UNIVERSITY OF
LIVERPOOL

Frameworks for Utilising Computational Knowledge: Studying, Harnessing and Improving Techniques for Materials Property Prediction

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Samantha Durdy

October 2023

Abstract

Machine learning research has attracted an enormous amount of attention in recent years, both in academia and in the wider media. This, coupled with the creation of large datasets of materials and materials properties, has led to a boon in research at the intersection between machine learning and materials science. Much of this research has focused on prediction of materials properties; if the properties of a material are known before its synthesis, then the discovery of new materials becomes inexpensive, fast, and efficient.

This thesis explores methods for the prediction of material properties, and how these methods can be evaluated, improved, and incorporated into the discovery of new materials. By focussing on computational research (as opposed to experimental) the thesis explores several case studies on the use of machine learning on non-structural descriptors to predict properties. This non-structural approach is coupled with the novel use of deep learning to directly interpret electronic band structure data, for the prediction of materials properties.

The examples presented are grounded in the process of material discovery. Workflows are suggested to incorporate machine learning models into collaborative projects. These collaborations result in the synthesis of new materials and the production of an online web platform that enables easy sharing of computational tools.

These proposed methods are complimented by in-depth analysis of the complexities of analysing machine learning models in a materials science context. Evaluation methods such as r^2_{comp} are suggested, which quantifies the predictive power of models in the context of substitutional studies, where the unknown properties of a new material are predicted in comparison to the known parent material.

Leave one cluster out cross validation (LOCO-CV) is an existing method for quantifying models' abilities to extrapolate predictions to classes of materials unlike those on which the model was trained. Following observations that clusters of materials used in LOCO-CV cause unreliable results, an adaptation is proposed "kernelised LOCO-CV" which applied a non-linear function *a priori* to LOCO-CV to improve reliability.

Overall, varying uses of computational and machine learning methods for the prediction of materials properties are explored. This interdisciplinary approach enables new computational methods to be discussed in the context of materials science and new methods of interpreting materials to be discussed from the perspective of the data used to represent them.

Acknowledgements

I would like to thank people. Firstly, I would like to acknowledge the concept of energy drinks. I never want to open a can of Monster again. Unfortunately, I probably will.

More earnestly, I would like to thank my supervisors, Professor Matthew J. Rosseinsky, Professor Danushka Bollegala, Dr. Vladimir Gusev, and Dr. Michael Gaultois. They have all been very supportive throughout and guided my research while leaving me space to go down various rabbit holes. I would also like to thank the Leverhulme Research Centre for Functional Materials Design for funding my research and being understanding when personal leave was needed. I would also like to thank the entire Rosseinsky group for being lovely co-workers, particularly Katie Scott, who was a life saver in keeping me on track with administrative tasks.

I would like to thank Cameron Hargreaves for being not only a pleasure to work with, but also a good housemate and friend. Also, for keeping me sane, I would like to thank my housemate Jay Griffiths and my cat Molotov.

Thanks to everyone at Alliance Muay Thai and Northside Muay Thai, for providing a supportive community and stress relief, without which I would not have been able to complete this thesis. I would also like to thank the entire Die Another Day discord server for being the void to rant into when needed. Particularly special thanks are given to Daryl Hodge, who's lockdown, socially distanced walks kept me sane during the pandemic.

I'd like to thank my amazing girlfriend Sree, who has been so supportive of me and who has brightened every day since we met. I would like to thank my brother James and congratulate him on finishing his degree. Finally, I would like to thank my parents for putting up with me the last 27 years and housing me the last 6 months while I was at peak thesis gremlin. They have always been extremely supportive, especially when things have been difficult for the family for the past two and a half years, they have shown amazing strength and love.

I would like to dedicate this thesis to my sister Jess, who sadly took her own life in October 2020 while under the care of the Bristol Mental Health team. Jess was let down at every turn by a system more set on cost saving than lives. We may not have always got along, but the world is immeasurably worse without her here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis background and research question

Machine learning (ML) has gained widespread recognition in recent years, revolutionising many areas of life. Accordingly, the intersection between ML and materials science, sometimes referred to as materials informatics, has rapidly gained momentum in the past decade [193, 166]. With the increasing availability of datasets of materials [78, 32], researchers have sought to leverage data science techniques to advance the field [166]. These investigations have largely focused on predicting properties of materials. This is often used as a screening method to identify materials with desirable properties as candidates for chemical synthesis.

At the time this thesis was initiated, the materials informatics revolution was already underway and, while significant progress had been made, the development of new materials had not yet been fully realised; the ratio of papers discussing ML to those discussing ML and producing actual materials remained (and remains) low. Many research papers suggest candidates for synthesis, but without that synthesis being carried out, these candidates remain suggestions.

The challenge lies in bridging the gap between ML and materials synthesis, as both require specialised but largely disparate skill sets. A ML expert could learn basic synthesis (or a synthetic chemist could learn ML skills). But both fields are fast moving, it would be somewhere between unrealistic and conceited to believe one could be at the cutting edge of both fields.

It is conceivable to go from that premise and produce a thesis which tries to balance learning both computational and synthetic skillsets and lies somewhere on a Pareto front between the fields of synthetic chemistry and data science. However, this is not that thesis. Instead, this thesis explores the collaboration between ML developers and synthetic chemists as a means to achieve progress.

Against this backdrop, the research question driving this thesis is as follows: How can ML be used effectively to enhance materials discovery? Although an ultimate goal could be to develop a fully automated system for materials discovery, the current state of technology necessitates the collaborative approach discussed. As such, this thesis concentrates on the methods and rationale behind ML. Specifically, explorations focus on how to formulate appropriate questions to make precise predictions and how to assess the efficacy of these predictions.

The evaluation will ask whether existing models are being justified based on their performance compared to alternative ML approaches or their performance compared to the lack of ML in traditional materials science literature or if there are opportunities for improvement. Models may offer novelty in performance on test datasets, but candidate predictions from a model are of little practical value if synthesis is not performed. Millions of candidate materials can be predicted (and millions of candidates will be predicted over the course of this thesis), but in the rapidly evolving landscape of ML in materials science, there is no reason that existing predictions will be used for synthesis over newer predictions.

Thus, it is critical to consider how to use models in collaborative projects. This requires models to be thoroughly justified and for data to be well explored.

Rather than solely focussing on developing state-of-the-art models and trying to reach top benchmarks, this thesis aims to explore how ML can be used more effectively to drive materials discovery. Through a critical evaluation of existing models and the development of novel approaches, this thesis seeks to contribute to the broader goal of advancing materials science through the application of ML.

1.2 Thesis structure

Each chapter will contain a section that describes how that chapter fits the larger narrative of the thesis. Despite this, it is helpful to begin with an outline of the thesis.

The thesis begins with a brief overview of the fundamental machine learning techniques relevant to the research. This will serve as a foundation for understanding the subsequent discussions. As classes of materials investigated throughout vary, explanations of materials will be given in the relevant chapters rather than focused on at the start of the thesis.

The thesis then examines the application of random forests in materials science. A specific focus is given to how the phrasing and scope of the questions posed to the algorithms can significantly impact their effectiveness. The emphasis is not on improving algorithms, but on incorporating elements unique to the materials science domain to improve applicability. In limiting the scope of the prediction task, performance is improved without increasing computational cost and while keeping the predictions relevant for a synthesis environment. RFs are widely used, as such, considering ways in which they can be adjusted for more effective use in a materials context serves the overall research question outlined above.

The subsequent chapter focusses on the prediction of the superconducting critical temperature of materials. The existing literature on this topic is sometimes problematic and vague, so this chapter aims to provide a more comprehensive understanding of the subject matter. Techniques such as garbage in are poorly explored in the literature [88], and research has been published with clear data leaks [64] Rather than chasing numerical performance, the chapter considers how to facilitate successful synthesis. Workflows for screening predictions, incorporating feedback, and working with chemists with differing interests and motivations are considered. Through incorporation of feedback loops into workflows, random forests were able to inspire the synthesis of various materials despite not being considered state-of-the-art in this field. By focussing on justifying techniques, and implementing them into a synthetic workflow, this contribution serves the overarching research question by addressing a lack of emphasis on synthetic results in the literature.

One way to streamline collaboration is by deploying models in formats accessible to people without coding experience, such as web applications. ML experts may not possess sufficient networking or web development knowledge to create such applications, as such templates and frameworks are created to ease development. By encouraging the use of accessible formats for ML models, the organisational overhead associated with collaboration is reduced, which in turn contributes towards answering the outlined research question.

The next chapter scrutinises the assumptions underlying the random forest models from the first chapter. Methods for representing materials as vectors are investigated, and it is found that, for many materials science tasks, feature engineering is unnecessary, in many cases. This work serves the overall research question by investigating underlying assumptions in current literature.

Methods for evaluating these models are also examined. A method for evaluating models' performance on predicting new classes of materials, leave-one-cluster-out cross-validation (LOCO-CV), was noted in previous chapters to produce unreliable results. Specifically, the LOCO-CV algorithm relies on clusters of data used to be similar in size to produce reliable results, but the algorithm does not provide mechanisms to ensure this. This work presents findings suggesting that many areas of materials science would result in unreliable LOCO-CV application. A mechanism for improving the evenness of clusters, radial basis function approximation, is suggested. It is observed that application of radial basis function approximation results in more evenly sized clusters in all of the datasets investigated. By improving evaluation methods proposed for this domain, the effective use of ML is encouraged (as per the research question above) by enabling researchers to better scrutinise their ML algorithms.

One aspect of clusters that the radial basis function approximation was qualitatively observed to change was the shape of the clusters. After applying the radial basis function approximation, clusters are seen to be more isotropic. The next chapter focuses on methods to quantify this observation.

Drawing on methodologies from the fields of medical imaging and natural language processing, two different methods are found to be candidates for this quantification: isotropy measure $I_{c,\text{vec}}$ and fractional anisotropy (FA). $I_{c,\text{vec}}$ is adjusted to make it more mathematically sound, and a variant, $I_{c,\text{rnd}}$ which is more computationally efficient in higher dimensions is suggested. Further adaptation is suggested to allow FA, $I_{c,\text{vec}}$, and $I_{c,\text{rnd}}$ to measure isotropy across a set of clusters rather than only one cluster. Investigations were conducted in the use of these measures in both a data science and a materials science context. FA was observed to behave counterintuitively in high dimensions, and explanations for this behaviour were mathematically derived using random matrix theory.

In the context of inorganic chemistrty, crystal structures are often described as being "isotropic" or "anisotropic", based on their unit cells. As such explorations of mathematical measures of isotropy in a materials science context would be lacking without attempting to quantify the isotropy of the unit cell of a material. As such, the applicability of FA, $I_{c,\text{vec}}$, and $I_{c,\text{rnd}}$ in measuring the isotropy of materials was investigated, finding no evidence of efficacy. While this work does link investigations of isotropy to the materials science domain, overall work in this chapter is somewhat tangential to the research question. This said, techniques explored may serve the purpose of enhancing ML for discovery of materials, by enhancing ML more generally (by enabling quantitative analyse of learnt embeddings). Outside of the research question outlined above investigations into isotropy remain highly interesting and a significant (if low impact) contribution to the scientific domain.

The final experimental chapter explores unutilised data in electronic band structures and how ML techniques can be applied to extract valuable information from these data. The chapter addresses potential caveats associated with band structures and explores how ideas from other areas of ML, such as image recognition and natural language processing, can be adapted for use in this context. This chapter serves the question of how to effectively use ML to enhance materials structure, by examining, and proposing methods for processing underutilised data.

Before the thesis is complete, reflections are made on this work and on the field as a whole. Future exploration areas are suggested, and closing remarks are considered.

This thesis offers a comprehensive examination of the application of ML in materials science and provides insights into how to optimise the process of collaborating with synthetic chemists. In such an exciting and rapidly moving field, it is hoped that this work will be as interesting and fun to read as it was to research.

1.3 Broad contributions of this thesis

Having outlined the structure of the thesis, it is useful to cement how that structure translates to the scientific contribution of this thesis. While, smaller, specific contributions are detailed both later in this chapter and throughout the thesis, broader (but more consise) contributions can be described as follows:

- Suggesting an exploring ways to adapt substitution chemical studies into machine learning tasks (Chapter 3).
- Exploring how the presentation of ML tasks (as regression, or classification tasks) impact performance in a materials context (Chapter 3)
- Further justifying the “garbage in” technique, for providing negative examples a dataset of positive examples (Chapter 4).
- Providing workflows for generating and screening large numbers of candidate materials, while focussing on creating outputs which are useful for synthetic chemists to use (Chapter 4).
- Creating a web platform for increasing acccess to novel computational tools, and providing information to encourage others to do the same (Chapter 4).

- Added to a body of evidence suggesting that in many cases feature engineering may not significantly impact the performance of ML algorithms in this field (Chapter 5).
- Explored use of random projection as featurisation for ML algorithms in this field (Chapter 5).
- Suggested a simple technique (kernel approximation methods) to address the problem of uneven cluster sizes in application of LOCO-CV (Chapter 5).
- Explored, adapted, and improved measurements for the isotropy of clusters (Chapter 6).
- Provided mathematical explainations for trends seen in use of the fractional anisotropy measure in high dimensions (Chapter 6).
- Explored numerous paradigms by which electronic band structure data can be interpretted (Chapter 7).
- Developped novel, high performing models to process electronic band structure data to extract properties (Chapter 7).

1.4 Publications

In the writing of this thesis numerous works were published in peer-reviewed journals. As such, many of results in presented in this thesis were previously published. Sections which contain results which were partially or wholely published are preceeded by notes to credit the publications that the results are from. For conveinience, and as demonstration of scientific contribution, the list of peer reviewed publications which contain results discussed in this thesis are as follows (listed in order of publication):

- Philip A. E. Murgatroyd, Kieran Routledge, Samantha Durdy, Michael W. Gaultois, T. Wesley Surta, Matthew S. Dyer, John B. Claridge, Stanislav N. Savvin, Denis Pelloquin, Sylvie Hébert, Jonathan Alaria. "Chemically Controllable Magnetic Transition Temperature and Magneto-Elastic Coupling in MnZnSb Compounds." Advanced Functional Materials, 2100108 (2021). [125].
- Rémi Pétuya, Samantha Durdy, Dmytro Antypov, Michael W. Gaultois, Neil G. Berry, George R. Darling, Alexandros P. Katsoulidis, Matthew S. Dyer, and Matthew

J. Rosseinsky. “Machine-Learning Prediction of Metal–Organic Framework Guest Accessibility from Linker and Metal Chemistry.” Angewandte Chemie International Edition 61, no. 9 (2022) [144].

- Samantha Durdy, Michael W. Gaultois, Vladimir V. Gusev, Danushka Bollegala, and Matthew J. Rosseinsky. “Random projections and kernelised leave one cluster out cross validation: universal baselines and evaluation tools for supervised machine learning of material properties.” Digital Discovery 1, no. 6 (2022): 763-778 [45].
- Samantha Durdy, Cameron J. Hargreaves, Mark Dennison, Benjamin Wagg, Michael Moran, Jon A. Newnham, Michael W. Gaultois, Matthew J. Rosseinsky, and Matthew S. Dyer. “The Liverpool materials discovery server: a suite of computational tools for the collaborative discovery of materials.” Digital Discovery 2, no. 5 (2023): 1601-1611 [46].

In addition to the above, the following paper is published as preprint, and is currently under peer review for publication:

- Metrics for quantifying isotropy in high dimensional unsupervised clustering tasks in a materials context [44] (under peer review for publication in Applied Intelligence).

1.5 Published Code Repositories

Throughout the investigations in this thesis, several code repositories were published. These publications were made not only for transparancy and reproducibilty purposes, but also as a contribution to the scientific community. As such, review of these code repositories is welcomed with review of this thesis. Each code repository is intended to be easy to use, and code should be understandable and documented to a high standard. Criticisms of this code or pull requests are welcomed.

Repositories are cited in the places relevant to their contents. For ease of access and as evidence of scientific contribution, they are also listed here. Repositories associated with this thesis are as follows:

- A demonstration of using random forests to predict unit cells properties [41].
- A web app for the prediction of MOF porosity from SMILES string and metal species [164].

- An implementation and demonstration of the kernelised LOCO-CV method [40].
- An impletation and demonstration of tools to measure the isotropy of clusters [43]:
- A web app for modelling the heat of a material [47].
- Scripts to set up a basic reverse proxy and service to manage Flask applications [42].

1.6 Specific contributions

Specific contributions will be listed in each chapter. For convenience, all lists of specific contributions contained in this thesis have been compiled, but note that points in this list will be repeated in the context of their respective chapters. The specific contributions are as follows.

1.6.1 Chapter 3: Use of random forests for prediction of material properties

- Exploration of solutions to problems with duplicated and conflicting data points which can arise when using ML for materials property prediction, particularly, but not exclusively in the organic synthesis domain (demonstrated in Section 3.2.1 and further discussed in Section 3.2.4).
- Exploration of use of sequential application of RFs for a well performing yet simple to implement model (Section 3.2.2).
- Discussion of the advantages and disadvantages of simplifying tasks to classification tasks (Section 3.2.4).
- Discussion of the benefits and drawbacks of models which aim to demonstrate datasets (Section 3.2.4).
- Investigating the ability of RFs to predict the $\frac{c}{a}$ ratio of a conventional unit cell.
- Presenting a comparative prediction method for using ML algorithms. This method is particularly suited for substitutional synthesis studies.
- Presenting and investigating performance metrics r^2_{comp} and ordinal accuracy which examine the performance of a machine learning algorithm in the context of comparative predictions.

1.6.2 Chapter 4: Collaborative workflows for discovery of superconductors and other functional materials

- A thorough investigation of literature surrounding the prediction of superconducting critical temperature.

- Exploring the "garbage in" method found in the literature, and questioning its underlying assumptions (namely that the Crystalographic open database does not contain any superconducting materials) and investigating the effects of those assumptions.

- Definition of a workflow to allow for collaboration and feedback with experimentalists.

- Screening of over 1 billion candidate materials for superconductivity using ML methods.

- Collaborating with experimentalists to identify areas of interest in the chemical space, helping to rationalise results from the screened materials.

- Creation of a filter to identify materials which could be potentially arc-welded.

- Inspiring the synthesis of possible candidate materials in the Sr-Cu-Sn and Sr-Cu phase fields.

1.6.3 Chapter 5: Random projections and leave one cluster out cross validations: improving evaluations of machine learning for materials properties

- Comparing the influence of composition based feature vectors (CBFVs) on ML model performance in practical tasks (explained further in Section 5.2.1, before being carried out in section Section 5.2). It was found that CBFVs with engineered features (*i.e.*, imbued with domain knowledge) do see some benefit in certain tasks, particularly band gap prediction tasks. While *magpie* representations [194] were seen to outperform other CBFVs in many tasks, this finding was not universal across tasks.

- Examining the effectiveness of random projections as featurisation methods for property prediction from chemical composition. Random projections can be used as a baseline against which to justify more involved featurisation methods (explained further in Section 5.2.1 before being carried out in Section 5.2). It was found that in many tasks, CBFVs with engineered features do not perform substantially better than random projections.

- Studying the effect of kernel approximation functions (explained further in Section 5.3) on the application of K-means clustering to materials data, and presenting a workflow to incorporate these methods into the LOCO-CV algorithm (Section 5.3). It was found that kernel approximation functions are a good way to reduce the variance between sizes of clusters found by K-means clustering on materials data. Using kernel approximation functions in the suggested workflow (kernelised LOCO-CV) results in a more robust evaluation method than LOCO-CV with no kernels.

- It can be recommended to use radial basis function (RBF) approximation when clustering for LOCO-CV, as clusterings found after application of RBF are seen to be more even in size than with no kernel method applied, and models are trained more reliably for property prediction. This helps to reduce the risk that performance differences on predicting an unseen cluster of data are caused by the training set size as opposed to the intrinsic inability of a model to perform well on that cluster of data.

