inducing a rotation of the spin

when acting on the spinor $\boldsymbol{\psi}$:

$$U_z(\phi) = \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{pmatrix}. \tag{D.2}$$

A generalised version of the Bloch theorem can be proved according to which it exists a set of eigenstates of the problem such that [194, 195]:

$$\boldsymbol{\psi}_{\mathbf{k}j}(\mathbf{r}|\mathbf{q}) = \begin{pmatrix} \psi^{(\uparrow)}_{\mathbf{k}j}(\mathbf{r}) \\ \psi^{(\downarrow)}_{\mathbf{k}j}(\mathbf{r}) \end{pmatrix} = \begin{pmatrix} e^{i(\mathbf{k}-\mathbf{q}/2)\cdot\mathbf{r}} u^{(\uparrow)}_{\mathbf{k}j}(\mathbf{r}) \\ e^{i(\mathbf{k}+\mathbf{q}/2)\cdot\mathbf{r}} u^{(\downarrow)}_{\mathbf{k}j}(\mathbf{r}) \end{pmatrix} \tag{D.3}$$

where the functions $u^{(\sigma)}_{\mathbf{k}j}(\mathbf{r})$ with $\sigma \in \{\uparrow, \downarrow\}$ are periodic functions with the same periodicity of the lattice. The $\boldsymbol{q}$-dependent phase factor appearing in Eq. (D.3) differentiates the spin-spiral solution.

Bibliography

[1] Luke P. J. Gilligan, Matteo Cobelli, Valentin Taufour, and Stefano Sanvito. A rule-free workflow for the automated generation of databases from scientific literature. *npj Computational Materials*, 9(1):222, Dec 2023.

[2] Michail Minotakis, Hugo Rossignol, Matteo Cobelli, and Stefano Sanvito. Machine-learning surrogate model for accelerating the search of stable ternary alloys. *Phys. Rev. Mater.*, 7:093802, Sep 2023.

[3] Hugo Rossignol, Michail Minotakis, Matteo Cobelli, and Stefano Sanvito. Machine-learning-assisted construction of ternary convex hull diagrams. *Journal of Chemical Information and Modeling*, Jan 2024.

[4] Matteo Cobelli, Paddy Cahalane, and Stefano Sanvito. Local inversion of the chemical environment representations. *Phys. Rev. B*, 106:035402, Jul 2022.

[5] Michelangelo Domina, Urvesh Patil, Matteo Cobelli, and Stefano Sanvito. Cluster expansion constructed over jacobi-legendre polynomials for accurate force fields. *Phys. Rev. B*, 108:094102, Sep 2023.

[6] Michelangelo Domina, Matteo Cobelli, and Stefano Sanvito. Spectral neighbor representation for vector fields: Machine learning potentials including spin. *Phys. Rev. B*, 105:214439, Jun 2022.

[7] Alex Zunger. Inverse design in search of materials with target functionalities. *Nature Reviews Chemistry*, 2(4):0121, Mar 2018.

[8] Robert G. Parr and Weitao Yang. *Density-Functional Theory of Atoms and Molecules (International Series of Monographs on Chemistry)*. Oxford University Press, USA, 1994.

[9] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[10] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.

[11] W. Kohn. Density functional and density matrix method scaling linearly with the number of atoms. *Phys. Rev. Lett.*, 76:3168–3171, Apr 1996.

[12] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1), 07 2013. 011002.

[13] Stefano Curtarolo, Wahyu Setyawan, Shidong Wang, Junkai Xue, Kesong Yang, Richard H. Taylor, Lance J. Nelson, Gus L.W. Hart, Stefano Sanvito, Marco Buongiorno-Nardelli, Natalio Mingo, and Ohad Levy. Aflowlib.org: A distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.*, 58:227–235, 2012.

[14] S. Kirklin, J.E. Saal, B. Meredig, A. Thompson, J.W. Doak, M. Aykol, S. Rühl, and C. Wolverton. The open quantum materials database (OQMD): assessing the accuracy of dft formation energies. *npj Comput. Mater.*, 1:15010, 2015.

[15] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, Aug 2021.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv*, 2103.00020, 2021.

