SMALL DATASETS, BIG PREDICTIONS: LEARNING METHODS FOR UNCERTAINTY-AWARE MODELLING OF MULTI-FIDELITY MATERIAL PROPERTIES

Thèse présentée en vue de l'obtention du grade de Docteur en Sciences de l'Ingénieur

DE BREUCK PIERRE-PAUL

Supervisor:

Prof. Gian-Marco Rignanese

Members of the jury:

Prof. Christophe de Vleeschouwer
Prof. John A. Lee (secretary)
Prof. Miguel Marques
Prof. Xavier Urbain (chairman)
Prof. Pierre Vandergheynst

UCLouvain
Institute of Condensed Matter and Nanosciences

March 2024 – v24.01.23

ABSTRACT

Functional materials play a critical role in various technological applications, and the availability of open databases has paved new paths for materials design. In this thesis, I address the challenge of supervised materials design with limited datasets by investigating various learning methods and proposing an all-encompassing framework called MODNet. Traditional approaches often rely on large datasets, which may not be readily available in practice. MODNet overcomes this limitation by combining a feedforward neural network with physically meaningful feature selection and joint learning. This results in faster training and superior performance on small datasets compared to existing graph-network models. MODNet showcases excellent performance on the Matbench test suite and achieves a mean absolute test error of 0.009 meV/K/atom on the vibrational entropy of crystals at 305 K, significantly outperforming previous studies.

The thesis also addresses the critical task of uncertainty assessment in material science, utilizing an ensemble MODNet model to build confidence intervals and quantify uncertainty in individual predictions. Furthermore, the potential of active learning is explored by using Bayesian Optimization to iteratively explore metals within the materials space. The significance of considering imbalance and bias in the training set for successful real-world applications of machine learning in materials science is emphasized.

Additionally, the thesis investigates techniques that leverage multiple quality sources for the same property, with a focus on the electronic band gap. By learning from differences between high- and low-quality values as a correction, substantial improvements in results are achieved compared to relying solely on high-quality experimental data.

This research significantly contributes to advancing machine learning in material property predictions and offers valuable insights into uncertainty assessment and data quality aspects in the field. By providing a comprehensive framework and effectively addressing critical challenges, this research opens new opportunities for accelerating materials design and discovery.

ACKNOWLEDGMENTS

First and foremost, I would like to express my profound gratitude to my advisor, Prof. Gian-Marco Rignanese. I recall beginning my internship at MIT with little to no knowledge about Machine Learning, a project that he further welcomed with enthusiasm. From that point forward, we embarked on a journey of joyful discovery, united by our shared passion for the subject. His relentless drive, perseverance, and commitment to excellence have been a source of inspiration for me. I am immensely grateful for his guidance and, above all, the independence that he gradually entrusted me with. This trust allowed me to explore my interests within the field freely, and for that, I am truly thankful.

I am thankful to the jury members, Christophe de Vleeschouwer, John A. Lee, Miguel Marques, Xavier Urbain, and Pierre Vandergheynst, for graciously accepting to review my thesis manuscript. Their valuable feedback and insights are truly appreciated.

I would like to express my deep appreciation to Tian Xie, whose passion for Materials Informatics have been instrumental in shaping my interest and understanding in the field.

A special word of thanks goes to Matthew L. Evans, whose guidance and mentorship have been invaluable in my professional growth. He has taught me various aspects of Python development, taking my skills to new heights. Beyond that, I deeply appreciate his genuine qualities as a human being and the way he approaches science with passion and dedication.

I am grateful to the entire MODL lab, many of whom have become friends, for their engaging discussions, camaraderie, and enjoyable moments. Yuqing, thank you for being my officemate and withstanding my good and bad jokes. Alexandre, Bogdan, Guillaume, Julien, Laura, Maryam, Mathilde, Romain, thank you for the nice time spent at Cagliari after a long lock-down.

I am grateful to the entire administrative team, including Vinciane, Bénédicte, Sadia, and Jean-Michel for their dedicated efforts in running the lab and for the enjoyable moments we have shared.

I extend my thanks to all the students I had the opportunity to advise: Romain, Grégoire, Cabrel, Rodrigo, Jordi, Najlae, Patrick, Victor and Hao.

Over the last two years, I had the privilege of serving as president of the ACIM, and I am thankful for the devoted team that evolved significantly during this time. Despite the challenges, the experience has been incredibly rewarding, and I am proud of our collective efforts in contributing to the institute's social life.

A special mention to Victor, not only for being a wonderful colleague but also for being my sports buddy throughout this past year.

J'aimerais exprimer ma reconnaissance à Alexis et Constance pour l'année et demie passée en colocation avec eux, malgré les défis d'une crise sanitaire difficile à gérer. Les souvenirs resteront gravés dans ma mémoire, et je suis profondément reconnaissant pour leur respect, leur calme et leur simplicité.

Je tiens à exprimer ma profonde gratitude envers Adrien, Gabriel et Thomas pour l'année incroyable passée à la Colove. Cette période a été particulièrement significative dans ma vie, marquée par de nombreux moments forts ensemble. Je chérirai toujours ces bons souvenirs.

J'aimerais remercier ma famille pour son soutien, et surtout,
à ma mère qui m'a tout donné et qui semble toujours trouver un moyen de donner plus.

PUBLICATIONS

All publications authored during this PhD thesis are listed below, ordered by publication date. The first three publications closely align with the content presented in Chapters 2, 3, and 4, respectively.

[P1] **De Breuck, P.-P.**, Hautier, G. & Rignanese, G.-M., *Materials property prediction for limited datasets enabled by feature selection and joint learning with MODNet*, npj Comput. Mater. **7**, 83 (2021)

[P2] **De Breuck, P.-P.**, Evans, M. L. & Rignanese, G.-M., *Robust model benchmarking and bias-imbalance in data-driven materials science: a case study on MODNet*, J. Phys.: Condens. Matter **33**, 404002 (2021)

[P3] **De Breuck, P.-P.**, Heymans, G. & Rignanese, G.-M, *Accurate experimental band gap predictions with multifidelity correction learning*, J Mater. Inf. **2**, 10 (2022)

[P4] Liu, X., **De Breuck, P.-P.**, Wang, L. & Rignanese, G.-M., *A simple denoising approach to exploit multi-fidelity data for machine learning materials properties*, npj Comput. Mater. **8**, 233 (2022)

[P5] Cruz, P. J., **De Breuck, P.-P.**, Rignanese, G.-M., Glinel, K. & Jonas, *Influence of roughness and coating on the rebound of droplets on fabrics*, Surfaces and Interfaces **36**, 102524 (2023)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AF	Acceleration Factor
AI	Artificial Intelligence
AL	Active Learning
ANN	Artificial Neural Networks
BO	Bayesian Optimization
CGCNN	Crystal Graph Convolutional Neural Network
CI	Confidence Interval
DFPT	Density Functional Perturbation Theory
DFT	Density Functional Theory
DOS	Density of States
EF	Enhancement Factor
GA	Genetic Algorithm
HSE	Heyd–Scuseria–Ernzerhof
ICSD	Inorganic Crystallographic Structure Database
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
MI	Material Informatics
ML	Machine Learning

MODNet	Materials Optimal Descriptor Network
MP	Materials Project
MPID	Materials Project ID
MSE	Mean Square Error
NMI	Normalized Mutual Information
PBE	Perdew-Burke-Ernzerhof
PCA	Principal Component Analysis
pdf	Probability Density Function
ReLU	Rectified Linear Unit Function
RF	Random Forest
ROC-AUC	Receiver Operating Characteristic Area Under the Curve
RR	Relevance and Redundancy
SGD	Stochastic Gradient Descent
SISSO	Sure Independence Screening and Sparsifying Operator
TF	Top Fraction

GLOSSARY

This section aims to provide definitions for selected technical terms that may vary in their usage across different research fields, rather than providing an exhaustive list.

Multi-Fidelity Learning involves the integration of diverse information sources, each with varying levels of accuracy, to optimize the performance of a machine learning model. In the present context, it uses different estimations (e.g., experimental, simulated) for a particular property (e.g. electronic band gap), often leveraging more abundant, low-cost, low-fidelity data.

Related concepts: related to weak supervised learning (using noisy labels) in computer vision, multi-fidelity learning, however, focuses on integrating data from multiple supervised sources. Finally, training strategies such as transfer learning or stacking are particular implementations of multi-fidelity learning.

Active Learning is a machine learning paradigm where the algorithm actively selects the data it learns from. The model starts with a small labeled dataset and iteratively requests the labeling of additional data points by an oracle, focusing on the most informative examples. This approach is particularly useful when labeled data is scarce or expensive to obtain.

Related concepts: active learning is closely associated with semi-supervised learning but is distinct in its dynamic data querying process. It also differs from reinforcement learning in that it focuses on efficient and scarce data labeling rather than learning from many actions and rewards.

Multi-Target Learning focuses on predicting multiple outcomes within one model, capturing potential correlations and interactions between targets. This approach contrasts with separate models for each outcome.

Related concepts: synonymous with multi-task or multi-output learning, this approach differs from multi-modal learning, which integrates various sensory inputs like visual and auditory data.

Joint Learning is a neural network architecture for predicting multiple outcomes with distinct output layers and a joint loss function. It uses shared layers for extracting common features, then branches into task-specific layers. The joint loss function aggregates losses from all tasks, leveraging interdependencies.

Related concepts: related to *multi-task learning* (simultaneous learning of multiple tasks) but distinct in its specific architecture and joint loss function. It differs from *transfer learning* (sequential task learning) by simultaneously learning interrelated tasks.

Ad-Hoc models are customized for specific material classes using selected features (e.g., atomic radii or bond lengths). These specialized models are tailored for narrow applications, relying on domain knowledge. They offer fine-tuned accuracy for their target material class but lack broader applicability.

Related concepts: these models are also known as handcrafted features models or feature engineered models.

1 INTRODUCTION

The periodic table. A palette of 118 elements. The building blocks of any material in the universe. Pair two elements to form binaries, and you're presented with 6,903 unique combinations. Expand to ternaries for 266,916 possibilities. Delve into quaternaries, filter out unstable compounds with simple rules, and you are left with a staggering 10^{10} *potential materials. Among all possibilities, which ones might be manufacturable, eco-friendly and room-temperature superconducting ? Today, we only have this data for 26,000 compounds - a grain of sand on an infinite beach. Experimentally testing all possibilities is a Sisyphean task. Yet, even a rough approximation could revolutionize discovery. Fortunately, with the advent of materials informatics - integrating algorithms with existing materials databases - we can quickly predict properties in a fraction of a second. Rapid screening. Tailored experimentation. Less time wasted. The challenge partially lies in designing accurate and applicable machine learning models tailored to limited datasets, the focus of this work - a complexity set to redefine materials science !*

This chapter provides a gentle introduction into the field of machine learning for solid-state materials science. It will cover the materials discovery paradigms, a short intro to learning algorithms, the concept of materials space, a brief literature overview and the main challenges and goals of this thesis. One might be interested in complementing this read by the excellent reviews by Butler *et al.* [1], Schleder *et al.* [2], Schmidt *et al.* [3] and more recently Choudhary *et al.* [4].

1.1 MOTIVATION

Materials, in their many forms and functions, are the backbone of our society. Consider, for example, the complex technology present in an everyday object such as your mobile phone. Each component, from the silicon chip to the nearly unscratchable front glass, all the way to the lightweight yet sturdy aluminum casing, shows the sophistication of materials engineering. Such functional materials are distinct from their structural counterparts. While structural materials primarily provide support (and therefore selected based on their mechanical

properties), functional materials are used for their larger set of chemical and physical properties, including electrical, magnetic, optical, thermal, piezoelectric, among others.

From the Stone Age, through the Bronze and Iron Ages, to the modern era characterized by silicon and the comprehensive utilization of the periodic table, humanity has consistently sought to develop innovative materials to surmount the challenges faced by civilization. The birth of semiconductors, often considered among the first functional materials, revolutionized electronics and led to the Information Age [5].

Since then, functional materials have been influential in various sectors, driven by technological demands. In the energy sector, they lead the advancement of renewable energy technologies where their unique properties are used to enhance efficiency and reduce costs [6, 7]. The development of thermoelectric materials has empowered more efficient energy harvesting, while advancements in optical materials have fuelled innovations in telecommunications and data storage [8]. Biomaterials, nanometrials exhibiting unique bioactivities, offer novel biomedical applications [9]. Finally, in the face of contemporary environmental challenges, the importance of sustainable and recyclable functional materials has become important. Materials promise not only technological breakthroughs but also pathways to a greener future.

More quantitatively, it is estimated that materials and process have contributed to 2/3 of computational advancements in the past 40 years, and more loosely to 20% of the progress in all areas[10]. Furthermore, functional materials not only possess far-reaching technical implications but also profound economic consequences. As per a report by Technavio, the market size of functional materials is projected to grow from 2021 to 2026 by USD 48 billion at a progressing CAGR of 9.02% [11].

Despite the significance of these materials, their design and development often prove to be a painstaking process. On average, it takes around 20 years for a material to progress from the lab to the market. Once a material is adopted, it tends not to be replaced in the short term due to the considerable cost associated with modifying production infrastructure. Thus, the early introduction of high-performance materials is a key success factor for various technological niches in need of potential new materials.

1.1.1 *The Vast Universe of Materials Combinations*

The preceding discussion underscores the central role of functional materials in our society. However, it is important to note that the materials known to us today represent only a minuscule fraction of the potential combinations. To illustrate this, consider the case of stoichiometric inorganic materials, which are formed by combining elements from the periodic table.

If we restrict ourselves to the first 103 elements of the periodic table, the combinations of two, three, and four components amount to 5,253, 176,851, and 4,421,275, respectively. If we extend this to five components, the combinations exceed a staggering 87 million. Furthermore, considering that elements can combine in different ratios and adopt multiple oxidation states, the number of four-component candidate materials surpasses 10^{12}. Even after applying chemical filters to narrow down the possibilities, such as electronegativity and charge neutrality, the potential space for four-component materials still exceeds 10^{10} combinations, and the five-component space surpasses 10^{13} combinations [12]. While it is true that most of these compounds will form competing phases, with only a part being thermodynamically stable, the sheer quantity remains vast.

This vast number of possibilities introduces a critical concept in materials science: the configurational phase space of materials, which is immense due to the intrinsic combinatorial explosion.

In contrast, popular databases such as the Inorganic Crystal Structure Database (ICSD) and the Materials Project currently contain around 200,000 and 154,000 entries, respectively, adding on average 8,000 new materials each year [13, 14]. The significant discrepancy between the number of potential and reported materials suggests that vast areas of unexplored compositional space may exist and contain stable and useful materials.

A recent, very publicized, example of this is LK-99, a potential room-temperature superconductor, with a rather simple formula and process, yet possessing entirely unknown properties. This illustrates that most materials remain unknown and yet to be discovered.

This vast "materials universe" presents both challenges and opportunities. The challenge lies in exploring this universe, which is impossible using traditional methods. The opportunity, however, lies in the potential to discover new materials that could revolutionize various fields of technology. As we continue to navigate this vast universe, we are

bound to uncover new materials that could shape the future of our society.

1.1.2 *Traditional Approaches to Materials Discovery*

In order to address these problems, the field of materials science emerged, aiming at the characterization of materials through process, structure, and properties. Over the years, it has both benefited from and shaped the four science paradigms, as shown in Figure 1.1. Initially, most knowledge was gained through experiments which empirically showed the advantages of some materials over others. This mainly consisted of a trial and error approach. A prime historical example is the first light bulb. In 1879, Thomas Edison experimented with many different materials, eventually discovering that a carbonized filament could last many hours. Since then, over 40 elements from the periodic table have been experimented with, and ultimately, tungsten was chosen as the filament for light bulbs.

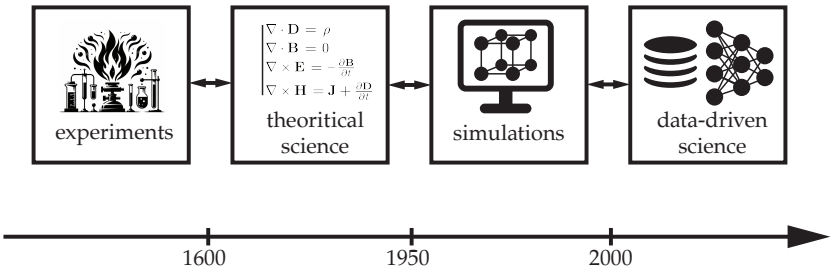


Figure 1.1: The four paradigms in materials science: empirical, theoretical, computational, and data-driven. Each paradigm both benefits from and contributes to the others. Image adapted from [15].

The emergence of a theoretical basis enabled the rational design of new compounds. Quantum Mechanics, in particular, made it possible to understand microscopic properties such as atomic bonding. However, a precise resolution of these properties for common systems is challenging, as stated by Dirac in 1929 [16]: *'The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in*

the fact that application of these laws leads to equations that are too complex to be solved.'

Major efforts were then focused on computational rather than theoretical aspects. Approximations to the Schrödinger equation, such as the Born-Oppenheimer approximation or the Hartree-Fock method, were introduced. With the rise of computational machines, a quantum computational model, known as the Density Functional Theory (DFT), was developed. DFT treats many-body systems by using functionals, i.e., functions of functions. It states that the electronic density of any system determines all ground-state properties of the system (proposed and demonstrated by Hohenber and Kohn in 1964, see Ref. [17]). Since its initial developments, DFT has evolved into an accurate predictive algorithm, capable of predicting various properties of materials, including those used in drug design, solar cells, and water splitting materials. Thus, DFT can be seen as the third paradigm of science.

1.1.3 *The Advent of Machine Learning in Materials Science*

In parallel, during the 1950s, an important subfield of artificial intelligence (AI) emerged known as machine learning (ML), inspired by developments in statistics, computer science, and neuroscience. ML comprises various algorithms capable of finding patterns and relationships in data, ranging from simple linear regressions to complex deep neural networks. By extracting knowledge from data, these algorithms can predict unknown entries or assist in decision-making processes under uncertainty, without requiring explicit programming. They learn in the sense that they will get better at a certain task by increasing the training set size without being explicitly programmed. Although the theoretical foundations of most ML models existed for over 50 years, they have gained significant popularity in recent times, especially deep learning. This can be attributed to the increase in computational power and the availability of large amounts of data. A broader introduction to ML will be provided in a subsequent section of this chapter.

The very precise reason of bringing ML-methods to the materials field is their remarkable speed, surpassing traditional methods. Conventional first-principles approaches often involve solving computationally intensive self-consistent equations, leading to slow computations that can take days to determine certain properties of solids. In contrast, ML can approximate these properties in milliseconds. The tradeoff,

however, lies in accuracy. Despite this, ML can predict materials properties before they are synthesized in the lab, thereby reducing time and resources spent on unsuccessful syntheses.

However, ML requires data to be trained on. Fortunately, the success of density functional theory (DFT) along with high-throughput methods and advancements in computational capabilities have generated large datasets containing information on numerous materials and their corresponding properties. These datasets, along with existing and continually increasing experimental data, have naturally led to the development of materials databases such as the Materials Project [14], OQMD [18], ICSD [19], AFLOWLIB [20], and many others.

This convergence of materials science and machine learning has given rise to the fourth paradigm of science: big data-driven science, or materials informatics (MI). It has led to various developments and applications in different fields, including fast computation of interatomic potentials [21], screening of high-temperature superconductors [22, 23], and identifying efficient solar cell materials [24].

Initially, the field of molecular science, primarily in pharmaceutical and medicinal applications, experienced a stronger uptake in ML compared to crystalline solids. This can be attributed partially to the difficulty of representing crystals in a format suitable for ML and partly to the high applicability of molecular science in the pharmaceutical industry [1, 25]. Currently, both fields are well-established with numerous studies, but there still remains considerable room for further advancements.

1.2 THE MATERIALS DESIGN PIPELINE

The materials design pipeline is a multistep process that involves both theory and experiments to identify promising compounds. This process is crucial for navigating the vast space of materials and efficiently discovering novel compounds with desirable properties.

The first major step in the pipeline is *in silico* screening, where theoretical models are used to filter promising compounds. This begins with defining the required properties for a given application. For instance, in the context of battery electrolytes, high ion conductivity is a crucial property, and stability, often assessed by calculating the free energy, must always be considered too. Inexpensive theoretical models, often based on machine learning (ML) methods, are then applied on a large pool of materials to provide a quick initial assessment of the

required properties. Many materials in the pool do not meet the desired criteria, leading to a smaller set of promising candidates.

The next step involves further screening of this subset using more accurate but computationally expensive methods, such as high-throughput density functional theory (DFT). This can be done by leveraging existing databases or conducting new calculations. This step refines the list, leaving a smaller set of promising compounds. At this point, individual DFT calculations are often performed for the selected candidates to obtain more detailed and accurate information.

Once a limited set of candidates is identified theoretically, the materials must undergo experimental characterization. This includes determining the synthesis parameters, which is an ongoing challenge in materials science, followed by comprehensive characterization of the required properties. Many compounds may fail this experimental step, further narrowing down the selection.

A material that successfully navigates this combined theoretical and experimental screening process can be considered a promising newly discovered material. However, it is essential to note that even at this stage, much more experimental analysis is required before a material can potentially be brought to the market.

In the context of this thesis, it is important to remember that property prediction models play a pivotal role as the initial step in the screening pipeline. Any improvement in their performance is of great significance. Keeping false positives low is crucial to prevent subsequent overwhelming, more expensive screening steps, and minimizing false negatives ensures that promising compounds are not overlooked. In essence, enhancements in the accuracy and efficiency of property prediction models contribute substantially to the overall effectiveness of the materials design pipeline.

1.3 Typical applications of machine learning in materials science

Both materials science and machine learning are both vast fields with diverse applications. Before exploring property prediction models for solids, it is worthwhile to provide a general overview of typical applications. The following categories, which may overlap and lack strict boundaries, can be identified:

Property prediction This forms the central focus of the present work. The objective is to create models capable of predicting specific materials

properties solely based on the crystal structure or composition. These models offer significantly faster execution compared to first-principle simulations or experimental measurements. For a comprehensive understanding of the latest developments, refer to Section 1.5.

ML potentials Machine learning models are designed to predict forces acting on individual atoms within a structure, thereby modelling force fields. ML potentials combine the accuracy of *ab-initio* methods with the speed of classical force fields [26]. Although this category shares similarities with the previous one, it stands out due to its distinct methodology.

Interpretable Learning: A common critique of machine learning is its perceived lack of interpretability. However, there are several techniques within this field that help us understand how materials structures influence their performance. These techniques include unsupervised machine learning (e.g., PCA) as demonstrated in the work of Helfrecht *et al.*, where carbon structures are projected into a low-dimensional space, and clusters are formed based on bond type and geometry (sp, sp2, and sp3) [27]. Other options include interpretable models that utilize meaningful features and low-dimensional models that employ simple algebra, such as SISSO [28].

Active Learning: Active learning is used for exploring a complex materials space with limited or no existing data. The idea is to use machine learning to iteratively sample novel experimental candidates. The idea behind it is to use machine learning to iteratively sample novel experimental candidates. This symbiotic approach significantly reduces the number of experiments needed to achieve a desired performance threshold compared to random search.

5. Inverse Design: In this approach, the focus shifts to solving the property prediction problem in reverse. Instead of predicting properties based on structures, the algorithm aims to *generate* novel compounds that do not exist in the original training set, given a desired property or an optimal combination of properties. However, this is a highly challenging task as materials structures must adhere to numerous rules to be valid.

1.4 INTRODUCTION TO LEARNING ALGORITHMS

In this section, we will delve into the fundamentals of supervised machine learning. We will introduce crucial concepts such as learning, features, testing procedures, and overfitting. It is worth noting that

other types of learning also exist, such as unsupervised learning, which addresses the task of identifying patterns in unlabelled data (e.g., PCA). We will briefly explore unsupervised learning in Chapter 3.

1.4.1 *Learning*

In *supervised* learning, one wants to map a set $\mathbf{X}$ to a *target* y, given some unknown relation $y = f(\mathbf{X})$. The set $\mathbf{X}$ is called the feature space while an element x from it is commonly called *feature*, *descriptor* or simply input. In other words, the idea is to find some approximate function $\hat{f}$, such that the difference between the predicted value $\hat{y} = \hat{f}(\mathbf{X})$ and the true value $y = f(\mathbf{X})$ is minimal. This discrepancy between y and $\hat{y}$ is commonly assessed by a *loss function* $J(y, \hat{y})$. If y is discrete, this is called a *classification* problem, and if y is continuous this is called a *regression* problem.

In order to do so, many samples (x, y) are given, and are known as the *training set*. Typically, a supervised ML-algorithm will have many degrees of freedom (i.e. *weights*) that will be iteratively tuned in order to minimize the loss function J. Parameters that are not optimized during training, but rather fixed *a priori* are called *hyperparameters*. Many different algorithms exist that try to model a function f. It can be as simple as a linear regression,

$$\hat{y} = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + ...\alpha_n x_n \tag{1.1}$$

where x_i are the scalar features and α_i the set of weights to be found by minimizing the loss function. More complex algorithms exist, such as the neural network explained in a further section.

1.4.2 *Validation and Testing*

Once a model is trained, it is commonly validated on a *validation* set, consisting of unknown samples (x, y), i.e. they were excluded from the training phase. This enables one to find out how well a model performs on new data, given a set of hyperparameters. This is repeated many times to find out which hyperparemeters are best.

Once the best hyperparameters have been identified, it is essential to assess the generalization performance. The validation error tends to be biased towards the best model. To obtain an unbiased estimate of the final performance, an independent *test* set is used, which remains

unseen during both training and validation. This approach enables one to provide an estimate of how well the final model can be expected to perform.

A typical validation and testing strategy is the *k-fold cross-validation*. This involves splitting the data into k folds (i.e., subgroups), training the model on $k-1$ folds, and then evaluating the model on the remaining single fold. This process is repeated k times, with each fold serving as the validation or test set in one iteration. Finally, the mean error is calculated over all folds. Compared to a single validation set, the k-fold cross-validation offers the advantage of reducing the variance in the estimate. A schematic representation of a 10-fold cross-validation is shown in Figure 1.2. Additionally, one can combine the k-fold strategy for both validation and testing, which is referred to as *nested cross-validation*.

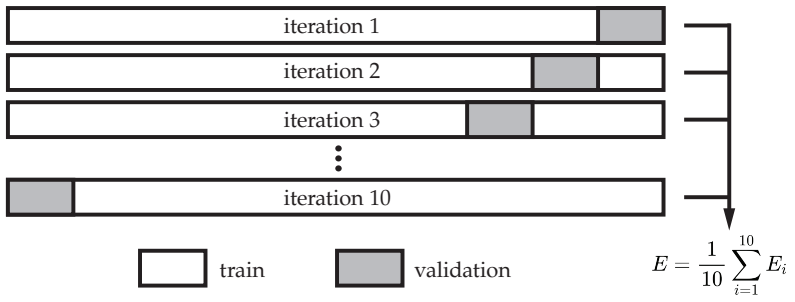


Figure 1.2: Schematic illustration of a 10-fold cross-validation strategy. A model is iteratively trained on the 9 folds while tested on the remaining fold. The final error is the mean over the 10 folds.

1.4.3 *On Training Size, Model Complexity and the Bias-Variance Tradeoff*

According to the 'No Free Lunch Theorems' [29], no ML algorithm is universally superior. Therefore, when building models, it is essential to take a case-by-case approach. Numerous algorithms based on different methods and complexities exist, some of which will be presented in the next section. When choosing or constructing a model, it's crucial to consider its complexity in relation to the dataset size. A common way to estimate complexity is by counting the number of parameters (i.e., weights). Simpler models like linear regression, LASSO, or ridge regression typically have a few tens to a hundred parameters to

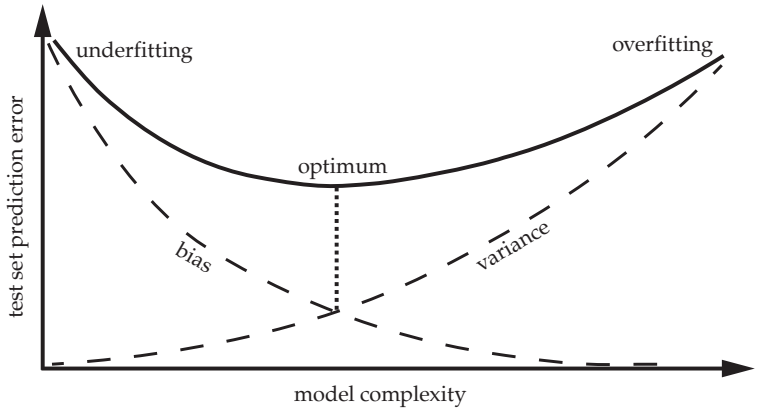


Figure 1.3: Bias-variance tradeoff. Depending on the model complexity and training size a trade-off between bias and variance is found on the prediction error given by the test set.

fit. In contrast, deep convolutional neural networks can have tens of thousands of parameters or even more.

This model complexity significantly impacts the test error, which reflects a model's ability to generalize to new data. Figure 1.3 illustrates the bias-variance trade-off. Very simple algorithms (or when few features are present) might struggle to learn non-trivial relations and tend to have a biased test error, known as underfitting. On the other hand, highly complex models with many input features can capture every detail in the input data, including noise, resulting in a very low training error. However, they often generalize poorly and suffer from high variance, known as overfitting (the model could become a mere databank that memorizes the samples). These two extreme cases are depicted in Figure 1.3. Ideally, one aims to strike a balance in the middle of this trade-off, close to the optimal test error. In practice, this can be achieved by relying on regularization (e.g., drop-out) of more complex (deep) models.

1.4.4 *Learning Algorithms*

This section does not aim to provide an exhaustive list of all available models but rather intends to give the reader two brief examples of regression algorithms. The Random Forest (RF) and the feed-forward

artificial neural network (ANN) are chosen as they are often popular choices.

Random Forest

The random forest is a simple model to understand and is often used in the literature, especially in the materials science field, due to its simplicity. It is non-parametric and requires very few optimizations. Random Forests can serve as both classification and regression algorithms.

The RF is an ensemble technique where several decision trees are trained and their predictions are averaged. A decision tree partitions the data space at each node into two subsets. Each node represents a question about the features that eliminates some possible targets. This process is repeated until only one possible target remains, referred to as the leaf of the tree, which forms the predicted value.[1]

Various implementations exist for the algorithm that partitions the space, but they are generally based on reducing the information entropy of the subsets. For more information, see Refs. [30] and [31]. An example of a decision tree applied to materials property prediction is given in the work of Oliynyk *et al.* [32], which will be detailed later in this chapter.