- It was found that the use of RBF approximation in clustering for LOCO-CV leads to more reliable and consistent model training, compared to using LOCO-CV without any kernel approximation methods.

- Use of random projections as a baseline against which to compare engineered feature vectors is recommended. It is noted that commonly used CBFVs have little to no advantage over random projections in most tasks investigated.

- The use of random projections as a featurisation method for clustering compositions in LOCO-CV was investigated, finding that random projections have no clear advantage over other CBFVs tested here.

1.6.4 Chapter 6: Mathematically quantifying isotropy

- Exploring how metrics used for measuring isotropy in 3 dimensions [10] generalise to higher dimensions.

- Providing a new implementation for an isotropy measure based on an existing mathematical derivation (Section 6.3.1).

- Proposing adaptions to the measures of isotropy for single clusters such that one can measure the average isotropy across a set of clusters (Section 6.3.2).
- Highlighting the need for analysis of representation when clustering datasets relating to materials (Section 6.4.1).
- Demonstrating analysis of isotropy in a supervised learning context using a foundational data science dataset (Section 6.4.2).
- Examining the robustness of the metrics under random noise perturbations (Section 6.4.3).
- Using random matrix theory to prove that the measurements of isotropy are related to the dimensionality of data, especially if the data are noisy (Section 6.4.3).

1.6.5 Chapter 7: Machine learning with electronic band structures

- Creation of a dataset of relating EBS data to resistivity.
- Exploration of the caveats surrounding using machine with EBS data.
- Suggesting numerous paradigms through which EBS data can considered. This allows easy adaptation of ML algorithms intended for those paradigms.
- Novel implementation of deep learning algorithms across datasets of EBS data.
- Creation of two RNNs to interpret EBS data.
- Use of a set transformer model in two different arrangements to interpret EBS data.
- Improving performance in predicting resistivity, and band gap by several orders of magnitude compared to a RF.

Chapter 2

Background and Related Work

This chapter aims to provide an overview of fundamental concepts in both data science/machine learning (ML) and materials science, which will be necessary to understand the subsequent chapters of this thesis. While pertinent background knowledge will be briefly revisited when used in context, the reader is encouraged to refer back to this section as needed throughout the thesis.

Basic concepts of ML, including types of ML, and ML evaluation, are covered. The most important algorithms for this thesis are explored, including principal component analysis, random projections, K-means clustering, random forests (RFs), and neural networks.

2.1 Machine Learning theory and related work

This section will outline basic ML concepts before describing several classes of algorithms that will be relevant to the thesis. Note that where other areas of ML, computer science, or mathematics are prerequisite knowledge to the thesis, those concepts are introduced and explained in context rather than in this chapter.

What is machine learning?

ML at its heart is the use of algorithms to understand and take advantage of data. A key difference in ML and other forms of artificial intelligence is that ML focusses on performing a task without explicit programming on how that task should be done. Instead, ML models identify trends in the data to learn how to complete the task. In ML, trends learnt by models are stored in parameters, not to be confused with hyperparameters, which control aspects of how a model is trained, or a more traditional definition of parameters which would imply that parameters are passed as arguments to the model. Machine learning tasks can generally be categorised into one of four categories: supervised learning, unsupervised learning, reinforcement learning, and semi-supervised learning.

Supervised learning involves input data and a corresponding target value. Algorithms are trained on labelled data, where the correct output, or ground truth, is known for each input. The goal is to learn a mapping between the input data and the target values to accurately predict the target values for new unseen input data. In the field of materials science, a typical supervised learning task would be predicting a property of a material after having been trained on a dataset of materials and corresponding values for that property. Algorithms in this category that will be investigated include random forests (RFs) and neural networks (often called deep learning).

Unsupervised learning deals with data that do not have target values associated with it. Instead, algorithms must identify patterns and relationships within the data. A typical example of unsupervised learning in the field of materials science would be to identify clusters of materials in a dataset, which can help explore data. Unsupervised algorithms include K-means clusterings and principal component analysis (PCA).

Semi-supervised learning combines supervised and unsupervised learning. Some data have target labels, while the rest are unlabelled. Algorithms must use the labelled data in conjunction with the unlabelled data to perform a task. A typical example in the field of materials science is iterative label spreading, which can be used to cluster materials based on a few labelled data points.

Throughout this thesis, machine learning algorithms that fall into one or more of these categories will be discussed. Depending on the specific task and dataset at hand, different algorithms may be more appropriate and effective than others.

2.1.1 Bias in machine learning

The trends identified in the data can be helpful in allowing ML models to complete tasks. But some trends are considered bias, which can introduce unwanted features into a ML model. ML faces the ongoing challenge of distinguishing between helpful trends in data and those that introduce bias.

Unwanted biases in the data often reflect biases in society, such as sexism, racism, homophobia, and transphobia. In the context of predicting the properties of the materials,

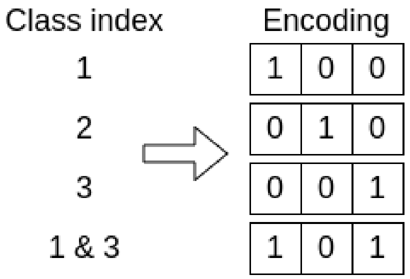


Figure 2.1: An example of 1-hot and n-hot encoding for a 3 class classification problem.

a large bias is publication bias. Publication bias arises from the tendency of researchers to publish only positive results. Negative results, such as materials that were not successfully synthesised or were found to be uninteresting on synthesis, are much less likely to be published. Data for these materials are more scarce, which means that entire areas of the possible chemical space are not seen in datasets. This means existing datasets are skewed towards popular, or easy to explore materials.

A common way to address this in materials science is by using uncertainty estimates [1]. Uncertainty estimates are a class of algorithms which aim to quantify how sure a model is about its predictions. Several of these algorithms exist, some of which are only applicable to certain types of ML models [153]. Areas of chemical space that are more sparsely represented in a model's training dataset will result in more uncertain predictions. Measurement of uncertainty allows these areas to be avoided or explored. In ML, the trade-off between avoiding or exploring certain areas of a domain is often called exploration versus exploitation.

2.1.2 Types of tasks in machine learning

Algorithms in ML can often be grouped according to the tasks they are performing. The groups do not always align with whether the task is considered supervised or unsupervised. Three tasks that occur multiple times in this thesis are classification, regression, and clustering. Classification involves taking an input piece of data and performing discrete predictions. This can be two discrete classes (binary classification) or more (multiclass classification). The target classes can be represented as integer indices (*i.e.* outputting 0 is a prediction of class 0, outputting 1 is a prediction of class 1 *etc.*). A problem with representing output classes as integers is the implication that the classes are sequential.

This would imply a distance metric between classes, which may not accurately reflect their relationship. As such, “one-hot” encodings are a common representation of the targets for these tasks. One-hot encodings are vectors with one entry for each class, and the vector will all be zero-valued, except for the column of the class which that vector is meant to represent, which will be one (Figure 2.1). Where a single data point can be in multiple classes, a one can be present in each class that data point represents. This is sometimes called “n-hot” encodings. An example classification task in materials science might be predicting whether a material has conductivity above or below a desired threshold.

Regression tasks are tasks in which an algorithm needs to produce a number in a continuous output domain. An example regression task in materials science is to predict the resistivity of a material.

Clustering involves separating data into groups (clusters) on the basis of some geometric criteria. Whereas classification separates points based on what that point represents, clustering separates points on how they exist in relation to each other. Clustering algorithms often require the number of target groups to be input. An example of clustering in materials science would be to cluster a dataset into different classes of materials to see if the resulting clusters are in line with expectations.

2.1.3 Unsupervised machine learning

This subsection will explore several unsupervised ML algorithms that will be recurrent or relevant to this thesis. The evaluation of unsupervised algorithms in ML is highly task dependent, as without target labels it can be unclear what can be considered successful. As such, the specific evaluation metrics for these algorithms will be discussed throughout this thesis in the context of their use, rather than in this section.

First, dimensionality reduction techniques will be discussed. Specifically, principal component analysis (PCA) and random projections. Then clustering algorithms K-means clustering and iterative label spreading (ILS) will be explored. Only algorithms particularly relevant to the thesis will be discussed; other notable algorithms that fall under these categories, such as t-distributed stochastic neighbour embedding [180], Density-Based Spatial Clustering of Applications with Noise [49], or hierarchical agglomerative clustering [111], will be left for the reader to explore in their own time.

Principal Component Analysis

PCA is a dimensionality reduction algorithm commonly used to visualise high-dimensional data, while retaining as much relevant information as possible [56]. Intuitively, PCA finds the set of orthogonal axes which best describe the variation seen in the data. To perform PCA, first a covariance matrix, $\mathbf{C}$, is calculated for the data. For a dataset of n data points existing in m dimensions, represented as a $n \times m$ matrix, $\mathbf{X}$, this can be expressed as:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^{\intercal} \tag{2.1}$$

$\mathbf{C}$ represents the variance relationship between each dimension of the dataset. The next step in PCA is to apply singular value decomposition (SVD) to $\mathbf{C}$. This decomposes $\mathbf{C}$ into three matrices of the form:

$$\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{V}^{\intercal} \tag{2.2}$$

Where $\mathbf{U}$ is a $m \times m$ matrix representing unit vectors of the principal components of the data. That is, the rows of $\mathbf{U}$ represent a set of orthogonal unit vectors (known as orthonormal vectors) that point in the directions in which $\mathbf{X}$ vary the most. $\mathbf{S}$ is a diagonal matrix containing the singular values of $\mathbf{X}$. Singular values represent the amount of variance in the data that is explained by each principal component. The diagonal entries of $\mathbf{S}$ are ordered from the largest to the smallest, so the first few singular values capture most of the variance in the data, and the remaining singular values capture less and less variance. Although $\mathbf{V}$ is not used at any point in this thesis, it is interesting to note that $\mathbf{V}$ is a $n \times n$ matrix whose rows represent how much of each principal component is present in each data point in $\mathbf{X}$. There are several SVD algorithms (and implementations of SVD algorithms) available [140, 139, 68]. How these implementations work and the differences between them will not be explored in this thesis. In practise, an application of SVD is done by calling a code library.

Once the principal components have been found, the first k principal components can be used to project $\mathbf{X}$ into lower dimensions. The result is a representation of the dataset that is in fewer dimensions than the original dataset, while minimising the amount of information lost.

Random Projections

Another technique which will be in this thesis is random projections. Random projection is a dimensionality reduction technique that uses the observation that in high dimensions random vectors approach orthogonality [154, 83]. A dataset of n data points existing in m dimensions, represented as a $n \times m$ matrix, $\mathbf{X}$, can be projected into k dimensions by multiplying it by a random matrix, $\mathbf{R}$, of shape $m \times k$:

$$\text{Random Projection} = \mathbf{X} \cdot \mathbf{R} \tag{2.3}$$

When the columns of $\mathbf{R}$ are normalised to be unit vectors, $\mathbf{X} \cdot \mathbf{R}$ becomes an approximately linear projection of $\mathbf{X}$. Another way to closely approximate normalisation of the columns of a random matrix, such as $\mathbf{R}$, is to sample the values of that matrix from a Gaussian distribution of mean 0 and variance $\frac{1}{k}$ $(\sim \mathcal{N}\left(0, \frac{1}{k}\right))$ where k is the size of the projection. This is mathematically justified by the Johnson-Lindenstrauss lemma, which states that for a set of M dimensional data points there exists a linear mapping that will embed these points into an k dimensional data space while preserving distances between data points within some error value, ϵ. This value of ϵ is shown to decrease as k increases [33].

In other words, since random matrix multiplication is approximately a linear projection and linear projections maintain distances between data points (within some error value), by multiplying a dataset by a random matrix, one can change the number of dimensions in a dataset, while maintaining the distance relationships between points. Thus, a random projection allows for the reduction in dimensionality of a dataset.

K-means clustering

K-means clustering [108] is an unsupervised clustering algorithm which iteratively groups data into K different clusters. The algorithm is very simple:

1. Place K points (centroids) randomly.
2. Group the dataset into clusters based on which centroid they are closest to.
3. Redefine the centroids as the mean of the points in their clusters.
4. Repeat steps 2-3 for a fixed number of iterations or until the clusters or centroids no longer change between iterations.

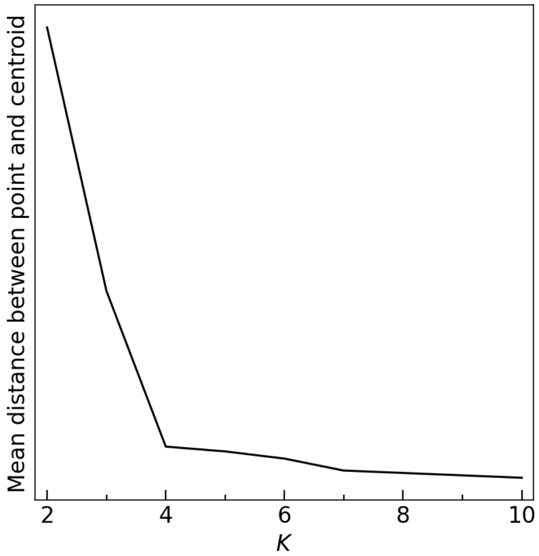


Figure 2.2: An example of an elbow plot. K-means has been applied to data with values of K between 2 and 10. The sharp change in gradient when K is 4 indicates that there are 4 clusters.

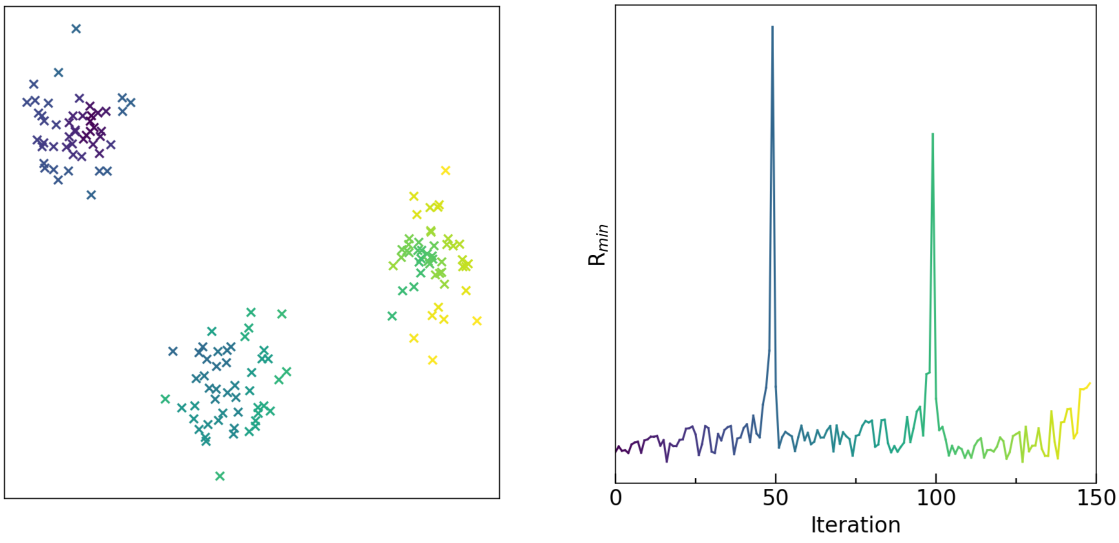
This algorithm results in a Voronoi diagram representing the data space; thus this process is sometimes called a Voronoi tessellation of the data. The parameter K can be chosen by the researcher if it is already known. If K is unknown, it is often established by repeating the algorithm with several different values of K, and measuring the average distance between a point and its centroid. The average distance between a point and its centroid can then be plotted against K, in what is called an “elbow plot” (Figure 2.2). The point at which increasing K no longer causes a large decrease in the average distance between a point in a cluster and its centroid is a good choice for deciding K.

Iterative label spreading (ILS)

ILS [138] is a clustering algorithm which can be used in an unsupervised or semi-supervised manner. Given a set of labelled points, $\mathcal{L}$ in a dataset, $\mathcal{D}$, ILS can be described as follows:

1. Let $R_{\min}(x, y)$ the smallest distance (by some measure) between any unlabelled point x in $\mathcal{D}$ to any point, y in $\mathcal{L}$

2. Let x have the same label as y



(a) (b)

Figure 2.3: An example of clustering 2D clusters using iterative label spreading (ILS). (a) The clusters, coloured by the order in which they were labelled (b) R$_{min}$ plotted against the iteration. The two peaks indicate points at which the label spreading has jumped from one cluster to another, suggesting that there are three clusters (as is the case).

3. Repeat steps 1-2 until all points are labelled.

To use ILS in an unsupervised manner, a random point is chosen to be the only labelled point. R_{min} can then be plotted against the iteration (Figure 2.3b). The number of peaks seen in the plot can represent the number of different clusters. Assign the lowest points between each peak to a different label and rerun ILS using these labels as the set $\mathcal{L}$

In materials science, datasets are often not fully labelled, or it may not be clear whether a representation reveals clusters that would be expected based on assigned labels. As such, ILS allows both issues to be investigated.

While iterative label spreading is not examined in detail in this thesis, it is included here for two reasons. Firstly, it is an excellent example of how general algorithms can come from the materials science domain because of the specific issues with materials science data. Secondly, parts of this thesis do relate to clustering, so it is pertinent to note related literature in the field.

2.1.4 Evaluation metrics in supervised machine learning

Table 2.1: Regression metrics to measure the similarity between the true value x and the predicted value y. Including whether the metric has lower bounds (LB) or upper bounds (UB), what the optimal value is, and the formula for each metric.

Metric	Formula	Bounded	Optimum
Mean Absolute Error	$\frac{1}{n}\sum_{i=1}^{n}\lvert x_i - y_i\rvert$	LB = 0	0
Mean Relative Error	$\frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{x_i - y_i}{x_i}\right\rvert$	LB = 0	0
Mean Squared Error	$\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$	LB = 0	0
r^2 correlation coefficient	$\left(\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}\right)^2$	UB = 1	1

Evaluation of supervised ML is non-trivial, particularly in the context of materials science. A notable choice is which measures (or metrics) are used to quantify success; this section will explore this choice.

There are a variety of metrics which can be used to quantify ML performance, though the choice of metrics is largely defined by the task at hand. For regression tasks, common metrics include the mean absolute error, the mean relative error, the mean squared error, and the correlation coefficient, r^2 (Table 2.1).

Choosing which regression metrics are considered most important imparts a bias, as the best performing model in one metric may not be the best performing in all metrics. Consider two models, a and b, that perform a regression task to predict the resistivity of two materials. One material, material w, has a resistivity of 10 Ωm and the other material, x, has a resistivity 1 Ωm. Model a predicts w and x to have resistivities of 11 Ωm and 2.0 Ωm, respectively. The model b predicts that w and x have resistivities of 12 Ωm and 1.2 Ωm. Using the mean absolute error as a metric model, a appears to be better (Table 2.2), but using the mean relative error model b appears to be better. Neither model is inherently better than the other; instead it is up to a researcher to decide which model would be better suited to their needs.

Table 2.2: Example of how different metrics for success favour different models. Here, two models a and b predict the resistivity for two materials, w and x. The true and predicted (pred.) values are compared as well as the absolute (Abs.) and relative (Rel.) errors.

		Model a			Model b		
Material	True (Ωm)	Pred. (Ωm)	Abs. Error (Ωm)	Rel. Error (%)	Pred. (Ωm)	Abs. Error (Ωm)	Rel. Error (%)
w	10	11	1	10	12	2	20
x	1	2	1	100	1.2	0.2	20
Mean			1	50.5		1.1	20

Table 2.3: An example confusion matrix for a binary classification task. The contents of the matrix are usually numbers that relate to the quantity of each type of prediction, although sometimes these are expressed as percentages.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

When assessing classification performance, many metrics can be considered. The choice in metrics does depend on whether the task is a binary classification task, or a multi-class classification task.