[17] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[18] Laura Gambini, Tiarnan Mullarkey, Lewys Jones, and Stefano Sanvito. Machine-learning approach for quantified resolvability enhancement of low-dose stem data. *Machine Learning: Science and Technology*, 4(1):015025, feb 2023.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[20] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct 1986.

[21] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, Ryota Tomioka, and Tian Xie. Mattergen: a generative model for inorganic materials design. *arXiv*, 2312.03687, 2023.

[22] James Nelson and Stefano Sanvito. Predicting the curie temperature of ferromagnets using machine learning. *Phys. Rev. Materials*, 3:104405, Oct 2019.

[23] Jingzi Zhang, Zhuoxuan Zhu, X.-D. Xiang, Ke Zhang, Shangchao Huang, Chengquan Zhong, Hua-Jun Qiu, Kailong Hu, and Xi Lin. Machine learning prediction of superconducting critical temperature through the structural descriptor. *J. Phys. Chem. C*, 126(20):8922–8927, 2022.

[24] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1):16028, Aug 2016.

[25] Cormac Toher, Corey Oses, David Hicks, and Stefano Curtarolo. Unavoidable disorder and entropy in multi-component systems. *npj Computational Materials*, 5(1):69, Jul 2019.

[26] Gabriele C. Sosso, Giacomo Miceli, Sebastiano Caravati, Jörg Behler, and Marco Bernasconi. Neural network interatomic potential for the phase change material gete. *Phys. Rev. B*, 85:174103, May 2012.

[27] Volker L. Deringer and Gábor Csányi. Machine learning based interatomic potential for amorphous carbon. *Phys. Rev. B*, 95:094203, Mar 2017.

[28] Rodrigo Freitas and Yifan Cao. Machine-learning potentials for crystal defects. *MRS Communications*, 12(5):510–520, Oct 2022.

[29] R. P. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, Aug 1939.

[30] S. J. Plimpton A. P. Thompson and W. Mattson. General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *J. Chem. Phys.*, 131:154107, 2009.

[31] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967.

[32] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv*, 2104.13478, 2021.

[33] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, Apr 2007.

[34] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Phys. Rev. B*, 87:184115, May 2013.

[35] Mitchell A. Wood and Aidan P. Thompson. Extending the accuracy of the SNAP interatomic potential form. *The Journal of Chemical Physics*, 148(24):241721, 03 2018.

[36] Sergey N. Pozdnyakov, Michael J. Willatt, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Incompleteness of atomic structure representations. *Phys. Rev. Lett.*, 125:166001, Oct 2020.

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

[38] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B*, 99:014104, Jan 2019.

[39] Yury Lysogorskiy, Cas van der Oord, Anton Bochkarev, Sarath Menon, Matteo Rinaldi, Thomas Hammerschmidt, Matous Mrovec, Aidan Thompson, Gábor Csányi, Christoph Ortner, and Ralf Drautz. Performant implementation of the atomic cluster expansion (pace) and application to copper and silicon. *npj Computational Materials*, 7(1):97, Jun 2021.

[40] Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *arXiv*, 2206.07697, 2023.

[41] Anton Bochkarev, Yury Lysogorskiy, Sarath Menon, Minaam Qamar, Matous Mrovec, and Ralf Drautz. Efficient parametrization of the atomic cluster expansion. *Phys. Rev. Mater.*, 6:013804, Jan 2022.

[42] Minaam Qamar, Matous Mrovec, Yury Lysogorskiy, Anton Bochkarev, and Ralf Drautz. Atomic cluster expansion for quantum-accurate large-scale simulations of carbon. *Journal of Chemical Theory and Computation*, 19(15):5151–5167, 2023. PMID: 37347981.

[43] Ngoc Cuong Nguyen and Andrew Rohskopf. Proper orthogonal descriptors for efficient and accurate interatomic potentials. *Journal of Computational Physics*, 480:112030, 2023.

[44] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[45] A.P. Thompson, L.P. Swiler, C.R. Trott, S.M. Foiles, and G.J. Tucker. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics*, 285:316–330, 2015.

[46] Volker L. Deringer, Albert P. Bartók, Noam Bernstein, David M. Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021. PMID: 34398616.