The problem with decision trees is that they often suffer from overfitting. To address this, the Random Forest algorithm trains a multitude of smaller decision trees, each on a random training subset (or bootstrap sample) and a random feature subset, hence the name *Random* Forest. In this sense, it generates a bunch of weak classifiers that, when taken together, increase the validation accuracy

Artificial Neural Networks

Artificial Neural Networks (ANN) were initially inspired by the human brain, particularly by the neuron cell and the synapse. They are based on the concept of a 'neuron', which takes a vector as input, performs a linear combination on it, followed by the application of a non-linear function called the activation function. These neurons are then organized in layers, where the output from one layer becomes the input of the next layer. This organization forms a directed weighted graph. A schematic illustration of an ANN is provided in Figure 1.4.

1 It is possible to have multiple samples at the leaves, and in this case, the mean is taken as the output value.

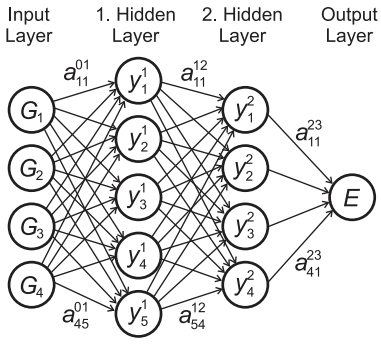


Figure 1.4: Schematic illustration of an ANN consisting of 2 hidden layers. The $\{G_i\}$ forms the input vector, while E forms the output value. The fitting parameters a_{ij}^{kl} are represented by the arrows. For clarity, the bias nodes are not drawn. Figure taken from Ref. [21].

The following notations are introduced: the weight a_{ij}^{kl} connects neuron i in layer k with neuron j in layer l. The input layer corresponds to the superscript 0. Additionally, each neuron i in layer j has a bias term b_j^i.

Any ANN can be mathematically represented as:

$$y_i^j = f_i^j \left(b_i^j + \sum_k a_{ki}^{j-1,j} \cdot y_k^{j-1} \right) \tag{1.2}$$

where y_i^j is the output of neuron i in layer j, and $f_i^j(\,\cdot\,)$ is a non-linear activation function. It is thanks to this non-linearity that arbitrary functions can be fitted. This formula clearly shows how the different layers are interconnected.

To choose the values for the weights a_{ij}^{kl} and the biases b_j^i, a loss function is optimized with respect to these weights. For example, the mean absolute error loss function is written as:

$$J = \frac{1}{N} \sum_{i=1}^{n} \|\hat{y}_i - y_i\| \tag{1.3}$$

where N is the number of training samples, $\hat{y}_i$ is the predicted value for sample i, and y_i is the actual value for this sample. To optimize this function, a gradient descent approach is typically followed, where each weight a_{ij}^{kl} is iteratively updated at time $t+1$ as:

$$a_{ij}^{kl}(t+1) = a_{ij}^{kl}(t) - \alpha \frac{\partial J}{\partial a_{ij}^{kl}} \tag{1.4}$$

where α is called the *learning rate*. The weights are usually initialized by sampling from a Gaussian distribution. It can be shown that this equation corresponds to the derivative of the error with respect to the output fed into the neural network in the opposite direction, extracted at the corresponding weights (the last layer's weights are replaced by the actual value of the corresponding neurons). This can be easily proven by the chain rule and is commonly referred to as *backpropagation*.

Another option is to use a mini-batch stochastic gradient descent approach, where samples are passed through the network in several subgroups, called batches, one after another, instead of passing all training samples at once. The ANN is sequentially optimized following Eq. 1.4 on each batch. The term "epoch" is used to denote the number of times all batches have been passed through the network. The advantage of this technique is that it requires less memory and, thanks to its stochastic nature, it often improves convergence speed for non-convex optimization problems [33].

Finally, ANNs differ mainly in their architecture, i.e., the number of layers and neurons, activation functions, optimizers, loss functions, regularization, among other factors. Other types of layers also exist, such as convolutional layers, which form the basis of *deep* neural networks. More information about ANNs and deep learning can be found in the excellent review given by LeCun *et al.* in Ref. [34].

1.5 PROPERTY PREDICTION MODELS: A BRIEF LITERATURE STUDY

Tracing the early origins of the field is challenging due to the gradual development of techniques. Nevertheless, a notably early paper concerns the work of Kiselyova *et al.*, published in 1998 [35]. They applied early ML methods to compound formation ability and crystal structure prediction. However, the number of studies really started to increase significantly only since 2010. These early reports relied on a case-by-case study and are therefore referred to as "ad-hoc" models, which are detailed below. The development of more generalized models, which target any property and rely on sophisticated methods, has only begun more recently. Depending on their type, these models are presented below under the following categories: *matminer*, *graph*-based, and *transformer*-based models.

1.5.1 *"Ad-hoc" Models*

Ad-hoc methods are machine learning algorithms that have a specific purpose in mind for a particular type of crystal structure, e.g., predicting the band gap of a certain class of materials. Typically, handcrafted descriptors are tailored to suit the physics and are the major focus, while common, simple-to-use ML models are chosen. Learning is commonly performed on experimental data or on a theoretically computed dataset.

There are two reasons why these ad-hoc methods were initially so popular. First, the difficulty of representing crystalline solids in an effective way that is easily understandable by the statistical learning procedure. Developing flexible and transferable representations is one of the most important research areas in machine learning for crystalline solids [1]. Secondly, due to the limited number of samples for many problems, much better performance is achieved by restricting the analysis to a particular structure, which is therefore inherently built into the model.

Finally, it is important to note that most published ML works can be mainly differentiated and analyzed by three key contributions: descriptors, ML model, and dataset. The latter often restricts the generalization space and thus the extent of applicability to new materials.

A typical "ad-hoc" example concerns the identification of Heusler compounds, which are intermetallics exhibiting diverse physical properties attractive for applications in thermoelectric and spintronic materials. They are difficult if not impossible to detect by standard diffraction techniques. Therefore, Oliynyk *et al.* proposed a machine learning model that identifies Heusler compounds of the type AB_2C [32]. A random forest is used as the ML model, while various compound-only descriptors are used, involving the group number of element B, the total number of p-electrons, or the radius difference between species A and B. It was then trained on 1948 samples from experimental data. It showed a remarkable true positive rate of 0.94 and a false positive rate of 0.01. An example of a simplified decision tree for this classification problem is shown in figure 1.5. In particular, novel predicted candidates MRu_2Ga and RuM_2Ga were synthesized and confirmed to be Heusler compounds. Moreover, it demonstrates how fast ML-based predictions can be compared to exact calculations by taking only 45 minutes to make the full set of predictions on over 400,000 candidates, that is, less

than 0.01 seconds per compound. Finally, this model illustrates well the 'ad-hoc'-method as it is restricted to AB_2C-compounds and uses a well-selected specific set of features known to be determining for Heusler compounds.

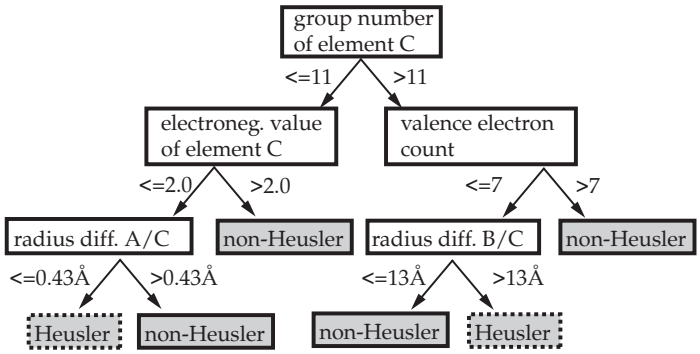


Figure 1.5: Illustration of a single decision tree, trained only on a fraction of the descriptors, for the prediction of Heusler structures for candidates of the type AB_2C, as presented in the work of Oliynyk *et al.* [32].

Another example concerns formation energies, where Rupp *et al.* proposed a framework to predict atomization energies for small molecules [36]. Any molecule is first transformed into a Coulomb matrix (see Eqs. (A.10) and (A.11) for an exact definition), which represents the Coulomb interaction between species and is therefore both a spatial and elemental representation. This matrix is flattened by computing the corresponding eigenvalues, which form the feature vector. From this, the distance $d(\mathbf{M}, \mathbf{M}')$ between two molecules is defined as the Euclidean norm of the difference between both feature vectors: $d(\mathbf{M}, \mathbf{M}') = \sqrt{\sum_I \|\epsilon_I - \epsilon'_I\|^2}$, where ϵ represents the previously mentioned eigenvalues in order of decreasing absolute value. For matrices of different dimensionality, the smallest vector is extended by zeros. Finally, atomization energies are predicted as a sum over weighted Gaussians:

$$E^{est}(\mathbf{M}) = \sum_{i=1}^{N} \alpha_i \exp\left[-\frac{1}{2\sigma^2} d(\mathbf{M}, \mathbf{M}_i)^2\right] \tag{1.5}$$

where i runs over all molecules $\mathbf{M}_i$ in the training set. The parameter σ is fixed a priori (and is thus a hyperparameter), while regression coefficients $\{\alpha_i\}$ are set to minimize the squared training error in addition

to L^2-regularization. This is nothing but a kernel ridge regression with a Gaussian kernel. Regarding accuracy, when training on more than 1000 molecules, the ML model becomes competitive with mean-field electronic structure theory, at a fraction of the computational cost. It is remarkable to see how this ML model can be seen as an approximation to the time-independent Schrödinger equation, $H\Psi = E\Psi$, where the Hamiltonian, defined by a set of nuclear charges $\{Z_I\}$ and atomic positions $\{\mathbf{R}_I\}$, is replaced by the Coulomb matrix. In other words, the ground-state potential energy found by solving the Schrödinger equation, $H\left(\{Z_I, \mathbf{R}_I\}\right) \overset{\Psi}{\longmapsto} E$, can be replaced by a ML-approach: $\{Z_I, \mathbf{R}_I\} \overset{\text{ML}}{\longrightarrow} E$.

This framework could be easily transferred to solid crystals by taking the unit cell as the molecule in the previous algorithm, which works remarkably well. However, this does not take into account long-range Coulomb interactions over many cells in the lattice. Therefore, extensions to periodic crystals have been proposed, such as the sine-Coulomb matrix [37].

Other examples are briefly mentionned. Hautier *et al.* used a machine learning model based on experimental data to predict possible novel ternary oxides, which are then simulated by ab-initio computations to confirm stability [38]. Saad *et al.* applied unsupervised learning to binary compounds, for example, grouping compounds into characteristically-similar peers [39]. Concerning magnetic properties, Lam Pham *et al.* developed a novel way to represent materials named "orbital-field matrix (OFM)" based on the distribution of the valence shell electrons [40]. By using the OFM with a random forest, they predicted the DFT-computed magnetic moment for lanthanide-transition metal alloys, with an MAE of 0.05 μ_B. Finally, Stanev *et al.* tackled the discovery of high T_c superconductors by proposing a model that predicts the critical temperature based on the composition alone [23]. More precisely, a random forest is used together with the Magpie elemental features on the SuperCon database [41, 42]. Then the Inorganic Crystallographic Structure Database (ICSD) was screened for new superconductor discovery. In particular, 35 non-cuprate and non-iron-based oxides were identified as candidate materials.

Further examples include models predicting melting temperatures of binary inorganic compounds [43], formation enthalpy of crystalline compounds [44, 45], crystal structure stability at certain compositions

[46–49], band gap energies of certain classes of crystals [50, 51], and mechanical properties of certain alloys [52, 53], among others.

These methods demonstrate the promise of machine learning for materials science. However, they covered only a fraction of the properties and materials classes. There was a clear lack of broadly-applicable machine learning models, such as a general band-gap energy prediction for all classes of materials. In particular, routine methods for transforming raw materials into numerical vectors were missing to exploit existing machine learning models. This led to the development of more general models and the *matminer* framework.

1.5.2 *Matminer*

The models presented in this section and upcoming are labeled 'advanced' in the sense that they are very flexible and general. They are designed with no specific purpose in mind and can therefore be trained on any given dataset. Different classes of materials or a mixture of classes can be used with the desired targets.

General-purpose methods that automatically predict materials properties based on Machine Learning techniques are based on three major ingredients: a database, a set of attributes generated for each material, and a machine learning algorithm. Fortunately, materials databases are now quite numerous, mainly built from first principles, such as the Materials Project [14] or OQMD [18], containing several thousands of materials. Machine learning models have a relatively long history, becoming increasingly popular in the last 15 years, thanks to growing databases and computational performance. This popularity has led to the development of many high-level frameworks, such as the scikit-learn python package. With these two ingredients steadily established, probabilistic materials property learning became quite straightforward. However, it required the development of the third ingredient, i.e., feature generation.

Initially, as discussed in the previous section, feature creation and engineering were a closed case-by-case job. They were property-dependent, with no open-source code or distributed across many repositories.

As a first step towards the general solution, Ward *et al.* proposed in 2016 a general-purpose machine learning framework for predicting properties of inorganic materials consisting mainly of 145 elemental property-related features [42]. This enabled one to easily featurize a

material into a fixed-length numerical vector and subsequently use an adequate predictive model. By using a broad variety of elemental features, very different properties could be successfully predicted, such as band gap energy (principally based on s, p, d, and f-electron occupancy) and formation energy (principally based on the difference of electronegativity of the constituent elements). However, the composition alone does not perfectly describe a material. It should be extended to additional features accounting for the structure and local environments, which resulted in additional features published later in Ref. [54]. Convinced that a general-purpose library representing materials characteristics could be created, Ward *et al.* came up with a python package named *matminer* in 2018 [55]. It not only offers the ability to generate a family of descriptors but also gives easy access to materials databases and ML methods. Furthermore, matminer proposes a uniform interface, despite the diversity of methods, making it easy for researchers to quickly iterate through data and features to determine which ones are suited for their application. In 2020, Dunn *et al.* proposed an automatic framework around matminer features [56] using the TPOT package [57].

Matminer will be heavily used throughout this work. In order to grasp an idea of what they are, some common descriptors are given below. For a full description of matminer features, the reader is referred to Appendix A.

ElementFraction. This feature is solely based on the composition. It is transformed into a vector V where V_i is set to the molar fraction of element i. In practice, each column represents a chemical element and is set to nonzero if it composes the considered compound. It should be noted that it creates a rather large sparse vector, as a compound typically contains only a few elements out of the approximately hundred possible elements.

ElectronegativityDiff. This feature is also composition-based. For each cation i, the following quantity is computed:

$$D_{\text{cation},i} = \chi_{\text{cation},i} - M_{\text{ions}} \tag{1.6}$$

where M_{ions} is defined as

$$M_{\text{ions}} = \sum_i a_i \cdot \chi_{\text{anion},i} \tag{1.7}$$

where χ is the electronegativity, and a_i is the molar fraction of cation i. From the vector D_i, different statistics can be computed as final features, such as the mean and/or standard deviation.

GlobalSymmetryFeatures. Three subfeatures are generated related to the crystal symmetry. The first feature is the spacegroup number of the crystal. The second is the crystal system (cubic, tetragonal, orthorhombic, hexagonal, trigonal, triclinic, or monoclinic), where the seven possible systems are mapped to an integer (1 to 7). The last feature is a boolean representing whether the structure is centrosymmetric, i.e., whether it has an inversion symmetry.

DensityFeatures. Generates three density-related features (full structure thus required). The first is the density of the material (mass per unit volume). The second feature is the volume per atom or 'vpa' computed by dividing the unit cell volume by the number of sites present. The third and last feature is the packing fraction computed by dividing the sum of the atomic volumes (estimated by $\frac{4}{3}\pi r_i^3$, with r_i being the species' radius) by the unit cell volume.

1.5.3 *Graph Networks*

In the search for a general model, Xie and Grossman presented the Crystal Graph Convolutional Neural Network (CGCNN) in 2018 [58]. It is a deep-learning approach that extracts features directly from a raw materials representation (e.g., CIF-file). These models are very flexible and powerful; however, they are computationally expensive to learn and often require more than 10,000 samples to be effective.

The CGCNN works as follows. A crystal structure is transformed into a graph, after which a convolutional neural network is applied, as schematically depicted in figure 1.6. The structure is first given as input, providing the species and atomic positions in the unit cell. This is transformed into an undirected multigraph where atoms become vertices and bonds become edges. It is a *multi*graph as it is possible to have multiple edges between the same pair of vertices, due to the periodicity of the crystal structure. The next step is to assign a vector $\boldsymbol{v}_i$ to each vertex i (atom) and a vector $\boldsymbol{u}(i,j)_k$ to each edge (bond), corresponding to the k-th bond between atom i and atom j. A choice for $\boldsymbol{v}$ could be the vector formed by zeros everywhere except at the atomic number, where a 1 is placed. A similar choice can be made for $\boldsymbol{u}$, but based on the bond length. These first two steps are represented in figure 1.6 (a).

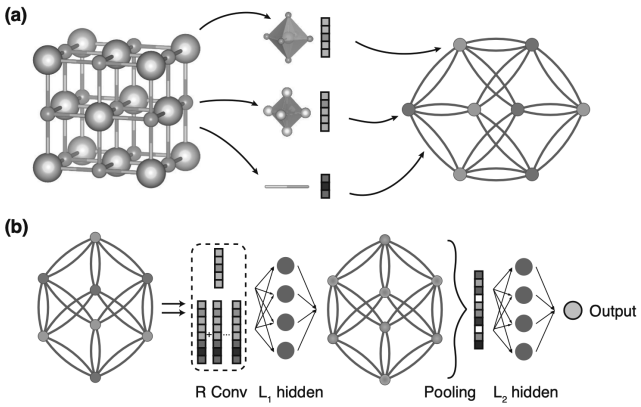


Figure 1.6: Illustration of the crystal graph convolutional neural network (CGCNN). (a) Construction of the crystal graph. Crystals are converted to graphs with nodes representing atoms in the unit cell and edges representing atom connections. Nodes and edges are characterized by vectors corresponding to the atoms and bonds in the crystal, respectively. (b) Structure of the convolutional neural network on top of the crystal graph. R convolutional layers and L_1 hidden layers are built on top of each node, resulting in a new graph with each node representing the local environment of each atom. After pooling, a vector representing the entire crystal is connected to L_2 hidden layers, followed by the output layer to provide the prediction. Taken from Ref. [58]

After this, a convolution is performed on each vertex by taking its chemical environment into account. More precisely, each vector $\boldsymbol{v}_i$ is updated as follows:

$$\boldsymbol{v}_i^{(t+1)} = \boldsymbol{v}_i^{(t)} + \sum_{j,k} \sigma\left(\boldsymbol{z}_{(i,j)_k}^{(t)} \boldsymbol{W}_f^{(t)} + \boldsymbol{b}_f^{(t)}\right) \odot g\left(\boldsymbol{z}_{(i,j)_k}^{(t)} \boldsymbol{W}_s^{(t)} + \boldsymbol{b}_s^{(t)}\right) \quad (1.8)$$

with $\boldsymbol{z}_{(i,j)_k}^{(t)} = \boldsymbol{v}_i^{(t)} \oplus \boldsymbol{v}_j^{(t)} \oplus \boldsymbol{u}_{(i,j)_k}$.
Here $\oplus$ represents concatenation, $\odot$ represents element-wise multiplication, $g(\,\cdot\,)$ is any non-linear activation function while $\sigma(\,\cdot\,)$ represents a sigmoid function. The $\boldsymbol{W}$-matrices are learnable weights. Intuitively, $\boldsymbol{W}_f$ and $\boldsymbol{W}_s$ can be seen as two convolution kernels that take into account the environment of atom i, while $g(\,\cdot\,)$ and $\sigma(\,\cdot\,)$ introduce some non-linearities between convolution layers. In fine, one obtains for each atom i (or vertex) a new representation characterized by a series of vectors $\boldsymbol{v}_i^1, \boldsymbol{v}_i^2, ..., \boldsymbol{v}_i^R$, which is represented by the second graph in figure 1.6 (b).

A pooling layer is then used to produce a single overall feature vector $\boldsymbol{v}_c$ for the crystal, which can be represented by a pooling function,

$$\boldsymbol{v}_c = \text{Pool}\left(\boldsymbol{v}_0^{(0)}, \boldsymbol{v}_1^{(0)}, \ldots, \boldsymbol{v}_N^{(0)}, \ldots, \boldsymbol{v}_N^{(R)}\right), \quad (1.9)$$

which satisfies permutational invariance with respect to atom indexing and size invariance with respect to the unit cell choice. The original work uses a normalized element-wise summation function for simplicity, and seems to work well.

Finally, from $\boldsymbol{v}_c$, two hidden fully connected layers predict the output target. All weights (including those for the convolution) are trained to minimize the prediction error (i.e., the difference between the predicted value and the 'true' labeled value) using backpropagation and stochastic gradient descent (SGD).

The CGCNN was trained on 8 different DFT calculated properties, with approximately 10^4 crystals per property having various structure types and compositions. The mean absolute error (MAE) of the CGCNN predictions with respect to the DFT values was in the same order as one can expect from DFT compared to experimental data, which demonstrates the high accuracy of the model. However, it systematically requires huge datasets of at least 10^4 samples.

Furthermore, the model was shown to be interpretable in some cases. By replacing the last hidden layer with a linear function, one can easily

find the contribution of each atom (and its environment, represented by v_c) to the final outputted value. This enables, for example, the determination of the mean contribution of each elemental species to the energy above hull of perovskites, as illustrated in figure 3 of the original paper. This information can be utilized to discover empirical rules for materials design, and in general, the chemical insights gained from CGCNN can significantly reduce the search space for high throughput screening.

The CGCNN has played a central role in the development of many other models based on the architecture. Chen *et al.* extended the idea to include state attributes and embeddings with their model MEGNet [59], providing a general, composable framework for quantitative structure-state-property relationship prediction in materials. Park and Wolverton proposed the improved CGCNN (iCGCNN), which includes Voronoi tessellations, three-body interactions, and improved bond attributes [60]. Choudhary *et al.* proposed ALIGNN, a graph neural network including bond angles [61]. Velickovic added the attention mechanism (GAT) [62], and Gasteiger *et al.* embedded messages passed between nodes (DimeNet) [63].

In conclusion, it is important to note that the previously enumerated models and groups are far from exhaustive, but they represent the main categories in solid-state property prediction. Other important models include CrabNet [64], which uses a transformer-based architecture on compositions, SchNet [65], using continuous convolutions, and euclidean neural networks [66].

1.6 CHALLENGES

After reviewing many concepts and models in the field, it is now worth taking a step back and looking at the global challenges. Materials informatics presents numerous challenges:

First and foremost, as mentioned earlier, the materials space is very sparse, with most datasets being relatively small. For instance, only 80 crystals have had their band gap computed within *GW* [67], 101 compounds have data on lattice thermal conductivity [68], and vibrational properties have been studied for 1245 materials [69].

The small size of datasets has two implications. First, due to the limited number of data points, handling such scarcity requires the development of specific techniques (e.g., by relying on meaningful features instead of embeddings).

Second, making predictions in regions of the materials space with limited data becomes challenging. Extrapolating to different crystal systems or compositions requires careful consideration and may lead to less reliable results due to the lack of representative data points.

Another challenge, closely related to the extrapolation problem mentioned earlier, arises from the intrinsic bias in most materials datasets. These datasets often favor smaller systems, and certain elements (e.g., oxygen) are overrepresented due to the ease of computation and historical biases towards materials like oxides.

Furthermore, besides the issue of data quantity, data quality poses another problem. Data can come from various sources and may be simulated or measured, leading to inherent differences in accuracy and reliability. Moreover, within each type of data, different measurement protocols and simulation methods may exist, making the data multi-fidelity in nature. Integrating these diverse data types and levels of accuracy presents a challenge in ensuring consistency and reliability in predictions.

Apart from the quantity and quality of data, various materials properties exist, and they may or may not be closely related. Datasets may contain information on specific properties for some materials while lacking data on other relevant properties for those materials. A major challenge lies in unifying this scattered knowledge to form a so-called materials genome, consolidating all relevant information about materials to facilitate better understanding and prediction.

In summary, applying machine learning to materials data involves dealing with sparse datasets, addressing biases, handling data of varying quality and types, and integrating various properties. Overcoming these challenges requires innovative techniques, careful consideration of data limitations, and the development of robust models capable of handling uncertainty to make reliable predictions in the face of sparse and heterogeneous materials information.

1.7 THESIS OVERVIEW

The aim of this thesis is to develop general methods to address diverse learning challenges in predicting properties of solid materials, with particular emphasis on effectively handling small datasets. Each chapter will focus on a different problem, providing an in-depth exploration of the motivation, methodology, and practical applications.

Question 1: *What approach can lead to a robust general learning framework that is well-suited for limited datasets and applicable to any property?* An answer to this question will be developed in Chapter 2, laying the foundation for our work. We propose a model based on feature selection and joint learning that can effectively handle limited datasets. We will demonstrate its applicability to various properties, such as experimental band gaps and vibrational thermodynamics.

Question 2: *How can the reliability of machine learning predictions be assessed?* This question of trust is crucial for the practical usage of machine learning methods. It introduces the concept of uncertainty, the main topic of Chapter 3. We will study the problem of bias and imbalance in the materials space and propose a quantitative method to predict uncertainty, employing an ensemble method.

Question 3: *How can small datasets be augmented by incorporating lower quality data, e.g., simulations?* This addresses a practical problem frequently encountered in materials science. Properties often come in different degrees of quality, ranging from precise experimental measurements to lower-quality simulations. Chapter 4 will investigate various techniques that combine multiple quality sources for the same property. In particular, focusing on the electronic band gap, we aim to achieve the lowest error by taking advantage of all available experimental measurements and density-functional theory calculations.

2 MODNET, A GENERAL SUPERVISED MODEL FOR LIMITED DATASETS

This chapter introduces the Material Optimal Descriptor Network (MODNet), a supervised machine learning framework for predicting materials properties based on composition or crystal structure. MODNet is well-suited for handling limited datasets and enables joint learning for multiple property predictions. The chapter begins by presenting the motivations and theoretical background behind the model. It then provides a detailed methodology, including the feature selection algorithm, joint learning, and genetic hyperoptimization algorithm. The performance of MODNet is assessed using the matbench test suite, and a practical example is given for vibrational thermodynamics. Additionally, the chapter discusses feature selection and interpretation, highlighting MODNet's ability to extract meaningful insights from materials data.

This chapter is a detailed elaboration of the concepts and findings introduced in my publication, as listed in reference [P1].

2.1 MOTIVATIONS

MODNet aims to solve the following simple supervised learning problem:

$$\hat{\boldsymbol{y}} = \hat{f}(\boldsymbol{x}), \tag{2.1}$$

where $\hat{f}$ is a neural network, $\boldsymbol{x}$ is a vector representation of a material $\boldsymbol{m}$, and $\hat{\boldsymbol{y}}$ represents the properties of interest.

MODNet bridges the gap between traditional "ad-hoc" methods and more advanced graph-based models. It maintains the accuracy of the former while retaining the generality of the latter. This is achieved through four key aspects. First, it uses chemical and geometrical descriptors to transform $\boldsymbol{m}$ into $\boldsymbol{x}$. These descriptors provide more information than a raw graph representation, particularly for small datasets where they can exploit existing chemical knowledge, aiding the

learning process. Second, it reduces the dimensionality of x through feature selection. The importance of tackling the curse of dimensionality [70] becomes greater with smaller datasets. Third, MODNet introduces the option to train in joint-learning, which helps mitigate the lack of data for a given target. By combining different properties for the same compound, the model can form a more general representation in the hidden layers, more efficiently utilizing the information provided to the learning algorithm and improving model performance for a given dataset size. Furthermore, this makes it easy to predict more complex objects such as temperature-, pressure-, or energy-dependent functions (such as the density of states).

Finally, a neural network is chosen for $\hat{f}$ and heavily optimized with a genetic algorithm. This approach is flexible, powerful, and has shown excellent performance on small datasets when well-tuned.

2.2 THE MODNET MODEL

The model is illustrated in Figure 2.1 and is referred to as the Material Optimal Descriptor Network (MODNet). The concepts of feature selection and the joint-learning architecture will now be elaborated further.

The raw structure is first transformed into a machine-understandable representation. This representation should fulfill several constraints, such as rotational, translational, and permutational invariances, and must be unique. In this study, the structure is represented by a list of descriptors based on physical, chemical, and geometrical properties. Unlike more flexible graph representations, these features contain preprocessed knowledge driven by physical and chemical intuition. This allows the machine to find their connection to the target more directly, which is crucial when dealing with limited datasets. While general graph-based frameworks could potentially learn these physical and chemical representations automatically, they would require much larger amounts of data, which are often not available. In other words, part of the learning is already done before training the neural network.

To achieve this, we rely on a large number of features previously published in the literature, which have been centralized into the matminer project [55] and are further detailed in Appendix A. These features cover a broad spectrum of physical, chemical, and geometrical properties, such as elemental (e.g., atomic mass or electronegativity), structural (e.g., space group), and site-related (i.e., local environments) features.

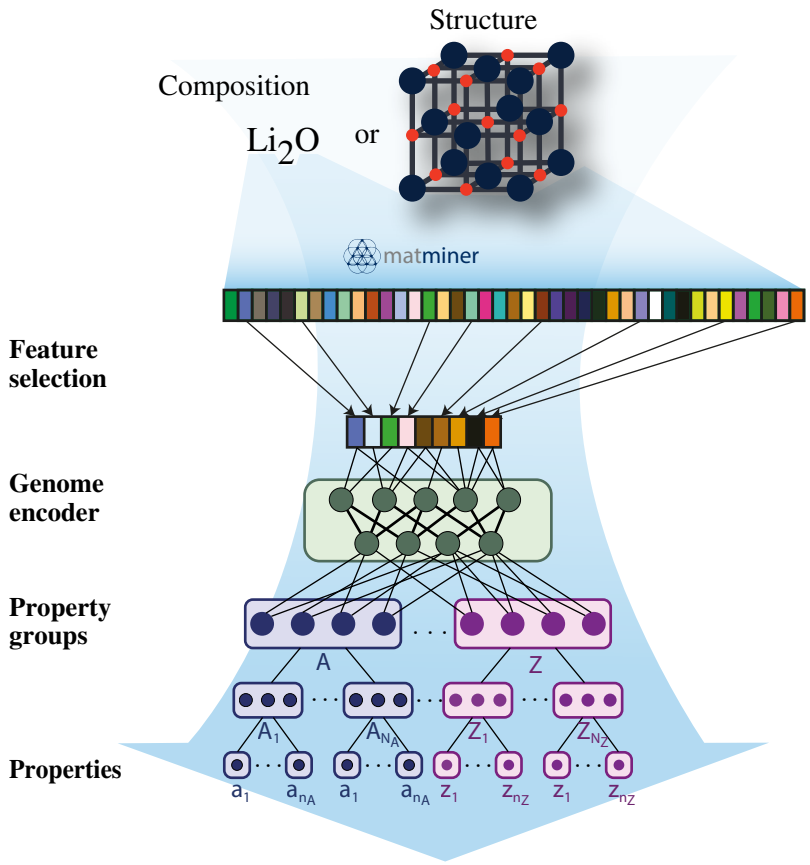


Figure 2.1: Schematic of the MODNet model. The feature selection on matminer is followed by a hierarchical tree-like neural network. Various properties $A_1, \ldots, A_{N_A}, \ldots, Z_1, \ldots, Z_{N_Z}$ (e.g., Young's modulus, refractive index, ...) are gathered in groups from A to Z of similar nature (e.g., mechanical, optical, ...). Each of these may depend on a parameter (e.g., temperature, pressure, ...): $A(a), \ldots, Z(z)$. The properties are available for various values of the parameters $a_1, \ldots, a_{n_A}, \ldots, z_1, \ldots, z_{n_Z}$. The first green block of the neural network encodes a material in an appropriate all-round vector, while subsequent blocks decode and re-encode this representation in a more target-specific nature.