In binary classification, predictions are often split into “positive” or “negative” predictions. For example, consider the task of predicting whether a material is a superconductor, a positive result would be a prediction that a material is a superconductor, and a negative result is that a material is not a superconductor. It can then be said that a true positive (TP) prediction would be a positive prediction where the material is indeed a superconductor. A false positive (FP) would be a material given a positive prediction that turns out not to be a superconductor. A true negative (TN) would be a negative prediction for a material that is not a superconductor, and a false negative (FN) would be a negative prediction for a material that actually was a superconductor. From these measures many informative metrics can be derived. These measures can also be put into a table, called a confusion matrix, indicating the number of each type of results that were present in a given task (Table 2.3).

Measures derived from these four measures (TP, FP, TN, and FN) include sensitivity (or recall), specificity, precision, F_1 score, and accuracy (Table 2.4). Further measures such

Table 2.4: Binary classification metrics derived from the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) rates. Whether the metric has an is bounded and what it's optimal values are also included.

Metric	Formula	Bounded	Optimum
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	0, 1	1
Specificity	$\frac{TN}{TN+FP}$	0, 1	1
Precision	$\frac{TP}{TP+FP}$	0, 1	1
F$_1$ Score	$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$	0, 1	1
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	0, 1	1
Matthew's corr. coef.	$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$	-1, 1	1
Balanced accuracy	$\frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$	0, 1	1

as the Matthew's correlation coefficient or balanced accuracy control for class imbalance, which can be a particular problem in materials science. For example, if 99% of the data in a dataset are in the positive class, then a model could get 99% accuracy by always returning a positive prediction. Many of the same quantification techniques used in binary classification can be used in multi-class classification. Confusion matrices can be constructed Table 2.5. Similarly, previously discussed metrics (Table 2.4) can be adapted for multi-class use. This is done by adapting the problem into a binary classification problem. Two different methods of adaptation are commonly used, the macro and micro averages. In macro averaging, the prediction of each class is considered a binary classification task, binary classification metrics are used, and a mean of the result is taken. In micro averaging, each prediction is considered as a binary classification problem, and the total number of TPs, FPs, TNs, and FNs are aggregated before the calculation of the metric. The result is that micro-averaging controls for class imbalance, where macro averaging does not.

Table 2.5: Confusion matrix for multiclass classification. Here n_{ij} is the number (or sometimes the percentage) of data points that were predicted to be in class i, but were actually in class j. The correct predictions in this case would be seen in cells $n_{1,1}$, $n_{2,2}$, and $n_{3,3}$.

		Predicted class		
		1	2	3
Actual Class	1	n_{11}	n_{12}	n_{13}
	2	n_{21}	n_{22}	n_{23}
	3	n_{31}	n_{32}	n_{33}

Table 2.6: Prominent classification loss functions for machine learning given a target value, y, a predicted value, $\widehat{y}$, and n possible target classes. All loss functions shown here have a lower bounds of 0, no upper bounds. All loss values here are designed to be minimised, thus, for all loss functions here the optimum value is 0.

Loss Function	Formula
Cross Entropy	$-\sum_{i=1}^{n} y_i \log \widehat{y_i}$
Binary Cross Entropy	$-y \log \widehat{y} - (1-y)\log(1-\widehat{y})$
Kullback–Leibler Divergence	$\sum_{i=1}^{n} y_i \log \frac{y_i}{\widehat{y_i}}$

Similarly to how choice in regression metrics imparts bias, so does choice of classification metrics. Although accuracy may seem the easiest to explain, other options may be more appropriate. For materials science, recall or precision may be more important. Consider the example of a model that predicts whether a material is a superconductor. If an experimental chemist will take a prediction and synthesise it to test the result, choosing models with the highest recall will minimise the chances that a superconductor is missed. In contrast, if the aim is to minimise the number of experiments carried out before any superconductor is found, the model with the highest precision would instead be a better choice. This example will be revisited (Chapter 4). This is not to say that any choice in metric is better than any other, but instead that the advantages and disadvantages of each metric should be considered in the context that the metric is due to be used in.

2.1.5 Loss functions

Loss functions in supervised ML are considered metrics but tend to be selected such that they are convex, as convex loss functions have a single global optimal soultion and thus are easier to optimise than non-convex functions. While, at times, the phrase loss function and metric can be used in similar situations, in the context of this thesis a loss function will always refer to a metric which is used as part of the training process.

Because of the non-convex requirement often put on loss functions, different metrics to the ones outlined above are more common. Prominent loss functions not previously mentioned previously include the cross-entropy, binary cross-entropy, and Kullback-Leibler divergence loss [94], which all measure the difference between the true and predicted probability distributions for classification problems (Table 2.6).

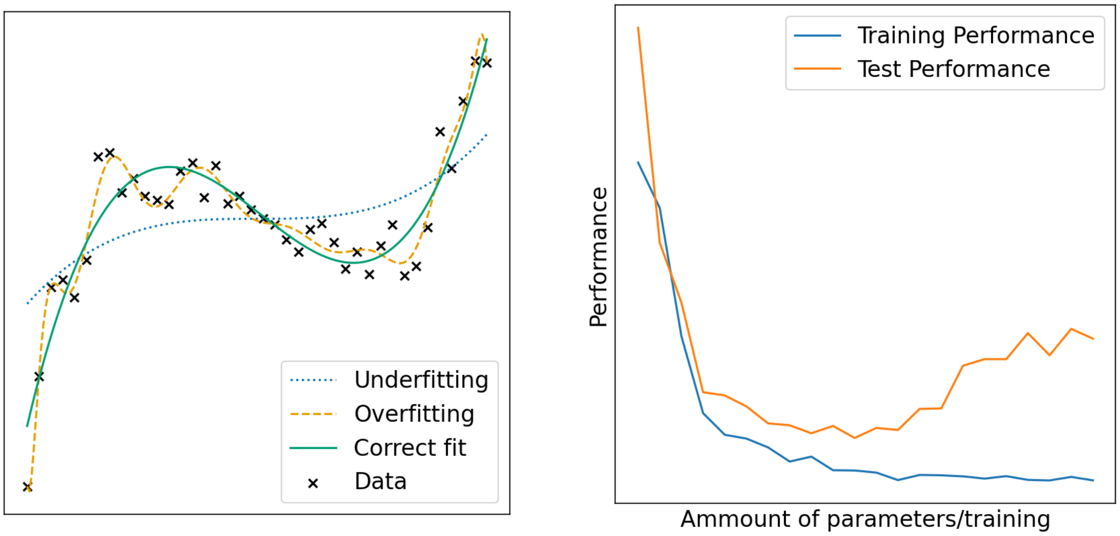
Prominent loss functions for regression functions have largely been discussed already. Specifically, the mean squared, mean relative, and mean absolute error are all convex and thus are used as loss functions (Table 2.1).

2.1.6 Evaluation schema for supervised machine learning

Having detailed several metrics for evaluating supervised ML, this section investigates how to use these metrics to gauge model performance. Another notable choice is how those metrics are applied. By definition, ML requires data to learn from, and testing on the same data from which it has learnt will give no insight into how well it has learnt to complete a task, just how well it has learnt the input data. If a model can only complete a task on the training data and fails to generalise to other data, this is known as overfitting; failing to learn at all is known as underfitting, and a large part of creating effective ML models is balancing the trade-off between the two (Figure 2.4a). This is sometimes called the bias-variance trade-off, but to avoid confusion with the social biases outlined above (Section 2.1.1) the terms overfitting and underfitting will be used here.

Most data points will include some amount of random noise, which can be caused by many different factors. In materials science, this is often instrument noise. An overfit model will learn the noise and data trend.

To test whether a model is overfit, or underfit, and to test the generalisability of a model, it is necessary to define processes to separate the data on which the model is trained from the data that are used to test the model. A common method of doing this is to separate the dataset into train, test, and validation sets. Models are trained with training data, tested,



(a)

(b)

Figure 2.4: (a) An example of models overfitting and underfitting to a noisy data trend. (b) An example of a training/test process for a ML model. Depending on the type of ML model being considered, the x axis could be training time, number of parameters in a model, or it could be amount of regularisation (with the left hand side being more regularisation).

and have hyperparameters optimised on the test set, and final models are evaluated on the validation set. Where models are not going into a production environment, the validation set is often left out. By establishing when increasing numbers of parameters, or training time does not result in improved performance on the test set, overfitting can be avoided (Figure 2.4b).

It could be that a test set is not representative of the entire dataset. K-fold cross validation is an alternative evaluation schema that aims to address this. In K-fold cross-validation, data is split into K random sets (or folds). K different models are trained, each using a different fold as the test set (and being trained on all data points). The final evaluation scores are the mean measurements taken at each fold. If classes in a dataset are imbalanced, the stratified variant of K-fold cross-validation ensures that each class is represented equally in all folds.

Evaluating the generalisability of ML models is a known challenge across data science, and is of particular concern in materials science, where data sets are of limited size com-

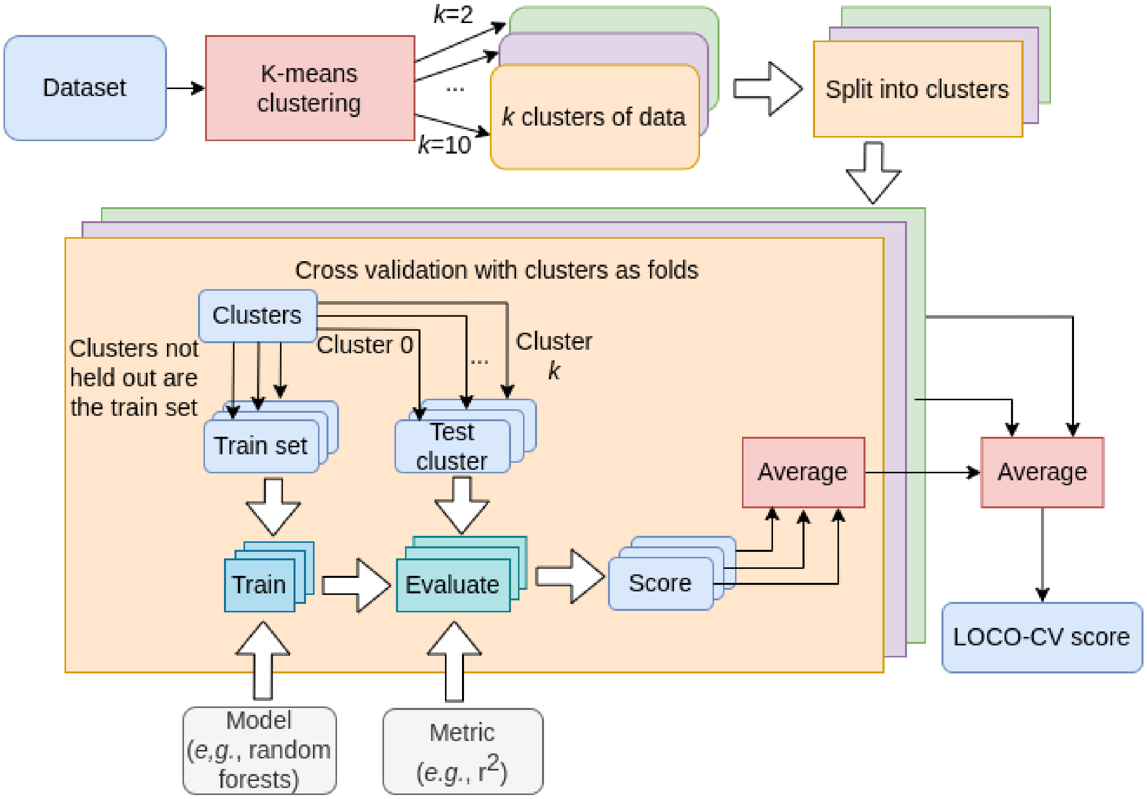


Figure 2.5: A flow chart demonstrating how the leave one cluster out cross validation (LOCO-CV) algorithm is used to measure the effectiveness of a ML model at extrapolating

pared with other application areas for ML, and often biased towards historically interesting materials or those closely related to known high-performance materials for certain performance metrics. Typically, models are evaluated on test sets separate from their training data, through a consistent train:test split or K-fold cross validation. However, this does not consider skew in a dataset. In chemical datasets, families of promising materials are often explored more thoroughly than the domain as a whole, which introduces bias and reduces the generalisability of ML models because the data they are trained and tested on are not sampled in a way representative of the domain of target chemistries to be screened with these models. Investigations into how such skew can affect ML models has seen that this skew can result in overfitting [184] and that more skewed datasets require more data points in order to train models to achieve achieve similar predictive performance when compared to models trained on less skewed datasets [149].

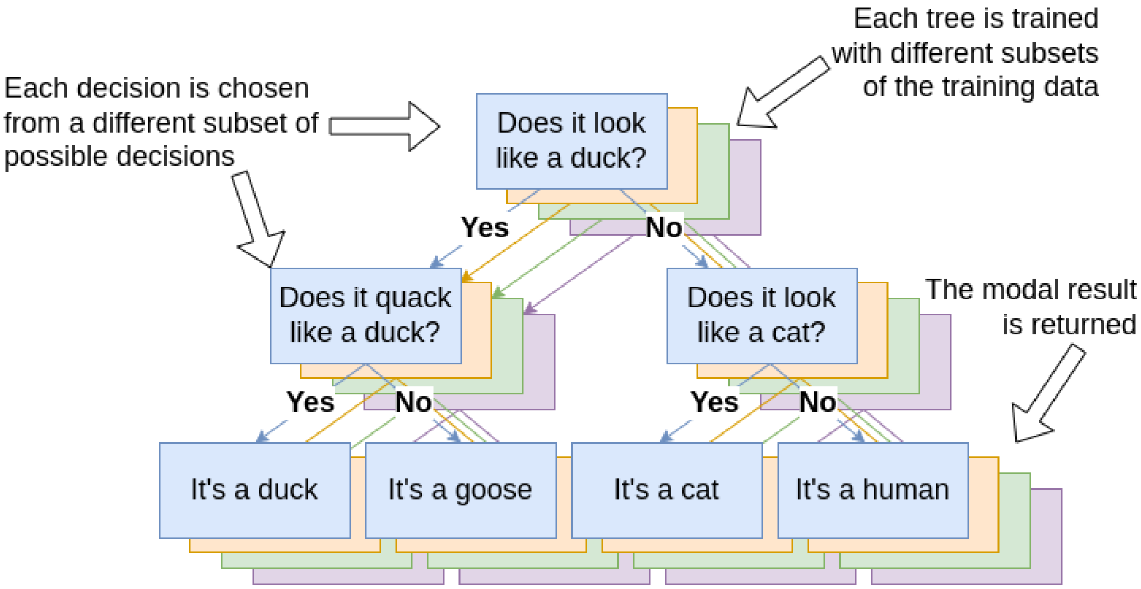


Figure 2.6: An example random forest classifier for classifier types of animals. Note that if this was a regression task the mean result would be returned rather than the mode.

Leave one cluster out cross validation (LOCO-CV) [117] is an evaluation schema developed in the materials science domain. The schema aims to measure the ability of models to extrapolate to data that are unlike data that it has seen before. To do this, rather than splitting data into random folds, data are clustered, and then clusters of data are left out as training sets (Figure 2.5). Clusters may be predefined or may be found using K-means clustering. Where the number of clusters is unclear, K can be iterated between 2 and 10.

2.1.7 Random Forests

RFs [19] are supervised ML algorithms that excel in a variety of fields [11, 179]. Although they are more than 20 years old, they are still a popular and useful tool. They have been used extensively in materials science [170, 144, 166] and are seen to perform well on data which are considered to be "tabular" (*i.e.*, data where each column represents a different feature describing a data point, and the order of the columns does not matter). They are fast, easy to use, and readily implemented [140].

RFs can be described as an ensemble of decision trees where each tree trained with bagging, and each decision is made with boosting (Figure 2.6). That is, each tree is trained with a random sample of all of the training data (sampled with replacement), and each decision is made with a random sample of all of the features. Classification and regression RFs return (respectively) the modal and mean results of their constituent trees.

Descision trees are ML algorithms which recursively split their training data based on which features best discriminate between target results. This is usually done by measuring differences in entropy between potential splits. Predictions are then made by applying these discriminations to input data, and return the training data which was discirminated in the most similar way. So, at each point in a decision tree a definite and explainable reason can be extracted as to the decision tree's performance (Figure 2.6).

RFs ensemble application of decision trees does reduce their explainability, but individual trees can still be extracted to explain performance. Introduction of randomness means that noise is (to an extent) cancelled out because problematic data points or features of data are excluded in at least some of the trees in a RF. This leads to the claim from the author that they are "impervious to overfitting" with a succinct accompanying proof [19]. More precisely what it means is that adding more trees to the RF will never cause that RF to overfit to the data. This is in contrast to neural networks, in which adding more parameters can cause overfitting if there are limited data.

Other aspects of overfitting (for example, the ability of a single incorrect datapoint to throw off accuracy) do affect RFs. As such, several strategies have been suggested to reduce these effects and improve performance on small and/or noisy datasets. These modifications include setting a maximum on the depth of the decision tree, specifying the minimum number of samples required on a leaf node to stop the recursive splitting process, or specifying the maximum number of leaf nodes. These modifications to the algorithm are considered (in the scope of this thesis), to be hyperparameters. Other hyperparameters include the number of trees in the forest, the number of sample of the data set used to train each tree, and the number of features available in each split of a tree.

2.1.8 Deep Learning and Neural Networks

Neural networks are a wide and varied class of ML algorithms [165]. This subsection will discuss high-level concepts needed for understanding appropriate uses of neural networks, specific architectures will be discussed in the context of their use in this thesis.

At their most basic, neural networks can be seen as a successive application of weighted sums, and non-linear (activation) functions, where the weights, w, of the sums are learnt

(Figure 2.7). As many weighted sums are taken at once, in practise these weighted sums are often implemented as matrix multiplications, where inputs are multiplied by matrices of learnt weights before the activation function. Each matrix multiplication and activation function pair are referred to as “layers”. Deep learning is a broad term used to refer to neural networks with many layers. After all layers have processed an input (known as the forward pass), the output is compared to the ground truth with a differentiable loss function, and the result is known as the loss, l of the network. l can be differentiated with respect to the weights of the matrix to get $\frac{\partial l}{\partial w}$, and by successive application of the chain rule, the loss caused by each weight in each layer, $\frac{\partial l}{\partial w}$, can be passed back through the network (known as the backward pass) (Figure 2.7b).

Forward and backward passes of the whole training dataset are done in batches (also called mini-batches). After each batch has been procesed the mean $\frac{\partial l}{\partial w}$ across the batch can be calculated, and the gradient of the loss function can be descended by adjusting the w (various weight adjustment schema exist [37, 85, 109]), hopefully reducing loss for future predictions. After all batches have been processed, a forward pass is done on the test data, and performance is measured. This process is then repeated for a number of cycles (epochs) until the model stops improving on the test set (Figure 2.4b).