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

[48] Mackay D. Information theory, inference, and learning algorithms. *Cambridge Univ. Press: Cambridge, United Kingdom*, 2003.

[49] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104:136403, Apr 2010.

[50] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[51] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017.

[52] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 03 2018.

[53] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, Nov 2022.

[54] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *arXiv*, 1802.08219, 2018.

[55] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, May 2022.

[56] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *arXiv*, 2102.09844, 2022.

[57] Vu Ha Anh Nguyen and Alessandro Lunghi. Predicting tensorial molecular properties with equivariant machine learning models. *Phys. Rev. B*, 105:165131, Apr 2022.

[58] J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.*, 8:3192–3203, 2017.

[59] Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017.

[60] Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature Communications*, 9(1):3887, Sep 2018.

[61] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120:143001, Apr 2018.

[62] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.

[63] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9377–9388. PMLR, 18–24 Jul 2021.

[64] Muhammed Shuaibi, Adeesh Kolluru, Abhishek Das, Aditya Grover, Anuroop Sriram, Zachary Ulissi, and C. Lawrence Zitnick. Rotation invariant graph neural networks using spin convolutions. *arXiv*, 2106.09575, 2021.

[65] Chris J. Pickard. Ephemeral data derived potentials for random structure search. *Phys. Rev. B*, 106:014102, Jul 2022.

[66] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171, 2022.

[67] J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam. Accelerating finite-temperature kohn-sham density functional theory with deep neural networks. *Phys. Rev. B*, 104:035120, Jul 2021.

[68] Sara Hooker. The hardware lottery. *arXiv*, 2020.

[69] A.I. Liechtenstein, M.I. Katsnelson, V.P. Antropov, and V.A. Gubanov. Lsdf-approach to the theory of exchange interactions in magnetic metals. *Journal of Magnetism and Magnetic Materials*, 54-57:965–966, 1986.

[70] J. F. Janak. Proof that $\frac{\partial e}{\partial n_i} = \epsilon$ in density-functional theory. *Phys. Rev. B*, 18:7165–7168, Dec 1978.

[71] John P. Perdew. Density functional theory and the band gap problem. *International Journal of Quantum Chemistry*, 28(S19):497–523, 1985.

[72] Ya Zhuo, Aria Mansouri Tehrani, and Jakoah Brgoch. Predicting the band gaps of inorganic solids by machine learning. *The Journal of Physical Chemistry Letters*, 9(7):1668–1673, 2018. PMID: 29532658.

[73] Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98, Jul 2019.

[74] Anthony Yu-Tung Wang, Steven K. Kauwe, Ryan J. Murdock, and Taylor D. Sparks. Compositionally restricted attention-based network for materials property predictions. *npj Computational Materials*, 7(1):77, May 2021.

[75] Charles J. Stone R.A. Olshen Leo Breiman, Jerome Friedman. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[76] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[77] Valentin Stanev, Corey Oses, A. Gilad Kusne, Efrain Rodriguez, Johnpierre Paglione, Stefano Curtarolo, and Ichiro Takeuchi. Machine learning modeling of superconducting critical temperature. *npj Computational Materials*, 4(1):29, Jun 2018.

[78] J.M.D. Coey. *Magnetism and Magnetic Materials*. Cambridge University Press, Cambridge, 2010.

[79] Matthew C. Swain and Jacqueline M. Cole. ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature. 56(10):1894–1904, 2016. PMID: 27669338.

[80] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv*, 1301.3781, 2013.

[81] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv*, 1706.03762, 2017.

[82] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[83] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv*, 2005.14165, 2020.

[84] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv*, 2203.02155, 2022.

[85] OpenAI ChatGPT. `https://chat.openai.com`.

[86] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv*, 1310.4546, 2013.

[87] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[88] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[89] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv*, 2022.

[90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv*, 1512.03385, 2015.

[91] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 1810.04805, 2018.

[92] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society.

[93] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv*, 1907.11692, 2019.

[94] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, oct 2021.

[95] Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, May 2022.

[96] Pranav Shetty, Arunkumar Chitteth Rajan, Christopher Kuenneth, Sonkakshi Gupta, Lakshmi Prerana Panchumarti, Lauren Holm, Chao Zhang, and Rampi Ramprasad. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *arXiv*, 2022.