We believe that they are diverse and descriptive enough to predict any property with excellent accuracy.

2.2.1 *Feature Selection*

Importantly, a subset of relevant features is then selected to reduce redundancy and, therefore, limit the curse of dimensionality [71], a phenomenon that inhibits generalization accuracy. Previous works have shown the benefits of feature selection when learning on materials properties [28, 44].

To build an effective selection procedure, we first need a metric to compute the relationship between features. Specifically, given two random variables X and Y, we are interested in a score that measures how the knowledge of Y could reduce our uncertainty about the value of X. One well-known statistical score is the Pearson correlation, defined as follows:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.2}$$

where $Cov(X,Y)$ is the covariance between X and Y, and σ the standard deviation. Unfortunately the Pearson correlation has some displeasing properties for our goal.

First, the correlation is a *linear* measure, which means that an absence of correlation does not mean absence of relation. As an example, $y = x^2$, has a zero correlation. Second, it is parametric (makes the hypothesis of a linear model) and is moreover very sensitive to outliers.

To go beyond these limitations, the more general concept of mutual information (MI) is used, which measures the decrease in *information entropy* when the second random variable is known. Formally, the information entropy H of a continuous random variable X is defined as:

$$H(X) = -\int P_X(x) log(P_X(x)) dx \tag{2.3}$$

where $P_X(x)$ is the probability density function (pdf) of X. From this the mutual information $\mathrm{MI}(X,Y)$ between two continuous random variables X and Y can be defined as,

$$\mathrm{MI}(X,Y) = H(X) - H(X|Y). \tag{2.4}$$

It is always non negative and smaller than $\min(H(X), H(Y))$. When two random variables are independent the MI is zero.

The difficulty lies in estimating the mutual information (MI) from a sample representing a random variable, which requires estimating the probability density function (pdf) using non-parametric methods.

In the present work, a K-nearest neighbors (KNN)-based method is used for entropy estimation, as explained in Ref. [72], and implemented by the scikit-learn library. From this, the information entropy can be computed as follows:

$$\mathrm{H}(X) = \mathrm{MI}(X, X). \tag{2.5}$$

An additional difficulty arises from the fact that the MI is not upper bounded. We therefore introduce the normalized mutual information (NMI) defined as,

$$\mathrm{NMI}(X, Y) = \frac{\mathrm{MI}(X, Y)}{(H(X) + H(Y))/2} \tag{2.6}$$

The Pearson correlation is bounded between 0 and 1. It is important to note that both the MI and NMI are symmetric measures. However, in practice, due to non-parametric estimation, the obtained results are often not perfectly symmetric, though they are typically close to symmetry (which can sometimes cause the NMI to go beyond 1). To limit these impracticalities, averages are taken over pairs of corresponding NMI's by considering the symmetric part of the NMI matrix.

Given a set of features $\mathcal{F}$, the selection process for extracting the subset $\mathcal{F}_S$ proceeds as follows: When $\mathcal{F}_S$ is empty, the first chosen feature will be the one having the highest NMI with the target variable y. Once $\mathcal{F}_S$ is non-empty, the next chosen feature f is selected based on its relevance and redundancy (RR) score, with the goal of maximizing this score:

$$\mathrm{RR}(f) = \frac{\mathrm{NMI}(f, y)}{\left[\max_{f_s \in \mathcal{F}_S} \left(\mathrm{NMI}(f, f_s)\right)\right]^p + c} \tag{2.7}$$

where (p, c) are two hyperparameters determining the balance between relevance and redundancy. In practice, varying these two parameters dynamically seems to work better, as redundancy is a bigger issue with a small amount of features. Practically, after some empirical testing, we decided to set $p = \max[0.1, 4.5 - n^{0.4}]$ and $c = 10^{-6} n^3$ when $\mathcal{F}_S$ includes n features, but other functions might even work better. The selection proceeds until the number of features reaches a threshold which can be fixed arbitrarily or, better, optimized such that the model error is minimized. When dealing with multiple properties, the union of relevant features over all targets is taken. Our selection process is in principle very similar to the mRMR-algorithm [73], but it goes beyond by combining both redundancy and relevance in a more flexible

way by introducing the parameters p and c. Furthermore, it is less computationally expensive than the Correlation-based Feature Selection (CFS) [74] and provides a global ranking.

2.2.2 *Joint Learning*

In contrast with what is usually done, we take advantage of learning on multiple properties simultaneously, as proposed for SISSO [75]. This could be used, for instance, to predict temperature-curves for a particular property.

In order to do so, we use the architecture presented in Figure 2.1. Here, the neural network consists of successive blocks (each composed of a succession of fully connected and batch normalization layers) that split on the different properties depending on their similarity, in a tree-like architecture. The successive layers decode and encode the representation from general (genome encoder) to very specific (individual properties). Layers closer to the input are shared by more properties and are thus optimized on a larger set of samples, imitating a virtually larger dataset. These first layers gather knowledge from multiple properties, known as joint-transfer learning [76]. This limits overfitting and slightly improves accuracy compared to single target prediction.

Taking vibrational properties as an example, the first-level block converts the features in a condensed all-round vector representing the material. Then, a second-level block transforms this representation into a more specific thermodynamic representation that is shared by many third-level predictor blocks, predicting different thermodynamic properties (specific heat, entropy, enthalpy, energy at various temperatures). A fourth-level block splits different predictors based on the actual property, but sharing different temperature predictors. Optionally, another second-level block could be built shared by mechanical third-level predictors.

2.2.3 *Genetic Hyperparameter Optimization*

For selecting the hyperparameters of MODNet, we employ either manual tuning or an automatic optimization using a Genetic Algorithm (GA). The GA starts by creating an initial population of individuals with different randomly set hyperparameters. The best individuals,

based on an external validation loss, are then propagated through mutations and crossover operations to generate further generations. Eventually, the model architecture with the lowest validation loss is selected.

The activation function, loss function, and number of layers are fixed to an exponential linear unit, mean absolute error, and 4, respectively. However, the number of neurons (ranging from 8 to 320), the number of input features (i.e., the first f features from the ranked relevance-redundancy list), learning rate (ranging from 0.001 to 0.1), batch size (ranging from 32 to 256), and input scaling (min-max or standard) are optimized through the GA.[1]

The GA introduces randomness while giving more importance to local optima. Consequently, a satisfactory set of hyperparameters is found faster and with reduced computational cost compared to standard grid- or random-search methods that were used previously. This approach results in a relative improvement of up to 12% on the matbench tasks compared to the previously used grid search [P3]. Moreover, the neural networks are always small (4 layers), resulting in fast training and prediction times.

2.3 PERFORMANCE ASSESSMENT

To investigate the predictive performance of MODNet, we first benchmark it on matbench and then present a use-case for properties originating from the Materials Project (MP) [14, 69, 78–80].

2.3.1 *Matbench*

Matbench is a pivotal initiative aimed at proper testing and comparison in the field of materials science. The diversity of approaches in this field highlights the need to establish a consistent benchmarking pipeline, drawing inspiration from successful practices in other domains, such as ImageNet for computer vision (Deng et al., 2009) [81]. Unfortunately, models are often tested on only a small number of datasets, and the validation and testing procedures are unclear or non-repeatable. This lack of standardization leads to significant variation in generalization performance for different hold-out sets, as will be demonstrated subsequently.

1 The code for MODNet and the GA can be found at `ppdebreuck/modnet` on GitHub [77].

To ensure a fair comparison between competing models, they should not only be benchmarked on the same dataset but also cross-validated on identical training and testing subsets. A recommended approach is using a 5-fold cross-validation with a fixed random seed to maintain consistency. Similarly, hyperparameter choices are frequently opaque and should instead adhere to a reproducible pipeline. This would enable researchers to better understand and replicate model performances. Moreover, testing a diverse range of tasks reveals that no model is universally superior. Instead, the performance of a given architecture depends on factors such as the amount of data, feature availability, the complexity of the target space, and the computational resources dedicated to tuning.

For the aforementioned reasons, Dunn *et al.* released the MatBench v0.1 test suite in 2020, playing a crucial role in elevating the standards of testing and comparison in the materials science field.

Data

Matbench v0.1 consists of 13 materials science-specific prediction tasks, taken from 10 datasets with size varying from 300 to 130,000. The MatBench suite covers various observables in materials science driven by different length scales, from microscopic simulations of elastic, electronic, optical and vibrational properties, to macroscopic measurements of steel yield strengths and metallic glass formation. For a detailed description of the contents and curation of each dataset, the reader is referred to the MatBench paper itself [56]. Each task provides either a composition or a structural model per sample. The target is always a single property, either continuous or a discrete class label (such as metal or non-metal). Almost all properties are therefore trained in single-target mode with MODNet, i.e., one target per task. The only exceptions are the elastic properties $\log_{10} K$ and $\log_{10} G$, where both bulk and shear modulus are provided for each sample (across two MatBench datasets); here model performance is improved by joint-learning both targets in a single model. Regression models were trained using the mean absolute error as a loss function. For classification, the output activation is set to a softmax function and the categorical cross-entropy is used as loss function. MODNet has been applied to all 13 Matbench tasks.

Training Methodology

The following sections present the results for Matbench v0.1. It is important to note that all computations were conducted and compared with the leaderboard as it stood in July 2022. However, Matbench is a dynamic benchmark that undergoes continuous updates. Our testing procedure adheres to the MatBench recommendations, wherein an outer loop of five-fold cross-validation is used to determine the generalization error (test error). Additionally, an inner five-fold validation is performed for both feature and hyperparameter selection, ensuring an unbiased generalization error. In other words, nested cross-validation is used.

All benchmarking data and the scripts used to featurize and train models for each dataset are available on GitHub at `modl-uclouvain/modnet-matbench`.

Results

Table 2.1 displays the results of MODNet with GA, along with four other state-of-the-art models (ALIGNN [61], AMME [56], CrabNet [64], and CGCNN [58]), which are currently leading in at least one of the tasks in Matbench.

The Atomistic Line Graph Neural Network (ALIGNN) is a graph convolution network that explicitly models two- and three-body interactions. It achieves this by composing two edge-gated graph convolution layers, the first applied to the atomistic line graph (representing triplet interactions) and the second applied to the atomistic bond graph (representing pair interactions) [61]. Automatminer Express (AMME) creates automated end-to-end pipelines that combine materials datasets decorated with Matminer descriptors with an AutoML search using the TPOT package [57]. This approach generates ensembles of tree-based models for making predictions and has the advantage of being fully automated, requiring minimal ML expertise to run on a new dataset. The Compositionally Restricted Attention-Based network (CrabNet) is a self-attention-based model, recently reported as a state-of-the-art model for predicting materials properties based on composition only [64]. For tasks involving the crystal structure as input, the CGCNN [58] model is also compared against. This model performs convolution and pooling operations on a graph representation of the crystal structure, as described in the introduction.

The performance of each model is evaluated using the mean absolute error (MAE) for regression tasks or the receiver operating characteristic area under the curve (ROC-AUC) for classification tasks. The best score for each task is indicated in bold. Additionally, we provide baseline metrics using (i) results obtained with a Random Forest (RF) regressor using features from the Sine Coulomb Matrix and MagPie featurization algorithms, and (ii) a Dummy model that predicts the mean of the training set for regression tasks or randomly selects a label in proportion to the distribution of the training set for classification tasks [56].

The presented results demonstrate that MODNet significantly improves the error on six tasks (steel yield strength, exp. band gap, exp. metallicity, refractive index, glass-forming ability, and 2D exfoliation energy), and achieves comparable or better performance (within 2%) than the existing leader for two tasks. However, MODNet has higher error for five tasks, all of which have larger datasets, except for the phDOS peak dataset. Notably, the Dummy algorithm is consistently outperformed by all models by a factor of two to three.

As a general trend, MODNet performs relatively well on tasks that rely solely on composition data. This success can be attributed to the use of a limited input neural network, where a latent representation is learned for the different elemental contributions, acting as a dimensionality reduction technique. This representation is more flexible than the equivalent obtained through feature engineering for tree-based approaches. Recent deep representation learning approaches may further exploit this effect, especially if transfer learning is utilized [64, 89].

2.3.2 *Vibrational Thermodynamics*

In this section, we consider the multi-property facet of MODNet. We train it on the vibrational energy, enthalpy, entropy, and specific heat at 40 different temperatures as well as the formation energy, as the latter was found to be beneficial to the overall performance. The accuracy is compared with that of MODNet on the vibrational entropy at 305K.

Data

The vibrational properties for 1,245 inorganic compounds were computed by Petretto *et al.* [69], in the harmonic approximation based on Density Functional Perturbation Theory (DFPT). This dataset contains

Target Property (Unit)	Samples	Type	MODNet	ALIGNN	AMME	CrabNet	CGCNN	RF	Dummy
Steel yield strength (MPa) [82]	312	[R]	**87.8**	—	97.5	107.3	—	103.5	229.7
Exfoliation energy (meV/atom) [83]	636	[R]	**33.2**	43.4	39.8	45.6	49.2	50.0	67.3
Freq. at last phonon PhDOS peak (cm^{-1}) [69]	1 265	[R]	34.3	**29.5**	56.2	55.1	57.8	67.6	324.0
Expt. band gap (eV) [84]	4 604	[R]	**0.325**	—	0.416	0.346	—	0.406	1.144
Refractive index [14, 85]	4 767	[R]	**0.271**	0.345	0.315	0.323	0.599	0.420	0.809
Expt. metallicity [84]	4 921	[C]	**0.968**	—	0.921	—	—	0.917	0.492
Bulk metallic glass formation [42, 86]	5 680	[C]	**0.990**	—	0.861	—	—	0.859	0.492
Shear modulus (GPa) [87]	10 987	[R]	0.073	**0.072**	0.087	0.101	0.090	0.104	0.293
Bulk modulus (GPa) [87]	10 987	[R]	**0.055**	0.057	0.065	0.076	0.071	0.082	0.290
Formation energy of Perovskite cell (eV) [88]	18 928	[R]	0.091	**0.028**	0.201	0.406	0.045	0.236	0.566
MP band gap (eV) [14]	106 113	[R]	0.220	**0.186**	0.282	0.266	0.297	0.345	1.327
MP metallicity [14]	106 113	[C]	**0.964**	0.913	0.909	—	0.952	0.899	0.501
Formation energy (eV/atom) [14]	132 752	[R]	0.045	**0.022**	0.173	0.086	0.034	0.117	1.006

Table 2.1: Matbench v0.1 results for MODNet, Automatminer Express (AMME), CrabNet, CGCNN, MEGNet, a Random Forest (RF) regressor, and a Dummy predictor. The scores are MAE for regression [R] or ROC-AUC for classification [C] tasks. The tasks are ordered by increasing number of samples in the dataset.

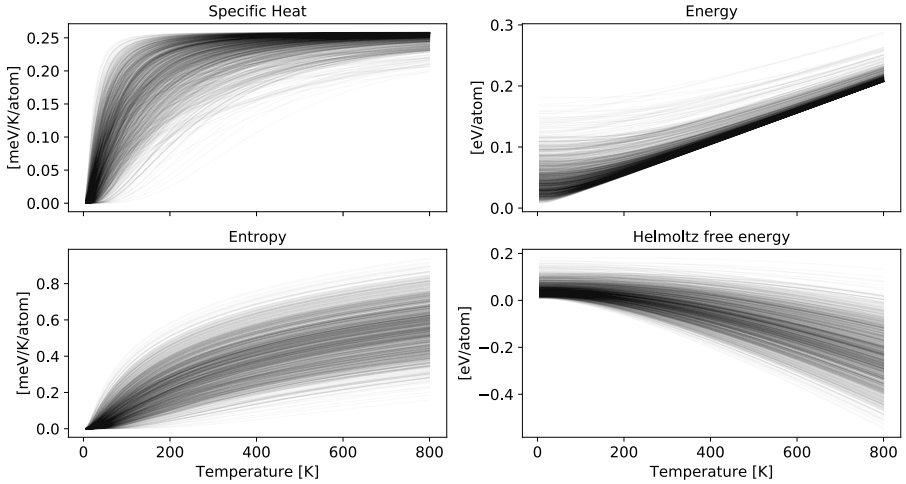


Figure 2.2: Visual representation of values encountered in the vibrational thermodynamic dataset (from Ref. [69]). Four vibrational properties from 5 to 800 K for 1245 materials are present. A wide range of values are reached, with clear differences in Debye temperatures.

the following thermodynamic properties: vibrational entropy (S), Helmholtz free energy (H), internal energy (U) and heat capacity (C_v) from 5 to 800 K in steps of 5 K. One important fact is that all four properties can be derived from the density of states (DOS) g(ω):

$$
\begin{aligned}
S &= 3nNk_B \int_0^{\omega_L} \left(\frac{\hbar\omega}{2k_BT} \coth\left(\frac{\hbar\omega}{2k_BT}\right) - \ln\left(2\sinh\frac{\hbar\omega}{2k_BT}\right)\right) g(\omega)d\omega \\
\Delta U_{\text{ph}} &= 3nN\frac{\hbar}{2} \int_0^{\omega_L} \omega \coth\left(\frac{\hbar\omega}{2k_BT}\right) g(\omega)d\omega \\
\Delta F &= 3nNk_BT \int_0^{\omega_L} \ln\left(2\sinh\frac{\hbar\omega}{2k_BT}\right) g(\omega)d\omega \\
C_v &= 3nNk_B \int_0^{\omega_L} \left(\frac{\hbar\omega}{2k_BT}\right)^2 \operatorname{csch}^2\left(\frac{\hbar\omega}{2k_BT}\right) g(\omega)d\omega
\end{aligned}
\tag{2.8}
$$

Here, k_B is the Boltzmann constant, and ω_L is the largest phonon frequency. The entropy represents the disorder (i.e., the number of possible configurations over the lattice) of the phonons.[2] The Helmholtz

2 Most solids also have an additional configurational entropy, due to the different species, which adds up to the total entropy of a compound. We only focus on the vibrational contribution in this work.

free energy[3] is of high importance when considering the stability of crystals. It is similar to the Gibbs free energy ($G = H - TS = U + PV - TS$), but considers the volume to be constant rather than the pressure. When considering the stability of crystals, it can, therefore, be seen as a close approximation to the Gibbs free energy, determining the stability at constant pressure and temperature. The specific heat represents the amount of vibrational energy to be added at constant volume to increase the temperature by 1 Kelvin.

From a machine learning perspective, the fact that Eqs. 2.8 are all based on the DOS gives a hint that they may be learned simultaneously. In other words, a common representation might be built in order to predict them.

Figure 2.2 graphically represents these four properties from 5 to 800 K for all materials contained in the dataset in meV/atom or meV/K/atom. A wide variety of values is reached, with different Debye temperatures as can be seen from the specific heat. This indicates no significant bias, giving us confidence in generalizing on unseen data.

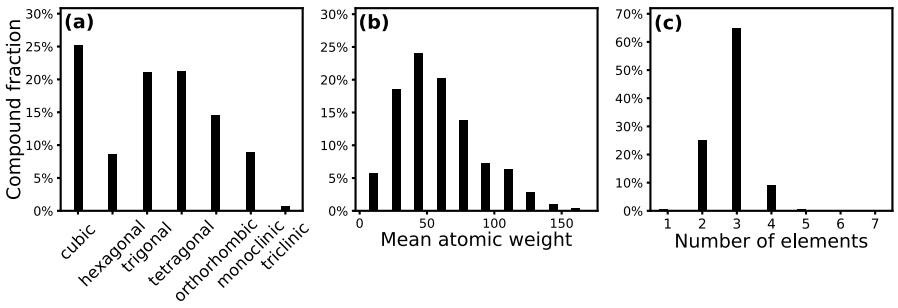


Figure 2.3: Relative distribution of (a) the crystal system, (b) mean atomic mass and (c) number of elements for the thermodynamic dataset.

Figure 2.3 contains various histograms representing the materials distribution. The dataset includes various crystalline compounds, ranging from simple mono-elemental compounds to complex semiconductors, as shown in Figure 2.3. The thermodynamic data cover 7 symmetry groups, 84 space groups, and 64 elements. Most compounds are ternary alloys, and there are few materials with more than 5 different elements. The mean atomic mass has a wide range, confirming the diversity of elements in the two datasets.

3 This quantity is, in fact, just $\Delta U_{\text{ph}} - TS$.

Training Methodology

A 10% hold-out test set is used for the vibrational thermodynamics. A separate 10-fold validation scheme is employed on the remaining data to optimize the hyperparameters, while the test set is used to obtain an unbiased generalization performance for the best hyperparameters.

It is important to note that the network is optimized on all targets together. Therefore, the different MAEs corresponding to different properties are combined to form the following metrics:

- **WMAE**: A Weighted summed MAE of the different properties such that the property-temperature curves are in the same range:

$$\begin{aligned} \text{WSMAE} &= \sum_{p \in (S, C_v, H, U)} w_p \, MAE_p, \\ (w_S, w_{Cv}, w_H, w_U) &= (0.3, 1, 0.004, 0.001) \end{aligned} \tag{2.9}$$

- **MAX WSMAE**: The WSMAE, but only taking the 5% worst predicted samples.

- **MAPE**: A mean absolute percentage error over all properties.

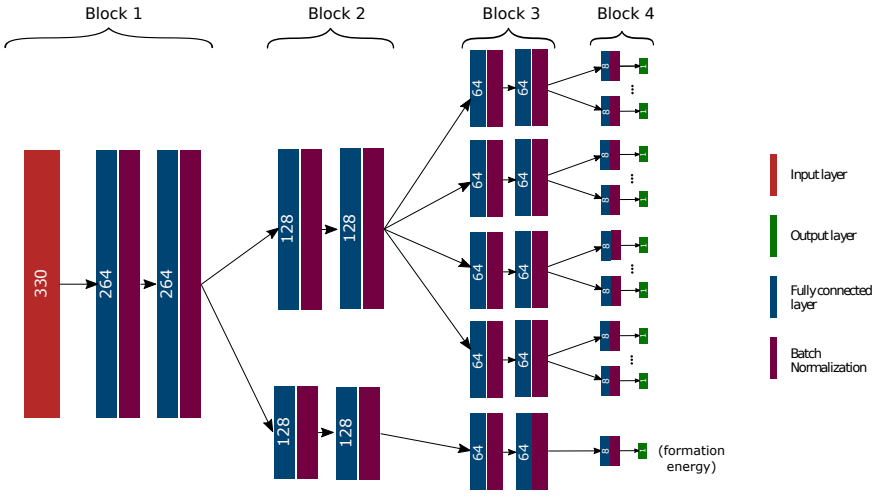


Figure 2.4: MODNet architecture for the vibrational properties. Architecture of the MODNet (composed of 4 blocks) when learning on the vibrational properties. The formation energy is added by adding a second order block to the first block.

In contrast to the previous section, hyperparameters were manually tuned here. First, the learning rate of MODNet was optimized to 0.01 from the set 0.1, 0.1, 0.005, 0.001, by fixing all other hyperparameters on arbitrarily chosen values.

Next, we study the effect of the following hyperparameters: input scaling, output scaling, loss function, and loss weights. These latter two parameters are very important as the peculiarity of our network consists in optimizing the predictions of multiple targets. Therefore, each target should have an equal contribution in the loss functions. For example, the vibrational energy and enthalpy are typically three orders of magnitude larger than the vibrational entropy and specific heat, which causes an imbalance in the summation of the loss function. Different solutions exist, such as scaling the targets or weighting targets in the loss function.

Table 2.2 presents the MAE (over 10 folds) for various combinations of the previous loss parameters. Scaling was tested using min-max or standard scaling, and the loss function was compared between Mean Square Error (MSE) or MAE. Next to scaling, we also test to weight the four different properties such that the average property-temperature curves are in the same range against scaling. Based on the results, we chose to use a min-max scaling of the input variables, a MSE loss function, and weighting of the different properties instead of scaling.

Once the learning rate, scaling, loss function, and weighting were fixed, the number of layers as well as the number of neurons is optimized. We chose two layers per block (except for the last one), with 256, 128, 64, and 8 neurons in the successive blocks, as shown in Figure 2.4. It was observed that adding or removing a layer, as well as doubling or halving the number of neurons, did not improve accuracy, as seen in Figure 2.5. From the same figure, the batch size was fixed to 256. Other architectures that were tested are listed in Table 2.3 with the corresponding performance criteria. Finally, the number of features was optimized, as described in the next section.

The final architecture is depicted in Figure 2.4. A rectified linear unit function (ReLU) is used as the activation for each layer. Learning is performed using an Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $decay = 0$) over 600 epochs.

Regarding MEGNet, in addition to the equivalent hyperparameters present in MODNet, the number of MEGNet-blocks is also optimized. As for the Random Forest, the number of trees is the only hyperparameter taken into consideration.

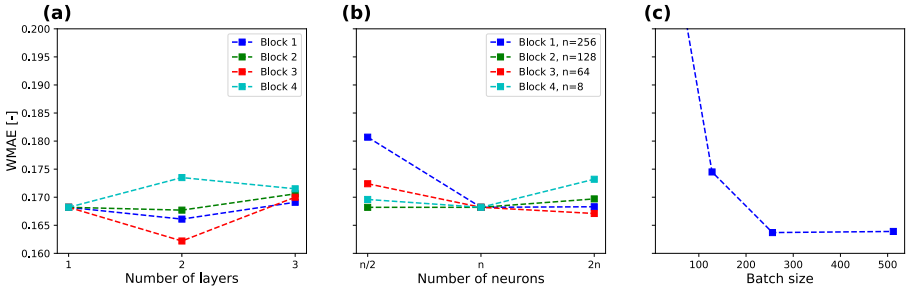


Figure 2.5: Optimization of the MODNet architecture . The effect of adding or removing layers (a) at each block is shown as well as reducing or increasing the number of neurons (b). The effect of changing the batch size is also shown (c). When one hyperparameter is changed, others are kept fixed at respectively 256, 128, 64 and 8 neurons (and 1 layer) per block.

Table 2.2: Scaling and loss optimization of MODNet when learning on the four vibrational properties and formation energy. The validation error (three custom defined criteria, see text, and S_{305K}) is reported for each set of hyper-parameters. [1]:When applied, the following weights are set: S: 0.3, C_v: 1, U: 0.001, H: 0.004. If the MSE is the loss function, corresponding weights are squared.

Input scale	Output scale	Loss function	Loss weights[1]	WMAE	MAX WMAE	MAPE	S_{305K} [meV/K/atom]
min-max	No	MSE	Yes, squared	0.167	0.435	0.0653	0.00860
Standard	No	MSE	Yes, squared	0.185	0.482	0.0760	0.00959
No	No	MSE	Yes, squared	0.416	1.220	0.158	0.0207
min-max	No	MAE	Yes	0.166	0.447	0.0683	0.00881
min-max	min-max	MAE	No	0.172	0.454	0.0700	0.00894
min-max	min-max	MAE	No	0.164	0.441	0.0680	0.00890
min-max	min-max	MSE	No	0.259	0.630	0.109	0.01253
min-max	min-max	MSE	No	0.1677	0.440	0.0666	0.00894

Results

Figure 2.6 shows the absolute error distribution on the vibrational entropy at 305K (S_{305K}) at three training sizes (200, 500, and 1100 samples) for different strategies, for a systematic identical test set of 145 samples. Furthermore, Figure 2.7 reports the test MAEs as a function of the training size for the same different strategies. MODNet is compared with a Random Forest (RF) learned on the composition alone (i.e. a vector representing the elemental stoichiometry) similar to a previous work relying on 300 vibrational data [90]. This strategy is referred to as c-RF in order to distinguish it from another strategy,

Table 2.3: Supplementary architectures for optimization of MODNet when learning on the four vibrational properties and formation energy. The validation error (three custom defined criteria, see text, and S_{305K}) is reported for each set of hyperparameters.

Block 1	Block 2	Block 3	Block 4	WMAE	WMAE MAX	MAPE	S_{305K} [meV/K/atom]
256,256	128,128	64,64	8	0.166	0.434	0.0666	0.00873
256,256	128,128	64,64	8,8	0.169	0.441	0.0679	0.00881
128,128,128	64,64,64	32,32,32	8,8	0.182	0.468	0.0730	0.00945
512	512	128	64	0.1646	0.429	0.0668	0.00866
256, 128, 32	32, 32	32, 64	16	0.174	0.455	0.0712	0.00892
512	128	64	8, 8, 8	0.1698	0.441	0.0688	0.00876
512	128	64	4, 4, 4	0.1651	0.433	0.0658	0.00840
512	128	16	3, 3, 3	0.1702	0.440	0.0686	0.00863

labeled RF, which consists in a RF learned on all computed features (covering compositional and structural features). Note that, for both c-RF and RF, performing feature selection on the input space has no effect on the results as a RF intrinsically selects optimal features while learning. This strategy can be seen as the baseline performance. The state-of-the art methods MEGNet *with transfer learning* (i.e. using the embedding trained from the formation energy) and SISSO are also used in the comparison. Another strategy, labelled AllNet, is considered which consists of a single-output feedforward neural network, taking *all* computed features into account. Finally, the results obtained with m-MODNet and m-SISSO, taking all thermodynamic data and formation energies, are also reported. [4]

The lowest mean absolute error and variance is systematically found for the MODNet models, with a significant ($\sim$ 8%) gain in accuracy for our joint-learning approach, more noticeable at lower training sizes. The RF approaches are performing worst in our tests, with a large spread

4 The SISSO framework typically starts from a few tens of features, iteratively applies operators on them in order to generate an enormous amount of combined features before proceeding to the SISSO screening itself. In our work, we automatically generate thousands of features for each material. Using all of them as starting point in the SISSO-approach is in fact unfeasible as the amount of secondary and ternary features grow too quickly. Therefore, one should first select the most appropriate primary features before running the SISSO framework.
We chose to proceed as follow. A first SISSO is executed without any operators (i.e. rung set to 0), and with a SIS-subspace of 3 features. The 50-dimensional model is then used to select the corresponding 50 most significant features. Then a second SISSO is executed on these 50 features with a rung of 2 and SIS-space of 20 features. Then the best model equal or below 5D is chosen before being evaluated on the hold-out test set. In other words we use a double SISSO, with the first one used to do feature selection.