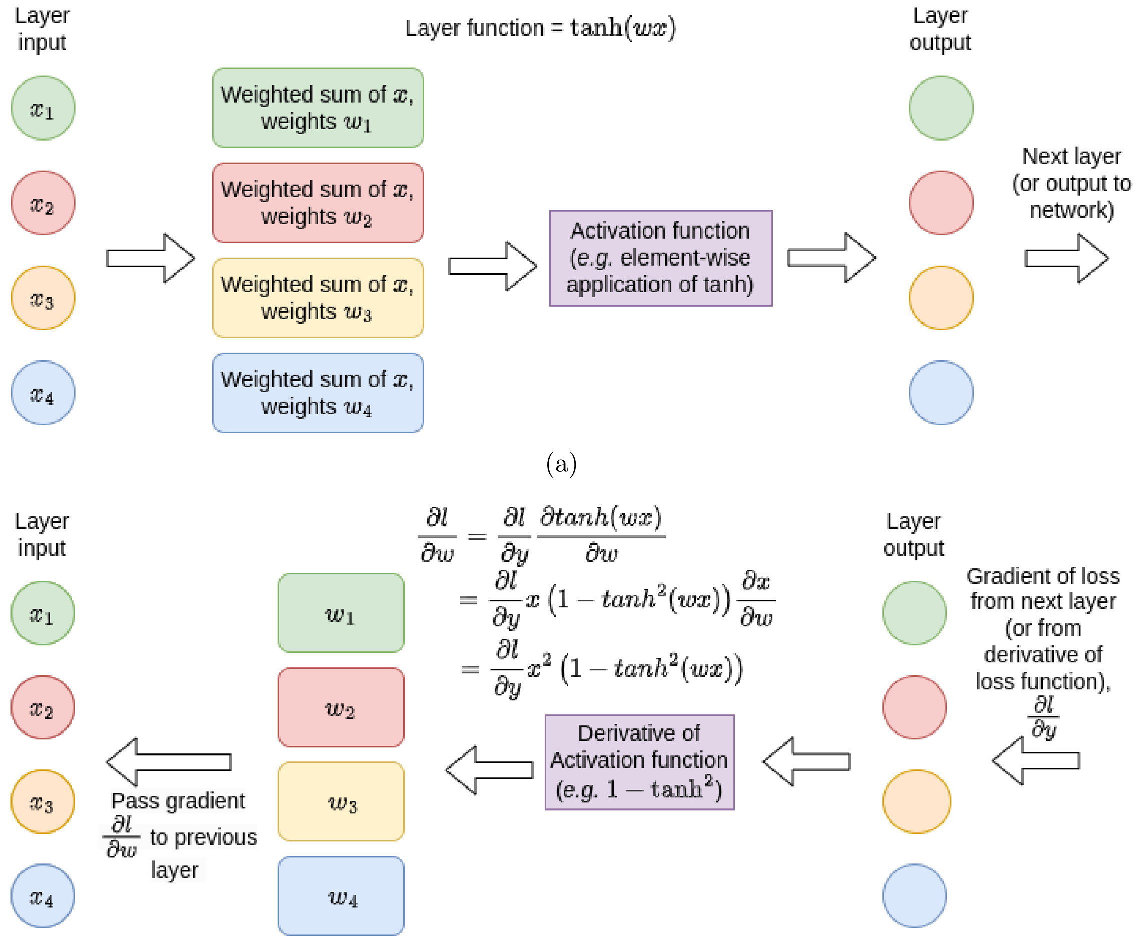
The above process describes “dense” networks (also called linear or fully connected networks). But variants which apply different linear and non-linear functions are specialised for different functions. Some examples which will be explored in this thesis include:

- Convolutional neural networks apply sliding windows (“kernels”) across data, weights of these kernels are learnt. These are commonly used for image processing. [98]

- Recurrent neural networks (RNNs) are applied to series of data. The series is processed in time steps, and each recurrent layer takes in the input at the current time step and the output of the layer at the previous time step. These networks are commonly used in tasks where a sequence needs to be considered. [70]

- Encoder-decoder networks take in a sequence of data, and use networks (such as RNNs) to encode data into a single representation and then a separate network to decode the data (sometimes into another sequence of data). [82]

- Auto-encoders are encoder decoder networks which are trained by taking an input, encoding it through some bottleneck, and then trying to reconstruct the original input. [165]


Layer input
Layer function = $\tanh(wx)$
Layer output
x_1
x_2
x_3
x_4
Weighted sum of x, weights w_1
Weighted sum of x, weights w_2
Weighted sum of x, weights w_3
Weighted sum of x, weights w_4
Activation function (*e.g.* element-wise application of tanh)
Next layer (or output to network)
Layer input
$\frac{\partial l}{\partial w} = \frac{\partial l}{\partial y}\frac{\partial tanh(wx)}{\partial w}$
$= \frac{\partial l}{\partial y} x\left(1 - tanh^2(wx)\right)\frac{\partial x}{\partial w}$
$= \frac{\partial l}{\partial y} x^2\left(1 - tanh^2(wx)\right)$
Layer output
Gradient of loss from next layer (or from derivative of loss function), $\frac{\partial l}{\partial y}$
w_1
w_2
w_3
w_4
Derivative of Activation function (*e.g.* $1 - \tanh^2$)
Pass gradient $\frac{\partial l}{\partial w}$ to previous layer

(a)

(b)

Figure 2.7: (a) The forward pass of a layer of a neural network. (b) The backward pass of a layer of a neural network. The gradient of the loss function with respect to the weights, $\frac{\partial l}{\partial w}$, is calculated and passed back to the previous layer. Note that line 2 of the derivation is the application of the chain rule, and line 3 involves application of the product rule.

- Variational auto-encoders are similar to auto-encoders but learn a distribution of data, rather than a single vector. This distribution is then sampled from to pass to the decoder. [86]

- Transformer networks consider a sequence but introduce the possibility of non-sequential interaction between points in the sequence. The process of deciding which parts of the sequence interact with each other is learnt. Transformers are often in an encoder-decoder layout and are currently considered the state of the art in many fields including natural language processing. [181]

.

Neural networks are a rapidly evolving field and are commonly used in the literature. As such, areas of this thesis will explore their use.

2.1.9 Repeatability in machine learning

Most machine learning algorithns (including all of the supervised algorithms examined in this thesis), require some degree of stocastic input. Whether that be the bagging and boosting of RFs, or the random initial weights of neural networks. As such when evaluating these models, performance will differ between runs. There is some evidence that this is particularly relevant in the materials domain with, differences in r^2 of up to 0.15 seen dependant on the train, test split [187].

Differences in performance between runs, train/test splits, or test folds is important to communicate and is often done either with seperate tables or using error bars on graphs. While this information can be seen in a couple of points in this thesis (Sections 3.2.2 and 5.4), generally the reporting of this uncertainty is not in this thesis. Performance numbers, may be taken with a grain of salt and readers rellying on these results are encouraged to recreate any findings using provided resources.

2.2 Key chemistry and materials science theories and related work

While most of the required background knowledge on materials science will be introduced in the context of its use, this section serves as background knowledge that may prime a reader with a computer science background. Broad categories of materials will be introduced, as will methods of representing these materials. Pitfalls with material representation and background about types of superconductors will also be introduced. The materials examined in this thesis fall into two broad categories, inorganic crystals, and metal organic frameworks (MOFs).

Inorganic crystals are lattices which do not contain carbon in combination with hyrdogen, oxygen or nitrogren. They are varried in properties and, as such, most description of specfic classes of materials will be given in the context for their use throughout the thesis. Information about these materials are often stored digitally in crystallographic information file (CIF) format [62]. The regular arrangements of atoms result in unique properties such as high hardness, high melting points, and excellent electrical conductivity. Inorganic materials include metals and non-metal.

Inorganic materials may also exhibit interesting electronic phenomena, some of which will be explored in this thesis. In particular, two phenomena which will be explored are megneto-caloric materials and superconductors.

MOFs feature metal metals, in regular arrangements joined together by organic molecules reffered to as linkers or ligands. These ligands are often characterised by long chains of carbon atoms as well as other elements such as hydrogen, oxygen, nitrogen, and sulphur. Organic materials have wide-ranging applications depending on their molecular structure [55].

2.2.1 SMILES strings

Simplified Molecular Input Line Entry System SMILES is a string format used to represent the structure of organic molecules (such as MOF linkers) in a concise and unambiguous way (Figure 2.8) [198]. SMILES strings consist of a sequence of characters that represent atoms, bonds, and other molecular features.

In an SMILES string, the atoms are represented by their atomic symbols, and the bonds between atoms are represented by their order. Additional characters give information about types of bonds between atoms (*e.g.*, single bond is represented by "-", double bond by '="). The SMILES string also includes additional symbols to represent ring structures and other molecular features.

SMILES strings provide a compact and standardised way of representing molecular structures that can be easily read and parsed by code. As such, they are commonly used in datasets of organic materials.

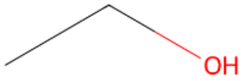
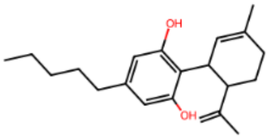
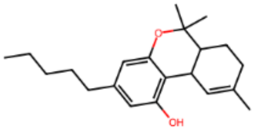
Chemical Formula	SMILES string	Visualisation
C_2H_6O	CCO	
$C_{21}H_{30}O_2$	CCCCCC1=CC(=C(C(=C1)O)C2C=C(CCC2C(=C)C)C)O	
$C_{21}H_{30}O_2$	CCCCCC1=CC(=C2C3C=C(CCC3C(OC2=C1)(C)C)C)O	

Figure 2.8: Examples of SMILES strings for popular chemicals. SMILES strings allow the encoding of structural information, even when two organic compounds have the same chemical formulae.

2.2.2 Unit Cells and Lattices

A lattice (or bravais lattices) can be said to be “a set of points where the environment of any given point is equivalent to the environment of any other given point.” [167]. In the context of materials the set of points would be atoms, equivalent environments would be the atoms of the same elements in another part of the crystal structure.

This allows for deffinition of a unit cell as “a region of space such that when many identical units are stacked together it tiles (completely fills) all of space and reconstructs the full structure” [167]. In the context of materials, this would be the region of space surrounded the atoms which is repeated throughout a crystal.

The smallest possible unit cell is referred to as the primitive unit cell. However, often a different unit cell is picked to better show the symmetry of the matterial; this is called the conventional unit cell [167]. There is some subjectivity as to which unit cell is the conventional unit cell, as the strictness of the symmetries to include means that some materials could be considered to have multiple conventional unit cells. The conventional unit cell is widely used in materials science to describe the structure and properties of crystalline materials.

As unit cells exist in 3 dimensions, they are quantified by three lengths, listed in size order as a, b, and c. These, together with angles are α, β, and γ that are the angles between lengths a, b, and c, can define the volume of a cell, V.

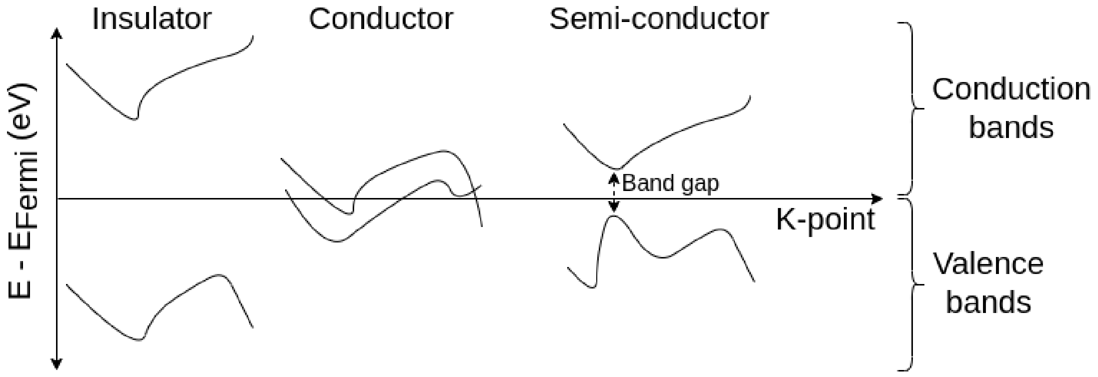


Figure 2.9: Example of an electronic band structure.

The Niggli reduced unit cell reduced unit cell [61] is a way of simplifying the description of a crystal lattice by reducing the size of the unit cell while preserving the symmetry of the lattice. It is obtained by a sequence of operations that transform the lattice into a simpler lattice that has the same symmetry, but with a smaller unit cell volume.

2.2.3 The Reciprocal Lattice

As wave properties of electrons are causes of electronic phenomena, it is normal to consider the reciprical lattice of a material. The Brilliouin zone is "any primitive unit cell of the reciprocal lattice" [167]. By describing a wave K in terms of the reciprocal lattice, translating K by a reciprocal lattice vector G will result in the same wave (as waves are preiodic).

This means that in reciprocal space, an area can be defined that describes a set of waves such that each wave occurs once and only once. In other words: "Start with the reciprocal lattice point G = 0. All K-points which are closer to 0 than any other reciprocal lattice point define the first Brillouin zone. Similarly, all K-points where the point 0 is the second closest reciprocal lattice point to that point constitute the second Brillouin zone, and so forth. Zone boundaries are defined in terms of this definition of Brillouin zones." [167].

2.2.4 Electronic band structure

The previous subsection outlined the way that a set of waves can be considered to be a set of points that lie within the first Brillouin zone, which is in the reciprocal space. Electrons form standing waves, and it is possible to compute the energy levels that an electron could have at a given point in the reciprocal space (K-space) [167]. This is done using density functional theory (DFT) [80], which is considered beyond the scope of this thesis.

Electornic band structures are often displayed in two dimensions, with one dimension being a path through the K-space, and the other dimension being the energy levels at which electrons can exist (Figure 2.9). These are normalised to the Fermi level, E_{Fermi}, which is "is the chemical potential at temperature T = 0." [167]. This means that when temperatures are above 0 K, electrons may occupy bands above 0 on the plot and may. Bands below 0 on this plot are considered to be valence bands, with bands above 0 being conduction bands. Bands crossing 0 implies that a material is electrically conductive. The gap between bands above and below the Fermi level is called the band gap, and large band gaps result in very insulating materials. Where band gaps are small electrons do not need to have much energy to cross from occupying valence bands to conduction bands, and as such the material will be a semi-conductor.

2.2.5 Composition-based representations of materials

Numerous suggestions for suitable representations have been made, both including and excluding structural information. Examples of these include Coulomb matrices (and variants there of), density of states fingerprints [22].

Although representations of materials will be explored in more depth (Chapter 5), a basic introduction is needed to provide context to the rest of the thesis. Composition-based representations are representations of a material based solely on composition and do not include any structural information.

As structural information is usually not known *a priori* to chemical synthesis, the ML predictions made using structural information are of more limited use than those made using a composition-based representation. Using composition as the basis for the representation in an ML model means that arbitrary compositions can be generated and screened.

The caveat to this is that materials with the same composition can have vastly different structures and properties. In the most extreme case, diamond and graphite are both made of carbon and have very different properties (it would be unusual to propose to someone with a graphite ring). The consequence is that no composition-based predictor will be perfectly accurate.

Many composition-based representations exist [124, 192, 195, 27]. There are several libraries that exist to easily create such representations [192, 121, 38], in a process often called featurisation. As such, many ML papers create bespoke representations to tackle problems [170]. However, as will be discussed (Chapter 5) this reduces repeatability if the creation of these bespoke representations is not properly detailed and leads to many representations that have not been well justified by advantages over competing representations [124].

A common representation for a material is *magpie*. While initially being the name of the library used to create a representation [192], *magpie* has become synonymous with a representation of 115 elemental based attributes. The minimum, maximum, range, standard deviation, mode (property of the most prevalent element) and weighted average of 23 elemental properties taken across the elements present in a composition. The remaining features are derived from valence orbital occupation and ionic compound attributes (which are based on differences between electronegativity between constituent elements in a compound).

Although featurisation is common, another popular way of representing a composition is an n-hot encoding of its elements [124, 79, 45]. This results in a (usually) 119 long sparse vector, with each entry in the vector corresponding to an element and the value of that entry representing the proportion of that element in a composition. Sometimes this composition is normalised so that the sum of the values in the vector is 1 [124, 45], other times this is not the case [79].

2.3 Thesis Context

Having provided a comprehensive examination of the background information relevant to the research presented in this thesis, the groundwork has been established to transition towards the contributions.

Chapter 3

Use of random forests for prediction of material properties

Random forests (RFs) are strong and versatile predictors which often outperform more recent and computationally intensive algorithms [155, 144]. They provide good out of the box performance, with little hyperparameter tuning needed, and are readily implemented [140].

RFs are supervised machine learning (ML) algorithms, which take a fixed sized input vector and a target value, with each value in the vector having a fixed meaning across vectors. As such data suited to RFs is often referred to as being "tabular," with each row representing a data point and each column representing a feature. RFs are an ensemble of decision trees, with each tree being trained on a subset of the training data, which is sampled with replacement (this subset is referred to as the tree's bag). At each node in the decision tree a random subset of features are used and of those the one which is calculated to be the most discriminatory to the target labels (Figure 2.6). Data is then split into two partitions based on the most discriminatory feature, with each partition being used to create child nodes. Child nodes are recursively added, until the tree reaches a fixed depth, until the leaf nodes are left with a pool of data of prespecified size, or until the bag has been fully discriminated. Depending on whether an RF is being used as a classifier or regressor, it will return the modal or mean value returned by its decision trees, respectively. More details can be found in Section 2.1.7.

RF's robust nature has made them often used in materials science [170, 166]. This chapter explores two examples of their use and discusses issues that arise when applying RFs in this field.

3.1 Predicting the shape of unit cells

Note: This section is adapted from results published in Advanced Functional Materials [125]

3.1.1 Motivation and background

The magneto-caloric effect is phenomena of a material to change temperature when exposed to a magnetic field [125]. In the PbFCl family of materials this has been seen to correlate to the ratio of lengths within the conventional unit cell (further explored in Section 2.2.2), namely the transition temperature, T_c correlates to $\frac{c}{a}$ ratio [125, 20].

This is pertinent as datasets of compositions and associated unit cells are widely available [69, 78, 134], unlike datasets associating magneto-caloric materials with their T_c. A lack of magneto-caloric datasets makes direct prediction of T_c infeasible. As such $\frac{c}{a}$ was seen as useful descriptor to calculate as proxy for the T_c value in the PbFCl family of materials.

Using chemical intuition, or physical formulae, predicting the $\frac{c}{a}$ ratio of structures from a wide variety of bonding characters (e.g., ionic, covalent, metallic) can be difficult. Standard models for such a prediction are often limited to a specific structure-type and often fail without prior knowledge of the adequate descriptor to use [20]. Owing to the importance of the $\frac{c}{a}$ parameter [20], RFs were used to predict $\frac{c}{a}$ using the experimental data available in the ICSD [69] as well as a hand curated dataset of 65 PbFCl type structures obtained from Pearson's database. [134].

RFs were used as they have been used successfully in a wide variety of applications both chemical and non chemical [166, 58]. This combined with their simplicity, relative hyperparameter ambivalence, and computational simplicity when compared with competing supervised machine learning algorithms such as deep neural networks led to the choice use the sci-kit learn [140] implementation of RFs with default hyperparameters (as of sci-kit learn version 0.22) for the following investigations. While improvements could be made to metrics for success by tuning hyperparameters, the novelty of this investigation comes from its apathy towards what are often focused on in papers (algorithms, representations, hyperparameters), in favour of focusing on applicability to the problem at hand.

Properties such as the $\frac{c}{a}$ ratio are can targeted by substituting one element in a similar (parent) composition with another, and adjusting ratios of elements so that the result is a charge balanced (child) composition. In the absence of machine learning, the Shannon ionic radius may be used by chemists as a predictor of how the unit cell will change with substitution elemental substitution in a compound. This proxy is compared to the predictive ability of RFs.

The context of such substitutional studies allows use of RFs as binary classifiers to measure the direction of change of the target property between parent and child compound, rather than regressors to predict the value of the target property in the child compound. In this case, that would mean predicting whether a chemical substitution would result in an increase or decrease in the $\frac{c}{a}$ ratio between the parent and child, rather than predicting the $\frac{c}{a}$ ratio of the child compound.