[97] Shu Huang and Jacqueline M. Cole. BatteryBERT: A pretrained language model for battery database enhancement. 62:6365–6377, May 2022.

[98] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[99] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003.

[100] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[101] BigScience Workshop: Teven Le Scao et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv*, 2211.05100, 2023.

[102] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv*, 2302.13971, 2023.

[103] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2307.09288, 2023.

[104] Leopold Talirz, Snehal Kumbhar, Elsa Passaro, Aliaksandr V. Yakutovich, Valeria Granata, Fernando Gargiulo, Marco Borelli, Martin Uhrin, Sebastiaan P. Huber, Spyros Zoupanos, Carl S. Adorf, Casper Welzel Andersen, Ole Schütt, Carlo A. Pignedoli, Daniele Passerone, Joost VandeVondele, Thomas C. Schulthess, Berend Smit, Giovanni Pizzi, and Nicola Marzari. Materials cloud, a platform for open computational science. *Sci. Data*, 7:299, 2020.

[105] Ya Zhuo, Aria Mansouri Tehrani, and Jakoah Brgoch. Predicting the band gaps of inorganic solids by machine learning. 9(7):1668–1673, Apr 2018.

[106] Antanas Vaitkus, Andrius Merkys, and Saulius Gražulis. Validation of the Crystallography Open Database using the Crystallographic Information Framework. *J. Appl. Crystallogr.*, 54(2):661–672, Apr 2021.

[107] D. Zagorac, H. Müller, S. Ruehl, J. Zagorac, and S. Rehme. Recent developments in the Inorganic Crystal Structure Database: theoretical crystal structure data and related features. *J. Appl. Crystallogr.*, 52(5):918–925, Oct 2019.

[108] Colin R Groom, Ian J Bruno, Matthew P Lightfoot, and Suzanna C Ward. The cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72(2):171–179, 2016.

[109] Samuel V. Gallego, J. Manuel Perez-Mato, Luis Elcoro, Emre S. Tasci, Robert M. Hanson, Koichi Momma, Mois I. Aroyo, and Gotzon Madariaga. *MAGNDATA*: towards a database of magnetic structures. I.The commensurate case. *J. Appl. Crystallogr.*, 49(5):1750–1776, Oct 2016.

[110] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[111] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, page 85–94, New York, NY, USA, 2000. Association for Computing Machinery.

[112] Callum J. Court and Jacqueline M. Cole. Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction. *Sci. Data*, 5:180111, 2018.

[113] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. 28(1):11–21, Jan 1972.

[114] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy, August 2019. Association for Computational Linguistics.

[115] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[116] Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S. Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv*, 2212.05238, 2022.

[117] Nicholas Walker, John Dagdelen, Kevin Cruse, Sanghoon Lee, Samuel Gleason, Alexander Dunn, Gerbrand Ceder, A. Paul Alivisatos, Kristin A. Persson, and Anubhav Jain. Extracting structured seed-mediated gold nanorod growth procedures from literature with gpt-3. *arXiv*, 2304.13846, 2023.

[118] Maciej P. Polak and Dane Morgan. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *arXiv*, 2303.05352, 2023.

[119] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. *arXiv*, page 1906.03158, Jun 2019.

[120] Steven Bird, Edward Loper, and Ewan Klein. Natural language processing with Python. *O'Reilly Media Inc.*, 2009.

[121] arXiv Dataset. `https://www.kaggle.com/datasets/Cornell-University/arxiv`. Accessed: 2023-02-24.

[122] Qingyang Dong and Jacqueline M. Cole. Auto-generated database of semiconductor band gaps using chemdataextractor. *Scientific Data*, 9(1):193, May 2022.

[123] Yusuke Shinyama. PDFMiner - Python PDF Parser, 2007.

[124] Yibin Xu, Masayoshi Yamazaki, and Pierre Villars. Inorganic materials database for exploring the nature of material. *Jpn. J. App. Phys.*, 50:11RH02, 2011.

[125] T. F. Connolly. *Bibliography of Magnetic Materials and Tabulation of Magnetic Transition Temperatures*. Springer Science & Business Media, New York, US, 2012.