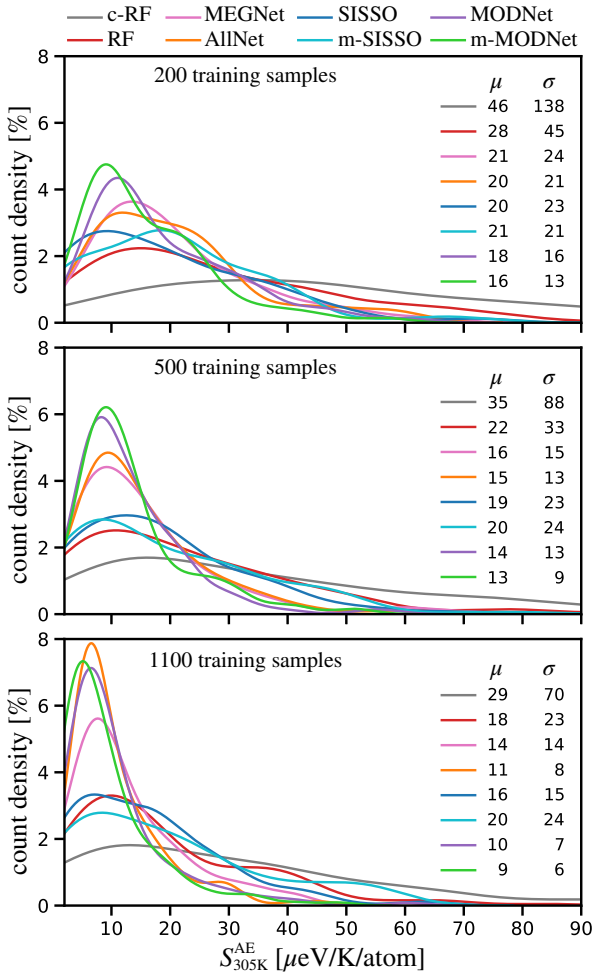


Figure 2.6: Comparison of the test error distributions on the vibrational entropy for different models. Absolute error distribution on the vibrational entropy at 305K (S_{305K} in μeV/K/atom) at three training sizes and for various strategies (see text for a detailed description). The density is obtained from a kernel density estimation with Gaussian kernel. The mean μ (equal to the MAE) and variance σ of each distribution are also reported in μeV/K/atom.

and maximum error, especially when considering only the composition. This is confirmed by a subsequent analysis of the features retained by the selection algorithm (see below): typically, the bond lengths are an important feature. Besides the MODNet models, AllNet, which is also based on physical descriptors, provides a baseline to measure the gain in performance achieved thanks to feature selection. In Figure 2.6 and even more clearly in Figure 2.7 , it can be seen that the usefulness

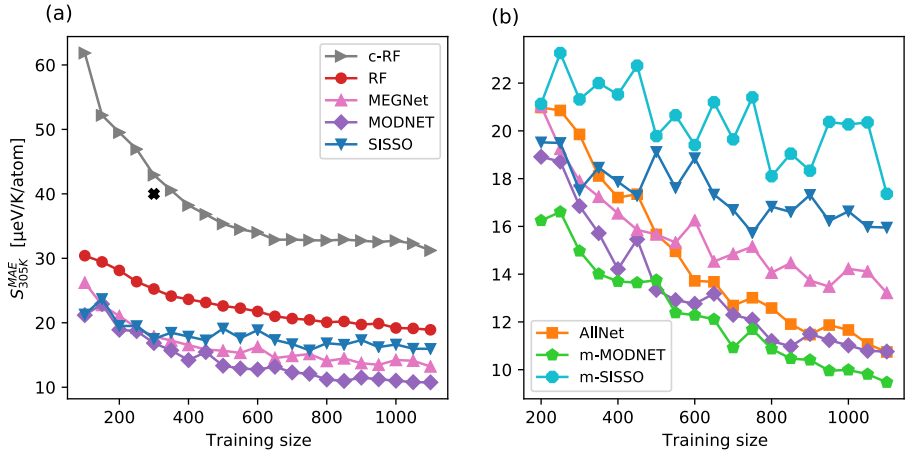


Figure 2.7: Test MAE on the vibrational entropy at 305K (S_{305K} in μeV/K/atom) as a function of the training size for various strategies (see main text for a detailed description). (a) Random Forest (RF) approaches are compared to neural-network models (Nets) and SISSO. The black cross indicates the RF result of Legrain *et al.* [90] based on 300 materials. (b) Best performing models and multi-target approaches are confronted.

of feature selection decreases with the training size. While, for 200 training samples, the gain is $\sim$12%, it reduces to $\sim$5% for 1000 training samples.

It is worth noting that, at the lower end of the training-set size (see 200 samples), SISSO has a comparable error with the other methods while offering a simpler analytic formula, which can be valuable. However, when increasing the training-set size, its error distribution does not seem to improve significantly in contrast with the other methods. Furthermore, contrary to m-MODNet, m-SISSO does not seem to provide any noticeable improvement with respect to SISSO in this example.

The m-MODNet was trained on four vibrational properties from 5 to 800 K: entropy, enthalpy, specific heat and Helmholtz free energy. Although the vibrational entropy at 305 K was systematically used to compare against other models, excellent performance was also found on the other properties. Table 2.4 contains the MAE for these four properties at 25, 305 and 705 K. Typical values of the corresponding properties found in the dataset are also given to compare against the error. As an example, we illustrate the prediction on Li_2O in Figure 2.8, which is a good representation of the typical observed error.

Table 2.4: MODNet errors on various vibrational properties. MAE and MaxMAE for the vibrational entropy, Helmholtz energy, specific heat and internal energy at different temperatures as predicted with MODNet. The MaxMAE is defined as the mean over worst 5% predicted samples.

Property	Typical values	MAE ($\times 10^{-3}$)	MaxMAE ($\times 10^{-3}$)
S_{25K} [meV/K/atom]	$\sim 10^{-5}$ - 0.1	2.5	2.9
S_{305K} [meV/K/atom]	$\sim$ 0.03 - 0.7	9.5	11.3
S_{705K} [meV/K/atom]	$\sim$ 0.1 - 0.9	10.8	12.6
H_{25K} [eV/atom]	$\sim$ 7 - 180	2.6	2.9
H_{305K} [eV/atom]	~ -130 - 180	6.9	7.5
H_{705K} [eV/atom]	~ -460 - 150	8.2	9.7
$C_{v,25K}$ [meV/K/atom]	$\sim 10^{-5}$ - 0.16	3.2	3.8
$C_{v,305K}$ [meV/K/atom]	$\sim$ 0.07 - 0.26	2.6	3.0
$C_{v,705K}$ [meV/K/atom]	$\sim$ 0.18 - 0.26	1.6	1.9
U_{25K} [eV/atom]	$\sim$ 10 - 180	2.2	2.6
U_{305K} [eV/atom]	$\sim$ 80 - 200	1.6	1.6
U_{705K} [eV/atom]	$\sim$ 180 - 270	1.2	1.4

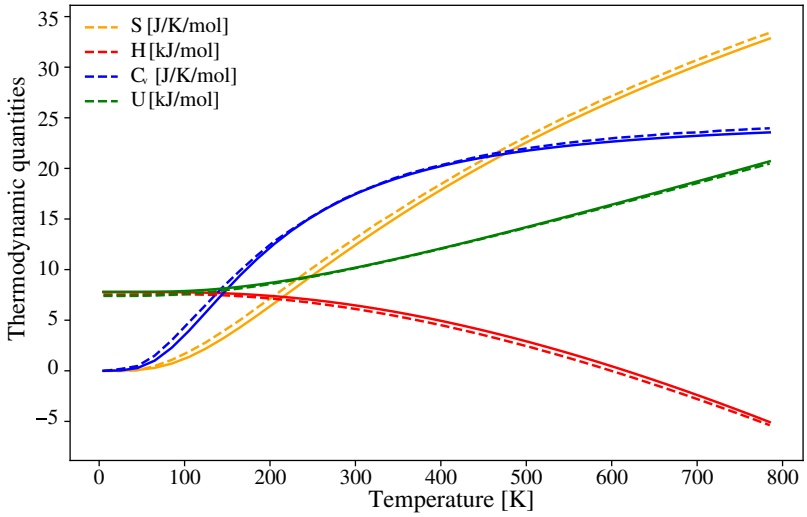


Figure 2.8: Example prediction of MODNet on vibrational thermodynamics. MODNet predictions (dashed line) and DFPT values (solid line) for the thermodynamic quantities of Li_2O (MPID: mp-1960) as a function of the temperature. Observed errors on this particular sample are close to the overall MAE of the test set.

We want to emphasize that the gain in accuracy provided by joint-learning is strongly influenced by the architecture choice. The similarity between target properties is used to decide where the tree splits, i.e., the layer up to which properties share an internal representation. In all generality, one can count the number of neurons and layers that separates two properties. This determines to which degree those two properties are related. Increasing this distance (i.e. more layers and neurons between them) gives more freedom to the weights and improves learning of dissimilar properties. However, increasing it too much will tend to make the predictions independent, and no common hidden representation can be used to improve generalization. A good balance thus needs to be found between freedom and generalization. Note that increasing the architecture-distance between two properties will always decrease training error (up to convergence), but the validation error will have a minimum. Unfortunately finding this minimum based on a quantitative analysis of the dataset is rarely feasible, similarly as finding the right architecture a priori for a single-target model. It is therefore considered as a hyperparameter, as it is commonly done in the ML field. In practice, we suggest to first gather the properties in groups and subgroups based on their similarity. This will define the splits in the tree-like architecture. Then, various sizes for the layers and number of neurons (which will define the intra-property distance in architectural space) should be included in the regular hyperparameter optimization of the model.

2.3.3 *Feature selection*

Feature selection is a valuable asset of MODNet and has two main advantages. First, it was shown in Figure 2.6 that an average 12% improvement in error can be obtained by removing irrelevant features. This is far from negligible. This increase in performance is achieved by reducing the noise to signal ratio, caused by the curse of dimensionality. This is especially the case for small datasets. Figure 2.7 shows that the gain in performance by feature selection reduces as the training size increases. We therefore expect that feature selection will be less important for larger datasets.

Second, feature selection (compared to feature extraction) has the advantage of keeping the input space understandable. As they are chosen according to their relation (i.e. mutual information), important factors contributing to the target property can be found. Figure 2.9 shows a

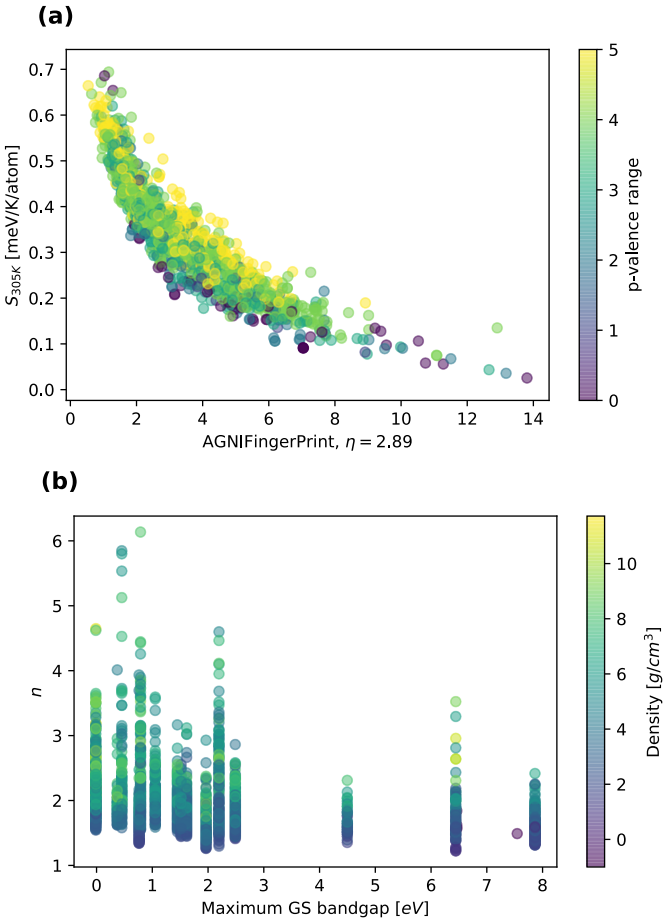


Figure 2.9: Visualization of selected features. Bivariate representation of the two most important features for two different properties: (a) vibrational entropy at 305 K and, (b) refractive index. Both features are complementary to narrow down the target output, although certainly not sufficient for an accurate estimation.

bivariate visualization for the vibrational entropy and refractive index (dataset from Naccarato *et al.* [80]) as a function of the two first selected features. Thanks to the redundancy criterion, both features are complementary to predict the target. A detailed description of these features can be found in Appendix A. Concerning the vibrational entropy, a strong correlation is seen with the first feature, namely AGNIFingerprint, which gives a measure of the inverse bond length. In other words, increasing the average bond length increases the vibrational entropy.

Similarly having a larger range of p-valence electrons (which is linked to ionicity) increases the vibrational entropy. Concerning the refractive index, two import factors are identified: the band gap and the density of the material. The band gap, although not explicitly given but instead approximated by the bandgap of the constituent elements, is known to be an important variable. Typically there is an inverse relation between the band gap energy and the refractive index, see Ref. [80]. Finding materials combining a high value for both properties remains a tedious task, and could therefore certainly benefit from Machine Learning. Overall, it is seen how common intuitive patterns for the physicist are indeed retrieved by the machine. Therefore, this strategy can be used to analyze and find underlying factors for all types of properties and datasets.

The feature selection algorithm presented in this work is based on relevance and redundancy and will be called *MOD-selection*. Other popular choices exist. Here, MOD-selection is compared to five other algorithms: (i) *corr-selection* in which features having the highest Pearson-correlation with the target are selected first; (ii) *NMI-selection* in which features having the highest NMI with the target are selected first; (iii) *RF-selection* where the data is first fitted with a Random Forest (300 trees) and features are ranked according to their impurity-based importance; (iv) *SISSO-selection* in which the data is first fitted by the SISSO model without applying any operator on the feature set, i.e. only primary features are used (rung set to 0) and each n^{th} dimension of the final model corresponds to the n-th descriptor; and (v) *OMP-selection* in which an orthogonal matching pursuit is applied by using the SISSO strategy with a SIS-space restricted to one.

It is worth noting that, although SISSO is a powerful dimensionality reduction technique, it can not be used as such for feature selection with the same generality as the other techniques. Indeed, SISSO provides a general framework for selecting the best few descriptors from an immense set of candidates but the selection is computationally limited to $\sim$10 features. This is not an issue for the original aim of SISSO (which consists in a low dimensional model), but it surely is when used together with a neural network, where the optimal amount is typically a few hundreds of features. Therefore, when going beyond the 10^{th} feature, we simplified SISSO to OMP, which scales linearly with the number of features.

Figure 2.10(a) shows the test error on the vibrational entropy at 305 K for the different models (MODNet substituted with different selection

algorithms) for the first 10 selected features. The training size is fixed to 1100 samples. There is a clear distinction between redundancy based techniques (MOD and SISSO) and non-redundancy based techniques (corr, NMI and RF). Accounting for redundancy is clearly important when using only a few features. In this particular scenario, SISSO outperforms MOD-selection.

However, in order to construct the best possible model, one should go much beyond 10 features. Figure 2.10(b) depicts the same error but as a function of the training size, all other hyperparameters being optimized. The number of optimal features is often chosen to be around 300 features. SISSO is therefore replaced by the OMP. It can be seen how MODNet outperforms all other selection algorithms, particularly at low sample size. The OMP method performs poorly due to wrongly chosen features which result in overfitting.

As an additional experiment, we aim to measure how well a feature selection algorithm is able to capture the important features, when only given a limited amount of labeled samples. We measure this by first running each selection algorithm on the *total dataset*, keeping the best 300 features, which forms the best approximation of optimal features for each algorithm. As a second step, we do the same but on sampled subsets of varying size. The Jaccardian similarity between the 300 features found on the subset and total dataset is represented in Figure 2.10(c). Note that this metric only represents how fast an algorithm converges in terms of chosen features, but does not necessarily mean that the selected features are worthwile. All methods (including Pearson-correlation) suffer significantly from small datasets, with a Jaccardian-similarity change of over 40% from 200 to 1000 samples. Similarity increases when the training size increases (with some exceptions due to sampling variance), as the sampled dataset approaches the total dataset. The correlation method provides the highest similarity for all training sizes. This can be explained by the simpler nature of the algorithm: measuring linear dependence requires less samples than more complex non-linear dependencies. The MOD and NMI approaches have a steeper increase in similarity than the RF-approach, while the three are non-linear approaches. Finally, the OMP algorithm has a low Jaccardian similarity, and this for the whole range of subsets. Additional experiments on the OMP showed that the similarity between features chosen on the different sampled subsets are also low, in contrast with the other methods. This clearly shows that a significant variance in selected features is found when slightly

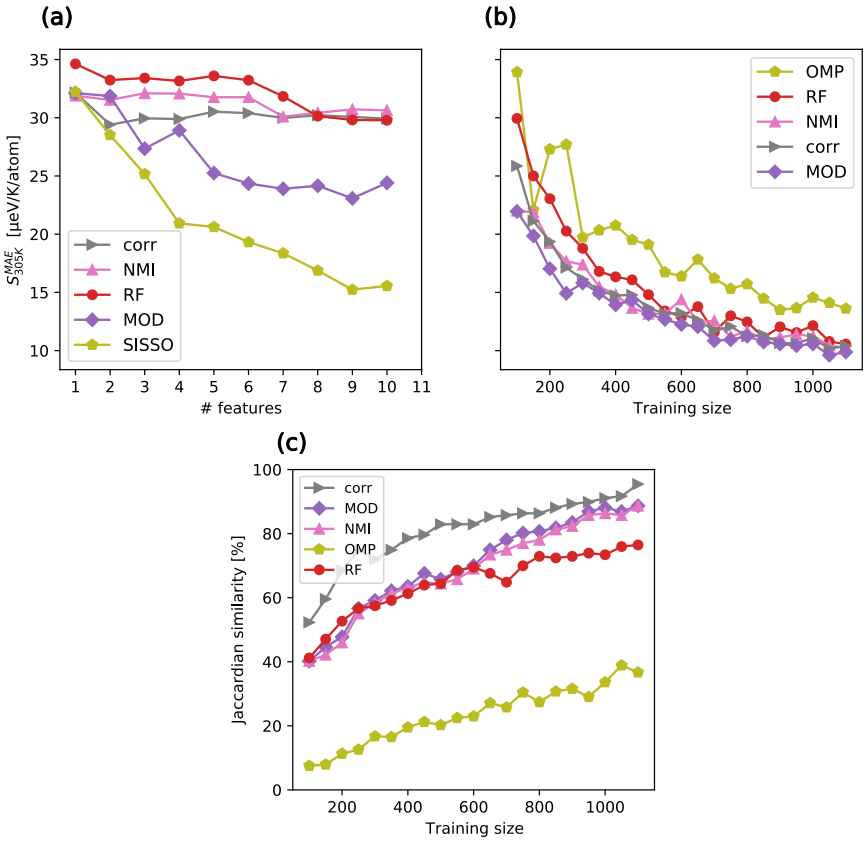


Figure 2.10: Performance comparison of different feature selection methods. (a) Test error on the vibrational entropy at 305K for different feature selection algorithms, as a function of the first few features for 1100 training samples. (b) Test error on the vibrational entropy at 305K for different feature selection algorithms as a function of the training size with other parameters being optimized over a fixed grid. Models are constructed by replacing the selection algorithm in MODNet by a Pearson correlation (corr), Normalized Mutual Information (NMI), Random Forest (RF), SISSO, and orthogonal matching pursuit (OMP). (c) Jaccardian similarity of the 300 first selected features on a sampled training set of size n, and the total dataset (1245 samples) as a function of n, for different feature selection algorithms.

changing the training set, which is not desirable. Therefore the OMP method should be avoided. This also explains the poor performance in Figure 2.10(b).

From these results, one can conclude that depending on the number of features to be selected, some algorithms work better than others.

As soon as there are more than 10 features (which is the case for most practical problems), MOD-selection performs the best. It should however be noted that, when selecting only a few features, accounting for redundancy is critical and, in this case, SISSO was found to be best. Unfortunately, it becomes computationally unaffordable above 10 features.

2.4 CONCLUSION

This chapter was rich in content, encompassing a wide array of methodologies and results. The findings revealed that state-of-the-art methods such as graph networks excel in handling large datasets but do not perform well on smaller datasets often encountered in physics. To address this limitation, we introduced MODNet, a novel approach that leverages prior knowledge, such as preprocessed meaningful features and multiple properties for the same material, to achieve excellent accuracy on limited datasets. Furthermore, m-MODNet demonstrated the convenience of constructing a single model for multiple properties, resulting in faster training and prediction times.

The significance of feature selection for small datasets was highlighted, with an observed improvement of 12% in vibrational thermodynamics when learning from only 200 samples. Additionally, the joint-learning mechanism of MODNet contributed to an additional 8% improvement in S_{305K}.

One of the most important contributions of our model is its outstanding performance in predicting vibrational entropies, achieving a mean absolute error (MAE) of 8.9 μeV/K/atom and a root mean square error (RMSE) of 12.0 μeV/K/atom on S_{305K} using a hold-out test set of 145 materials. Comparatively, this is four times lower than the results reported by Legrain *et al.* [90] (trained on 300 compounds) and 25 times lower than those reported by Tawfik *et al.* [91] on the exact same dataset used in our work.

Another key advantage of MODNet is its feature selection algorithm, which provides insights into the underlying physics. It identifies the most important and complementary variables related to the property of interest. For instance, in the case of vibrational entropy, it highlights the strong dependence on inter-atomic bond length and the valence range of the constituent elements, which reflects the ionicity of the bond. Similarly, for the refractive index, the algorithm relates it to an estimation of the band gap and density.

MODNet offers a versatile and flexible framework for predicting materials properties based on composition or structure. To facilitate its use, we have released it as an easy-to-use Python package on GitHub (see [77]). Since its initial release in May 2020, it has received continuous updates, including additional matminer presets, composition primitives, the genetic algorithm, multi-label, and multi-label classification capabilities, among others. The package remains actively maintained and has proven to be a reliable choice for materials property prediction.

In conclusion, our research has identified a frontier between physical-feature-based methods and graph-based models. While graph-based models are often considered state-of-the-art for many materials predictions, physical-feature-based methods demonstrate greater power when dealing with small datasets (below $\sim$4,000 samples). We have introduced a novel model, MODNet, that relies on optimal physical features selected based on mutual information with the target property, maximizing relevance and minimizing redundancy. The model, combined with a feedforward neural network, has shown remarkable performance in predicting properties, particularly when multiple properties are involved. For instance, in the case of vibrational properties of solids, our approach provides highly reliable predictions, significantly faster than conventional methods. Lastly, we highlighted how the feature selection algorithm in MODNet can offer valuable insights into the underlying physics of materials properties.

3

EPISTEMIC UNCERTAINTY IN MATERIALS PREDICTION: A BIAS-IMBALANCE ISSUE

This chapter focuses on exploring epistemic uncertainty, a critical aspect when predicting materials properties. Often, the MAE is used as a performance metric for a model. However, we observe significant variations in predictions for individual samples and subgroups. To explain this, we introduce the concept of materials classes, bias, and imbalance. To better understand and assess this uncertainty, we propose a quantitative approach using ensemble MODNet models. This approach will be thoroughly evaluated and compared using various uncertainty metrics. By doing so, we aim to enhance our understanding of uncertainty in materials predictions.

Finally, we will apply this newfound knowledge to tackle the challenge of active learning through Bayesian optimization, focusing on exploring metals within the materials space. This approach will help us make informed decisions on selecting the most informative samples for further experimentation and characterization.

Overall, this chapter sheds light on the importance of accounting for epistemic uncertainty and its potential impact on materials design.

This chapter is a detailed elaboration of the concepts and findings introduced in my publication, as listed in reference [P2].

3.1 MOTIVATIONS

We are interested in the following equation:

$$\hat{\boldsymbol{y}}, \boldsymbol{\sigma_e} = \hat{f}(\boldsymbol{x}) \tag{3.1}$$

Here, $\hat{f}$ represents a neural network, $\boldsymbol{x}$ is a vector representing a material $\boldsymbol{m}$, $\hat{\boldsymbol{y}}$ denotes the properties of interest, and $\boldsymbol{\sigma}_e^2$ refers to the epistemic variance.

In contrast to the previous chapter, the model now includes the epistemic variance, which refers to the uncertainty of the model, often due to the lack of data. This quantity enhances the interpretability of

predictions. Specifically, it helps answer the question: for an unseen sample, how reliable is the model's prediction? Epistemic uncertainty is distinct from aleatoric uncertainty, which represents the inherent randomness in the data that cannot be resolved and reduced. The current work does not address aleatoric uncertainty.

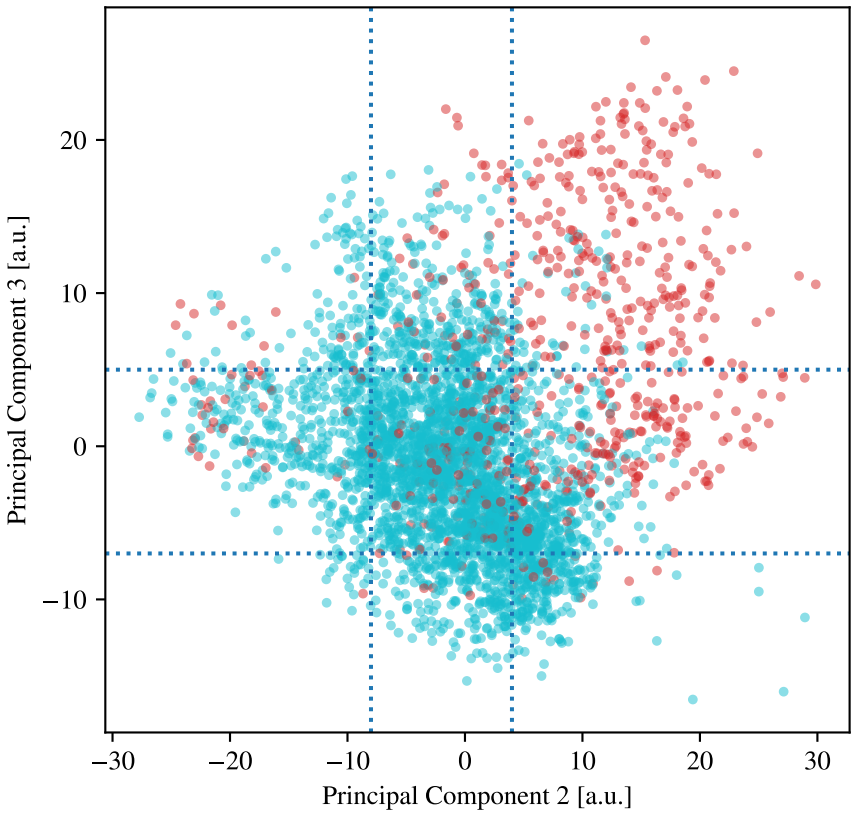
Epistemic variance is closely related to the applicability and generalization capabilities of a model. In the realm of materials science, addressing this challenge deserves significant attention, similar to the effort devoted to optimizing an error metric. Apart from providing users with information about the model's trust, another motivation stems from its practical importance in Bayesian optimization, enabling active learning, as we will explore later.

3.2 THE BIAS-IMBALANCE ISSUE

When predicting materials properties with a machine learning model, one should keep in mind that the prediction error can vary significantly from material to material. The MAE is merely an aggregate statistic, with sometimes huge differences (lower or higher) on single predictions. For instance, concerning the experimental band gap, the errors span 5 orders of magnitude from 10^{-4} to 10^{1}. One can be interested in better understanding this variability, and seek to model it. This requires one to better understand the concept of bias and variance in the materials space.

A striking example concerns the prediction of the refractive index, a DFT-computed dataset by Naccarato *et al.* [80]. A random test set of 200 samples yields an MAE of 0.051. However, when looking only at the non-oxides in this test set, the error jumps to 0.082. The oxides (defined loosely here as compounds containing at least one oxygen atom) in the test set have a much lower MAE of 0.035. This simple observation shows that the models seems to perform much better on oxides. From physical and chemical grounds, we learn that materials appear in distinct qualitative classes based on composition (e.g., oxides, phosphides, sulfides), structural classes (e.g., perovskite, wurtzite, zincblende) or by intrinsic properties (e.g., metals and non-metals). Other more subtle possibilities exists and are equally important as materials often behave differently depending on subgroup membership. Therefore, it could be expected that a machine learning model trained on a particular class will fail to make good predictions when applied to another class. However, many of the different classes mentioned previously are principally

Figure 3.1: PCA decomposition of the refractive index dataset (Naccarato *et al.* [80]) for the second and third component. The second component is linked to the bond-length, while the third component is linked to the ionicity of the compound. Data points corresponding to oxides and non-oxides are coloured blue and red respectively. Compounds having at least one oxygen atom tend to have shorter mean bond lengths and increased ionicity. An imbalance appears due to sampling: the dotted blue line divides the sample space in denser and coarser regions. The compounds in the well-sampled centre region yield on average the lowest error, regardless of whether they are an oxide. PCA components are provided in detail in Tables 3.1-3.3.



human made. They are based on human intuition and concepts such as bond type or the periodic table, and in some sense, do not exist for the machine [92].

In all generality, a machine learning model goes beyond these human concepts and forms its own materials representation, considering a material as a purely mathematical object in a high-dimensional space. For instance, one can use the feature space, or better, a condensed latent space. In this chapter, we consider materials either as points in the MODNet feature space, or as a PCA (principal component analysis) reduction in a lower dimensional space. Classes can now appear naturally as point clouds in these spaces and a similarity metric can be defined through a simple distance-based measure. Close materials are similar and should behave closely with respect to the learned property; if this is not the case, the feature space may not be sufficiently complete to fully discriminate samples.

The concepts of *imbalance* and *bias* are important in this context. A training set that has two or more distinct classes (i.e., clusters in the materials space) that are sampled in different proportions is *imbalanced*. For instance, 84% of the refractive index dataset is comprised of oxides and therefore constitutes an imbalanced training set. A training set is *biased* when it covers only a specific region (i.e., not all classes are covered) of the materials space, a more extreme case of imbalance. Both imbalance and bias have consequences to the generalisation of a model as they express potential dissimilarities between training and target domain.

In order to visualize imbalance-bias and the discrepancy in error between oxides and non-oxides for the refractive index DFT dataset of Naccarato *et al.* [80], we performed a PCA decomposition. PCA was selected over other methods such as tSNE [93], LDA, or PcovR [27] due to its understandability, as it allows for an easier correlation with physical or observable characteristics thanks to its linearity. Additionally, PCA supports unsupervised exploratory analysis of the feature space, focusing solely on variance and disregarding costly labels. This approach is therefore suitable for uncovering underlying patterns in the material space.