The introduction of a parent compound creates new opportunities to analyse the performance of an ML model. Inclusion of parent compounds in measuring performance of ML can help inform experimental chemists as to the expected performance of that model in the desired context [125]. New performance metrics, the ordinal accuracy and the comparative r^2 (r^2_{comp}) which incorporate the parent compound are defined and examined here.

The key investigations and contributions of this section are:

- Investigating the ability of RFs to predict the $\frac{c}{a}$ ratio of a conventional unit cell.
- Presenting a comparative prediction method for using ML algorithms. This method is particularly suited for substitutional synthesis studies.
- Presenting and investigating performance metrics r^2_{comp} and ordinal accuracy which examine the performance of a machine learning algorithm in the context of comparative predictions.

3.1.2 Comparative machine learning models for substitutional chemistry studies

Although the contributing factors that determine $\frac{c}{a}$ of a conventional unit cell cannot be easily enumerated, they are present to such an extent in data that ML methods can be used to approximate them. One approach to making such an approximation would be choosing a representation for a material and training an RF to predict known values of c and a either separately or in the form $\frac{c}{a}$ (Figure 3.1a). This could then be used to screen potential candidates for chemical synthesis.

However, most functional properties are tuned through substitution studies, where the researcher wishes to alter a known (parent) compound to find a better performing child compound. If prediction of the value of this child's functional property is framed

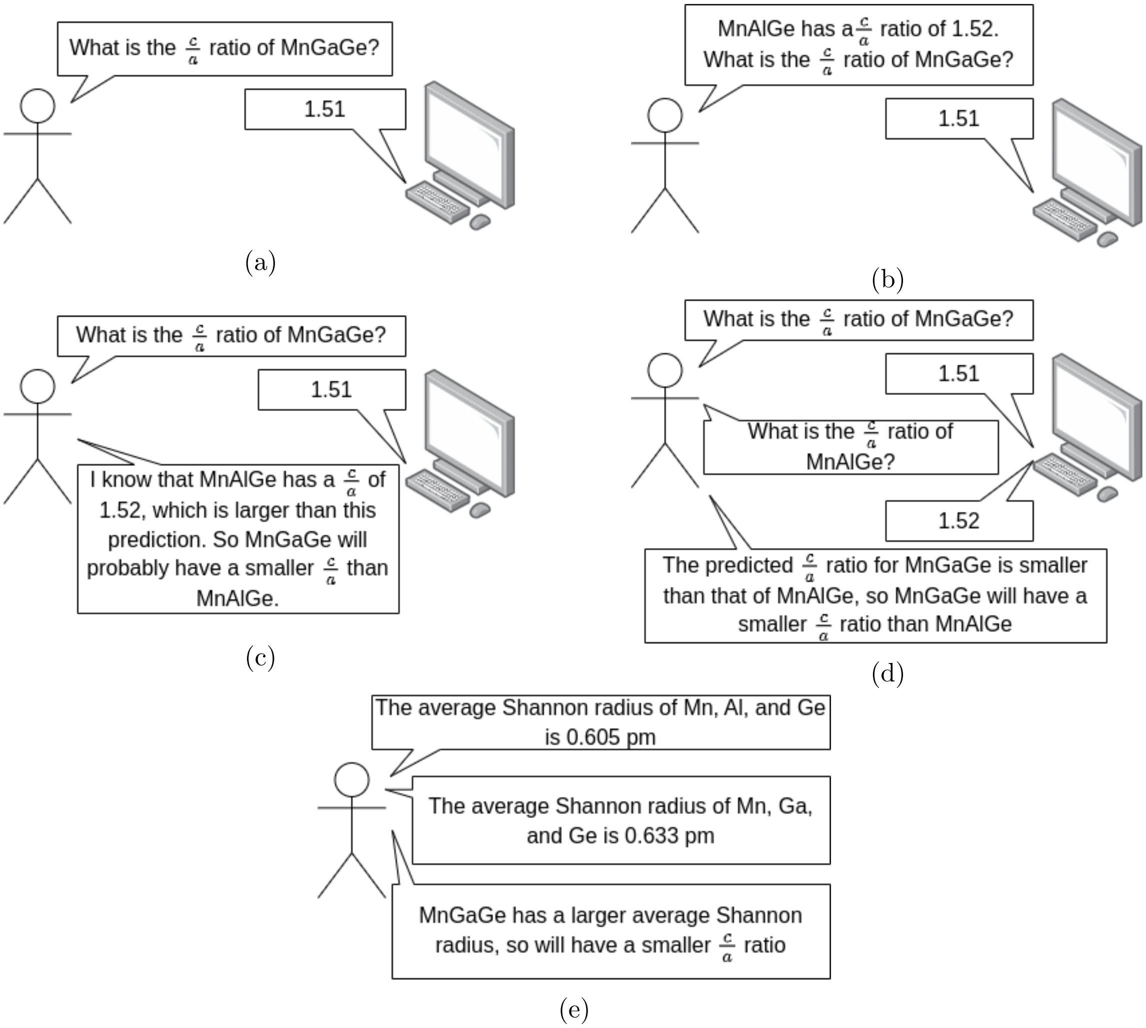

What is the $\frac{c}{a}$ ratio of MnGaGe?
1.51
(a)
MnAlGe has a $\frac{c}{a}$ ratio of 1.52. What is the $\frac{c}{a}$ ratio of MnGaGe?
1.51
(b)
What is the $\frac{c}{a}$ ratio of MnGaGe?
1.51
I know that MnAlGe has a $\frac{c}{a}$ of 1.52, which is larger than this prediction. So MnGaGe will probably have a smaller $\frac{c}{a}$ than MnAlGe.
(c)
What is the $\frac{c}{a}$ ratio of MnGaGe?
1.51
What is the $\frac{c}{a}$ ratio of MnAlGe?
1.52
The predicted $\frac{c}{a}$ ratio for MnGaGe is smaller than that of MnAlGe, so MnGaGe will have a smaller $\frac{c}{a}$ ratio than MnAlGe
(d)
The average Shannon radius of Mn, Al, and Ge is 0.605 pm
The average Shannon radius of Mn, Ga, and Ge is 0.633 pm
MnGaGe has a larger average Shannon radius, so will have a smaller $\frac{c}{a}$ ratio
(e)

Figure 3.1: Strategies for property prediction using RF. (a) Directly predicting the property. (b) Supplying information about a parent compound to directly predict the property of a child compound. (c) Using knowledge of a parent compound in comparison to the prediction of a child compound. Using that comparison as an indicator for the direction of change of that property. (d) Predicting a property for both a parent and child compound, comparing the output and using that comparison as an indicator for the direction of change of that property. (e) Comparing known statistics about the constituents of a compound and using that comparison as an indicator for the direction of change of a property.

as a regression task, the inclusion of information from the parent compound allows the reframing of this task(Figures 3.1b and 3.1c). The ML algorithm being used could be used to trained as a regressor to find the child property relative to that of its parent, or the ML algorithm could be trained in binary classification task of whether the magnitude of this property will increase or decrease between the parent and the child. This reframing of the task is dubbed “comparative prediction” and is explored in this section. While this comparative prediction has been developed for use in substitutional chemistry studies, it can be generalised to any object where a perturbation is applied to a parent object to create a child object whose properties are to be predicted.

For clarity, this comparative prediction method will first be defined for an arbitrary property, y, in a parent-child system where the child is some parametric alteration of the parent. y_1 is the value of y in a parent object, and y_2 is the value of y in a child object.

When examining a model trained in this paradigm, it is useful to examine whether large changes in y between the parent and child in ground truth data are reflected by the model’s predicted changes. The r^2 metric (discussed further in Section 2.1.4) can be used to examine these trends, allowing the definition of “comparative r^2”, r^2_{comp} as:

$$r^2_{comp} = r^2\left(\left(y_{1,\text{true}} - y_{2,\text{true}}\right), \left(y_{1,\text{true}} - y_{2,\text{pred}}\right)\right)$$

Where $y_{j,\text{true}}$ is the true value of y_j and $y_{j,\text{pred}}$ is the predicted value of y_j and $j \in \{1, 2\}$. When evaluating the comparative prediction paradigm, it could be helpful to also evaluate models which just predict y_2, without knowledge of y_1. In such cases, it is unclear whether the definitions of r^2_{comp} should use the r^2_{metric} defined above, or if knowledge of $y_{1,\text{true}}$ should be ignored by the model (Figure 3.1d) (as similar errors in prediction of $y_1, pred$ and $y_2, pred$ may cancel each other out, explored further in Section 3.1.4). Both definitions are explored in this chapter, with the models evaluated with the latter schema will be referred to as models which are “comparing to prediction”. Thus, the final definition of r^2_{comp} is as follows:

$$r^2_{\text{comp}} = \begin{cases} r^2\left(\left(y_{1,\text{true}} - y_{2,\text{true}}\right), \left(y_{1,\text{pred}} - y_{2,\text{pred}}\right)\right), & \text{if model compares to prediction} \\ r^2\left(\left(y_{1,\text{true}} - y_{2,\text{true}}\right), \left(y_{1,\text{true}} - y_{2,\text{pred}}\right)\right), & \text{otherwise} \end{cases} \tag{3.1}$$

Although r^2_{comp} provides a metric for investigating the comparative prediction paradigm as a regression task, it is possible to simplify this task to a binary classification task. Rather than training a ML model to predict y_2, one can use a target value of $y_1 > y_2$,to limit the output to a Boolean domain. The accuracy of this prediction can be said to be the proportion of predictions in which the direction of change was predicted correctly. We label this metric the "ordinal accuracy."

Consider the context of predicting $\frac{c}{a}$. If $\left(\frac{c}{a}\right)_1 > \left(\frac{c}{a}\right)_2$ then an ordinally accurate prediction of $\left(\frac{c}{a}\right)_2$ ($\frac{c}{a}$ of the child compound) would be less than the value of $\left(\frac{c}{a}\right)_1$. Conversely $\left(\frac{c}{a}\right)_1 < \left(\frac{c}{a}\right)_2$ then a ordinally accurate prediction of $\left(\frac{c}{a}\right)_2$ would be greater than the true value of $\left(\frac{c}{a}\right)$. For example, if the true value of y_1 is 10 and the true value of y_2 is 7, an ordinally accurate prediction of y_2 would be any value less than 10. Ordinal accuracy for a dataset can thus be said to be the ratio of predictions which are ordinally accurate, divided by the number of predictions made.

Similarly to the definition for r^2_{comp} (Equation (3.1)), it is unclear if the definitions of ordinal accuracy should ignore knowledge of $y_{1,\text{true}}$ (as similar errors in the prediction of $y_{1,\text{pred}}$ and $y_{2,\text{pred}}$ may cancel each other out). Like r^2_{comp}, this will be explored in a schema labelled "comparing to prediction." When considering the comparing to prediction schema, an ordinally accurate prediction would be one where $\left(\frac{c}{a}\right)_{1,\text{pred}} > \left(\frac{c}{a}\right)_{2,\text{pred}}$ in the case where $\left(\frac{c}{a}\right)_{1,\text{true}} > \left(\frac{c}{a}\right)_{2,\text{true}}$.

Using elemental statistics for comparative studies

The above outlined the comparative paradigm that this investigation will examine and noted two new metrics of success that can be used to evaluate this paradigm (r^2_{comp} and comparative accuracy). Although this investigation focusses on RFs, other models may be more suited to this task. There is also no need for these models to be ML models. Simple proxies can be used.

Although the use of ML for the comparison of parent and child materials in a chemical substitution study is novel, the practise of using the chemical properties of the constituents of the material as a proxy for its potential properties is prevalent in materials science [](Figure 3.1e). For example, when looking to increase the cell size of a material, one may target a substitution of one element of the material for an element with a larger atomic radius. The common practise of using elemental statistics can also be evaluated with the r^2_{comp} and comparative accuracy metrics discussed.

Shannon radius of an atom was suggested as a proxy for tuning the $\frac{c}{a}$ ratio of PbFCl type compounds. As such, this will be investigated and compared to the models developed for this study.

3.1.3 Material representation

To train RFs to predict $\frac{c}{a}$ of a material, a representation for that material must be chosen. Previous work used purely chemical descriptors to predict unit cell volume with some level of accuracy [28]. Although this level of accuracy may be improved by including structural information, this previous success justifies a focus on chemical descriptors when looking at other unit cell parameters such as $\frac{c}{a}$. The literature has also reported predictions of various chemical properties based only on elemental composition [79]. This seemed to be the simplest possible chemical descriptor as it can be derived purely from chemical formulae with no further domain knowledge required. This representation was used as a basis in all investigations presented here on the prediction of unit cell properties. The results reported in this chapter inspired a more thorough review of composition-based representation which can be found in Chapter 5.

All the models examined took one of two inputs. The simplest models received the elemental composition for a compound encoded as a vector of floating points, where each floating point represented the number of atoms for an element divided by the total number of atoms in the compound (this representation is referred to as $compVec$). From this vector, the models predict unit cell parameters for this composition.

This vector cannot be used with “comparative” models as information about the parent compound must also be included. Comparative models were proposed for use in substitution studies where a “parent” compound is known and the researchers wish to examine an altered “child” form of that compound. The input into the models was a vector encoding of the composition of the parent and the child, and the parameters of the unit cells for the parent compound ($a, b, c, v, \alpha, \beta$, and γ).

Benchmarks, Data splitting, and Metrics

Data used

Data obtained for this study were split into 3 data sets depending on the origin:

1. ICSD - all compounds in the inorganic crystal database as of 2018 [69].
2. ICSD and Materials Project - a union of experimentally measured structures in the ICSD and density functional theory (DFT) measured structures with a tetragonal symmetry in the Materials Project [78].[1]

[1] While all models were tested with this dataset and consistently converged, none of the models performed

3. PbFCl structure-type - All PbFCl type structures in Pearson's Crystal Database[134]

Each data point (in the form of a cif) was preprocessed to extract the relative elemental constituents of the compound from its formula, as well as the unit cell parameters. Data were then split into training and test sets and into clusters for use with leave one cluster out cross validation (LOCO-CV) [117].

For all datasets, most of the models were trained on a random 80% selection of the data and tested on the remaining entries in all experiments. This split was consistent for all experiments. For comparative models, which rely on data points being paired into parent and child compounds, it was ensured that both parent and child compounds were either both in the training set or both in the test set.

For a measure of the extrapolatory power of the models, LOCO-CV was also performed with K-means clustering (clustered on the inputs to the model) for values of k between 2 and 10 inclusive (Figure 2.5). The implementation of K-means clustering used was that found in sci-kit learn all but two hyperparameters set to default as of version 0.22 [140]. First, the number of clusters was varied (as required to perform LOCO-CV measurement), and second, the initial placement of the clusters was set to random as opposed to the default which is K-means++ [7].

Caution should be taken when analysing the LOCO-CV scores obtained as the clusters were universally uneven in size. Although marginal (but not decisive) improvements were observed using random placement rather than K-means++, cluster sizes almost always varied by at least two orders of magnitude. The implications of this finding will be explored more in the discussion section (Section 6.5).

3.1.4 Results

Models were tested for their ability to predict $\frac{c}{a}$. In total, five methods for estimating $\frac{c}{a}$ or change in $\frac{c}{a}$ were tested:

- Using a RF to predict $\frac{c}{a}$ of a material encoded with *compVec* (the results are under the subheading Random forest prediction).

better than when trained using data from the ICSD alone because of this time will not be spent analysing these results. The most likely cause is due to the mixture of data of entirely different origin, as it is well known there are significant and systematic differences between unit cell parameters generated by DFT (in the Materials Project) and determined by experiment (in the ICSD).It is worthwhile noting here that augmenting a dataset should be attempted with care and a critical eye, and does not necessarily result in improved model performance.

- Using a RF to predict $\frac{c}{a}$ of a child material. As input, this RF takes $compVec$ encodings of the child and parent material and unit cell parameters of the parent material. The results are under the subheading Random forest prediction (trained on parent and child compounds).

- Using a RF to predict $\frac{c}{a}$ of a child material encoded with $compVec$, comparing the output of that with the knowledge of $\frac{c}{a}$ of the parent material and examining that comparison rather than the prediction itself. The results are under the subheading Comparing predicted unit cell values for a child to known unit cell values for a parent.

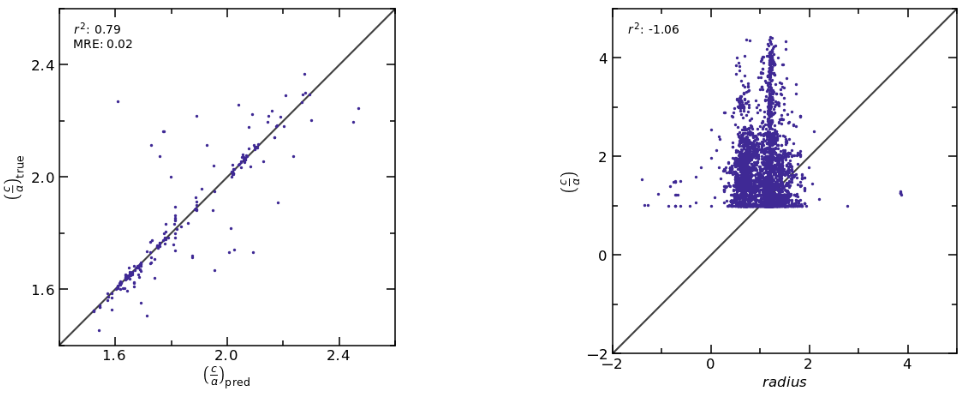
- Using a RF to predict $\frac{c}{a}$ of a child material encoded with $compVec$, using the same RF to predict $\frac{c}{a}$ of the parent material and comparing the two predictions. An examination is done as to the effectiveness of the comparison of the predictions rather than either of the predictions individually. The results are under the subheading Comparing predicted unit cell values for a child to predicted unit cell values for a parent.

- Comparing the average Shannon radius of the elements in the parent compound to the average Shannon radius of the elements in the child compound and examining that comparison as a proxy for the change in $\frac{c}{a}$. The results of this are given under the subheading Using Shannon radius as a benchmark.

Random forest prediction

Given the elemental makeup of a compound, using RFs to directly predict $\frac{c}{a}$ was found to be more effective when applied to the PbFCl structure-type compounds (dataset 3) than ICSD (dataset 1) (Figure 3.2 and Figure 3.4a respectively), with a r^2 of 0.79 compared to 0.75. This could be explained by the more limited range of chemistries in the PbFCl dataset. It is notable that while the prediction of *a, b,* and *c* separately often resulted in a similar r^2 and dividing the predicted c by the predicted a always resulted in a worse performance than directly predicting $\frac{c}{a}$.

Also of note is that the prediction of unit cell volume from models trained with PbFCl structure-type data to predict volume and on models trained to predict all unit cell parameters (a, b, c, α, β, γ, V) both perform similarly to results found in previous work (which also found a mean average error of 3.9%) [28]. However, this success was not found with



(a)

(b)

Figure 3.2: (a) Predicted $\frac{c}{a}$ against true $\frac{c}{a}$ using the model described in 3.1.4 on PbFCl structure-type dataset (b) (lack of) correlation between Shannon radii and $\frac{c}{a}$ as described in Section 3.1.4

the ICSD data, where the models exhibited a mean error of 17% in predicting the volume of units of cells. Previous work used data from the Materials Project [78] (for training) and the score obtained is based on a restricted selection of just 309 structures from the ICSD dataset. While performance on ICSD based data could be attributed a better machine learning and featurisation method, it could equally be attributed to the use of a restricted dataset with consistent chemistries. The full results found using this method are also listed (Table 3.1).