[126] K. Buschow and E. Wohlfarth, editors. *Handbook of Magnetic Materials, Volumes 4-16 and 18*. Elsevier, Amsterdam, Netherlands, 1988-2009.

[127] Journey K. Byland, Yunshu Shi, David S. Parker, Jingtai Zhao, Shaoqing Ding, Rogelio Mata, Haley E. Magliari, Andriy Palasyuk, Sergey L. Bud'ko, Paul C. Canfield, Peter Klavins, and Valentin Taufour. Statistics on magnetic properties of Co compounds: A database-driven method for discovering Co-based ferromagnets. *Phys. Rev. Mater.*, 6:063803, Jun 2022.

[128] Xiaodong Si, Yongsheng Liu, Wei Lei, Juan Xu, Wenlong Du, Jia Lin, Tao Zhou, and Li Zheng. First-principles investigation on the optoelectronic performance of mg doped and mg–al co-doped zno. *Materials & Design*, 93:128–132, 2016.

[129] Lanli Chen, Aiping Wang, Zhihua Xiong, Siqi Shi, and Yanfeng Gao. Effect of hole doping and strain modulations on electronic structure and magnetic properties in zno monolayer. *Applied Surface Science*, 467-468:22–29, 2019.

[130] W. Bludau, A. Onton, and W. Heinke. Temperature dependence of the band gap of silicon. *Journal of Applied Physics*, 45(4):1846–1848, 10 2003.

[131] Yoshio Nosaka and Atsuko Y Nosaka. Reconsideration of intrinsic band alignments within anatase and rutile TiO2. *J Phys Chem Lett*, 7(3):431–434, February 2016.

[132] Th. Böker, R. Severin, A. Müller, C. Janowitz, R. Manzke, D. Voß, P. Krüger, A. Mazur, and J. Pollmann. Band structure of mos_2, $mose_2$, and $\alpha - mote_2$: angle-resolved photoelectron spectroscopy and ab initio calculations. *Phys. Rev. B*, 64:235305, Nov 2001.

[133] B. Radisavljevic, A. Radenovic, J. Brivio, V. Giacometti, and A. Kis. Single-layer mos2 transistors. *Nature Nanotechnology*, 6(3):147–150, Mar 2011.

[134] Qing Tang and De-en Jiang. Stabilization and band-gap tuning of the 1t-mos2 monolayer by covalent functionalization. *Chemistry of Materials*, 27(10):3743–3748, 2015.

[135] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov. Electric field effect in atomically thin carbon films. *Science*, 306(5696):666–669, 2004.

[136] Chris J.H. Wort and Richard S. Balmer. Diamond as an electronic material. *Materials Today*, 11(1):22–28, 2008.

[137] Saeid Jalali-Asadabadi, E. Ghasemikhah, T. Ouahrani, B. Nourozi, M. Bayat-Bayatani, S. Javanbakht, H. A. Rahnamaye Aliabad, Iftikhar Ahmad, J. Nematollahi, and M. Yazdani-Kachoei. Electronic structure of crystalline buckyballs: fcc-c60. *Journal of Electronic Materials*, 45(1):339–348, Jan 2016.

[138] Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, Sep 2020.

[139] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[140] Corey Oses, Eric Gossett, David Hicks, Frisco Rose, Michael J. Mehl, Eric Perim, Ichiro Takeuchi, Stefano Sanvito, Matthias Scheffler, Yoav Lederer, Ohad Levy, Cormac Toher, and Stefano Curtarolo. Aflow-chull: Cloud-oriented platform for autonomous phase stability analysis. *Journal of Chemical Information and Modeling*, 58(12):2477–2490, 2018. PMID: 30188699.

[141] Michael J. Mehl, David Hicks, Cormac Toher, Ohad Levy, Robert M. Hanson, Gus Hart, and Stefano Curtarolo. The aflow library of crystallographic prototypes: Part 1. 136:S1–S828, 2017.

[142] David Hicks, Michael J. Mehl, Eric Gossett, Cormac Toher, Ohad Levy, Robert M. Hanson, Gus Hart, and Stefano Curtarolo. The aflow library of crystallographic prototypes: Part 2. 161:S1–S1011, 2019.