The first three components together account for 25 % of the variance. Tables 3.1-3.3 contains the list of descriptors and corresponding weights in the feature space. The first feature roughly corresponds to a mean of all features and therefore identifies outlier compounds in the feature space. They are not of particular interest in our study case. However, the second and third components are closely related to the concept of oxide. Figure 3.1 displays the second and third principal components of the PCA decomposition. Each data point, initially a high-dimensional

Table 3.1: 5 highest contributing features of $PC_1 = \sum w_{1,i} f_{1,i}$, with corresponding weights

$w_{1,i}$	Feature $f_{1,i}$
-0.0890	`ChemEnvSiteFingerprint|mean SC:12`
-0.0890	`ChemEnvSiteFingerprint|mean SH:11`
-0.0890	`ChemEnvSiteFingerprint|std_dev S:10`
-0.0890	`ChemEnvSiteFingerprint|mean DD:20`
-0.0890	`ChemEnvSiteFingerprint|mean H:10`

Table 3.2: 5 highest contributing features of $PC_2 = \sum w_{2,i} f_{2,i}$, with corresponding weights

$w_{2,i}$	Feature $f_{2,i}$
-0.1070	`GaussianSymmFunc|mean G2_20.0`
-0.1053	`AGNIFingerPrint|mean AGNI eta=1.23e+00`
-0.1049	`GeneralizedRDF|mean Gaussian center=1.0 width=1.0`
0.1047	`ElementProperty|MagpieData mean Row`
0.0996	`VoronoiFingerprint|mean Voro_dist_minimum`

Table 3.3: 5 highest contributing features of $PC_3 = \sum w_{3,i} f_{3,i}$, with corresponding weights

$w_{3,i}$	Feature $f_{3,i}$
-0.1194	`IonProperty|avg ionic char`
-0.1174	`ElementProperty|MagpieData avg_dev Electronegativity`
-0.1159	`LocalPropertyDifference|` `mean local difference in Electronegativity`
-0.1023	`IonProperty|max ionic char`
0.1020	`ElementProperty|MagpieData mean MendeleevNumber`

point in the MODNet feature space, is now projected onto two dimensions. The second component is proportional to the mean bond length (or mean atomic volume), with corresponding descriptors of radial distribution function peaks and elemental radii. The third component is related to the bond type, comprising of descriptors related to electronegativity and element identity. Small values indicate structures

tendentiously forming ionic bonds, while larger values indicate metallic bonding. The second component can be seen as being related to the structure while the third depends more on its composition. From the figure, it can be seen that regions with dense (lower left) and coarse (upper right) samplings appear, indicating an imbalance in the dataset. Notably, oxides (in blue) are on average in the lower left part and non-oxides are on average in the upper right corner of the decomposition. Indeed, oxygen being a quite small and electronegative atom, it tends to shorten the mean bond length of a compound and increases ionicity. These two observations explain the higher average error on non-oxides: they form a minority group in a region with a coarse sampling, with fewer neighbouring samples than oxides. Importantly, it is the density (thus the number of neighbours) that is the critical parameter here. Therefore, it is possible to have non-oxides that are well predicted and oxides that are poorly predicted; non-oxides in the middle region of Figure 3.1 have a MAE of 0.04, while oxides in the lower right region have an error of 2.5. Note that this latter region is mainly outside the main distribution of points, and corresponds thus to a biased training set.

The oxide vs. non-oxide distinction is used here only as an illustrative example, put forward by human intuition. We emphasize that the imbalance should be seen in the feature space directly, i.e., in terms of bond-length / bond-type imbalance.

3.3 THE ENSEMBLE MODNET MODEL

The previous section introduced a more qualitative understanding of the origin of wrongly predicted samples based on an unsupervised analysis. However, to answer the question of "How reliable is my prediction," one should conduct a more quantitative analysis of the uncertainty.

Different approaches exist in the literature for quantitative uncertainty estimations with neural networks. Notably, Bayesian networks offer a mathematical framework that incorporates uncertainty directly into the architecture of the network. They treat weights as distributions rather than fixed values, allowing for a probabilistic understanding of the model predictions. Another popular method is Monte Carlo (MC) dropout as a Bayesian approximation [94]. The dropout technique, initially designed to prevent overfitting, can be used at inference time, providing a measure of uncertainty by observing the variance

in predictions across multiple forward passes. This approach effectively creates a pseudo-ensemble of models. Deep Ensembles, another approach, involve training multiple neural networks independently and then aggregating their predictions [95]. By comparing the outputs of these diverse models, one can gauge the uncertainty of the predictions. Unlike Bayesian methods, deep ensembles do not require any modification to the network architecture or the training process.

Each of these methods has its strengths and weaknesses. Bayesian networks offer a theoretically grounded approach to uncertainty estimation but can be computationally intensive and complex to implement due to their intractable nature and reliance on approximation techniques such as Laplace approximation [96], Markov Chain Monte Carlo methods [97], variational Bayesian methods [98, 99], and Monte Carlo Batch Normalization [100]. MC Dropout provides a more practical solution, though it still inherits some of the limitations of the dropout technique (as it depends on the dropout rate and chosen layers for dropout). Deep Ensembles, although requiring to fit multiple models, are relatively simple to implement and have been shown to provide robust uncertainty estimates across a variety of tasks [95]. More specifically, Scalia *et al.* [101] showed it has consistent advantage in out-of-domain epistemic uncertainty calibration and generalization error when applied on the chemical space [101]. In line with this, this chapter proposes the ensemble MODNet model, as illustrated in Figure 3.2.

The model consists of a set of MODNet models, all trained on the same task, but with independently optimised hyperparameters, training set (by bootstrapping) and initial weights. In particular, we choose to take the 5 best architectures for each inner training fold. Moreover, multiple copies of each individual architecture are trained via bootstrap resampling of the given training set fold, resulting in a 125-model ensemble. This final ensemble model has the advantage of predicting a distribution of values, from which statistics such as the mean and standard deviation can be computed. The mean over the models is used as the final prediction, while the standard deviation is used as an epistemic uncertainty measure (σ_e). A Gaussian distribution is assumed (even if not always realistic) to create confidence intervals (CIs). For instance, a 95% confidence interval would fall within $2\sigma_e$.

The descriptors used by MODNet are not necessarily numerically robust across the space of all possible compositions and crystal structures and thus could lead to unphysical results for particular rare representations (e.g., unit cells with very acute angles). To mitigate this,

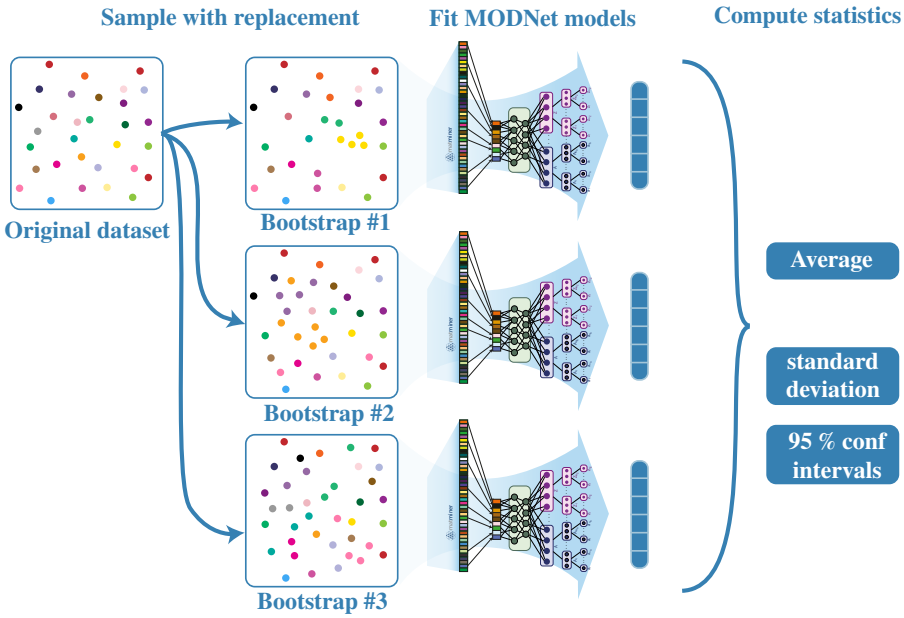
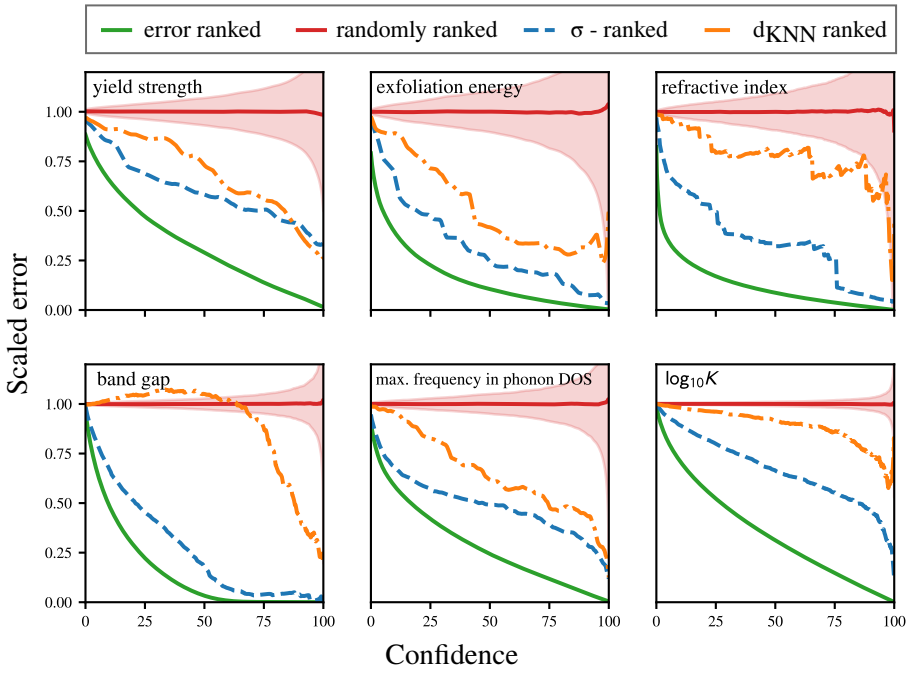


Figure 3.2: Schematic of the bootstrapping ensemble approach. From the initial dataset, bootstrap subsampling is performed. This leads to the training of a set of MODNet models, each with independently optimized hyperparameters and initial weights. From the distribution of predicted values, statistics such as the mean and standard deviation can be computed, offering a robust prediction together with an uncertainty estimate.

predictions from the final model that fall considerably outside of the range, $r = y_{\max} - y_{\min}$, of target values present in the training set (i.e., those outside of r padded by 25%) are replaced with a value drawn from a uniform distribution spanned by r. This makes the ensemble more robust; the remapping will have minor impact on the overall ensemble on average, but the variance will still be penalised with the number of pathological cases.

Finally, it should be known that beyond ensemble-based methods, there exist other model architectures that intrinsically perform uncertainty quantification, namely Bayesian learning approaches, Gaussian processes or random forests [102, 103].

Figure 3.3: Confidence error curves for six different regressions task found in MatBench: steel yield strength, 2D exfoliation energy, refractive index, experimental band gap, phonon DOS peak and bulk modulus. Each curve represents how the mean absolute error changes when test points are sequentially removed following different strategies. A baseline (red) is generated by randomly removing points one at a time (as if all points had equal confidence), with the shaded area showing the deviation across 1000 trials. The randomly ranked error (red) forms a baseline with the shaded area representing the standard deviation over 1000 random runs. The error ranked curve (green), where the highest error is removed sequentially, represent a lower limit. The std-ranked strategy follows the uncertainty predicted by the ensemble MODNet, while the d_{KNN} ranking is based on the 5-nearest neighbour cosine distance between each test point and training set.



3.4 PERFORMANCE ASSESSMENT

In order to assess the aforementioned method, we use Matbench v0.1 and compare the actual errors to the predicted standard deviation using various metrics.

Figure 3.3 represents the MAE as a function of the confidence percentile for different the different tasks. The MAE of a confidence percentile

p is computed by removing the p% most uncertain predictions, following different strategies. Randomly removing points (in red) forms a baseline, while a perfect ranked strategy based on the absolute errors forms a lower limit. The uncertainty provided by the ensemble MODNet model is depicted in blue. In addition, a strategy based on distances in the feature space is added. The idea is to determine a model's applicability domain via the distance of a query point and the k-nearest neighbours (KNN) in the training set, d_{KNN}. The space of optimal descriptors, as selected by MODNet is used. Here, after various tests, the cosine distance is used for simplicity with $k = 5$. Other distances are possible, although one should be cautious with the high dimensionality [104].

Both the ensemble's uncertainty and d_{KNN} distance are adding value to distinguish error magnitudes with respect to the random procedure. The ensembling techniques is seen to be performing particularly well, especially at the lower end of the confidence percentiles. The improvement of d_{KNN} wrt. random ranking confirms that feature space imbalance is indeed an important factor in assessing uncertainty. However, it has clear limitations. First, it is clear that it does not perform as well as the probabilistic ensemble method. Especially, for the experimental band gaps a poor performance is found. Second, the d_{KNN} method cannot provide any quantitative uncertainty estimate such as confidence intervals. Therefore, the ensembling technique is preferred for uncertainty estimations. Moreover, assuming Gaussian distributions, 95% confidence intervals can be built for each prediction from the predicted standard deviation.

More generally, one can compare a theoretical Gaussian proportion to the actual observed proportion of errors within the associated CI. This is called the calibration curve and is illustrated in Figure 3.4 for the Matbench dielectric task (refractive index). Given a theoretical proportion and assuming a Gaussian distribution, we count the fraction of points for which the target is indeed within the associated CI. For instance, if we set a theoretical proportion of 68%, we count the number of points for which the target lies between $[\mu - \sigma_e, \mu + \sigma_e]$. This process is repeated for multiple proportions. As one can see from Figure 3.4, the observed proportion is slightly higher than the theoretical one. The ground truth is more often within the CI than expected, and the model is underconfident. A calibration curve under the diagonal would indicate an overconfident model. Note that recalibration of the

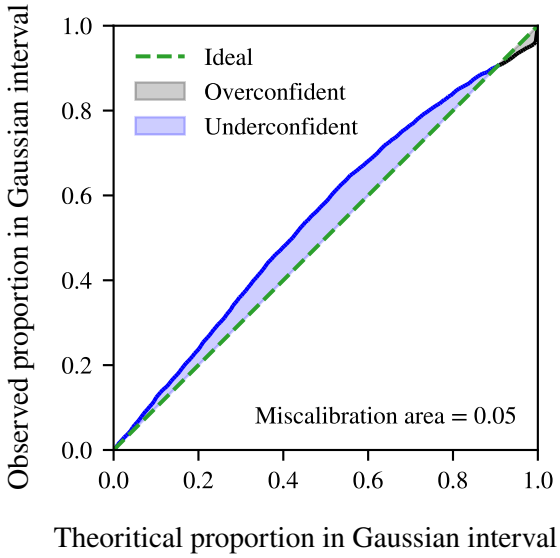


Figure 3.4: Calibration curve for the ensemble MODNet model trained on the matbench dielectric task.

model (i.e., remapping the predicted standard deviations based on the observed errors) could further improve accuracy.

Figure 3.5 represents the calibration curve for the following other Matbench tasks: bulk modulus, shear modulus, experimental band gap, 2D exfoliation energy, maximum phonon frequency, and experimental yield strength of steels. The miscalibration area (the area between the diagonal and calibration curve) is, however, small for all benchmark tasks, confirming the effectiveness of the ensemble approach at providing uncertainty estimates.

3.5 ACTIVE LEARNING

Thanks to the previously presented ensemble approach, we now have a quantitative measure of uncertainty. In this section, we will explore how this measure enables the model to perform active learning. In active learning, one iteratively explores a materials space to find a set of materials satisfying a specific requirement, often a property optimum. This section will demonstrate this using the example of finding metals within a pool of compositions.

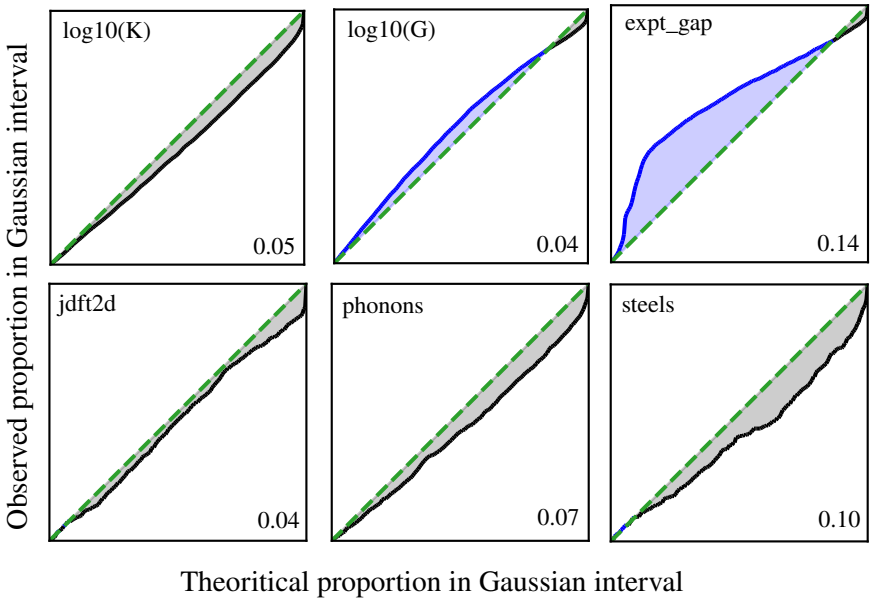


Figure 3.5: Calibration curves for the following matbench tasks: bulk modulus ($log_{10}(K)$), shear modulus ($log_{10}(G)$), experimental band gap (exptgap), 2d exfoliation energy (jdft2d), max. phonon freq. (phonons) and experimental yield strength (steels). The area under the curve is provided in the lower right corner for each task.

3.5.1 *Bayesian Optimization*

The present goal is to explore, in a minimum of iterations, all metals in a given set of compositions. This discrete problem can be translated to a continuous one by iteratively searching for compounds minimizing the band gap. We will therefore use a Bayesian optimization approach, as illustrated in Figure 3.6.

Let D be the total set of data under consideration. We first put a test set D_t apart. This will be helpful for evaluating the model in terms of MAE. The remaining data, D_a, is used for the active learning loop, such that $D = D_a \cup D_t$. At each learning cycle i, the model will be trained on a training set T_i, such that $T_i \subset D$. For this, we suppose that an experimenter is available to label instances in D with the corresponding band gap for each cycle i.

The model then uses this set T_i to train and generate predictions (μ) and uncertainties (σ_e) on the remaining space, called candidate

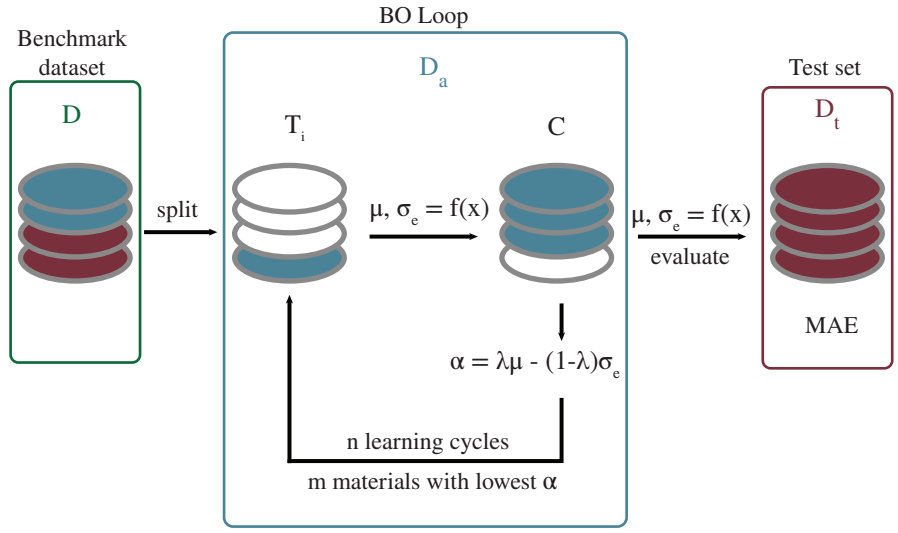


Figure 3.6: Schematic of the proposed active learning method, with Bayesian optimization (BO). A labelled dataset D is split into two parts: one for the AL loop (D_a) and one as an external test set (D_t). Inside the BO loop, at each learning cycle i, the model is used to select the best candidates in C through an acquisition function α.

space $C \equiv D_a \setminus T_i$, to select the most promising candidates for the next iteration. Of course, not all instances in the candidate space C are equally informative for the learning task. Therefore, an acquisition function α is created to determine the most valuable candidate, such that $argmin(\alpha)$ gives the most valuable candidate. Different acquisition functions exist, depending on the application. Here, we use a lower confidence bound (LCB):

$$\alpha = \lambda\mu - (1-\lambda)\sigma_e, \quad 0 \leq \lambda \leq 1. \tag{3.2}$$

Here, λ is a hyperparameter that balances exploitation and exploration. Indeed, a high value of λ closer to 1 emphasizes areas where the model has low predictions μ, focusing on exploitation. Conversely, a low value of λ closer to 0 encourages the model to probe regions with high uncertainties σ_e, emphasizing exploration. Intermediate values of λ provide a mix of both, allowing the model to explore and exploit the input space to find the most informative data points.

We decide to iteratively add the $n = 50$ candidates with the lowest α. For benchmarking purposes, we need to work with a finite set D with all

data already labeled. We use the experimental band gaps as provided by Zhuo *et al.* [84], which contains 4604 labeled entries. Labels are initially hidden except for the initial T_0 containing 50 random samples. The candidate space consists of randomly drawn 2050 samples, while the remaining 2139 samples are used as the external test set.

In order to assess the performance of the model, we use the Top Fraction (TF):

$$\mathrm{TF}(i) = \frac{\text{number of metals discovered}}{\text{total number of metals}} \in [0, 1] \tag{3.3}$$

this metric identifies the fraction of top candidates that we have recovered from the candidate space. It is in fact similar to the recall metric in classification.

Furthermore, we also define the Enhancement Factor (EF):

$$\mathrm{EF}(i) = \frac{\mathrm{TF}_{\mathrm{BO}}(i)}{\mathrm{TF}_{\mathrm{random}}(i)}, \tag{3.4}$$

which shows how much better (in terms of found top candidates) the model performs with respect to a random baseline, consisting in randomly selecting points as is sometimes done in materials design. Another useful metric is the Acceleration Factor (AF):

$$\mathrm{AF}(\mathrm{TF} = a) = \frac{i_{\mathrm{BO}}}{i_{\mathrm{random}}}, \tag{3.5}$$

which shows how much faster one would reach a desired TF with respect to a random search.

3.5.2 *Results*

Figure 3.7 shows the top fraction as a function of the learning cycle for different values of λ. This is compared to a random search, represented in grey, consisting of sampling $n = 50$ random samples in the candidate space, and a perfect approach consisting of adding 50 metals per cycle until all metals from the candidate space have been found. As can be seen, the random baseline does an average job by steadily providing a constant number of metals, which corresponds to the statistical fraction of metals in the datasets. This can be expected as it doesn't favor any particular compound, and due to the relatively large sample size (50), no fluctuations should appear. Interestingly, all models, except the one having an acquisition function only relying on the uncertainty, behave

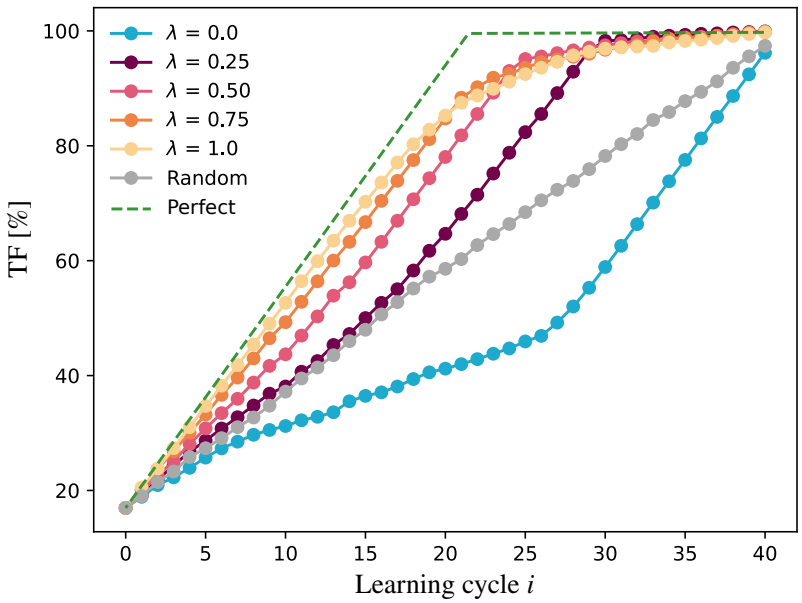


Figure 3.7: Top fraction of metals identified as a function of the learning cycle for different exploitation-exploration rates (λ). It is compared to a random and perfect search.

better than the baseline. Indeed, whenever the strategy is guided by a rough estimate of the band gap, better candidates can be identified. The slope, however, tends to decrease around cycle number 20; this is merely an artifact of the finite dataset. It becomes much harder to draw top candidates when the pool is small and only a very few metals are left.

Interestingly, the best performance is found for $\lambda = 1$. It has the highest slope on average for the first 20 cycles, reaching the Top 50% first (corresponding to an EF of 0.56), and has an EF of 1.7 at i=15. Uncertainty, and therefore exploration, does not seem to have a major impact on the present problem. However, this is not entirely true. Figure 3.8 represents the error of each model (differentiated by λ) on the hold-out test set. It can be seen that accounting for exploration improves the intrinsic model quality, as measured by the test MAE. This has minor implications for Figure 3.7, as all models seem to have enough accuracy to distinguish metals from non-metals. One slight implication is that the inflection point for $\lambda = 0.25$ occurs at a later

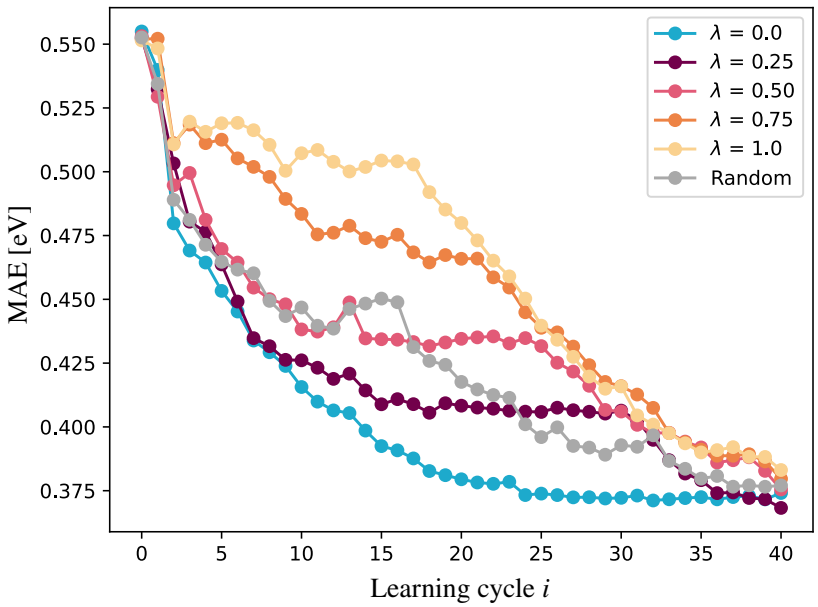


Figure 3.8: MAE of the MODNet model used in the BO loop, evaluated on an external test set, as a function of the learning cycle.

iteration state than $\lambda = 1$. As the model has better prediction accuracy, it achieves a higher TF threshold after saturating; i.e., even within a small pool of left-out "difficult" materials, it still finds metals.

Finally, one might ask why the full exploration variant ($\lambda = 0$) performs worse than the random baseline. A priori, the predicted variance should not be correlated with the target value. However, in the case of the band gap, it does exhibit a correlation, as it has a lower bound of 0 eV. Consequently, metals will have a smaller uncertainty by the design of the ensemble network.

3.6 CONCLUSION

In this chapter, we explored the bias-imbalance issue in materials science. We showed that depending on the test set sampling, significant variations in errors can be measured. These variations arise due to the high dimensionality of the feature space spanned by materials and the associated sparsity of a given dataset. Bias or imbalance can be easily introduced into the training set through this sampling. Therefore, the

assessment of applicability through uncertainty is a crucial aspect for model development.

We demonstrated that dataset imbalance can be examined using dimensionality reduction techniques (such as PCA), and uncertainty can be quantified through ensemble-based methods. This approach offers the possibility to set confidence bounds on individual predictions, allowing uncertain predictions to be flagged or removed without incurring significant computational or architectural costs. Furthermore, we discussed how active learning, when coupled with Bayesian optimization, can be applied to explore materials spaces, particularly in the context of metals.

However, it is essential to recognize that our approach is an initial step. Future work could explore more sophisticated techniques, integration with other approaches, and broader applications across other materials spaces. These prospects could profoundly impact the efficiency and effectiveness of materials design and discovery.

4

ACCURATE EXPERIMENTAL BAND-GAP PREDICTIONS WITH MULTI-FIDELITY LEARNING

In this chapter, we explore multi-fidelity learning techniques that aim to enhance band gap predictions by combining experimental and DFT data. By leveraging the strengths of each dataset, these methods extract valuable knowledge from both high-accuracy experimental measurements and larger, though less accurate, computational results.

We begin by addressing the challenges posed by datasets of varying quality. Next, we delve into various multi-fidelity techniques, including transfer learning, joint learning, stacking ensemble learning, and deep-stacking ensemble with correction learning. These techniques are compared against a single-fidelity MODNet model trained solely on experimental data. The simple yet effective *Correction Learning* method emerges as a particularly promising tool.

Overall, this chapter showcases the potential of multi-fidelity learning in materials informatics, particularly in enhancing predictions of electronic band gaps. The insights derived from this chapter open up exciting possibilities for applying multi-fidelity techniques to predict other materials properties, leading to accelerated materials discovery and design.

This chapter is a detailed elaboration of the concepts and findings introduced in my publication in collaboration with Gregoire Heymans, as listed in reference [P3].