Random forest prediction (trained on parent and child compounds)

Substitutional investigations, which take a known (parent) substance and wish to find the properties of a variant (child), are common. One model which was investigated for such a situation takes as input the elemental composition of the parent and child substance as well as parental unit cell parameters. A RF was trained to predict $\frac{c}{a}$ with an r^2 of 0.95 (Figure 3.4b) and predicts the change in $\frac{c}{a}$ with an r^2 of 0.97 (this will be referred to as the comparative r^2 or r^2_{comp}). The accuracy in predicting whether the child would have a larger or smaller $\frac{c}{a}$ than the parent (ordinal accuracy) was 0.97. This model was found to be more effective in ordinal accuracy than a guess regardless of the size of the difference in $\frac{c}{a}$ between the parent and the child; however, r^2_{comp} was only found to display a trend when child $\frac{c}{a}$ was more than 7% different from the parental $\frac{c}{a}$ (Figure 3.3). It should be

Table 3.1: Results of random forest regression (described in Section 3.1.4: Random forest prediction). Models are trained on certain unit cell parameters but are tested for prediction of $\frac{c}{a}$

metric	train target	ICSD		PbFCl	
		80/20 split	LOCO-CV	80/20 split	LOCO-CV
r^2	$\frac{c}{a}$	0.75	−4.9	0.79	−2.5
	a, b, c	0.75	−7.5	0.76	−1.4
	a, b, c, V	0.76	−12	0.78	−1.3
	a, b, c, α, β, γ	0.76	−11	0.77	−1.4
	a, b, c, α, β, γ, V	0.76	−11	0.79	−1.3
MRE	$\frac{c}{a}$	0.077	0.22	0.022	0.078
	a, b, c	0.074	0.21	0.024	0.073
	a, b, c, V	0.076	0.24	0.024	0.063
	a, b, c, α, β, γ	0.075	0.21	0.024	0.074
	a, b, c, α, β, γ, V	0.076	0.25	0.024	0.065

noted that this model was able to predict the volume of the child compound at an average error of 7.2% on the ICSD dataset and 3.7% on the PbFCl structure-type dataset. The full results found using this method are available (Table 3.2)

Comparing predicted unit cell values for a child to known unit cell values for a parent

The RF discussed in 3.1.4 was just trained with the child compound but could still be used for substitutional study by subtracting the model prediction for the parameters of the child unit cell from the known parameters of the unit cell for the parent. Application of the model in this way led to a good accuracy of 0.92 and an r^2_{comp} of 0.91 (Figure 3.4d). When the application of the model was changed, more knowledge was extracted. This would be useful for a chemist doing a substitutional investigation, while adding almost no increase in model complexity and no increase in the required training data and required training time. The complete results found using this method are available (Table 3.3)

Table 3.2: Results from prediction of $\frac{c}{a}$ using model trained on both parent and child compounds as described in Section 3.1.4: Random forest prediction (trained on parent and child compounds). Models are trained on certain unit cell parameters but sometimes tested on other unit cell parameters in order to evaluate different prediction strategies).

metric	train target	ICSD		PbFCl	
		80/20 split	LOCO-CV	80/20 split	LOCO-CV
r^2	$\frac{c}{a}$	0.95	0.52	0.75	−2.3
	a, b, c	0.95	0.44	0.74	−0.94
	a, b, c, V	0.94	0.21	0.78	−0.68
	a, b, c, α, β, γ	0.95	0.36	0.75	−0.95
	a, b, c, α, β, γ, V	0.94	0.23	0.77	−0.68
r^2_{comp}	$\frac{c}{a}$	0.97	0.88	0.87	0.21
	a, b, c	0.97	0.88	0.87	0.19
	a, b, c, V	0.97	0.89	0.89	0.24
	a, b, c, α, β, γ	0.98	0.88	0.87	0.16
	a, b, c, α, β, γ, V	0.97	0.88	0.89	0.23
MRE	$\frac{c}{a}$	0.029	0.045	0.021	0.083
	a, b, c	0.028	0.042	0.023	0.084
	a, b, c, V	0.032	0.044	0.021	0.077
	a, b, c, α, β, γ	0.029	0.043	0.023	0.08
	a, b, c, α, β, γ, V	0.031	0.045	0.022	0.075
Ordinal accuracy	$\frac{c}{a}$	0.97	0.94	0.95	0.75
	a, b, c	0.97	0.94	0.94	0.73
	a, b, c, V	0.96	0.94	0.94	0.73
	a, b, c, α, β, γ	0.97	0.95	0.94	0.73
	a, b, c, α, β, γ, V	0.97	0.94	0.94	0.73

Comparing predicted unit cell values for a child to predicted unit cell values for a parent

Training a RF to directly predict $\frac{c}{a}$ from a compound's elemental makeup as in 3.1.4, and then testing the model by predicting $\frac{c}{a}$ for both a parent and child compound and comparing the two yielded impressive results with r^2_{comp} of 0.79 on the PbFCl structure-type dataset and 0.91 on the ICSD dataset. Comparisons here were between the predictions to investigate whether a RF's inaccuracies could be taken advantage of. If the model was wrong in similar ways for the parent and child, by comparing a model's predictions for

Table 3.3: Results from comparison of prediction of $\frac{c}{a}$ of child to the true value of $\frac{c}{a}$ of parent as described in Section 3.1.4: Comparing predicted unit cell values for a child to known unit cell values for a parent

metric	train target	ICSD		PbFCl	
		80/20 split	LOCO-CV	80/20 split	LOCO-CV
r^2_{comp}	$\frac{c}{a}$	0.91	−2	0.79	−17
	a, b, c	0.91	−2.5	0.77	−12
	a, b, c, V	0.9	−4.8	0.79	−15
	a, b, c, α, β, γ	0.91	−3.1	0.76	−15
	a, b, c, α, β, γ, V	0.9	−3.9	0.8	−13
ordinal accuracy	$\frac{c}{a}$	0.92	0.71	0.9	0.61
	a, b, c	0.93	0.7	0.89	0.62
	a, b, c, V	0.93	0.72	0.89	0.61
	a, b, c, α, β, γ	0.93	0.71	0.89	0.62
	a, b, c, α, β, γ, V	0.92	0.72	0.89	0.62

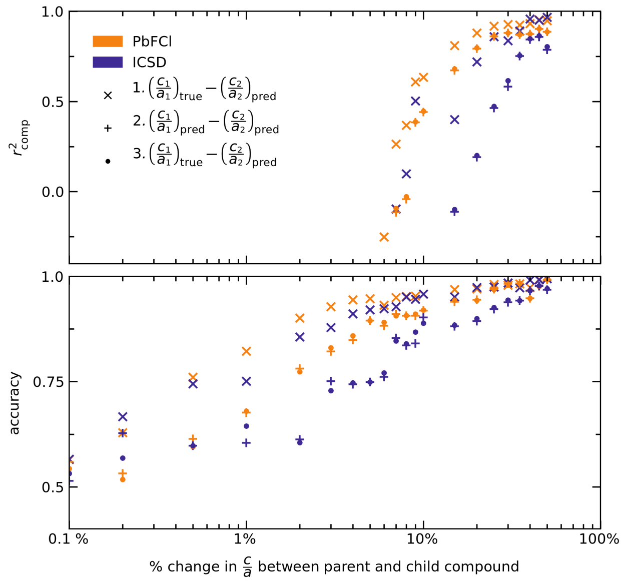


Figure 3.3: Performance of comparative models vs the distance between the compounds being compared, where model 1 is described in 3.1.4 model 2 in 3.1.4 and model 3 in 3.1.4

Table 3.4: Results from comparisons of predictions for $\frac{c}{a}$ of parent and child compound as described in Section 3.1.4: Comparing predicted unit cell values for a child to predicted unit cell values for a parent

metric	train target	ICSD		PbFCl	
		80/20 split	LOCO-CV	80/20 split	LOCO-CV
r^2_{comp}	$\frac{c}{a}$	0.91	-1.9	0.79	-17
	a, b, c	0.9	-2.5	0.76	-16
	a, b, c, V	0.9	-4.4	0.8	-16
	a, b, c, α, β, γ	0.91	-3.2	0.76	-15
	a, b, c, α, β, γ, V	0.9	-4.5	0.79	-14
Ordinal accuracy	$\frac{c}{a}$	0.92	0.71	0.90	0.62
	a, b, c	0.92	0.70	0.89	0.62
	a, b, c, V	0.92	0.72	0.89	0.61
	a, b, c, α, β, γ	0.93	0.72	0.89	0.61
	a, b, c, α, β, γ, V	0.93	0.71	0.89	0.62

Table 3.5: Results from weighted average Shannon radius benchmark described in Section 3.1.4: Using Shannon radius as a benchmark.

metric	ICSD		PbFCl	
	80/20 split	LOCO-CV	80/20 split	LOCO-CV
r^2	-1.1	-53	-13	-26
r^2_{comp}	-0.078	-1.1	-5.9	-8.9
ordinal accuracy	0.58	0.6	0.53	0.54

$\frac{c}{a}$ of both parent and child could these inaccuracies cancel each other out if the parent and child occupy a similar part of the data space?

However, this is not consistently the case, with almost identical performance between these two models regardless of the distance between the parent and child compounds (Figure 3.3). It is interesting to see that the introduction of more use of machine learning into the method does not seem to increase the uncertainty (Figure 3.4e). It could be argued that any uncertainty introduced in this method compared to that described in 3.1.4 is offset by taking advantage of similar model inaccuracies across parent and child unit cell parameter predictions, however, with regard to this hypothesis these experiments are inconclusive. The complete results for this model can be found in Table 3.4.

Using Shannon radius as a benchmark

To demonstrate the models to predict $\frac{c}{a}$ developed here are effective, RFs were compared against a standard method that might be employed by a chemist (as described Section 3.1.2). In particular, Shannon radii are commonly employed to determine the change in unit cell volume when substituting an ionic species by another. If a larger species is inserted, the unit cell generally expands. It is, of course, less clear whether there would be a general rule for the influence on the $\frac{c}{a}$ ratio. Thus, it is interesting to observe that there is no correlation between the Shannon radius and $\frac{c}{a}$ (Table 3.5 and fig. 3.2b). Although some accuracy was found when using the change in the average Shannon radius to investigate the change in $\frac{c}{a}$ between the parent and child compound, this is far outperformed by all other models discussed here.

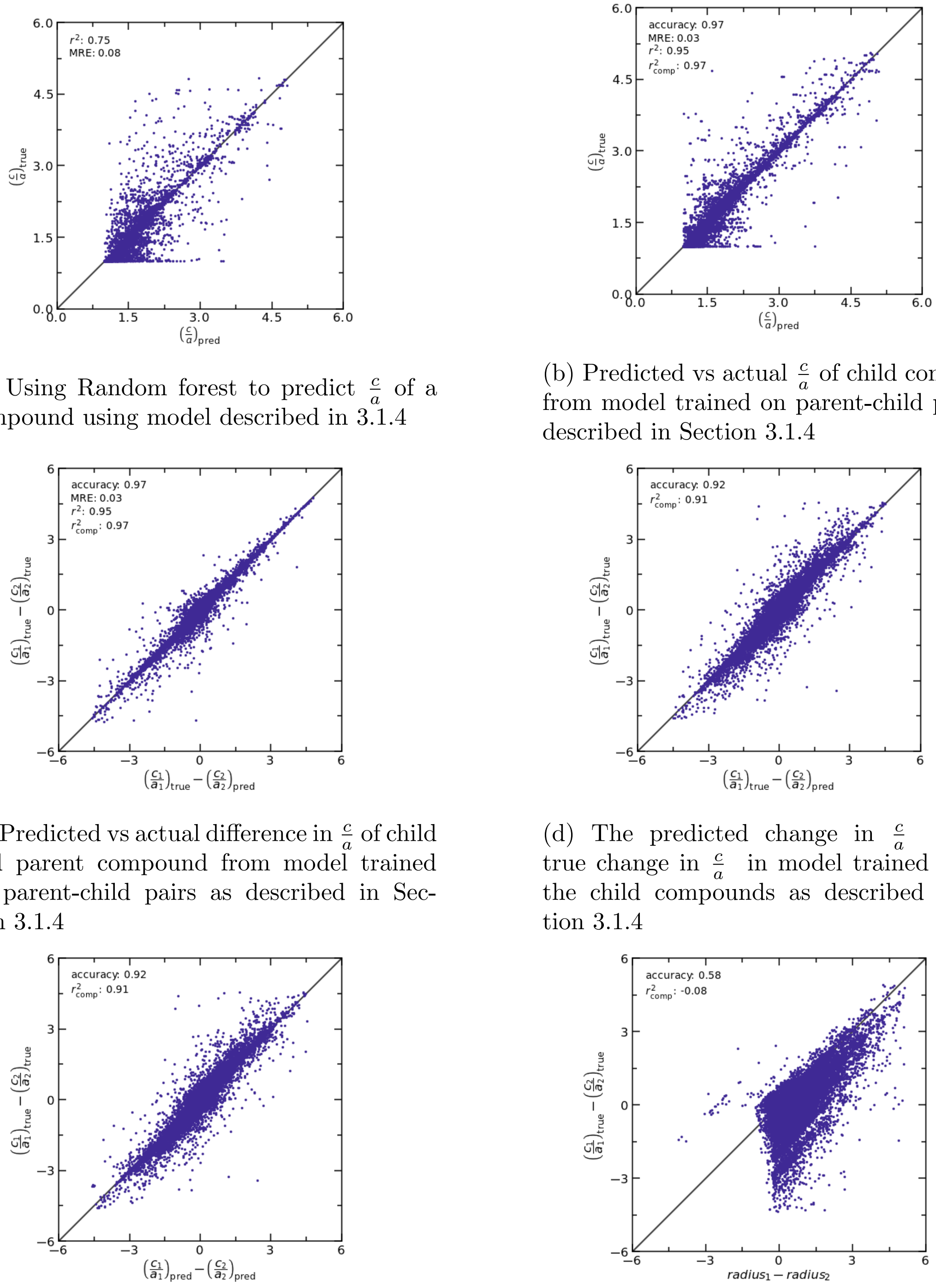
To obtain values for the radius, the 6-coordinate Shannon radius was used for all elements. Oxidation states were manually chosen to be chemically reasonable; the oxidation states of transition metals were chosen based on the most common corresponding metal-oxides. When high and low spin radii are present, the average is taken.

3.1.5 Discussion

Predicting $\frac{c}{a}$ of a material from the compound's formula was found to be effective using RFs (the best r^2 was 0.78 on PbFCl data, Figure 3.2; or 0.69 on ICSD data). This demonstrates the ability of RFs to work effectively on small datasets, where larger datasets are not available (*i.e.*, datasets of hundreds, rather than hundreds of thousands of materials).

However, most functional properties are tuned through substitution studies, where the researcher wishes to alter a known (parent) compound to find a better performing child compound. A more effective application of RFs was developed to predict $\frac{c}{a}$ of an arbitrary child composition relative to the $\frac{c}{a}$ of a known parent structure. Although the contributing factors that determine $\frac{c}{a}$ cannot be easily enumerated, the ability of ML models presented here to predict them show that these factors are present in the training data.

The most effective model to predict the change in the $\frac{c}{a}$ ratio between the parent compound and the child compound used the paired prediction schema (Section 3.1.4). This model was trained using 80,000 randomly selected parent-child pairs of compounds in the ICSD. As input the RF received compositions of the parent and child structures, as well as the lattice parameters for the parent compound. When evaluating such a model, one could examine the relationship between the predicted child $\frac{c}{a}$ $(\frac{c_2}{a_2})_{\text{pred}}$ and the true



(a) Using Random forest to predict $\frac{c}{a}$ of a compound using model described in 3.1.4

(b) Predicted vs actual $\frac{c}{a}$ of child compound from model trained on parent-child pairs as described in Section 3.1.4

(c) Predicted vs actual difference in $\frac{c}{a}$ of child and parent compound from model trained on parent-child pairs as described in Section 3.1.4

(d) The predicted change in $\frac{c}{a}$ against true change in $\frac{c}{a}$ in model trained on just the child compounds as described in Section 3.1.4

(e) Change in predictions of $\frac{c}{a}$ of parent and child compounds against true change in $\frac{c}{a}$ in model trained on just the child compounds as described in Section 3.1.4

(f) Difference in weighted average Shannon radius in parent and child vs difference in $\frac{c}{a}$ in child and parent compound as described in Section 3.1.4

Figure 3.4: Results of different models on ICSD dataset

child $\frac{c}{a}$ $(c_2/a_2)_{\text{true}}$; this leads to $r^2 = 0.95$.

Alternatively, one could examine the success of the model in making predictions for the child $\frac{c}{a}$ (c_2/a_2) compared to the parent compound (c_1/a_1), which is described as r^2_{comp}; this leads to $r^2 comp = 0.97$ (Figure3a). This model correctly predicted the direction of change in $\frac{c}{a}$ 97% of the time and is referred to as the accuracy, or ordinal accuracy. Also reported is the mean error of these models' $\frac{c}{a}$ relative to the size of the $\frac{c}{a}$ predicted (MRE); here, MRE = 0.03. Using just the PbFCl structure type and related families to train such a model (a training set size of only 878), resulted in $r^2 = 0.87$ and an accuracy of 0.95.

While substantial datasets such as the ICSD improve performance if they are available, changing the application of ML methods (for example through comparative predictions) can obtain comparable results. These methods are seen to work on smaller, more specialised datasets where chemistries and/or structures are more similar. The accuracy of the prediction of the direction of change in a substituted compound was found to improve with the size of the change in $\frac{c}{a}$ (Figure 3.3). For very small changes, the ordinal accuracy of the model is better than a guess (and also better than trying to use a naive model using the weighted average of the Shannon ionic radii, Figure 3.2b), and this improves with larger changes in $\frac{c}{a}$.

However, r^2 of the model shows no correlation for small changes, but improves dramatically after the difference in the values of $\frac{c}{a}$ is greater than 7% (Figure 3.3). Although this ordinal accuracy is always better than a guess, for small changes in $\frac{c}{a}$, the accuracy is not sufficient to be assured of correct predictions, particularly where small numbers of chemical reactions are involved such as in this case. This could be partially attributed to dataset noise; the ICSD contains materials synthesised and measured using different protocols and instruments. Changes of less than 1% of $\frac{c}{a}$ could be attributed to instrument error, making these data of insufficient precision to be helpful in many circumstances. Nevertheless, RFs were shown to be viable tools to guide synthetic experiments and demonstrate reasonable performance, producing a $\frac{c}{a}$ ratio when given an arbitrary composition without the need for researchers to enumerate contributing factors. In practise, an investigator will have a parent compound in mind and is looking for chemical substitution to produce a child compound. Accordingly, a tool was developed with input data well matched to these circumstances (taking parent and child compositions and parental lattice parameters); this model also performs the best of all models tested.

These conclusions are drawn for the models trained and tested with an 80/20 train / test split, and ignored measurements made with LOCO-CV, which were universally worse.

Comparative models performed better in LOCO-CV than a guess (Tables 3.2 to 3.4), which indicates that these models are able to extrapolate to domains different from those in which they are trained. However, examination of the clusters generated for LOCO-CV revealed that they vary a lot in size (by at least two orders of magnitude each), making it unclear whether extrapolation performance is being measured, or the effect of differing train/test set sizes. These differences in cluster sizes are an indication that the domains being studied here are not linearly separable through Voronoi tessellation. Further, this shows that use of LOCO-CV is not always appropriate and must be examined critically when being used to evaluate the effectiveness of a model. Exploration of this effect and proposals on how to fix it will be further explored (Chapter 5).

3.1.6 Conclusion

RFs clearly outperform average weighted Shannon radii as predictors of $\frac{c}{a}$ (with r^2 of 0.75 and -1.06 respectively). However, this improvement is overshadowed by the effectiveness of comparative models (Figures 3.4d to 3.4f) especially comparative models that train on both parental and child composition (where the parent compound is a known compound and the child is a variant on the parent to be investigated).