[143] David Hicks, Michael J. Mehl, Marco Esters, Corey Oses, Ohad Levy, Gus L.W. Hart, Cormac Toher, and Stefano Curtarolo. The aflow library of crystallographic prototypes: Part 3. 199:110450, 2021.

[144] G. Kresse and J. Hafner. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B*, 47:558–561, Jan 1993.

[145] G. Kresse and J. Furthmüller. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science*, 6(1):15–50, 1996.

[146] G. Kresse and J. Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54:11169–11186, Oct 1996.

[147] Camilo E. Calderon, Jose J. Plata, Cormac Toher, Corey Oses, Ohad Levy, Marco Fornari, Amir Natan, Michael J. Mehl, Gus Hart, Marco Buongiorno Nardelli, and Stefano Curtarolo. The aflow standard for high-throughput materials science calculations. *Computational Materials Science*, 108:233–238, 2015.

[148] Gus LW Hart and Rodney W Forcade. Algorithm for generating derivative structures. *Phys. Rev. B*, 77(22):224115, 2008.

[149] Gus L. W. Hart and Rodney W Forcade. Generating derivative structures from multilattices: Algorithm and application to hcp alloys. *Phys. Rev. B*, 80(1):014120, 2009.

[150] Gus LW Hart, Lance J Nelson, and Rodney W Forcade. Generating derivative structures at a fixed concentration. *Comput. Mater. Sci.*, 59:101–107, 2012.

[151] Alan Prince. Phase diagrams of ternary gold alloys. *Inst. Metals*, pages 7–42, 1990.

[152] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv*, 1312.6114, 2022.

[153] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[154] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. PMID: 29532027.

[155] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[156] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[157] Wenhao Gao and Connor W. Coley. The synthesizability of molecules proposed by generative models. *Journal of Chemical Information and Modeling*, 60(12):5714–5723, 2020. PMID: 32250616.

[158] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. drugan: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular Pharmaceutics*, 14(9):3098–3104, 2017. PMID: 28703000.

[159] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv*, 1805.11973, 2022.

[160] Yong Zhao, Mohammed Al-Fahdi, Ming Hu, Edirisuriya M. D. Siriwardane, Yuqi Song, Alireza Nasiri, and Jianjun Hu. High-throughput discovery of novel cubic crystal materials using deep generative neural networks. *Advanced Science*, 8(20):2100566, 2021.

[161] Asma Nouira, Nataliya Sokolovska, and Jean-Claude Crivello. Crystalgan: Learning to discover crystallographic structures with generative adversarial networks. *arXiv*, 1810.11203, 2019.

[162] Sungwon Kim, Juhwan Noh, Geun Ho Gu, Alan Aspuru-Guzik, and Yousung Jung. Generative adversarial networks for crystal structure prediction. *ACS Central Science*, 6(8):1412–1420, 2020. PMID: 32875082.

[163] Martin Uhrin. Through the eyes of a descriptor: Constructing complete, invertible descriptions of atomic environments. *Phys. Rev. B*, 104:144110, Oct 2021.

[164] Felix Musil, Andrea Grisafi, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Physics-inspired structural representations for molecules and materials. *Chemical Reviews*, 121(16):9759–9815, 2021. PMID: 34310133.

[165] Jörg Behler. Four generations of high-dimensional neural network potentials. *Chemical Reviews*, 121(16):10037–10072, 2021. PMID: 33779150.

[166] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth gpo printing edition, 1964.

[167] Atsushi Togo, Laurent Chaput, and Isao Tanaka. Distributions of phonon lifetimes in brillouin zones. *Phys. Rev. B*, 91:094306, Mar 2015.

[168] Atsushi Togo. First-principles phonon calculations with phonopy and phono3py. *Journal of the Physical Society of Japan*, 92(1):012001, 2023.

[169] Guido Petretto, Shyam Dwaraknath, Henrique P.C. Miranda, Donald Winston, Matteo Giantomassi, Michiel J. van Setten, Xavier Gonze, Kristin A. Persson, Geoffroy Hautier, and Gian-Marco Rignanese. High-throughput density-functional perturbation theory phonons for inorganic materials. *Scientific Data*, 5(1):180065, May 2018.