4.1 MOTIVATIONS

Obtaining reliable ML models typically requires large and high-accuracy datasets. Unfortunately, there is often an inverse correlation between quantity and quality. Large datasets are in many cases theoretical ones, such as those based on cheap density functional theory (DFT) functionals, while high-accuracy experimental datasets usually have a rather small size. For instance, if one considers the electronic band gap (i.e., without excitonic effects), the Materials Project [14] contains $\sim 10^5$ Perdew-Burke-Ernzerhof (PBE) [105] calculations, while the ex-

perimental dataset collected by Zhuo *et al*. [84] consists of two orders of magnitude less measurements. Band gaps estimated by DFT calculations typically lead to a systematic underestimation of 30 % to 100 % with respect to the experimental results [106]. Therefore, a model built on this data will present systematic errors with respect to reality. Alternatives exist, such as the rigorous many-body perturbation theory based on the *GW* approximation, that provides quite accurate results. However, it is computationally very demanding and typically leads to even smaller datasets (80 materials) [67].

More generally, materials properties are often bundled with different degrees of accuracy. The most straightforward case is a dataset gathering both experimental and DFT results, but it is not uncommon at all to see a dataset combining calculations computed with different exchange-correlation functionals such as PBE and the more accurate Heyd–Scuseria–Ernzerhof (HSE) [107].

For screening materials, one ideally wants to obtain the best estimate of the actual value (i.e., experimental) of the required property. This means that models should, in principle, only be built from scarce experimental data. In practice, it is, however, possible to gain knowledge from the larger, but less qualitative datasets, in order to improve predictions of the experimental quantity. This idea has already been investigated previously [108, 109]. Kauwe *et al*. [108] combined multiple learners, forming a so-called *ensemble learning*, which are trained on experimental or DFT band gap data. The different predictions are then combined in a separate model, a meta-learner, predicting the final higher quality (i.e., experimental) data. Chen *et al*. [109] built a convolutional graph neural network, based on MEGNet, where the fidelity of each sample is encoded through an embedding to form an additional state feature of the crystal. This method has the advantage to work on sets of compounds that can be very diverse, in the sense that all the compounds do not have to be present in each dataset (in contrast to Kauwe's method). However, given the complexity of the graph neural network, the errors are still slightly higher than with state-of-the-art methods relying on the smaller experimental dataset only [110].

Another popular strategy is to use transfer learning. In this approach, a neural network is first fitted on a large source dataset, followed by fine-tuning on a smaller target dataset. The network will transfer knowledge (embedded in the weights) from the source to the target task. This technique has successfully been applied in several studies, covering properties from Li-ion conductivity for solid electrolytes to

steel microstructure segmentation [59, 111–116]. In order to be effective, the source task should be closely related with the target task.

In this chapter, we compare different techniques that combine multiple quality sources for the same property, in order to improve the accuracy of the predictions. The property of interest is the experimental band gap. Various studies have tackled the band gap problem (experimental and simulated), from composition or structure-specific tasks to more general approaches[42, 50, 51, 56, 59, 84, 117, 118], with only more recently efforts on a multi-fidelity approach [109, 119]. In particular, we aim at having the lowest prediction error on the experimental band gap, for any structure, by combining experimental measurements with both PBE and HSE DFT calculations. We show that an improvement of 17% can be achieved, with respect to the predictions resulting from learning on the sole high-quality experimental data, when learning on the difference between high- and low-quality values. On the contrary, ensembling does not seem to be particularly helpful.

4.2 DATASETS

Three different datasets for the electronic band gap with varying levels of accuracy are used in this chapter: (i) DFT computational results using the PBE functional, (ii) DFT computational results using the HSE functional, and (iii) experimental measurements. These will be referred to as PBE, HSE, and EXP data, respectively.

The PBE data was retrieved from Matbench v0.1, covering a total of 106 113 samples [120]. Compounds containing noble gases or having a formation energy 150 meV above the convex hull were removed.

The HSE data was recovered from the work by Chen *et al*. [109]. The HSE functional typically provides more accurate results than the PBE ones, but it is computationally more expensive. The HSE dataset contains 5 987 samples.

For the experimental data, we started from the dataset gathered by Zhuo *et al*. [84], which covers 4 604 compositions. This dataset will be referred to as EXP^{c}. The typical ML models usually provide better results when the structures are known. Hence, we adopted the matching of the experimental compositions to the most likely structures performed by Kingsbury *et al*. [121], which is available through matminer [55]. The EXP dataset obtained in this way contains a total of

2 480 samples with an associated structure. These are considered to be the true values that the multi-fidelity ML models should predict.[1]

Figure 4.1 summarizes the dataset sizes by representing a Venn diagram for the three datasets used in this chapter. The intersections are computed based on the structures. As can be seen, all datasets overlap with 2480 samples in PBE ∩ EXP, 325 in HSE ∩ EXP, and 5987 in HSE ∩ PBE. Note, in particular, that there is no data in EXP or HSE that is not in PBE.

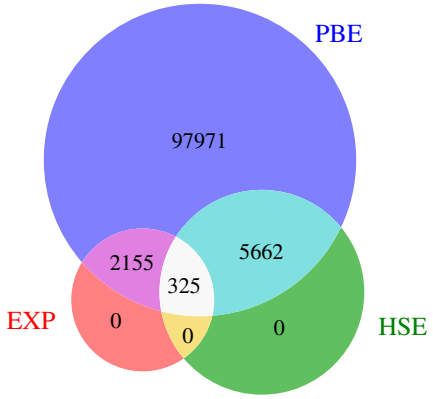


Figure 4.1: Venn diagram over the structures for the different fidelity datasets used in this chapter. Numbers represent the amount of samples in each corresponding intersection.

To test different learning approaches, we adopt the following systematic procedure. We hold out 20% of the EXP data as a test set and train on the remaining 80%. This is repeated five times and the final result is calculated as the average over the fivefold test data. This outer cross-testing guarantees a fair comparison of the different models.

4.3 MULTI-FIDELITY MODELS

We will compare a variety of multi-fidelity techniques with the standard single-fidelity MODNet approach. The single-fidelity model is trained only on the experimental data, whereas the multi-fidelity techniques also take advantage of the available knowledge from DFT calculations.

1 It is important to acknowledge that experimental values also inherently possess fidelity variations based on factors such as the employed method and crystal purity (e.g., frequently polycrystalline). In this analysis, we do not accommodate such fluctuations and treat the provided experimental values as our reference or ground truth.

The different multi-fidelity techniques investigated in this chapter are described below.

Transfer Learning — This technique, which is schematically illustrated in Figure 4.2(a), aims to take advantage of large datasets to pre-train the model. The model is first trained on a large dataset and then fine-tuned on higher-quality data. All weights of the model are free in the fine-tuning step (the same learning rate is used in the present work).

Here, a MODNet model is first trained on the dataset formed by PBE $\cup$ HSE $\cup$ EXP (i.e., regardless of the fidelity of the data). The model is further trained on the dataset formed by HSE $\cup$ EXP (i.e., with more reliable band gap values). A final fine-tuning step is performed using only the EXP dataset.

Joint Learning — This technique, which is schematically illustrated in Figure 4.2(b), consists of having a single model that predicts multiple targets at once with a shared architecture. It is similar to *Transfer Learning*, but the properties are learned in parallel rather than sequentially.

This approach can improve the accuracy of the predictions, compared to training a model for each target separately. In our case, instead of training only the EXP dataset, we also use the corresponding PBE values. Although technically nothing prevents it, we chose not to use the corresponding HSE values. Indeed, this would have considerably reduced the size of the training set since *Joint Learning* requires to use only the data that is common to all considered datasets (only 325 samples, if considering PBE $\cap$ HSE $\cap$ EXP). Note that it would also be possible to weight the different targets in the loss function.

Stacking Ensemble Learning — This technique, which is schematically illustrated in Figure 4.2(c), consists in combining the predictions of different (weak) learners (i.e., sub-models). We will train three submodels respectively on the PBE, HSE and EXP datasets. The predictions of these models will be referred to as $\hat{E}_{\mathrm{PBE}}$, $\hat{E}_{\mathrm{HSE}}$, and $\hat{E}_{\mathrm{EXP}}$. Then a linear regression is used to produce the final prediction $\hat{E}_{\mathrm{EXP,f}}$:

$$\hat{E}_{\mathrm{EXP,f}} = \alpha\hat{E}_{\mathrm{PBE}} + \beta\hat{E}_{\mathrm{HSE}} + \gamma\hat{E}_{\mathrm{EXP}} + \delta. \tag{4.1}$$

Deep-Stacking Ensemble Learning — This technique, which is schematically illustrated in Figure 4.2(d), is based on the previous one. Basically, the last hidden layer of the three neural networks are concatenated and used as the input of a new neural network for the final prediction. This can be seen as a feature extraction procedure, where the last hidden layers are high-level descriptors for the electronic band gap. Therefore, one could in principle use another model (e.g., random forest) on top

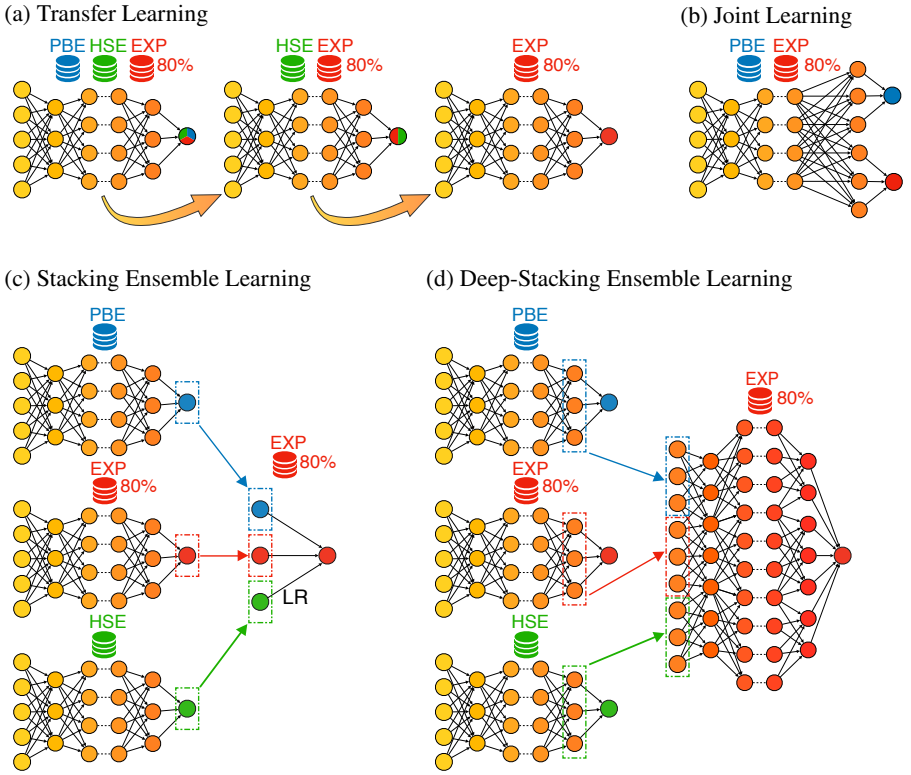


Figure 4.2: Schematic of the different multi-fidelity methods. (a) *Transfer Learning*: a model is sequentially trained on PBE ∪ HSE ∪ EXP, HSE ∪ EXP, and EXP. (b) *Joint Learning*: a model learns on multiple targets at once in parallel, by using a shared architecture. (c) *Stacking Ensemble Learning*: three submodels are separately trained on the three data sources. A linear regression (LR) is then fitted from the submodel predictions to the experimental value. (d) *Deep-Stacking Ensemble Learning*: same principle as stacking, except the last layer of each submodel is fed to a neural network.

of these extracted features. It has the advantage to have more input information than the stacking ensemble.

PBE as a feature — This technique consists in using the PBE results as an additional feature for the model that is trained on the EXP dataset. Note that, for new predictions, a PBE calculation is thus explicitly needed with this technique. In fact, the HSE results could also have been used as an additional feature. This would, however, reduce the size of the training set since all the structures in the EXP dataset are not necessarily in the HSE one.

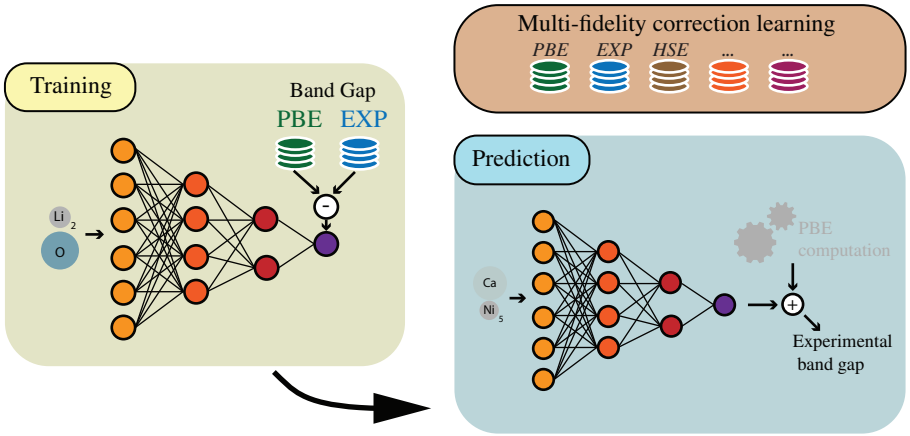


Figure 4.3: Schematic of the Correction Learning approach.

Correction Learning — This technique, illustrated in Figure 4.3, consists in learning the correction that needs to be applied to a low-fidelity data to obtain the high-fidelity one. Here, we thus use the difference $E_{\text{corr}} = E_{\text{EXP}} - E_{\text{PBE}}$ as the target of the model. Note that, for new predictions, a PBE calculation is thus explicitly needed with this technique. When evaluating the quality achieved, the MAE on E_{EXP} will be exactly that on E_{corr}. Indeed, for any given sample, the error ϵ_{EXP} on the prediction of the experimental value is actually exactly equal to the error ϵ_{corr} on the prediction of the correction:

$$\epsilon_{\text{EXP}} = E_{\text{EXP}} - \hat{E}_{\text{EXP}} = (E_{\text{PBE}} + E_{\text{corr}}) - (E_{\text{PBE}} + \hat{E}_{\text{corr}}) = E_{\text{corr}} - \hat{E}_{\text{corr}} = \epsilon_{\text{corr}} \tag{4.2}$$

4.4 RESULTS

Table 4.1 summarizes the MAE scores of band gaps using the different multi-fidelity methods. The MODNet model trained only on the EXP dataset shows a MAE of 0.382 eV (note that the dataset is different than in Table 2.1). It is chosen as the reference baseline against which all multi-fidelity learning techniques will be compared.

Despite the fact that it only contains the compositions, the MODNet model trained on the more-populated EXP^c dataset (with 4604 compositions) improves the MAE by 4%. This shows that the size of the training set (4604 vs. 2480) is actually more important than the

Learning Technique	Training sets	Samples	Test set (5-fold)	MAE (eV)	
Single-Fidelity	EXP	2480	EXP	0.382	0%
Single-Fidelity^c	EXP^c	4604	EXP	0.366	-4%
Transfer Learning	PBE $\cup$ HSE $\cup$ EXP $\rightarrow$ HSE $\cup$ EXP $\rightarrow$ EXP	2480	EXP	0.397	+4%
Joint learning	PBE $\cap$ EXP	2480	EXP	0.368	-4%
Stacking Ensemble Learning	PBE $\|\|$ HSE $\|\|$ EXP $\rightarrow$ EXP	2480	EXP	0.367	-4%
Deep-Stacking Ensemble Learning	PBE $\|\|$ HSE $\|\|$ EXP $\rightarrow$ EXP	2480	EXP	0.370	-3%
PBE as a feature	PBE $\cap$ EXP	2480	EXP	0.371	-3%
Correction Learning (PBE)	PBE $\cap$ EXP	2480	EXP	0.318	-17%
Single-Fidelity	PBE $\cap$ HSE $\cap$ EXP	325	HSE $\cap$ EXP	0.582	0%
Correction Learning (PBE)	PBE $\cap$ HSE $\cap$ EXP	325	HSE $\cap$ EXP	0.442	-24%
Correction Learning (HSE)	PBE $\cap$ HSE $\cap$ EXP	325	HSE $\cap$ EXP	0.402	-31%
Single-Fidelity	EXP^c	4604	HSE $\cap$ EXP	0.438	0%
Correction Learning (PBE)	PBE $\cap$ EXP	2480	HSE $\cap$ EXP	0.356	-19%
Correction Learning (HSE)	HSE $\cap$ EXP	325	HSE $\cap$ EXP	0.402	-8%

Table 4.1: MAE on the band gap for the different multi-fidelity learning techniques (see Figure 4.2) and for various training and test sets ($\|\|$ and $\rightarrow$ indicate that the training sets are used in parallel and one after the other, respectively). The MODNet models trained on the EXP dataset only (Single-Fidelity) is always considered as the reference baseline. Three reference baselines are available depending on the training and test sets. The MAE scores are obtained by five-fold nested cross-validation. The relative change compared to the reference baseline is indicated in the last column. In the first case, we also provide, for the sake of comparison, the results obtained with a MODNet trained using only the compositions (not the structures) on the EXP^c dataset (Single-Fidelity^c). The lowest error is achieved on Correction Learning (PBE) as tested on the full EXP dataset (MAE of 0.318 eV corresponding to a reduction of 17%).

knowledge of the structure. From this observation, we may consider that an improvement by more than 4% can be regarded as significant.

Most of the investigated learning techniques do not overcome this threshold. *Transfer Learning* even leads to a deterioration by 4%. The other techniques lead to an improvement of 3-4%, similar to what can be obtained by exploiting all the data with the compositions. On the contrary, the *Correction Learning* technique based on PBE calculations induces a strong enhancement of 17% in the MAE score (13% with respect to the results obtained with all compositions). This can probably be related to the fact that the PBE gap already corresponds to a large fraction of the experimental gap and the ML model only needs to account for a small fraction of it. Although the model taking the PBE value as a feature could mathematically reach the same performance, it does not seem to find this minimum by itself. As a result, explicitly providing the correction seems to be more favorable. This is a clear advantage over the other techniques. Its main drawback is that it requires to perform a PBE calculation.

Based on these findings, we further investigate whether *Correction Learning* based on HSE calculations, which are known to provide more accurate estimates of the experimental band gaps, improves the results further or not. To this end, we perform two series of tests. In the first, the training set is purposely limited to HSE $\cap$ EXP with 325 compounds for a fair comparison; while in the second, the training set consists of all available data. In both cases, the same test set HSE $\cap$ EXP is used.

In the first case, the *Correction Learning* technique leads to an improvement of 24% and 31% with respect to the baseline reference when based on PBE and HSE calculations, respectively. Therefore, we conclude that, when the lower-fidelity datasets contain the same number of samples, it is better to apply the *Correction Learning* technique starting from the best data among those.

However, in a real situation, such higher-fidelity data (here, the HSE band gaps) are scarcer than lower-fidelity ones. Therefore, we further benchmark the method taking into account all available data. Three realistic scenarios are considered (i) a composition MODNet model trained on the full experimental dataset (4604 compositions), (ii) *Correction Learning* based on PBE calculations (2480 structures) and (iii) *Correction Learning* based on HSE calculations (325 structures). Importantly, the three models are tested on a commonly compatible test set: a five-fold test on the inner 325 materials. In this second case, the *Correction Learning* technique leads to an improvement of 19% and 8% with respect to the baseline reference when based on PBE and HSE calculations, respectively. Although the HSE functional

is more accurate, the larger amount of PBE data gives the latter the edge. We conclude that in a realistic situation, the quantity of data can compensate for the lower quality. Note that this result might be (slightly) biased and limited to the 325-sample test set.

Finally, it is interesting to note that the same most relevant features are shared among all models (regardless of the target fidelity or strategy such as difference learning). This can be expected as they all are an approximation of the same physical property. They include the element and energy associated to the highest and lowest occupied molecular orbital (computed from the atomic orbitals), and various elemental statistics (such as atomic weight, column number and electronegativity). Oxidation state also plays an important role in the prediction. The 20 most relevant features are all composition based, with the sole exception of the spacegroup number.

4.5 CONCLUSION

Various techniques relying on multi-fidelity data have been presented, in order to improve band gap predictions. These techniques aim to take advantage of all the available data, from limited experimental datasets to large computational ones. Among the various methods investigated, the most promising results are obtained with the *Correction Learning* technique, which consists of training a model on the difference between the DFT calculations and the experimental measurements. This difference is considered to be a correction to the DFT results. Hence, any new prediction is obtained as the sum of the DFT band gap and the correction predicted by the ML model. Hence, a DFT calculation is still needed, which is a disadvantage compared to other techniques. However, if the PBE approximation is used, it is computationally not very expensive. Among the difference models, we studied the trade-off between the accuracy and availability of the computational data. It was concluded that the difference method trained on the PBE dataset yields better performance than the same method applied on scarce but more precise HSE data. In particular, a MAE of 0.318 eV is found, corresponding to a reduction of 17% w.r.t. training on the experimental band gap only.

Multi-fidelity correction learning can be applied to various other materials properties, with hopefully similar improvements obtained here. We therefore encourage and expect that it will be used in a wider context.

5 CONCLUSION

This PhD thesis focuses on the application of Machine Learning (ML) techniques in predicting and understanding materials properties. The main methodological contribution is the development of the MODNet model. The key difference from previous and current approaches is its generality and accuracy on limited datasets. Various materials primitives, such as structure, composition, or any numerical vector, can be provided, and any set of properties can be employed. The improved accuracy was made possible by introducing physically meaningful features, feature selection, joint learning, and efficient hyperparameter optimization with a genetic algorithm.

We further extended the framework by incorporating the concepts of uncertainty in Chapter 3. Given the vast materials space, biases or imbalances can easily be introduced into the training set. This chapter demonstrated how dataset imbalance and uncertainty can be examined and quantified through the use of dimensionality reduction techniques such as Principal Component Analysis (PCA) and ensemble-based methods. This approach led to the establishment of confidence bounds on individual predictions, paving the way for active learning through Bayesian optimization. Finally, multi-fidelity data-based techniques were introduced, with correction learning emerging as a simple yet powerful method applicable to any property. Overall, these developments furnish a robust framework to address a wide array of problems in supervised materials modelling.

From an application standpoint, MODNet emerged as the most accurate ML model for vibrational entropies to date, with an error of 0.009 meV/K/atom at room temperature. It could be particularly valuable in estimating stability at $T > 0\,\mathrm{K}$, a quantity rarely found in materials databases. We also explored the potential of active learning in the exploration of metals through the use of the ensemble model with Bayesian optimization. Despite its simplicity, it demonstrated potential for further applications and could even be integrated with multi-fidelity data. Finally, we established a powerful band gap predictor using correction learning, with a final MAE of 0.318 eV.

Future prospects

The pursuit of generality in materials science has gained significant attention in recent years. Generality is something that many other fields have seen before us. For instance, before the advent of deep convolutional neural networks in computer vision, more traditional features from image processing (e.g., edge detection, Gabor filters, color histograms) were used. Similarly in natural language processing, models such as BERT [122] or GPT [123] are used for a variety of tasks. This trend is now extending even further with the emergence of multimodal models that integrate knowledge from multiple fields, such as the successful combination of materials knowledge into Large Language Models as demonstrated in Ref. [124].

However, one prominent challenge facing materials science remains the limited amount of available data compared to image and text domains. To overcome this, concerted efforts are needed to expand materials data resources. Encouragingly, the landscape is evolving. With the advent of active learning, automated labs can dramatically increase the pace of research. Robotic systems equipped with AI algorithms can perform experiments around the clock, optimizing conditions and even designing new experiments based on the results. Moreover, advancements in computational power, including the potential impact of quantum computing, will significantly accelerate ab-initio high-throughput screening. Finally, emphasizing open and collaborative platforms is also critical for developing robust and powerful future methods, adhering to the FAIR principles (Findable, Accessible, Interoperable, and Reusable) for data management and sharing.

Amid these advancements, machine learning techniques already possess great applicative potential in materials science. However, their full potential remains unused, as many experimental scientists have yet to fully embrace their benefits for the design and synthesis of novel compounds. Nevertheless, a shift is beginning to occur, with industries progressively integrating AI into their pipelines.

Looking ahead, current and future developments in materials science and machine learning promise to unveil a new frontier. The vast space of materials holds many secrets, and a key challenge will be the autonomous and unbiased exploration of this immense domain. A symbiotic approach that leverages all available paradigms – experimental,

theoretical, simulation-based, and data-driven – will be essential for this. Only through this holistic integration can we hope to reveal novel materials with unprecedented properties, providing innovative solutions to society's most pressing challenges.

Appendices

A

MATMINER FEATURES

This chapter lists the *matminer* features used in this thesis. Matminer is, in principle, a comprehensive framework as it contains various routines for (i) obtaining materials properties from various databases, (ii) featurizing complex materials attributes (e.g., composition, crystal structure, band structure) into physically-relevant numerical quantities, and (iii) graphically analyzing the results of data mining. However, in this thesis, only the second aspect of the package is utilized.

The different available featurizers are presented with an in-depth physical explanation of the generated features, crucial for enhancing interpretation and expertise concerning the available features.

A.1 COMPOSITION

Composition-based features rely on the composition object, as defined by the `pymatgen` library. This composition object mainly describes which elements are present and in which proportions. Examples of compositional features are the mean atomic number or the mean electronegativity of the constituent elements of the material. Such type of features have been shown to work well on predicting inorganic crystal formation energies [42]. The complete list of different featurizers are detailed below.

ElementFraction. The composition is transformed to a vector V where V_i is set to the molar fraction of element i. In practice, each column represents a chemical element and is set nonzero if it composes the considered compound. It should be noted that it creates a rather large sparse vector, as a compound typically contains only a few elements out of the approximately hundred of possible elements.

ElementProperty, see Refs. [42], [78] and [125]. This class generates a list of various composing elemental properties related statistics. Examples are: the mean melting temperature of the constituent elements in their pure state, the mean row (or column, atomic number) in the periodic table of the constituent elements, mean number of electrons in the s, p, d or f-orbitals of the constituent elements, and many others. Often the standard deviation of the previous quantities are also taken

(and other statistics). For a complete list of the elemental properties, the reader is referred to the database that is used to generate them, which is `pymatgen` [78] in the present work.

Stoichiometry, see Ref. [42]. Each composition is first transformed to a vector V containing the elemental fractions V_i of the constituent elements. Then, various L^p norms are computed as features. For example SiO_2 is represented by the vector $(\frac{1}{3}, \frac{2}{3})$, which is subsequently transformed to the feature vector (2,0.745,...), corresponding to the $(L^0, L^2, ...)$-norm of the former vector.

AtomicOrbitals, see Ref. [126]. The highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO) are estimated from the atomic orbital energies of the composition. The feature consists in the HOMO/LUMO character (s, p, d or f) and the corresponding HOMO/LUMO element.

AtomicPackingEfficiency, see Ref. [127]. The first feature is an Atomic Packing efficiency (APE), which is based on the amorphous packing of hard spheres. It measures the difference between the ratio of the radii of the central atom to its neighbors and the ideal ratio of a cluster with the same number of atoms that has optimal packing efficiency. The APE can be positive or negative. The second feature is the L_2 distance between the material and a k-cluster that has a packing efficiency below a certain threshold from ideal (i.e. zero).

BandCenter, see Ref. [128]. This feature computes a rough estimation of the absolute position of the band center. It is found by taking a simple geometric mean of the electronegativities:

$$\prod_i X^{e_i/\Sigma_i} \tag{A.1}$$

where X is the Pauling electronegativity of species i, a_i is the molar fraction of the species i and $\prod_i$ is defined as $\sum_i = \sum_i e_i$.

CohesiveEnergy, see Ref. [129]. The cohesive energy of the crystal is defined as the energy (per unit cell) required to form the crystal starting from the elementary constituents (here taken as the elementary crystals).

For each compound, it is computed as:

$$E_{\text{coh}} = -E_{\text{form}} + a_i \cdot E_{\text{coh,i}} \tag{A.2}$$

where E_{form} is the formation energy of the unit cell (from the Materials Project), and $a_i * E_{\text{coh,i}}$ is the cohesive energy (taken from *matminer.utils.data.CohesiveEnergyData*) of the elemental species weighted

by their molar fraction. The last term is the change of energy required to go from the elementary crystals to the individual atoms, while the first term is the energy required to go from the individual atoms to the final compound.

ElectronAffinity, see Ref. [125]. Computes a mean electroaffinity multiplied by the formal charge of the composing anions:

$$E_{\text{aff}} = a_i \cdot E_{\text{aff},i} \cdot Q_{\text{anion},i} \tag{A.3}$$

where $E_{\text{aff},i}$ is the affinity of anion i, a_i the molar fraction ($\sum_i a_i = 1$) and $Q_{\text{anion},i}$ the formal charge (oxidation state).

ElectronegativityDiff, see Ref. [125]. For each cation i the following quantity is computed:

$$D_{\text{cation},i} = \chi_{\text{cation},i} - M_{\text{ions}} \tag{A.4}$$

with M_{ions} is defined as

$$M_{\text{ions}} = \sum_i a_i \cdot \chi_{\text{anion},i} \tag{A.5}$$

where χ is the electronegativity and a_i the molar fraction of cation i. From the vector D_i, different statistics can be computed as final features, such as the mean and/or standard deviation.

IonProperty, see Ref. [42]. This class produces three features. The first one is a boolean set to TRUE if a neutral ionic compound is possible, i.e. if the sum of the weighted oxidation states by their molar fraction sum up to zero. The second and third feature computes the maximum and average ionic character of the composing atomic bonds (one for each pair of atoms, only based on composition and thus neglecting neighbors). This ionic character is based on the difference in electronegativity between the pair of atoms.

Miedema, see Ref. [130]. This feature represents the formation enthalpy of intermetallic compounds, solid solutions, and amorphous phases using the semi-empirical Miedema model. It is based on a set of self-consistent equations (see Ref. [130]), solved numerically, from the knowledge of the elemental properties. The final feature is the formation enthalpy of the binary/ternary alloy.

OxidationStates, see Ref. [125]. This featurizer simply gathers the oxidation states of the constituent elements and computes various statistics about them such as mean, maximum, minimum and variance.

TMetalFraction, see Ref. [125]. This featurizer generates a single feature obtained by computing the molar fraction of all transition metals (Ti, Fe, Co, Ni, Cu, Nb,...) in the compound's composition. For example $CaNiGe_2$ is transformed into 0.25.

ValenceOrbital, see Ref. [42]. For each element of the composition the valence electrons are determined as well as their orbitals. Then a variety of statistics about these valence electrons are computed. Some examples are the mean number of s-electrons, fractions of p-electrons or the average deviation of the number of d-electrons of the constituent atoms.