Comparative models which only train on child compounds provide more useful metrics than their noncomparative counterparts, while adding no additional computational complexity. Comparing the predicted child $\frac{c}{a}$ to the true parent $\frac{c}{a}$ results in a similar performance to comparing predicted child $\frac{c}{a}$ to predicted parent $\frac{c}{a}$ however the latter performs better when child and parent $\frac{c}{a}$ are very similar.

Although comparative models performed better on the larger, more general ICSD dataset, simpler models performed better on the smaller, domain specific, PbFCl structure-type dataset. The $\frac{c}{a}$ of child compounds were easier to predict as the distance from the parent compound increased; however, even with very similar child and parent compounds, the models still outperform a random guess as to which $\frac{c}{a}$ is larger; parent or child. These results show the clear applicability of these models in guiding chemists to make informed decisions as to which compounds to test experimentally to obtain the desired $\frac{c}{a}$ ratios. As well as the aptitude of RFs to predict unit cell parameters just using the elemental composition of a compound.

The above has evaluated the usefulness of comparative predictions in the context of predicting lattice parameters. However, the usefulness of comparative predictions depends on how well this method would generalise to other algorithms, datasets or material properties.

RFs by their nature have no mechanism of comparing features within a single feature vector. As such it is expected that substituting RFs for a different algorithm which can do this (such as neural networks or linear regression) would, in theory, be better able to take advantage of comparative predictions.

It is not clear that there is any specific feature of lattice parameter prediction which would lead to this method performing particularly well for this task. Thus, comparative predictions can be suggested for any property prediction task concerning substitutional chemical studies. However, as this method has thus far only been used in this context, further research would be needed in other contexts to confirm the efficacy of this technique for that task.

Similarly there is no reason why comparative prediction would not generalise to other datasets, but efficacy of this technique should be checked in new contexts due to the limited scope of this study. Since completion of this work, the elemental movers distance (ElMD) has been suggested as a robust distance measurement between compositions [66]. Future work using the comparative prediction technique may find it useful, and informative to investigate how the ElMD between parent and child compositions effects the efficacy of comparative predictions.

In short there is no reason why comparative predictions should not generalise to other tasks concerning substitutional chemical studies. Due to the limitted scope of this study, it would be recommended to investigate (rather than assume) efficacy in any new contexts. However, the good performance of this method in predicting lattice parameters is promising.

3.2 Predicting the pore limiting diameter of metal organic frameworks

Note: This section is adapted from results published in Angewandte Chemie [144]. No credit is claimed for the conceptualising, or downloading of the dataset or featurising of metal species.

Metal-Organic Frameworks (MOFs) are a class of porous materials consisting of metal ions linked by organic ligands (often called linkers). MOFs offer a large surface area and have properties highly dependant on the size of their pores. As such, the ability to predict porosity is critical for designing MOFs with optimal gas storage, separation, and catalytic properties. ML offers a fast and scalable method of screening potential MOFs for desired properties.

Use of ML for prediction of MOF properties has been recently reviewed [77]. Many works [136, 50, 120] used feature vectors engineered for the description of MOFs (some of these features, such as pore sizes, require a priori knowledge of the MOF structure) to build a series of machine learning (ML) models for the prediction of CO_2 and CH_4 adsorption. At the earliest stage of MOF synthesis only the chemical identities of the organic linker and metal species are known, as such feature vectors should ideally avoid inclusion of information that can not be readily calculated from either the metal or linker.

Existing ML are trained either from databases of hypothetical MOF structures [199, 17, 18, 59] or from the Computation-Ready, Experimental (CoRE) MOF database of reported structures [29]. However, databases of, or based on, existing structures only cover a limited part of the potential design space [77] and new combinations of metal species and organic linkers will lead to new MOF structures that arise from their coupled chemistries. As such an experimental dataset of Metal-Linker combinations was derived from the Cambridge Structural Dataset (CSD) [119]. The derived dataset, focused on three-dimensionally connected MOF structures and associated one metal and one linker to a calculated pore limiting diameter (PLD), was dubbed 1M1L3D [144].

PLD can be calculated from a known MOF structure, and deterimines the size of the molecules which can be absorbed within a MOF structure (thus is helpful in designing MOFs for specific applications). While chemists may have target structures in mind during synthesis, whether that structure will be formed, or can be formed, as opposed to another structure with the same metal-linker combination is unknown a priori to synthesis. Therefore, prediction of the potential PLD from a given metal-linker combination is a good candidate task demonstrating the capabilities of ML in conjunction with the new 1M1L3D dataset.

This section describes the creation of a RF based ML model which trained on the 1M1L3D dataset which classifies a MOF structure's porosity into one of four categories with 80.5% accuracy. Specifically, the contributions of this section can be described as follows:

- Exploration of solutions to problems with duplicated and conflicting data points which can arise when using ML for materials property prediction, particularly, but not exclusively in the organic synthesis domain (demonstrated in Section 3.2.1 and further discussed in Section 3.2.4).

- Exploration of use of sequential application of RFs for a well performing yet simple to implement model (Section 3.2.2).
- Discussion of the advantages and disadvantages of simplifying tasks to classification tasks (Section 3.2.4).
- Discussion of the benefits and drawbacks of models which aim to demonstrate datasets (Section 3.2.4).

3.2.1 Data preparation

Featurisation

Much like magneto-caloric materials explored previously (Section 3.1), MOF's unique properties are defined by a combination of their structure, and chemical make up. However, MOFs, like other organic materials contain carbon atoms which can be arranged in chains of varying topologies. Organic linkers can be represented as diagrams, or SMILES strings (further discussed in Section 2.2.1), as is the case in the 1M1L3D dataset. While a ratio of a MOF's constituent elements could be extracted from this dataset, (similar to the ratio used with magneto-caloric materials), the variation in linker topologies means that such an approach is inappropriate to train ML models from, as it would discard valuable structural data present in the linker.

However, as a baseline it is still interesting to consider only chemical (rather than structural) information, so a fixed length vector was constructed to describe the chemical composition of entries in the 1M1L3D dataset. All chemical elements encountered throughout the entire dataset are accounted for in each entry: either by their number of atoms in each linker or by a 0 if the element is not present. Similarly, for the metal, the metal encountered in the given entry is marked as 1 whereas all other metals are marked as 0. The total length of this vector encoding composition is 70, which corresponds to the total number of distinct chemical elements encountered throughout the 1M1L3D dataset (17 distinct chemical elements for the linkers and 53 distinct metals).

While the ratio of elements in a MOF was explored as a baseline, further exploration of featurisation was needed. Six feature were previously selected to represent the metal species:

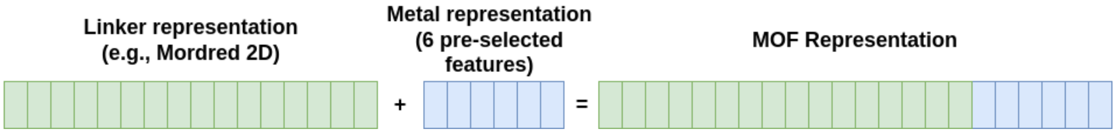


Figure 3.5: Creation of vector representations of MOFs

- Atomic number
- Atomic weight
- Atomic radius
- Polarizability
- Electron affinity
- Mulliken's electronegativity

This feature vector for the metal was then concatenated with several representations for a SMILES string (Figure 3.5). While more recent innovations have trained ML algorithms using SMILES strings as inputs, SMILES strings are of variable length and thus are not suited to as input to many algorithms. As this investigation was a preliminary demonstration of the capabilities of the 1M1L3D dataset, only algorithms that train from a fixed sized input were considered due to their ease of use. Therefore, several representations for SMILES strings, from two different software packages were considered:

- Molecular descriptors from Dragon6 using SMILES codes (Dragon 2D): Dragon6 [115] is a commercial package that can calculate molecular descriptors relying on the 2D molecular structures of the linkers. Using the SMILES codes provided as input, Dragon6 returned 2,098 descriptors.
- Molecular descriptors from Dragon6 using 3D molecular conformations (Dragon 2D & 3D): Dragon6 can also calculate molecular descriptors relying on the 3D molecular structures of the linkers. To calculate 3D descriptors, linkers' 3D molecular conformations were provided as input, as well as information of atomic partial charges for the linkers, which were assigned using Open Babel [129]. Dragon6 returned 3,582 descriptors using 3D structures.

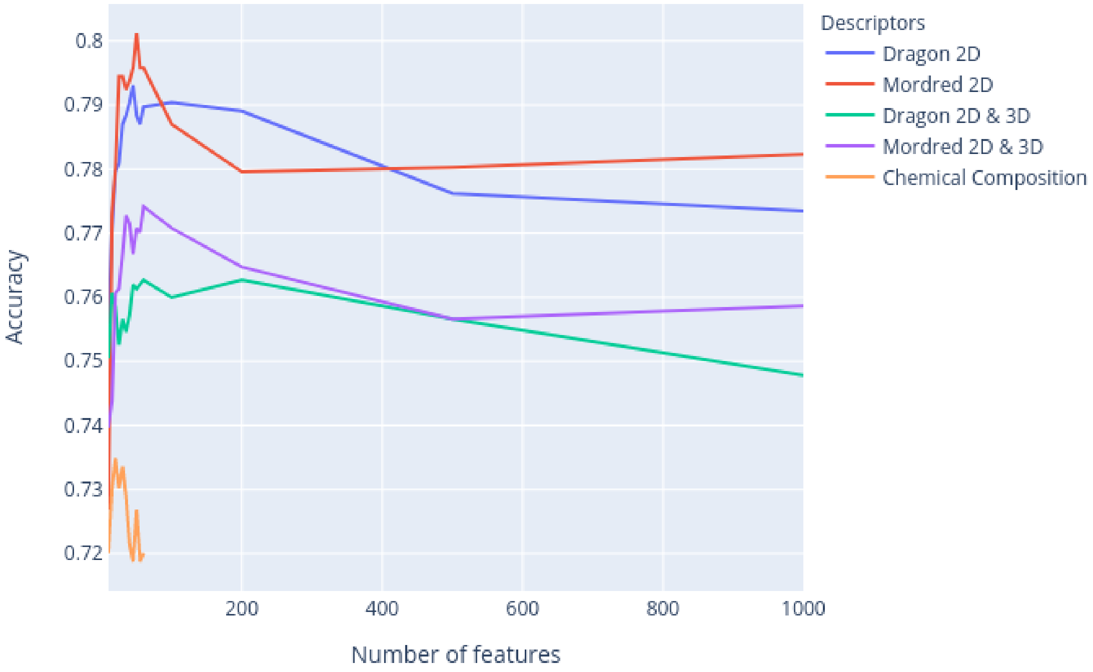


Figure 3.6: Performance of random forests in predicting MOF porosity using different featurisation libraries

- Molecular descriptors from Mordred using SMILES codes (Mordred 2D): Mordred [121] is a freely available package that can calculate molecular descriptors relying on the 2D molecular structures of the linkers. Using the SMILES codes provided as input, Mordred returned 1,613 descriptors.

- Molecular descriptors from Mordred using 3D molecular conformations (Mordred 2D & 3D): Mordred can also calculate molecular descriptors relying on the 3D molecular structures of the linkers. To calculate 3D descriptors, the linkers' 3D molecular conformations were provided as input. Mordred returned 1,826 descriptors using 3D structures.

While more descriptors could have been generated by both software packages, results shown here filter out low-variance descriptors (excluding descriptors with a variance across the dataset of 0.0001 or lower).

Conflicting data entries in 1L1M3D

A common problem in ML for predicting properties of inorganic materials is that just because two materials are constituted of the same elements, it does not mean they will have remotely similar properties. One of the more prominent examples is the difference between diamond and graphite, both of which are carbon. Similarly, just because a MOF is made of the same metal and linker combination does not imply that they will have the same PLD. As such, it is important to anticipate this to prevent potential data leakage when training ML models.

The 1L1M3D dataset contains 14,296 metal linker combinations, but only 7,391 unique metal-linker combinations. Often these duplicated combinations have vastly different porosities: in it's most extreme, there may be 179 different instances of a metal linker combination with PLD ranging from 0.91 Å to 4.09 Å (Figure 3.7b). Three possible courses of action were considered to account for repeated combinations:

- Removing any metal-linker combinations which occur more than once (leaving only 5,258 data points).
- Selecting one entry from repeated metal-linker combinations (such as the first entry) and removing any recurrences of that combination.
- Aggregating repeated metal-linker combinations using, for example, the mean or the median

All of these courses of actions have advantages and disadvantages. Removing any metal-linker combinations which occur more than once not only reduces the dataset size, but also induces a bias in the dataset. Metal-linker combinations which only occur once are likely to be combinations that have been studied less, combinations for which successful synthesis is hardest (and thus combinations which have been reported less), or combinations for which only one possible MOF structure can result.

Selecting one entry from repeated metal-linker combinations allows for all 7,391 unique metal-linker combinations to be investigated. However, the selection of which result to keep is likely to be arbitrary as the dataset is too large to manually select the most appropriate PLD for each repeated metal-linker combination. There is also no guarantee that a "most appropriate" PLD exists, and this selection process induces the biases of the researcher who is doing such manual investigation.

Aggregating repeated metal-linker combinations using the mean, or median systematic, and can be done without manual interrogation of the dataset. However, the nature of these aggregation functions means that reverse lookup of entries can become impossible. This means that if a researcher wanted to look up the publication associated with a metal-linker to PLD combination, such a publication may not exist.

For the purposes of this investigation, the median PLD for a metal-linker combination was used. As this model was intended to be a demonstration of the dataset, reverse look-up of publications was seen as a less important factor. Median was selected over the mean PLD to ignore outliers. Following the aggregation by median PLD the dataset could be split into a training set (of 5912 MOFs) and a test set (of 1479 MOFs) , without fear of data leakage.

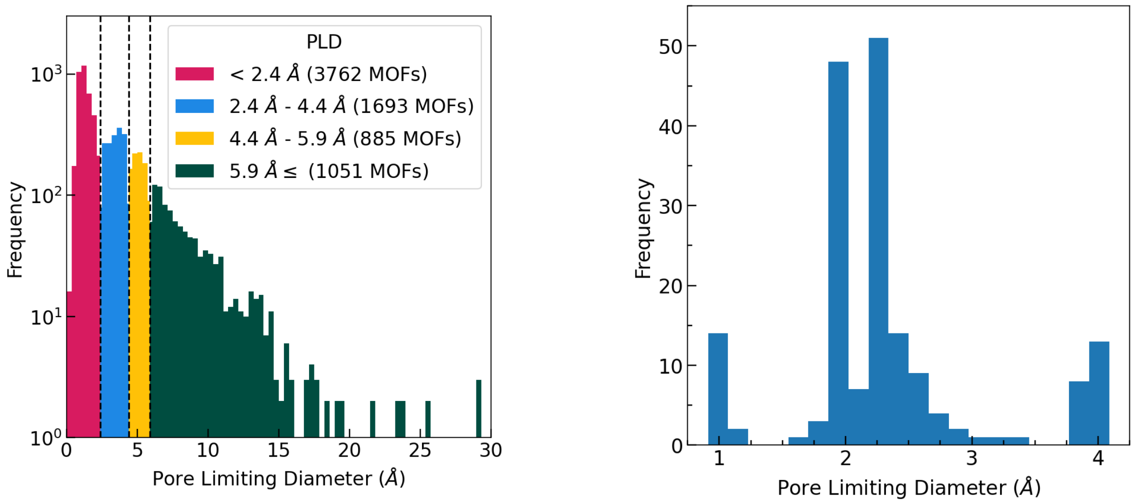
Categorising porosity

As a continuous value, porosity could be predicted using a regression model. However, as discussed (Section 3.1) and will be noted as a theme for this chapter (Section 3.3) classification is a simpler task, with a more limited range of outputs. As such many models (and RFs in particular) will show improved performance in a classification task as opposed to a regression task in the same domain. However, while training of a regression model to predict porosity will be explored to demonstrate this point (Section 3.2.2), considering prediction of porosity as a classification task will be the primary concern of this section.

In order to simplify this task from a regression task to a classification task, PLD must be categorised. Following convention [29], MOF structures with a PLD larger than 2.4Å(approximately the van der Waals diameter of H_2) were considered porous. This split the dataset approximately in half (3629 porous and 3762 non-porous). In order to further discriminate between porous materials further categories were created, splitting the at the 4.4 Åand 5.9 Å, creating four categories (Figure 3.7a).

3.2.2 Model creation

To justify RFs as the choice in classification algorithm, several models were tested in their ability to predict where of MOF was porous (had a PLD larger than 2.4 Å). To do this, the Mordred 2D descriptors were used. Algorithms were implemented with Scikit-learn [140]



(a)

(b)

Figure 3.7: (a) A histogram of the pore limiting diameters (PLDs) of MOFs in the 1M1L3D dataset (b) A histogram the PLDs of the modal MOF in the 1M1L3D dataset, which has ligand of OC=O linking magnesium atoms.

with diameters set to default unless otherwise specified, and their performance was compared via a 3-repeated stratified 10-fold cross validation procedure. As the performance of each algorithm depends on the number of features used to train the models, each of the classification algorithms was tested with a range of different numbers of features selected using the SelectKBest procedure [140].

The following classification algorithms were tested:

- logistic regression (LR): with a 'liblinear' optimization algorithm (solver='liblinear') to perform (multi_class='ovr').
- linear discriminant analysis (LDA): with default parameters.
- k-nearest neighbors (KNN): with default parameters.
- decision tree classifier (CART): with default parameters.
- Gaussian naive Bayes (NB): with default parameters.
- Support Vector Machine (SVM): with default parameters.

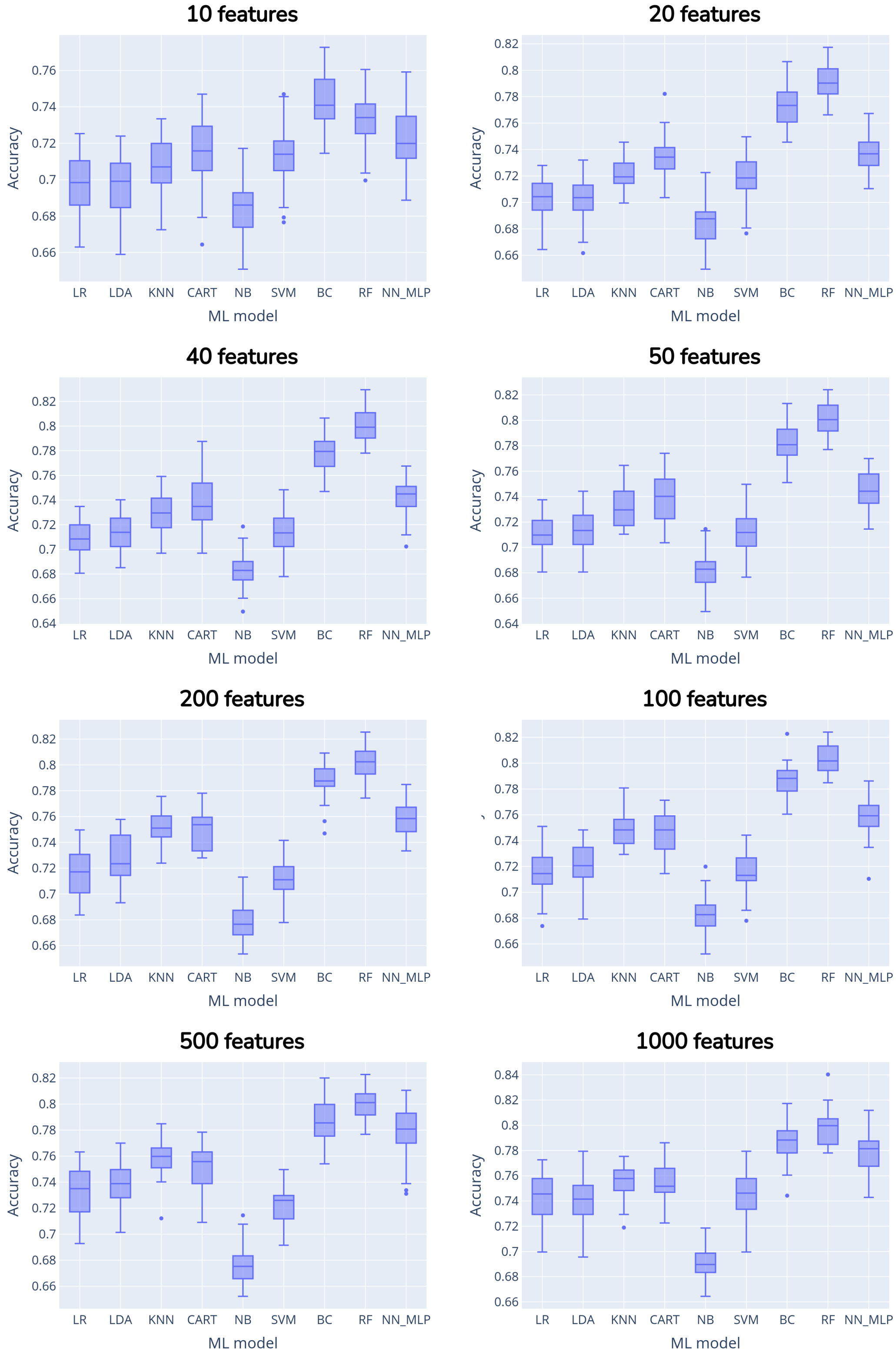


Figure 3.8: Performance of MOF porosity classifiers with different numbers of features. Random forests consistantly performed best using this feature set.