[170] Stewart J. Clark, Matthew D. Segall, Chris J. Pickard, Phil J. Hasnip, Matt I. J. Probert, Keith Refson, and Mike C. Payne. First principles methods using castep. *Zeitschrift für Kristallographie - Crystalline Materials*, 220(5-6):567–570, 2005.

[171] S. L. Dudarev and P. M. Derlet. Interatomic potentials for materials with interacting electrons. *Journal of Computer-Aided Materials Design*, 14(1):129–140, Dec 2007.

[172] Pui-Wai Ma, C. H. Woo, and S. L. Dudarev. Large-scale simulation of the spin-lattice dynamics in ferromagnetic iron. *Phys. Rev. B*, 78:024434, Jul 2008.

[173] David P. Landau and Kurt Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 4 edition, 2014.

[174] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 12 2004.

[175] Marco Eckhoff and Jörg Behler. High-dimensional neural network potentials for magnetic systems using spin-dependent atom-centered symmetry functions. *npj Computational Materials*, 7(1):170, Oct 2021.

[176] Svetoslav Nikolov, Mitchell A. Wood, Attila Cangi, Jean-Bernard Maillet, Mihai-Cosmin Marinica, Aidan P. Thompson, Michael P. Desjarlais, and Julien Tranchida. Data-driven magneto-elastic predictions with scalable classical spin-lattice dynamics. *npj Computational Materials*, 7(1):153, Sep 2021.

[177] Jacob B. J. Chapman and Pui-Wai Ma. A machine-learned spin-lattice potential for dynamic simulations of defective magnetic iron. *Scientific Reports*, 12(1):22451, Dec 2022.

[178] Keith Dalbey, Michael S. Eldred, Gianluca Geraci, John Davis Jakeman, Kathryn Anne Maupin, Jason A. Monschke, Daniel Thomas Seidl, Laura Painton Swiler, Anh Tran, Friedrich Menhorn, and Xiaoshu Zeng. Dakota a multilevel parallel object-oriented framework for design optimization parameter estimation uncertainty quantification and sensitivity analysis: Version 6.12 theory manual. 5 2020.

[179] M Weissbluth. Atoms and molecules. 1978.

[180] Emir Kocer, Jeremy K. Mason, and Hakan Erturk. Continuous and optimally complete description of chemical environments using Spherical Bessel descriptors. *AIP Advances*, 10(1):015021, 01 2020.

[181] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii. *Quantum Theory of Angular Momentum*. World Scientific, 1988.

[182] F. Bloch. Zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 61(3):206–219, Mar 1930.

[183] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013.

[184] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, jun 2017.

[185] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[186] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature*, 585(7825):357–362, Sep 2020.

[187] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[188] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science and Engg.*, 13(2):31–39, mar 2011.

[189] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[190] You-Sheng Lin, Guan-De Li, Shan-Ping Mao, and Jeng-Da Chai. Long-range corrected hybrid density functionals with improved dispersion corrections. *Journal of Chemical Theory and Computation*, 9(1):263–272, 2013. PMID: 26589028.

[191] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter notebooks - a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Scmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016.

[192] Victor Fung, Shuyi Jia, Jiaxin Zhang, Sirui Bi, Junqi Yin, and P Ganesh. Atomic structure generation from reconstructing structural fingerprints. *Machine Learning: Science and Technology*, 3(4):045018, nov 2022.

[193] Bruno Focassio, Michelangelo Domina, Urvesh Patil, Adalberto Fazzio, and Stefano Sanvito. Linear jacobi-legendre expansion of the charge density for machine learning-accelerated electronic structure calculations. *npj Computational Materials*, 9(1):87, May 2023.

[194] Ph. Kurz, F. Förster, L. Nordström, G. Bihlmayer, and S. Blügel. Ab initio treatment of noncollinear magnets with the full-potential linearized augmented plane wave method. *Phys. Rev. B*, 69:024415, Jan 2004.

[195] M. Heide, G. Bihlmayer, and S. Blügel. Describing dzyaloshinskii–moriya spirals from first principles. *Physica B: Condensed Matter*, 404(18):2678–2683, 2009.