YangSolidSolution, see Ref. [131]. Yang *et al.* developed two features that are useful for predicting whether metal alloys will form metallic glasses, crystalline solid solutions, or intermetallics. The idea is that the formation of compounds is both ruled by thermodynamics, and the sizes of the atoms and their respective mismatch. Therefore, the first feature, Ω, is related to mixing thermochimestry, by computing the balance between the mixing entropy and mixing enthalpy of the liquid phase:

$$\Omega = \frac{T_m \Delta S_{mix}}{\|\Delta H_{mix}\|} \tag{A.6}$$

Where T_m is the average melting temperature, ΔS_{mix} is the ideal mixing entropy, and ΔH_{mix} is the average mixing enthalpy taken over all pairs of elements in the considered alloy.

The second feature, δ, is related to the atomic size mismatch between the different elements in the material. It is defined as:

$$\delta = \sqrt{\sum_{i=1}^{n} c_i \left(1 - \frac{r_i}{\bar{r}}\right)^2} \tag{A.7}$$

where c_i and r_i are respectively the fraction and radius of element i, and $\bar{r}$ is the fraction-weighted average of the radii. This number deviates from zero all the more the radii are different. Atomic radii are taken from Ref. [132].

A.2 STRUCTURE

GlobalSymmetryFeatures, see Ref. [133]. Three features are generated related to the crystal symmetry. The first feature is the spacegroup number of the crystal. The second is the crystal system (cubic, tetragonal, orthorhombic, hexagonal, trigonal, triclinic or monoclinic) where the

seven possible systems are mapped to an integer (1 to 7). The last feature is a boolean representing whether the structure is centrosymmetric, i.e. whether it has an inversion symmetry.

MaximumPackingEfficiency, see Ref. [54]. This single feature represents the packing efficiency of the crystal by measuring the volume fraction of occupied atoms. The algorithm first determines the largest radius for each atom, which equals the distance from the center of the atom to the closest Voronoi face. Then, the packing efficiency is computed as the following volume ratio:

$$\frac{1}{V_{cell}} \sum_i \frac{4}{3} \pi r_i^3 \tag{A.8}$$

where V_{cell} is the volume of the unit cell and r_i the largest radius (as defined before) of each atom i in the unit cell.

RadialDistributionFunction. The radial distribution function (RDF) or pair correlation function $g(r)$ represents the density of atoms at a certain distance r from a reference atom, see figure A.1. More precisely, the number of atoms between two spheres of radius r and $r + dr$ centered around a reference atom is given by:

$$dN(r) = 4\pi r^2 g(r) dr. \tag{A.9}$$

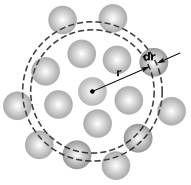


Figure A.1: Radial distribution function $g(r)$ represents the probability to find a particle between a distance r and $r + dr$, illustrated by the dotted circles. Image taken from Ref. [134].

It can thus be seen as the probability of finding a particle at a distance r away from a given reference particle, relative to that for an ideal gas. Moreover, it can be used to calculate the coordination number, crystallinity and other quantities.

In practice, the algorithm computing the RDF gathers the distances between all atom pairs in the unit cell and forms a corresponding histogram with a certain bin size. This corresponds to a mean RDF over all sites of the unit cell as reference atom. Finally, this histogram is represented as a vector, where each element is a bin, giving the final feature.

DensityFeatures. Generates three density related features. The first is the density of the material (mass per unit volume). The second feature is the volume per atom or 'vpa' computed by dividing the unit cell volume by the number of sites present. The third and last feature is the packing fraction computed by dividing the sum of the atomic

volumes (estimated by $\frac{4}{3}\pi r_i^3$, with r_i the species radius) by the unit cell volume.

BondFraction, see Ref. [135]. From the compound structure, all atomic bonds are identified by a nearest neighbor search. A bond is defined here only by the pair of atoms participating in this bond, for example Si-O. When all the bonds are enumerated, the fraction of each bond is returned as a feature element. For example for a structure containing 2 Li-O bonds and 3 Li-P bonds, this is represented by (Li-O: 0.4, Li-P: 0.6). It should be noted that for a complete dataset, the length of the feature vector (number of features) corresponds to the number of all possible occuring bonds, which is often in the order of a few thousands. Finally, this featurizer class requires for construction an instance of the pymatgen nearest neighbor class. In the present work, `CrystalNN()` is used.

CoulombMatrix, see Ref. [136]. The Coulomb-matrix, as it name suggests, represents the electrostatic Coulomb interactions within a structure or molecule in form of a matrix. It was originally proposed by Rupp *et al.* [136] for predicting atomization energies of a diverse set of organic molecules. The Coulomb matrix M off-diagonal elements are defined as,

$$M_{ij,i\neq j} = \frac{Z_i \cdot Z_j}{\|R_i - R_j\|} \tag{A.10}$$

and the diagonal elements are defined as,

$$M_{ii} = 0.5\ Z_i^{2.4} \tag{A.11}$$

where Z_i and R_i denote the nuclear charge and the position of atom i, respectively. Off-diagonal elements (M_{ij}) represents the Coulomb repulsion between two atoms i and j, while diagonal elements (M_{ii}) encode a polynomial fit of the atomic energies to nuclear charge of atom i. It can be very simply extended to crystals by computing the Coulomb matrix for the *unit cell* (instead of a molecule). In order the use the Coulomb matrix for ML purposes, it should be transformed to a fixed length vector. Therefore, the eigenvalues of the Coulomb matrix are computed and ordered from largest to smallest. This flattens the matrix but does still not make it size invariant, because not all unit cells have the same number of atoms and therefore not the same number of eigenvalues, which equals the dimensionality of the Coulomb matrix. To solve this, the right end of the vector is zero-padded, which corresponds to extending missing eigenvectors to zero, such that every

compound has the same length vector (this length is thus equal to the maximum number of species present in a particular compound).

Note that this feature has the following interesting properties [136]: (i) any system is uniquely encoded because stoichiometry as well as atomic configuration are explicitly accounted for, (ii) symmetrically equivalent atoms contribute equally, (iii) the diagonalized matrix M is invariant with respect to atomic permutations, translations, and rotations, and (iv) the eigenvectors are continuous with respect to small variations in inter-atomic distances or nuclear charges.

SineCoulombMatrix, see Ref. [37]. As described before, the Coulomb Matrix was initially designed for organic molecules. As it was shown to be successful, it was desirable to extend it to periodic crystals. Faber *et al.* [37] proposed the sine-Coulomb matrix, among others, in order to account for the interaction between repeating atoms in the crystal lattice. These long range electrostatic interactions over the crystal lattice can be represented by correcting the Coulomb matrix with an arbitrarily chosen two point potential $\Phi(\vec{r_i}, \vec{r_j})$:

$$M_{ij} = \begin{cases} Z_i \cdot Z_j \cdot \Phi(\vec{r_i}, \vec{r_j}) \\ 0.5 \cdot Z_i^{2.4} \end{cases} \tag{A.12}$$

This two-point potential is a function of the position of two atoms A and B, representing their interactions over the infinitely repeating grid. This potential should (i) be periodic with respect to the crystal lattice, (ii) have the same value for two equivalent atoms in neighboring cells and (iii) tend to infinity when these two atoms take the same position. A possible solution is found by setting

$$\Phi(\vec{r_i}, \vec{r_j}) = \|\mathbf{B} \cdot \sum_{k=\{x,y,z\}} \hat{e_k} sin^2[\pi \hat{e_k} \mathbf{B^{-1}} \cdot (\vec{r_i} - \vec{r_j})]\|_2^{-1} \tag{A.13}$$

where $\hat{e_x}, \hat{e_y}, \hat{e_z}$ are the coordinate unit vectors and $\mathbf{B}$ the matrix formed by taking for each line the basis vectors of the lattice.

Equations (A.12) and (A.13) defines the sine-Coulomb matrix representation of a periodic crystal structure. It has the benefit of depending only on the relative positions of the atoms in the unit cell and hence, has a minimal computational load.

ChemicalOrdering, see Ref. [54]. This featurizer generates a single feature describing the chemical order of the species within the structure, by computing how much the structure differs from random. The first step of this calculation is to determine the nearest neighbor shell of

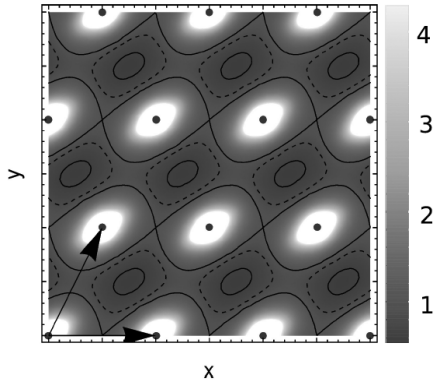


Figure A.2: Illustration of $\Phi(\vec{r_1}, \vec{r_2})$ used in the sine-Coulomb matrix representation, Eq. (A.13), for a 2D crystal lattice represented by the dark arrows and purple dots. One atom is moved following $r_1 = (x, y)$ while another is fixed at the origin $r_2 = 0$. Image taken from Ref. [37].

each site s. Then, a degree of order $\alpha(t, s)$ is determined, for each site s and chemical species t defined as:

$$\alpha(t, s) = 1 - \frac{\sum_n w_n \delta(t - t_n)}{x_t \sum_n w_n} \tag{A.14}$$

where w_n is the weight associated with a certain neighbor (i.e. how far it is from the central site), t_n is the type (chemical element) of the neighbor, and x_t is the molar fraction of type t in the structure. For atoms that are randomly dispersed in a structure or unary compounds, this formula yields 0 for all types. For structures where each site is surrounded only by atoms of another type, this formula yields large values of α. The mean absolute value of this parameter across all sites is used as a feature.

EwaldEnergy, see Ref. [133]. This featurizer generates a single feature representing the Ewald energy based on Coulombic interactions using the charges defined in the structure. For this, the corresponding `EwaldSummation` function from *pymatgen* is used.

StructuralHeterogeneity, see Ref. [54]. This featurizer computes several statistics derived from the Voronoi tessellation of a structure. First, the mean bond length of each site is computed. This is done by taking all neigbours (i.e. atoms that share a common Voronoi face) of a site and taking the corresponding mean. From this, three features are generated: the minimum average site bond length, the maximum average site bond length and the mean absolute deviation of

the average site bond length (over the sites). They are then normalized by dividing by the mean (over the sites) average (over the neighbors) site bond length. Finally, a last feature is formed by computing the mean absolute deviation of the Voronoi cell volume (over the sites) divided by the mean Voronoi cell volume.

XRDPowderPattern, see Ref. [37]. For each crystal a powder diffraction pattern can be obtained that forms a fingerprint of the crystal. The `XRDPowderPattern` featurizer represents this diffraction pattern as obtained from pymatgen where each feature $f_{2\theta}$ represents the diffraction intensity at angle 2θ. The intensities are smeared and normalized according to a Gaussian kernel, i.e. the intensities are seen as a probability density function (PDF) of a random variable (the angle here) and are estimated by the kernel density estimation in a non-parametric way.

A.3 SITE

Site based features are features based on the individual sites of the unit cell, i.e. a specific atomic position in the unit cell. Some examples are the coordination number of a site (number of neighbors) or the mean bond length of a specific site. More generally, one can introduce the concept of Order Parameter (OP). An OP is a mathematical construct that provides a numerical measure of the local environment of an atom. One of the simplest OP is the coordination number (CN) q_{CN} [137]:

$$q_{\mathrm{CN}} = \sum_{j \neq i} S\left(\left\|\mathbf{r}_j - \mathbf{r}_i\right\|\right)$$

where $\mathbf{r}_j$ is the position of neighbor atom j and S a function that is 1 if the argument is smaller then a threshold distance r_{cut} and $S = 0$ otherwise. This OP can be further sophisticated by using a distance-weighted function such as the Fermi function [137] :

$$q_{\mathrm{CN,\,Fermi}} = \frac{1}{\exp\left[\kappa\left(d_{i,j} - r_{\mathrm{cut},i}\right)\right] + 1}$$

where κ^{-1} represents the transition width from 1 towards 0 as $d_{i,j}$ increases. Note that these OPs do not encode any geometrical arrangement of the atoms. Therefore more advanced OP have been introduced, and will be discussed below.

Finally, because these type of features are intrinsically tailored for a specific site, they are more suited for applications involving individual

site characteristics, such as vacancy energies or diffusion coefficients. The different featurizers within matminer are detailed below.

OPSiteFingerprint, see Ref. [138]. In order to overcome the limitations of simpler OPs, Zimmerman *et al.* [138] introduced a set of OPs that provide information on how closely an environment resembles a given structural motif. Different common structural motifs are shown in figure A.3.

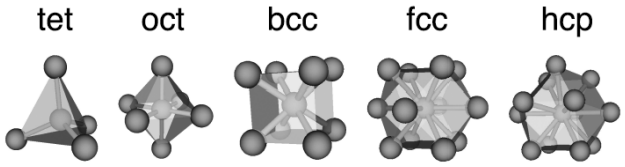


Figure A.3: Basic structural motifs that frequently recur in various material databases, from left to right: tetrahedron (tet), octahedron (oct) as well as body-centered cubic (bcc), face-centered cubic (fcc), and hexagonal close packed (hcp) motifs. The central atom and bonds are shown in orange, whereas the coordinating atoms are displayed in blue. From Ref. [138].

For each structural motif a corresponding OP is formed that describes how closely it matches the motif, from 0 (no match) to 1 (match). The computation is done in two steps. First, the first neighbor shell is localized. This is done by determining the distance $d_{min,i}$ of the nearest neighbor, of a given site i and subsequently considering all additional sites that are at maximum $r_{\text{cut},i} = (1+\delta)d_{\text{min},i}$ apart from site i where δ denotes a neighbor-finding tolerance. Two other approaches of finding the neighbors are presented in the reference article.

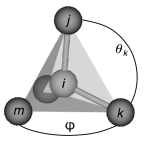


Figure A.4: Definition of atom indices, i, j, k and m as well as angles, θ (polar) and ϕ (azimuth). Used for the OPs introduced by Zimmerman *et al.* From Ref. [138].

The second step, once the first neighbor shell is determined, is to give a matching score to a specific motif. This is achieved by setting up a spherical coordinate system around the central atom i and two neighbors j and k that forms a plane from where the polar angles θ (within plane) and ϕ (out of plane) are defined, see figure A.4. This enables to localize other neighbors $l, m, ...$ If a neighbor is not located at angles that are commensurate with the expected positions of the underlying motif, the decline in reward for being away from the perfect position follows a Gaussian function. Conversely, a full reward is given if any expected remaining

position is exactly assumed. This procedure gives rotationally invariant OPs, because all possible pair of neighbors for defining the coordinate system are taken and averaged.

As an example, the tetragonal OP is given below:

$$q_{\text{tet}} = \frac{1}{N_{\text{ngh}}\left(N_{\text{ngh}}-1\right)\left(N_{\text{ngh}}-2\right)} \sum_{j\neq k}^{N_{\text{ngh}}} \left\{ \exp\left[\frac{-\left(\theta_k - 109.47^\circ\right)^2}{2\Delta\theta^2}\right] \sum_{m\neq j}^{N_{\text{ngh}}} \cos^2(1.5\varphi) \exp\left[\frac{-\left(\theta_m - 109.47^\circ\right)^2}{2\Delta\theta^2}\right] \right\}$$

where the angles θ and ϕ are defined as above, N_{ngh} denotes the number of neighbors bonded to the central atom i in a motif and $\Delta\theta = 12^\circ$ is the Gaussian width. It basically computes the deviation from the expected angles (109.47° polar and 120° azimuth) in a Gaussian fashion.

Finally, the `OPSiteFingerprint` featurizer computes a bunch of OPs corresponding to various structural motifs, which forms the outputted feature vector.

VoronoiFingerprint, see Ref. [139]. This featurizer computes various statistics about the Voronoi tesselation of the compound structure. In mathematics, Voronoi tesselations are geometrical constructs that partition the space into regions based on the distance to points in a specific subset of the space [140]. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called Voronoi cells. The Voronoi diagram of a set of points is dual to its Delaunay triangulation. It is named after Georgy Voronoi, and is also called a Voronoi tessellation, a Voronoi decomposition, a Voronoi partition, or a Dirichlet tessellation. Two examples are given in figure A.5 (2D and 3D).

Once the Voronoi tesselation determined, various statistics about the site Voronoi cell (or polyhedra) is computed. The first is a vector N where N_i, called the Voronoi indices, represents the number of i-edged facets around the considered site, where i goes from 3 to 10. For example, for a bcc lattice $N = [0, 6, 0, 8, ...]$ meaning that there are six

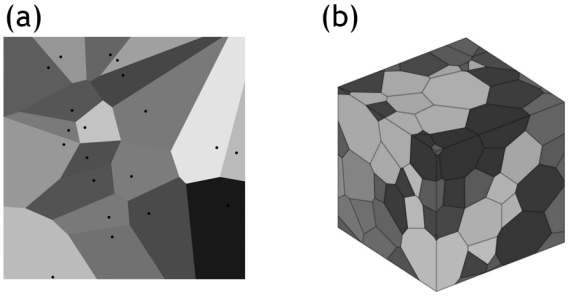


Figure A.5: Illustrative example of two Voronoi tesselations in 2D (a) and 3D (b), where the seeds are represented as black dots. Images taken from Refs. [140] (a) and [141] (b).

4-facets and eight 6-facets. For fcc/hcp lattice $N = [0, 12, 0, 0, ...]$ and for icosahedra $N = [0, 0, 12, 0, ...]$.

The second set of statistics are i-fold symmetry indices computed as:

$$S_i = \frac{N_i}{\sum_j N_j} \tag{A.15}$$

which are basically normalized Voronoi indices such that $\sum_j S_j = 1$. Optionally these indices can be weighted by their relative distance to the central site, see the matminer API for more information.

Other statistics are the Voronoi-polyhedron volume and surface, Voronoi-volume statistics of subpolyhedra formed by each facet and the center site, the facets area and finally nearest-neighbors distances (defined as the distance to the corresponding facet).

Combining these statistics into a single vector forms the feature of this featurizer.

CrystalNNFingerPrint. This featurizer first computes a score for various CN (0 to 10 for example), giving how well the local environment matches that CN. This score is computed using the `CrystalNN` class from pymatgen, which contains the `cn_weights` attribute giving a weight (or score) for a CN based on the neighbors distances. This gives a first set of features. Then, various other OPs are computed for that site that are more related to the structural motifs via the pymatgen `LocalStructOrderParams` class. This class is mainly a container to the previously explained OPs from Ref. [138], but other motifs are considered too such as the body-centered cubic OP from Ref. [142] or pentagonal bipyramids from Ref. [143]. Once those OPs are computed

they are multiplied by their corresponding coordination score to form the final feature vector.

GaussianSymmFunc, see Refs. [21] and [144]. This featurizer will encode the local environment of a site into various symmetry functions instead of Cartesian coordinates. It was originally proposed by Behler *et al.* [21] to fit potential energy surfaces for molecular dynamics. The general idea is to form a set of descriptors that encode the local environment of an atom (such as neighbor distances and angles) that is invariant with respect to rotations of the structure and has a fixed number of dimensions. This latter is due to the fact that most ML algorithms have a fixed input dimension. Moreover these descriptors should be continuous with respect to the atomic positions, in order to guarantee derivability, a necessary condition to compute force fields. The proposed solution is to use many-body symmetry functions that simultaneously depend on the positions of all atoms inside a certain cutoff sphere. The proposed set of radial symmetry functions are:

$$G_i^1 = \sum_j f_c\left(R_{ij}\right) \tag{A.16}$$

$$G_i^2 = \sum_j e^{-\eta\left(R_{ij}-R_s\right)^2} \cdot f_c\left(R_{ij}\right) \tag{A.17}$$

$$G_i^3 = \sum_j \cos\left(\kappa R_{ij}\right) \cdot f_c\left(R_{ij}\right) \tag{A.18}$$

where R_{ij} is the distance between the site i and every neighbor j, R_s a suitable shift distance, η and κ are parameters ruling respectively the width of the Gaussian and the periodicity of the cosinus and finally f_c is a cutoff function:

$$f(r) = \begin{cases} 0.5\left[cos(\frac{\pi r_{ij}}{R_c}) + 1\right] & \text{if} \quad r_{ij} \leq R_c \\ 0 & \text{if} \quad r_{ij} > R_c \end{cases} \tag{A.19}$$

with R_c is the cutoff radius. Equations (A.16)-(A.18) can be seen as the integral of the product of the RDF with a unit, Gaussian and cosine function respectively (itself multiplied by the cutoff function). In particular equation (A.16) is the sum of all neighbors weighted by the cutoff function and equation (A.18) can be seen as a Fourier decomposition. A graphical representation of the radial symmetry functions can be found in figure A.6.

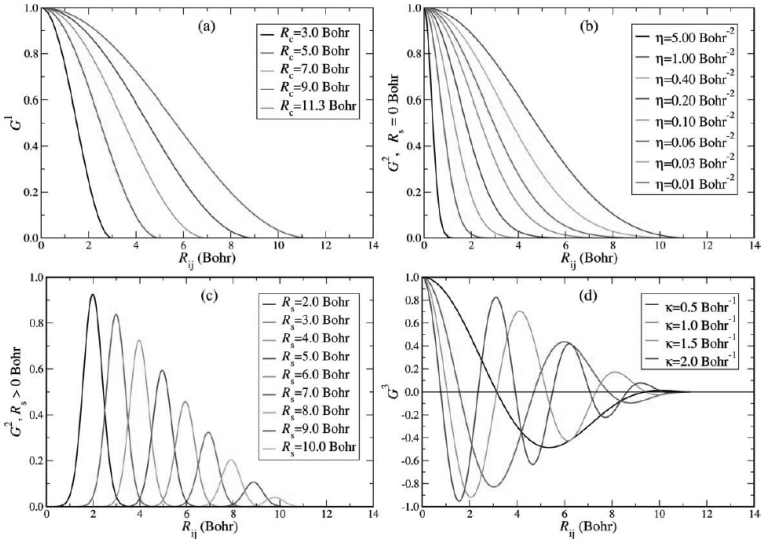


Figure A.6: Radial symmetry functions G^1, G^2, and G^3 for an atom with one neighbor only as used in the `GaussianSymmFunc` featurizer. For the G^2 and G^3 functions a cutoff $R_c = 11.3$ Bohr has been used, the Gaussian width of the shifted G^2 functions in panel (c) is $\eta = 3.0$ Bohr^2. From Ref. [21].

By computing $G^1, G^2, G3$ at various values for η and κ a feature vector encoding the radial information can be obtained.

Similarly a set of angular function is proposed:

$$G_i^4 = 2^{1-\zeta} \sum_{j,k\neq i}^{\text{all}} \left(1 + \lambda \cos\theta_{ijk}\right)^{\zeta} \cdot e^{-\eta\left(R_{ij}^2 + R_{ik}^2 + R_{jk}^2\right)} \cdot f_c\left(R_{ij}\right) \cdot f_c\left(R_{ik}\right) \cdot f_c\left(R_{jk}\right) \tag{A.20}$$

$$G_i^5 = 2^{1-\zeta} \sum_{j,k\neq i}^{\text{all}} \left(1 + \lambda \cos\theta_{ijk}\right)^{\zeta} \cdot e^{-\eta\left(R_{ij}^2 + R_{iz}^2\right)} \cdot f_c\left(R_{ij}\right) \cdot f_c\left(R_{ik}\right) \tag{A.21}$$

where θ_{ijk} is the angle formed by atoms i,j and k. The parameter λ is taken once as +1 and once as -1 (shifting the maxima of the cosine function). Parameters ζ and η are again taken over a varying range in order to obtain a set of features representing the angular geometry.

Commonly, around 50 features are generated by combining 50 different symmetry functions (G^1 to G^2 and their parameters) in order to represent a single environment.

ChemEnv, see Ref. [145]. This featurizer generates a set of OPs, the features, representing the closeness of the local environment to various

perfect coordination environment (tetrehedral, trigonal prism,...) in form of a percentage. It is therefore very similar to `OPSiteFingerprint`, but the actual implementation is different.

It first determines a set of neighbors of the central site through a Voronoi approach similar to what was proposed by O'Keeffe [146]. Atoms that share a common Voronoi face with the central atom are considered to be neighbors of the central site. Then, two cutoff parameters are introduced to discard some neighbors. The first cutoff parameter α discards neighbors that are localized further than αd_{min} where d_min is the distance to the closest neighbor. The second cutoff parameter γ discard atoms that have a solid angle formed by the common Voronoi face that is smaller than $\gamma\Omega^{max}$ where Ω^{max} is the largest solid angle formed by a common Voronoi face. Additionally other neighbors can be discarded based on their oxidation states. For example in ionic solids, cations neighboring other cations should be discarded.

Once a set of neighbors is established, it can be compared to various model polyhedra. For this, the Continuous Symmetry Measure (CSM) introduced by Pinsky and Avnir [147] is used. It defines a distance S between a distorted structure P and a perfect G-symmetric structure Q, where G is a specific symmetry group:

$$S = \min \frac{\sum_{k=1}^{N} \|Q_k - P_k\|^2}{\sum_{k=1}^{N} \|Q_k - Q_0\|^2} \times 100 \tag{A.22}$$

where $\{Q_k, k = 1, 2, \ldots, N\}$ are the coordinates of the N vertices of Q, $\left\{\hat{P}_k, k = 1, 2, \ldots, N\right\}$ the coordinates of the N vertices of P and Q_0 is the coordinate vector of the center of mass of the investigated structure:

$$Q_0 = \frac{1}{N} \sum_{k=1}^{N} Q_k \tag{A.23}$$

In order to have the best match with the perfect structure P, this latter should be rotated, dilated and translated, in order to minimize the distance S, which is represented with the *min* keyword. These operations can mathematically be written as:

$$P_k = A\mathbf{R}P_{0k} + \mathbf{T} \tag{A.24}$$

and the corresponding minimization problem:

$$J = \sum_{k=1}^{N} \|Q_k - P_k\|^2 = \sum_{k=1}^{N} \|Q_k - (A\mathbf{R}P_{0k} + \mathbf{T})\|^2 \tag{A.25}$$

with **R** a 3x3 rotation matrix, **T** a 3x3 translation matrix and A an isotropic scaling factor. This minimization is central to the CSM algorithm and is discussed in more detail in the original paper [147].

Once the optimal set of coordinates are found for P, the corresponding distance S can be computed. It should be noted that S is bounded within the interval $[0, 100]$. The symmetry measure attains its minimum value ($S = 0$) when the structure has exactly the investigated symmetry.

Going back to the `ChemEnv` featurizer [145], it uses a specific *strategy* to find the different percentages for the different coordination environments based on the CSM. First, for more robustness, different sets of neighbors are considered by varying the cutoff parameters: $\alpha \in [1.15, 2.0]$ and $\gamma \in [0.05, 0.75]$. Secondly, for each set of neighbors (corresponding to a set of cutoff parameters) different fractions or percentages are given for different possible polehedra. This is done by multiplying two factors. The first is a weight that is a decreasing monotonic mapping of the CSM distance, giving higher weights to closer structures. The second weight factor favor neighbor-sets of higher coordination (even if they are more distorted, i.e. higher CSM distance).

Finally, this framework enables one to generate a feature vector giving a percentage to the closeness of different local coordinations, and doing so, it considers local environments as a superposition of different polyhedra rather then one alone, giving a more robust descriptor.

AGNIFingerprints, see Ref. [148]. This feature represents the local environment of an atom by computing the integral of the product of the radial distribution function and a Gaussian window function. It is therefore very similar to the work of Behler *et al.* [21]. It has the advantage of giving a vector $G(\eta)$ that is not sparse compared to the RDF, and thus much more suited for similarity comparison that are computed by taking the L^2-norm of the difference or the scalar product between two vectors, which would respectively be one or zero for the RDF. Originally, Botu *et al.* [148] used it to fit empirical energy potentials. This OP is defined as:

$$G_i(\eta) = \int R_i(r) e^{-(\frac{r_{ij}}{\eta})^2} dr \tag{A.26}$$

where i is the considered site, $R(r)$ the RDF and r_{ij} the distance between site i and neighbor j. By introducing the same cutoff-function $f(r_{ij})$ as Behler *et al.* [21] and discretizing the integral, one has:

$$G_i(\eta) = \sum_{i \neq j} e^{-(\frac{r_{ij}}{\eta})^2} f(r_{ij}) \tag{A.27}$$

which forms the undirected AGNI fingerprint, a vector containing $G(\eta)$ at different η's, a first set of features. The cutoff-function $f(r_{ij})$ is taken as:

$$f(r) = \begin{cases} 0.5\left[cos(\frac{\pi r_{ij}}{R_c}) + 1\right] & \text{if} \quad r_{ij} \leq R_c \\ 0 & \text{if} \quad r_{ij} > R_c \end{cases} \tag{A.28}$$

In order to introduce directionality to the feature, which is interesting for fitting vector fields (e.g force fields), the previous equation is extended to direction-resolved atomic fingerprints $V_i(\eta) = \{V_i^x(\eta), V_i^y(\eta), V_i^z(\eta)\}$ as follows:

$$V_i^k(\eta) = \sum_{i \neq j} \frac{r_{ij}^k}{r_{ij}} e^{-(\frac{r_{ij}}{\eta})^2} f(r_{ij}) \quad , \quad k \in \{x, y, z\} \tag{A.29}$$

where r_{ij}^k is the k-th component of $(\vec{r}_i - \vec{r}_j)$. This gives a second set of features $V_i(\eta)$. It should however be noted that this latter set of features depend on the axes of the reference system and is therefore not invariant with respect to rotations.

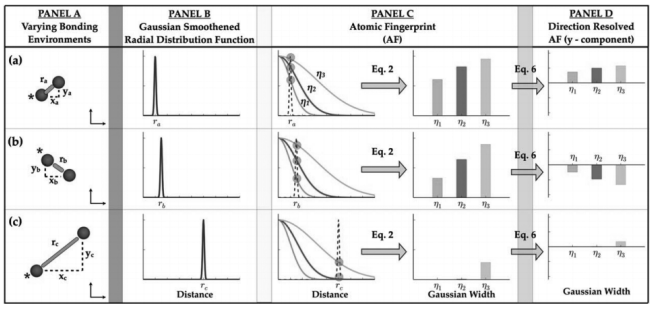


Figure A.7: Schematic representation of the AGNI features. They are formed by transformation the RDF using Gaussian functions on an eta-grid as indicated by the colored lines in panel C. The direction resolved fingerprints are showed in panel D. Image taken from Ref. [148].

CoordinationNumber, see Refs. [149] and [150]. This featurizer generates a unique feature representing the coordination number of a given site, i.e. the number of nearest neighbors. In order to determine the nearest neighbors, it uses one of pymatgen `NearNeighbor` classes such as `VoronoiNN`, `JmolNN`, `MinimumDistanceNN`, `MinimumOKeeffeNN` or `MinimumVIRENN`. In the present work, `VoronoiNN` is used. Neighbors are

determined based on the surface of the common Voronoi face with the central site. This surface can be translated into a weight for each neighbor and can optionally be incorporated into the coordination number by summing over the weights of the different neighbors. This weighted sum avoids that the coordination number changes abruptly with small perturbation of the local environment.

GeneralizedRDF, see Ref. [43]. This featurizer generalizes the RDF by using non-rectangular bins. Remember that for the RDF, each feature represents the number of bonds that are within a certain length interval, which can be written as:

$$x_k = \int_{k\Delta}^{(k+1)\Delta} g(r)dr \tag{A.30}$$

where g(r) is the RDF, Δ the bin width and x_k the k-th feature ($k \in [0, \lceil r_{cuttof}/\Delta \rceil]$, with $\lceil \cdot \rceil$ the ceiling function).

The GRDF generalizes this latter to:

$$x_k = \int_0^{r_{cuttof}} H(r - ((k + \frac{1}{2})\Delta)g(r)dr \tag{A.31}$$

where $H(r)$ is a binning function centered on the considered interval. The bins are now potentially overlapping and thus not mutually exclusive. In the present work the default Gaussian preset is used for $H(r)$. Note that taking $H(r) = 1$ from $k\Delta$ to $(k+1)\Delta$ and o otherwise is equivalent to equation A.30. The RDF is thus a particular case of the GRDF. Note that x_k can also be seen as:

$$\begin{cases} f(r) = \sum_j H(r_{ij} - r) \\ x_k = f((k + \frac{1}{2})\Delta) \quad , \quad k \in \mathbb{N} \end{cases} \tag{A.32}$$

where j are the neighbors of site i. In other words, the RDF is convoluted with H(r) (if one assumes that the RDF are Dirac-δ functions centered around the neighbor positions).

BondOrientationalParameter, see Ref. [143, 151]. In order to asses local symmetries to local bonds around a particular site, these bonds are associated to spherical harmonics. Spherical Harmonics are a set of radial functions Y_{lm} that are solution to the Laplace equation,

$$Y_{lm}(\theta, \phi), \tag{A.33}$$

where l and m are integer parameters with $m \in [-l, l]$. Spherical harmonics are therefore solutions of the angular part of the Schrödinger

equation for a radial potential (e.g hydrogen atom), where l is related to the angular momentum and m the magnetic momentum. A graphical representation of the first spherical harmonics are drawn in figure A.8.

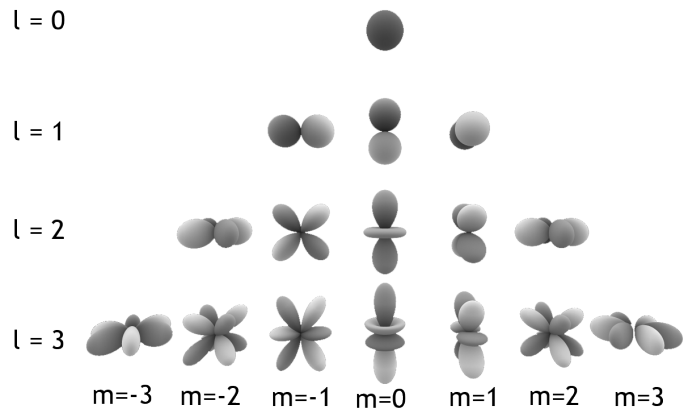


Figure A.8: Illustration of the first few real spherical harmonics $Y_{lm}(\theta, \phi)$. Blue zones (resp. yellow zones) represent regions where the function is positive (resp. negative). The distance of the surface from the origin indicates the value of $Y_{lm}(\theta, \phi)$ in the angular direction (θ, ϕ). Figure adapted from Ref. [152].

As spherical harmonics represent quite well different types of angular symmetries, Steinhardt *et al.* proposed to use them to describe the local environment of an atomic site. The idea is to associate as set of numbers Q_{lm} to a particular site, representing how well it matches with a particular spherical harmonic [143]

$$Q_{lm}(\overrightarrow{\mathbf{r}}) \equiv Y_{lm}(\theta(\overrightarrow{\mathbf{r}}), \phi(\overrightarrow{\mathbf{r}})) \tag{A.34}$$

where $\theta(\vec{r})$ and $\phi(\vec{r})$ are the polar angles of a particular bond with midpoint $\vec{r}$. In order to take into account all bonds of a site, a mean $\overline{Q}_{lm}$ is taken over all bonds:

$$\overline{Q}_{lm} \equiv \langle Q_{lm}(\vec{r}) \rangle \,. \tag{A.35}$$

Moreover, to guarantee rotational invariance, the following combination is defined:

$$Q_l \equiv \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} \left\| \overline{Q}_{lm} \right\|^2 \right)^{1/2} \tag{A.36}$$

which forms the final feature vector by taking l from 1 to 10.

This feature vector forms a fingerprint of the local symmetry, for example for a cubic symmetric environment, the first nonzero element is Q_4, i.e. when $l = 4$. For an icosahedrally oriented system the first nonzero element occur for $l = 6$.

The Bond Orientational Parameter was initially introduced by Steinhardt *et al.* to study order in liquids and glasses in function of the temperature. It has later been used by Seko *et al.* [43] in the machine learning field to predict melting temperatures of single and binary compounds.

LocalPropertyDifference, see Ref. [54]. This featurizer computes the difference in elemental properties between the site and its neighbors, such as the mean difference in atomic number. It is therefore necessary to first determine the neighbors, which is done here by the Voronoi-tesselation approach. Each neighbor is weighted proportionally to the Voronoi facet (face shared between the site's cell and the neighbor's cell). The local property difference is then computed by the following weighted average:

$$p_{diff} \frac{\sum_n A_n \, \|p_n - p_0\|}{\sum_n A_n} \tag{A.37}$$

where A_n is the weight of neighbor n, p_n the property corresponding to neighbor p and p_0 the property corresponding to the central site. This finally forms the feature vector by computing p_{diff} over different elemental properties.

AverageBondLength, see Ref. [153]. This class generates the mean bond length over all Voronoi neighbors, by means of a weighted average, similar to equation (A.37) where p_n is replaced by the corresponding bond length.

AverageBondAngle, see Ref. [153]. The mean bond angle is computed between the site and the set of pairs of closest neighbors. More precisely, a matrix A is first generated containing the bond angle A_{ij} between site s, neigbor i and neigbor j, for all neigbors determined with a pymatgen `NearNeigbor` class such as `VoronoiNN`. Then for each line (i.e. neighbor) only one angle is kept, the smallest, called the neighbor angle. Finally the average is taken over all neighbor angles. Note that the angles are not weighted here.

BIBLIOGRAPHY

[1] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).

[2] Schleder, G. R., Padilha, A. C. M., Acosta, C. M., Costa, M. & Fazzio, A. From DFT to machine learning: Recent approaches to materials science–a review. *J. Phys. Mater.* **2**, 032001 (2019).

[3] Schmidt, J., Marques, M. R. G., Botti, S. & Marques, M. A. L. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* **5**, 83 (2019).

[4] Choudhary, K. *et al.* Recent advances and applications of deep learning methods in materials science. *npj Comput. Mater.* **8**, 59 (2022).

[5] Bardeen, J. & Brattain, W. H. The Transistor, A Semi-Conductor Triode. *Phys. Rev.* **74**, 230–231 (1948).

[6] Lewis, N. S. Toward Cost-Effective Solar Energy Use. *Science* **315**, 798–801 (2007).

[7] Chen, X., Li, C., Grätzel, M., Kostecki, R. & Mao, S. S. Nanomaterials for renewable energy production and storage. *Chem. Soc. Rev.* **41**, 7909–7937 (2012).

[8] Snyder, G. J. & Toberer, E. S. Complex thermoelectric materials. In *Materials for Sustainable Energy*, 101–110 (Co-Published with Macmillan Publishers Ltd, UK, 2010).

[9] Zhao, Y., Zhang, Z., Pan, Z. & Liu, Y. Advanced bioactive nanomaterials for biomedical applications. *Exploration* **1**, 20210089 (2021).

[10] Magee, C. L. Towards quantification of the role of materials innovation in overall technological development. *Complexity* **18**, 10–25 (2012).

[11] Advanced Functional Materials Market Size, Trends | Industry Forecast 2026. https://www.technavio.com/report/advanced-functional-materials-market-industry-analysis. Last accessed: 2023-07-31.

[12] Davies, D. W. *et al.* Computational Screening of All Stoichiometric Inorganic Materials. *Chem* **1**, 617–627 (2016).

[13] Zagorac, D., Müller, H., Ruehl, S., Zagorac, J. & Rehme, S. Recent developments in the Inorganic Crystal Structure Database: Theoretical crystal structure data and related features. *J. Appl. Crystallogr.* **52**, 918 (2019).

[14] Jain, A. *et al.* The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* **1**, 011002 (2013).

[15] Agrawal, A. & Choudhary, A. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Materials* **4**, 053208 (2016).

[16] Dirac, P. A. M. Quantum mechanics of many-electron systems. *Proc. R. Soc. Lond. A* **123**, 714–733 (1929).

[17] Hohenberg, P. & Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **136**, B864–B871 (1964).

[18] Saal, J. E., Kirklin, S., Aykol, M., Meredig, B. & Wolverton, C. Materials design and discovery with high-throughput density functional theory: The open quantum materials database (OQMD). *JOM* **65**, 1501–1509 (2013).

[19] Hellenbrandt, M. The Inorganic Crystal Structure Database (ICSD)—Present and Future. *Crystallogr. Rev.* **10**, 17–22 (2004).

[20] Curtarolo, S. *et al.* AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Computational Materials Science* **58**, 227–235 (2012).

[21] Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **134**, 074106 (2011).

[22] Meredig, B. *et al.* Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. *Mol. Syst. Des. Eng.* **3**, 819–825 (2018).

[23] Stanev, V. *et al.* Machine learning modeling of superconducting critical temperature. *Npj Comput. Mater.* **4** (2018).

[24] Choudhary, K. *et al.* Accelerated Discovery of Efficient Solar Cell Materials Using Quantum and Machine-Learning Methods. *Chem. Mater.* **31**, 5900–5908 (2019).

[25] Alizade, M. A., Bressler, R. & Brendel, K. Stereochemistry of the hydrogen transfer to NAD catalyzed by (S)alanine dehydrogenase from Bacillus subtilis. *Biochim. Biophys. Acta* **397**, 5–8 (1975).

[26] Botu, V., Batra, R., Chapman, J. & Ramprasad, R. Machine Learning Force Fields: Construction, Validation, and Outlook. *J. Phys. Chem. C* **121**, 511–522 (2017).

[27] Helfrecht, B. A., Cersonsky, R. K., Fraux, G. & Ceriotti, M. Structure-property maps with Kernel principal covariates regression. *Mach. Learn.* 22 (2020).

[28] Ouyang, R., Curtarolo, S., Ahmetcik, E., Scheffler, M. & Ghiringhelli, L. M. SISSO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates. *Phys. Rev. Materials* **2**, 083802 (2018).

[29] Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997).

[30] Quinlan, J. R. *C4.5: Programs for Machine Learning* (Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993).

[31] Kohavi, R. & Quinlan, R. Decision Tree Discovery (1999).

[32] Oliynyk, A. O. *et al.* High-Throughput Machine-Learning-Driven Synthesis of Full-Heusler Compounds. *Chem. Mater.* **28**, 7324–7331 (2016).

[33] Bottou, L. Large-scale machine learning with stochastic gradient descent. In Lechevallier, Y. & Saporta, G. (eds.) *Proceedings of COMPSTAT*, 177–186 (Physica-Verlag HD, 2010).

[34] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

[35] Kiselyova, N., Gladun, V. & Vashchenko, N. Computational materials design using artificial intelligence methods. *J. Alloys Compd.* **279**, 8–13 (1998).

[36] Rupp, M., Tkatchenko, A., Müller, K. . R., Lilienfeld, V. & Anatole, O. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108** (2012).

[37] Faber, F., Lindmaa, A., von Lilienfeld, O. A. & Armiento, R. Crystal structure representations for machine learning models of formation energies. *Int. J. Quantum Chem.* **115**, 1094–1101 (2015).

[38] Hautier, G., Fischer, C. C., Jain, A., Mueller, T. & Ceder, G. Finding Nature's Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory. *Chem. Mater.* **22**, 3762–3767 (2010).

[39] Saad, Y. *et al.* Data mining for materials: Computational experiments with A B compounds. *Phys. Rev. B* **85** (2012).

[40] Lam Pham, T. *et al.* Machine learning reveals orbital interaction in materials. *Sci. Technol. Adv. Mater.* **18**, 756–765 (2017).

[41] Superconducting Material Database (SuperCon). `https://mdr.nims.go.jp/collections/5712mb227`.

[42] Ward, L., Agrawal, A., Choudhary, A. & Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *Npj Comput. Mater.* **2**, 16028 (2016).

[43] Seko, A., Maekawa, T., Tsuda, K. & Tanaka, I. Machine learning with systematic density-functional theory calculations: Application to melting temperatures of single- and binary-component solids. *Phys. Rev. B* **89**, 054303 (2014).

[44] Ghiringhelli, L. M., Vybiral, J., Levchenko, S. V., Draxl, C. & Scheffler, M. Big data of materials science: Critical role of the descriptor. *Phys. Rev. Lett.* **114**, 105503 (2015).

[45] Meredig, B. *et al.* Combinatorial screening for new materials in unconstrained composition space with machine learning. *Phys. Rev. B* **89**, 094104 (2014).

[46] Kong, C. S. *et al.* Information-Theoretic Approach for the Discovery of Design Rules for Crystal Chemistry. *J. Chem. Inf. Model.* **52**, 1812–1820 (2012).

[47] Curtarolo, S., Morgan, D., Persson, K., Rodgers, J. & Ceder, G. Predicting Crystal Structures with Data Mining of Quantum Calculations. *Phys. Rev. Lett.* **91**, 135503 (2003).

[48] Fischer, C. C., Tibbetts, K. J., Morgan, D. & Ceder, G. Predicting crystal structure by merging data mining with quantum mechanics. *Nat Mater* **5**, 641–646 (2006).

[49] Hautier, G., Fischer, C., Ehrlacher, V., Jain, A. & Ceder, G. Data Mined Ionic Substitutions for the Discovery of New Compounds. *Inorg. Chem.* **50**, 656–663 (2011).

[50] Dey, P. Informatics-aided bandgap engineering for solar materials. *Comput. Mater. Sci.* **83** (2014).

[51] Pilania, G. *et al.* Machine learning bandgaps of double perovskites. *Sci Rep* **6**, 19375 (2016).

[52] Bhadeshia, H. K. D. H., Dimitriu, R. C., Forsik, S., Pak, J. H. & Ryu, J. H. Performance of neural networks in materials science. *Mater. Sci. Technol.* **25**, 504–510 (2009).

[53] Chatterjee, S., Murugananth, M. & Bhadeshia, H. K. D. H. Delta-TRIP steel. *Mater. Sci. Technol.* **23**, 819–827 (2007).

[54] Ward, L. *et al.* Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations. *Phys. Rev. B* **96**, 024104 (2017).

[55] Ward, L. *et al.* Matminer: An open source toolkit for materials data mining. *Computational Materials Science* **152**, 60–69 (2018).

[56] Dunn, A., Wang, Q., Ganose, A., Dopp, D. & Jain, A. Benchmarking materials property prediction methods: The Matbench test set and Automatminer reference algorithm. *Npj Comput. Mater.* **6**, 1–10 (2020).

[57] Olson, R. S., Bartley, N., Urbanowicz, R. J. & Moore, J. H. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In *Proc. Genet. Evol. Comput. Conf. 2016*, GECCO

'16, 485–492 (Association for Computing Machinery, New York, NY, USA, 2016).

[58] Xie, T. & Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **120** (2018).

[59] Chen, C., Ye, W., Zuo, Y., Zheng, C. & Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.* **31**, 3564–3572 (2019).

[60] Park, C. W. & Wolverton, C. Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery. *Phys. Rev. Mater.* **4**, 063801 (2020).

[61] Choudhary, K. & DeCost, B. Atomistic Line Graph Neural Network for improved materials property predictions. *npj Comput. Mater.* **7**, 185 (2021).

[62] Veličković, P. *et al.* Graph Attention Networks. *International Conference on Learning Representations* (2018).

[63] Gasteiger, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. *arXiv2003.03123* (2022).

[64] Wang, A., Kauwe, S., Murdock, R. & Sparks, T. Compositionally-Restricted Attention-Based Network for Materials Property Prediction. Preprint (2020).

[65] Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **148**, 241722 (2018).

[66] Chen, Z. *et al.* Direct Prediction of Phonon Density of States With Euclidean Neural Networks. *Adv. Sci.* **8**, 2004214 (2021).

[67] van Setten, M. J., Giantomassi, M., Gonze, X., Rignanese, G.-M. & Hautier, G. Automation methodologies and large-scale validation for G W : Towards high-throughput G W calculations. *Phys. Rev. B* **96**, 155207 (2017).

[68] Seko, A. *et al.* Prediction of Low-Thermal-Conductivity Compounds with First-Principles Anharmonic Lattice-Dynamics Calculations and Bayesian Optimization. *Phys. Rev. Lett.* **115**, 205901 (2015).

[69] Petretto, G. *et al.* High-throughput density-functional perturbation theory phonons for inorganic materials. *Sci. Data* **5**, 180065 (2018).

[70] Jovic, A., Brkic, K. & Bogunovic, N. A review of feature selection methods with applications. In *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO*, 1200–1205 (IEEE, Opatija, Croatia, 2015).

[71] Verleysen, M. & François, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In Cabestany, J., Prieto, A. & Sandoval, F. (eds.) *Comput. Intell. Bioinspired Syst.*, Lecture Notes in Computer Science, 758–770 (Springer Berlin Heidelberg, 2005).

[72] Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Phys. Rev. E* **69** (2004).

[73] Hanchuan Peng, Fuhui Long & Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Machine Intell.* **27**, 1226–1238 (2005).

[74] Mangal, A. & Holm, E. A. A Comparative Study of Feature Selection Methods for Stress Hotspot Classification in Materials. *Integr Mater Manuf Innov* **7**, 87–95 (2018).

[75] Ouyang, R., Ahmetcik, E., Carbogno, C., Scheffler, M. & Ghiringhelli, L. M. Simultaneous learning of several materials properties from incomplete databases with multi-task SISSO. *J. Phys. Mater.* **2**, 024002 (2019).

[76] Li, Z. & Hoiem, D. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 2935–2947 (2018).

[77] MODNet github website. `https://github.com/ppdebreuck/modnet`.

[78] Ong, S. P. *et al.* Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science* **68**, 314–319 (2013).

[79] Ong, S. P. *et al.* The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials

data based on REpresentational State Transfer (REST) principles. *Computational Materials Science* **97**, 209–215 (2015).

[80] Naccarato, F. *et al.* Searching for materials with high refractive index and wide band gap: A first-principles high-throughput study. *Phys. Rev. Materials* **3**, 044602 (2019).

[81] Deng, J. *et al.* ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, 248–255 (2009).

[82] Mechanical properties of some steels. `https://citrination.com/datasets/153092/`.

[83] Choudhary, K., Kalish, I., Beams, R. & Tavazza, F. High-throughput Identification and Characterization of Two-dimensional Materials using Density functional theory. *Scientific Reports* **7**, 5179 (2017).

[84] Zhuo, Y., Mansouri Tehrani, A. & Brgoch, J. Predicting the Band Gaps of Inorganic Solids by Machine Learning. *J. Phys. Chem. Lett.* **9**, 1668–1673 (2018).

[85] Petousis, I. *et al.* High-throughput screening of inorganic compounds for the discovery of novel dielectric and optical materials. *Scientific Data* **4**, 160134 (2017).

[86] Kawazoe, Y., Masumoto, T., Tsai, A.-P., Yu, J.-Z. & Aihara Jr., T. Nonequilibrium phase diagrams of ternary amorphous alloys.

[87] de Jong, M. *et al.* Charting the complete elastic properties of inorganic crystalline compounds. *Sci Data* **2**, 150009 (2015).

[88] Castelli, I. E. *et al.* New cubic perovskites for one- and two-photon water splitting using the computational materials repository. *Energy Environ. Sci.* **5**, 9034 (2012).

[89] Goodall, R. E. A. & Lee, A. A. Predicting materials properties without crystal structure: Deep representation learning from stoichiometry. *Nat. Commun.* **11**, 6280 (2020).

[90] Legrain, F., Carrete, J., van Roekeghem, A., Curtarolo, S. & Mingo, N. How Chemical Composition Alone Can Predict Vibrational Free Energies and Entropies of Solids. *Chem. Mater.* **29**, 6220–6227 (2017).

[91] Tawfik, S. A., Isayev, O., Spencer, M. J. S. & Winkler, D. A. Predicting Thermal Properties of Crystals Using Machine Learning. *Adv. Theory Simul.* **3**, 1900208 (2020).

[92] George, J. & Hautier, G. Chemist versus Machine: Traditional Knowledge versus Machine Learning Techniques. *Trends in Chemistry* S2589597420302653 (2020).

[93] van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

[94] Gal, Y. & Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. 33rd Int. Conf. Mach. Learn.*, 1050–1059 (PMLR, 2016).

[95] Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, NIPS'17, 6405–6416 (Curran Associates Inc., Red Hook, NY, USA, 2017).

[96] Mackay, D. J. C. *Bayesian Methods for Adaptive Models* (California Institute of Technology, 1992).

[97] Neal, R. M. *Bayesian Learning for Neural Networks*, vol. 118 (Springer Science & Business Media, 2012).

[98] Graves, A. Practical variational inference for neural networks. *Adv. Neural Inf. Process. Syst.* **24** (2011).

[99] Louizos, C. & Welling, M. Structured and efficient variational deep learning with matrix gaussian posteriors. In *Int. Conf. Mach. Learn.*, 1708–1716 (PMLR, 2016).

[100] Teye, M., Azizpour, H. & Smith, K. Bayesian uncertainty estimation for batch normalized deep networks. In *Int. Conf. Mach. Learn.*, 4907–4916 (PMLR, 2018).

[101] Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P. & Green, W. H. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *J. Chem. Inf. Model.* **60**, 2697–2717 (2020).

[102] Abdar, M. *et al.* A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. *ArXiv2011.06225* (2021).

[103] Coulston, J. W., Blinn, C. E., Thomas, V. A. & Wynne, R. H. Approximating Prediction Uncertainty for Random Forest Regression Models. *Photogram Engng Rem Sens* **82**, 189–197 (2016).

[104] Aggarwal, C. C., Hinneburg, A. & Keim, D. A. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In Goos, G., Hartmanis, J., van Leeuwen, J., Van den Bussche, J. & Vianu, V. (eds.) *Database Theory — ICDT 2001*, vol. 1973, 420–434 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001).

[105] Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).

[106] Morales-García, Á., Valero, R. & Illas, F. An Empirical, yet Practical Way To Predict the Band Gap in Solids by Using Density Functional Band Structure Calculations. *J. Phys. Chem. C* **121**, 18862–18866 (2017).

[107] Heyd, J., Scuseria, G. E. & Ernzerhof, M. Hybrid functionals based on a screened Coulomb potential. *J. Chem. Phys.* **118**, 8207–8215 (2003).

[108] Kauwe, S. K., Welker, T. & Sparks, T. D. Extracting knowledge from DFT: Experimental band gap predictions through ensemble learning. *Integr. Mater. Manuf. Innov.* **9**, 213–220 (2020).

[109] Chen, C., Zuo, Y., Ye, W., Li, X. & Ong, S. P. Learning properties of ordered and disordered materials from multi-fidelity data. *Nat Comput Sci* **1**, 46–53 (2021).

[110] De Breuck, P.-P., Hautier, G. & Rignanese, G.-M. Materials property prediction for limited datasets enabled by feature selection and joint learning with MODNet. *npj Comput Mater* **7**, 83 (2021).

[111] Chen, C. & Ong, S. P. AtomSets as a hierarchical transfer learning framework for small and large materials datasets. *npj Comput Mater* **7**, 173 (2021).

[112] Cubuk, E. D., Sendek, A. D. & Reed, E. J. Screening billions of candidates for solid lithium-ion conductors: A transfer learning approach for small data. *J. Chem. Phys.* **150**, 214701 (2019).

[113] Goetz, A. *et al.* Addressing materials' microstructure diversity using transfer learning. *npj Comput Mater* **8**, 27 (2022).

[114] Gupta, V. *et al.* Cross-property deep transfer learning framework for enhanced predictive analytics on small materials data. *Nat Commun* **12**, 6595 (2021).

[115] Ju, S. *et al.* Exploring diamondlike lattice thermal conductivity crystals via feature-based transfer learning. *Phys. Rev. Materials* **5**, 053801 (2021).

[116] Kong, S., Guevarra, D., Gomes, C. P. & Gregoire, J. M. Materials representation and transfer learning for multi-property prediction. *Applied Physics Reviews* **8**, 021409 (2021).

[117] Fronzi, M. *et al.* Active Learning in Bayesian Neural Networks for Bandgap Predictions of Novel Van der Waals Heterostructures. *Advanced Intelligent Systems* **3**, 2100080 (2021).

[118] Rajan, A. C. *et al.* Machine-Learning-Assisted Accurate Band Gap Predictions of Functionalized MXene. *Chem. Mater.* **30**, 4031–4038 (2018).

[119] Kauwe, S. K., Welker, T. & Sparks, T. D. Extracting Knowledge from DFT: Experimental Band Gap Predictions Through Ensemble Learning. *Integr Mater Manuf Innov* **9**, 213–220 (2020).

[120] Dunn, A., Wang, Q., Ganose, A., Dopp, D. & Jain, A. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Comput. Mater.* **6**, 138 (2020).

[121] Kingsbury, R. *et al.* Performance comparison of r2scan and scan metagga density functionals for solid materials via an automated, high-throughput computational workflow. *Phys. Rev. Mater.* **6**, 013801 (2022).

[122] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805* (2019).

[123] Brown, T. B. *et al.* Language models are few-shot learners. *arXiv:2005.14165* (2020).

[124] Jablonka, K. M., Schwaller, P., Ortega-Guerrero, A. & Smit, B. Is GPT all you need for low-data discovery in chemistry? *ChemRxiv* (2023).

[125] Deml, A. M., O'Hayre, R., Wolverton, C. & Stevanović, V. Predicting density functional theory total energies and enthalpies of formation of metal-nonmetal compounds by linear regression. *Phys. Rev. B* **93** (2016).

[126] Kotochigova, S., Levine, Z. H., Shirley, E. L., Stiles, M. D. & Clark, C. W. Local-density-functional calculations of the energy of atoms. *Phys. Rev. A* **55**, 191–199 (1997).

[127] Laws, K. J., Miracle, D. B. & Ferry, M. A predictive structural model for bulk metallic glasses. *Nat. Commun.* **6** (2015).

[128] Butler, M. A. Prediction of Flatband Potentials at Semiconductor-Electrolyte Interfaces from Atomic Electronegativities. *J. Electrochem. Soc.* **125**, 228 (1978).

[129] Kittel, C. *Introduction to Solid State Physics* (Wiley, Hoboken, NJ, 2005), 8. ed edn.

[130] Zhang, R. F., Zhang, S. H., He, Z. J., Jing, J. & Sheng, S. H. Miedema Calculator: A thermodynamic platform for predicting formation enthalpies of alloys within framework of Miedema's Theory. *Comput. Phys. Commun.* **C**, 58–69 (2016).

[131] Yang, X. & Zhang, Y. Prediction of high-entropy stabilized solid-solution in multi-component alloys. *Mater. Chem. Phys.* **132**, 233–238 (2012).

[132] Miracle, D. B., Louzguine-Luzgin, D. V., Louzguina-Luzgina, L. V. & Inoue, A. An assessment of binary metallic glasses: Correlations between structure, glass forming ability and stability. *International Materials Reviews* **55**, 218–256 (2010).

[133] Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* **369**, 253–287 (1921).

[134] Radial distribution function - wikipedia. `https://en.wikipedia.org/w/index.php?title=Radial_distribution_function&oldid=887892627`. Last accessed: 2023-06-04.

[135] Hansen, K. *et al.* Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *J. Phys. Chem. Lett.* **6**, 2326–2331 (2015).

[136] Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **108**, 058301 (2012).

[137] Sprik, M. Coordination numbers as reaction coordinates in constrained molecular dynamics. *Faraday Discuss.* **110**, 437–445 (1998).

[138] Zimmermann, N. E. R., Horton, M. K., Jain, A. & Haranczyk, M. Assessing Local Structure Motifs Using Order Parameters for Motif Recognition, Interstitial Identification, and Diffusion Path Characterization. *Front. Mater.* **4** (2017).

[139] Okabe, A., Boots, B., Sugihara, K. & Chiu, S. N. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* (Wiley, Chichester ; New York, 2000), 2 edition edn.

[140] Voronoi diagram - wikipedia. `https://en.wikipedia.org/wiki/Voronoi_diagram`. Last accessed: 2023-08-07.

[141] Falco, S., Jiang, J., De Cola, F. & Petrinic, N. Generation of 3D polycrystalline microstructures with a conditioned Laguerre-Voronoi tessellation technique. *Comput. Mater. Sci.* **136**, 20–28 (2017).

[142] Peters, B. Competing nucleation pathways in a mixture of oppositely charged colloids: Out-of-equilibrium nucleation revisited. *J Chem Phys* **131**, 244103 (2009).

[143] Steinhardt, P. J., Nelson, D. R. & Ronchetti, M. Bond-orientational order in liquids and glasses. *Phys. Rev. B* **28**, 784–805 (1983).

[144] Khorshidi, A. & Peterson, A. A. Amp: A modular approach to machine learning in atomistic simulations. *Computer Physics Communications* **207**, 310–324 (2016).

[145] Waroquiers, D. *et al.* Statistical Analysis of Coordination Environments in Oxides. *Chem. Mater.* **29**, 8346–8360 (2017).

[146] O'Keeffe, M. A proposed rigorous definition of coordination number. *Acta Cryst A* **35**, 772–775 (1979).

[147] Pinsky, M. & Avnir, D. Continuous Symmetry Measures. 5. The Classical Polyhedra. *Inorg. Chem.* **37**, 5575–5582 (1998).

[148] Botu, V. & Ramprasad, R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *Int. J. Quantum Chem.* **115**, 1074–1083 (2015).

[149] Voronoi, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *J. Für Reine Angew. Math.* **134**, 198–287 (1908).

[150] Dirichlet, G. L. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *J. Für Reine Angew. Math.* **40**, 209–227 (1850).

[151] Seko, A., Hayashi, H., Nakayama, K., Takahashi, A. & Tanaka, I. Representation of compounds for machine-learning prediction of physical properties. *Phys. Rev. B* **95**, 144110 (2017).

[152] Spherical harmonics - wikipedia. `https://en.wikipedia.org/wiki/Spherical_harmonics`. Last accessed: 2023-06-03.

[153] de Jong, M. *et al.* A Statistical Learning Framework for Materials Science: Application to Elastic Moduli of k-nary Inorganic Polycrystalline Compounds. *Sci Rep* **6**, 34256 (2016).