- bagging classifier (BC): with default parameters.
- random forest classifier (RF): with default parameters.
- multi-layer perceptron classifier (NN_MLP): with a maximum iteration number increased to 500 (max_iter=500). By default, this is configured to have one hidden layer of size 100, and ReLU activation functions, optimised by the Adam gradient descent algorithm.

The accuracy of each algorithm was used to assess its performance. RFs consistently yielded the best accuracy across the entire range of features tested, from 20 to 1000 (Figure 3.8). The numerous reasons could be suggest for RFs' robust performance. One may be that RFs' bagging and boosting make it resilient to outliers, this can be evidenced that the second best performing algorithm (BCs) was also an ensemble that uses bagging. Examining the distribution of data points suggests there are some outliers (Figure 3.7a). Another explaination is taht RF's perform well with little data. There are 5912 training data points, which while a lot for the materials science domain is still relatively little for ML more generally. In particular, the NN_MLPs tested had a hidden layer of 100 parameters, it may be that more data was required to addeqauatly tune these networks.. Regardless of reason, RF's good performance, resulted in the algorithm being chosen to train the models that were tested in subsequent analyses.

Having selected RFs as the appropriate model, featurisation methods outlined in Section 3.2.1 were investigated. Mordred 2D descriptors were found to be the most effective, though the differences between all the featurisation methods tested were small (Figure 3.6). For this reason, 50 Mordred 2D descriptors (chosen using the SelectKBest algorithm) were used in model creation.

Once the ML algorithm and the set of features were chosen, the hyperparameters of the RF algorithm were investigated using a grid search. The RF algorithm performs well with the default settings, and the changes of the hyperparameters could produce only a slight increase in performance. Therefore, the only change to the default hyperparameter was to increase the number of trees in the forest. Thus RFs discussed in this section use:

- 2000 trees in the forest (chosen to balance performance with time complexity)
- at least 2 samples to split a node
- at least 1 sample per leaf

- a minimum weighted fraction per leaf of 0.0
- a maximum number of features considered per node split of log2
- an unlimited number of leaves per node

As discussed (Section 3.2.1) gaining further insights to MOF porosity can be advantagous. A sequential application of RFs was implemented, featurised inputs were sequentially passed through three RFs. Each RF is trained as in binary classification of the porosity of MOFs as above or below an increasing threshold of PLD. The first RF (referred to as model 1) classifies the porosity of a MOF as being porous or non-porous (PLD of greater or less than 2.4 Å). The second RF (model 2) classifies porosity as having large or small pores (PLD greater or less than 4.4 Å given porosity $>$ 2.4 Å. The last model (model 3) classifies the porosity pores as being large or very large (PLD greater or less than 5.9 Å given porosity $>$ 4.4 Å). In combination, these RFs assign a MOF porosity as one of four categories (porosity $<$ 2.4 Å, porosity $<$ 4.4 Å, porosity $<$ 5.9 Å, and porosity $\geq$ 5.9 Å).

To establish the effectiveness of this sequential application of RFs, two further RFs were trained:

- A multi-class classifier, categorising input into one of the 4 outlined categories.
- A regression model, whose output was then assigned into one of the four categories (use of RFs as regressors is discussed further in Section 2.1.7).

3.2.3 Results

Using an 80/20 train/test split, model 1 successfully classified MOFs in being porous on non-porous 80.5% of the time, with models 2 and 3 being accurate 76.3% and 68.5% of the time respectively (Table 3.6). These decrease in performance between models 1 and 3 can be explained by the smaller training sets, as models 2 approximately half the number of MOFs as model 1, and model 3 was trained with approximately a quarter of the MOFs (Figure 3.7a). While binary confusion matrices can be presented for each model individually (Figure 3.9), these can be combined into a multiclass matrix (Figure 3.10). Similarly, further metrics for the performance of this model are also reported (Tables 3.6 and 3.7).

Table 3.6: Summary of performance metrics for the task of Models 1-3 using The validation dataset was used to measure performance at predicting whether whether a MOF's porelimiting diameter (PLD) was greater or less than a certain threshold. For the 4.4Å and 5.9Å thresholds, the MOFs with PLDs less than 2.4Å and 4.4Å respectively were removed from the test dataset. This allows for comparison of three different approaches: Sequential (Seq.) (*i.e., Models 1,2 and 3 separately*), multi-class (M.C.) and regression (Reg.) approaches described in the text (Section 3.2.2).

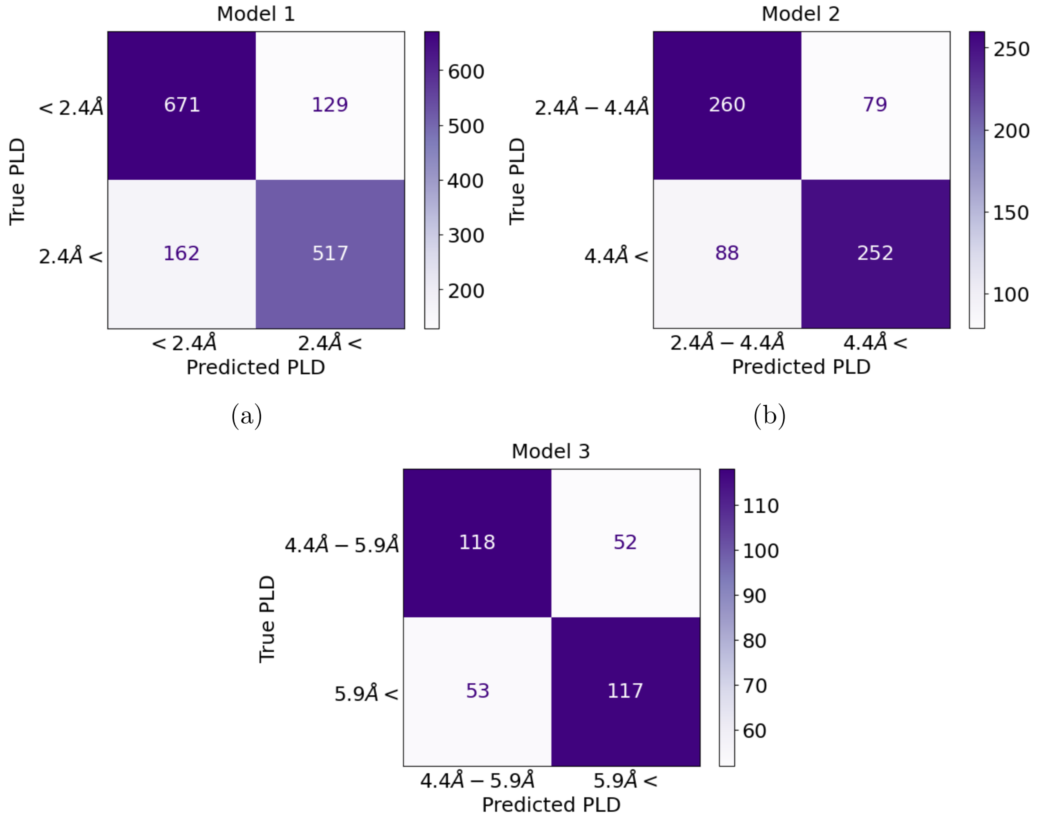
	PLD≥2.4Å			PLD≥4.4Å			PLD≥5.9Å		
	Seq.	M.C.	Reg.	Seq.	M.C.	Reg.	Seq.	M.C.	Reg.
Accuracy	0.81	0.78	0.75	0.77	0.72	0.73	0.69	0.66	0.63
Cohen's κ	0.60	0.55	0.51	0.54	0.43	0.47	0.38	0.32	0.26
Matthew's*	0.60	0.56	0.52	0.54	0.46	0.48	0.38	0.34	0.30
F_1	0.78	0.73	0.76	0.76	0.66	0.70	0.69	0.59	0.52
Hamming loss	0.20	0.22	0.25	0.23	0.28	0.27	0.31	0.34	0.37
Jaccard score	0.64	0.58	0.61	0.62	0.50	0.54	0.52	0.42	0.35
Precision	0.80	0.83	0.69	0.78	0.82	0.81	0.69	0.75	0.75
Recall	0.76	0.66	0.85	0.75	0.56	0.62	0.68	0.49	0.40
0-1 loss	0.20	0.22	0.25	0.23	0.28	0.27	0.31	0.34	0.37

* Matthew's Correlation Coefficient

Assigning outputs of a regression model to a class performed worse than the sequential and multi-class models with balanced accuracy of 51% compared to 58% and 54% accuracy respectively (Table 3.6). While the multi-class model had equal accuracy to the sequential models, when considering metrics which account for class imbalance in the overall dataset, the sequential model performs better (Table 3.7). Each binary classification task described in Section 3.2.2 was designed such that classes for each task were balanced. Thus, examining the performance of the multi-class model on each of these tasks highlights the superior performance of the sequential approach (Table 3.6). This justifies the sequential application of RFs as a successful approach to decompose the problem of predicting MOF porosity.

3.2.4 Discussion

RFs are seen to provide a good indication of MOF porosity using pre-existing feature descriptors. Mordred was seen to result in marginally better predictions than those generated with Dragon6 (Figure 3.6. This is a good example of open source software being equally



(a)

(b)

(c)

Figure 3.9: Confusion matrices for predicting pore limitting diameter (PLD) using a sequential approach (a) Model 1 predicts whether a MOF has pores of greater or less than 2.4 Å. (b) Model 2 is trained on MOFs with a PLD of greater than 2.4 Åand predicts whether a MOF has a PLD greater or less than 4.4 Å. (c) Model 3 is trained on MOFs with a PLD of greater than 4.4 Åand predicts whether a MOF has a PLD greater or less than 5.9 Å.

useful to commercial alternatives.

Due to long chains of hydrocarbons being very varied in structures, one would expect composition of a MOF to not be informative as to it's properties. However, representing MOFs using chemical composition still allowed for prediction of MOF porosity with 73% accuracy. As will be further explored (Chapter 5), the differences in prediction accuracy between feature engineered vectors and encodings of the composition is surprisingly small (approximately an 8% accuracy improvement).

Table 3.7: Comparison of the sequential, multi-class, and regression methods of predicting MOF porosity. While multi-class and sequential approaches have the same accuracy, sequential approach outperforms multi-class when measuring using metrics which account for class balance in the dataset.

	Sequential	Multi-class	Regression
Accuracy	0.68	0.68	0.59
Balanced accuracy	0.58	0.54	0.51
Cohen's κ	0.49	0.46	0.38
Matthew's*	0.49	0.46	0.39
F_1	0.68	0.68	0.59
Hamming loss	0.32	0.32	0.41
Jaccard score	0.52	0.51	0.42
Precision	0.68	0.68	0.59
Recall	0.68	0.68	0.59
0-1 loss	0.32	0.32	0.41

* Matthew's Correlation Coefficient

The impact of better featurisation on prediction of whether a material was porous or non-porous was not disimilar to the impact of using a sequential classification model over a regression model, which improved performance by 8% (Table 3.6). While the sequential application of RFs does increase the model training time, RFs are still fast to train when compared to other approaches such as deep neural networks. This is in line with trends from the literature which suggest that when using feature vector representations of materials, combining more, faster to train models will result in comparable if not improved performance when compared to deep learning methods [39].

One advantage of modelling the task of PLD prediction as a classification task rather than a regression task is more definite success critera. A classification prediction will either be incorrect or correct. In contrast, a regression model may only be able to provide an average level of accuracy (*i.e.*, mean error), which can be more difficult to interpret and may lead to lower adoption rates for the model. However, this also removes nuance from predictions. For example, a classification model may be able to predict whether a MOF is porous or not, but not the degree of porosity. This can be countered by introducing more classes, such as presented here, but there is a clear trade-off to be made between the nuance of prediction and the ease of model evaluation.

This trade-off is perhaps the most generalisable finding from this study. Regardless, of the property which is being predicted, or the ML model being used, the type of task and the metrics chosen for evaluation are vital in communicating the importance, and usefulness of an ML model.

Other findings are not so generalisable: It is not clear that sequential binary classification offers distinct advantages (or distadvantages) over multiclass classification for materials properties outside of MOF porosity. Researchers could examine these methods in other contexts, but as seen here impact seems to be minimal compared to other variables, such as choice of algorithm, or use of regression or classification tasks.

In the case of MOF porosity prediction presented here, it would have made no difference whether a regression or classification model was used, as no MOF's were made based on the prediction of this models. It is easy to conceive of a similar study to this that posed the prediction of MOF porosity as a regression problem. The conceived study would have chosen algorithms, and representations which resulted in the best regression performance, and reported results suggesting how useful the regression model was. As seen from the regression model trained for this study, such a model would likely perform well (Tables 3.6 and 3.7 and fig. 3.9). Metrics such as r^2, or mean error, could be used to justify this regression model, and the model would still have served the purpose of demonstrating the usefulness of the 1M1L3D dataset.

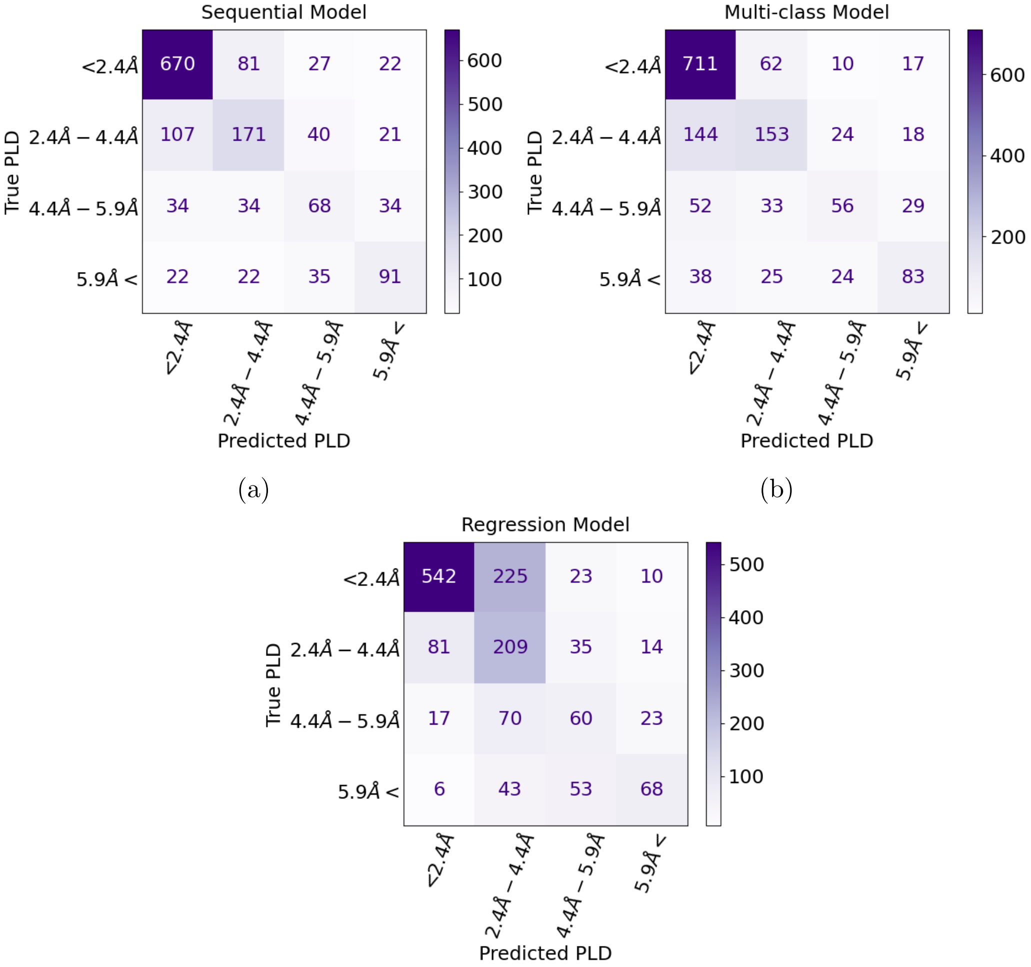
Without a clear use case and without incorporation into experimental workflows, while models such as the one presented here can at most demonstrate the potential of new techniques or datasets. In isolation, an ML model can never discover a new material, only predict the possible characteristics. Access to this dataset and the model's code are made free and open to the community so that researchers can incorporate this model into their experimental workflows, or build on it to gain new insights [164]. However, some coding knowledge is required to be able to use the model in this format. Improving access to ML models, and better designing workflows with ML in mind is key for their justification and will be discussed in subsequently (Chapter 4).

3.2.5 Conclusion

This section outlined the preparation of a new dataset for ML, and the implementation of a sequential RF approach to classification of MOF porosity. It was noted that in different MOFs may have the same linker and metal combination, but vastly different properties.

While in all areas of materials science, compounds may contain the same elements but differ in structure, this is particularly a problem for the fields of MOFs and organic chemistry, where hydro-carbon chains can form large and complicated structures.

As such, featurisation methods to represent linkers were tested, finding marginally better predictive performance for models trained using the open-source Mordred software's representations as opposed to the licenced Dragon6 software's representations. The model presented could be used to screen candidate MOFs for their predicted porosity, and is made open-source for this purpose. However, without a clear method of integrating this model into an experimental workflow this model is of limited use. Methods of integrating models such as this into experimental workflows will be explored in the next chapter (Chapter 4).



(a)

(b)

(c)

Figure 3.10: Confusion matrices for predicting pore limiting diameter (PLD) as a multi-class classification problem (a) Confusion matrix using a sequential model (combining models 1, 2 and 3) (b) Confusion matrix using a multi-class random forest (RF) (c) Confusion matrix for a regression RF who's output has been assigned into converted assigned to the relevant class.

3.3 Considerations in using random forests for prediction of material properties

This chapter has examined two examples of using RFs to predict materials properties, giving reflections on each. On their surface both examples are very different. One (Section 3.1) concerns inorganic chemistry, the other organic chemistry (Section 3.2). One encodes the materials using ratios of composition; the other uses an engineered feature vector. One approaches the task from the view of chemical substitution studies, the other takes a more general approach which only uses linker SMILES and metal ligand information.

A common theme between these two studies is that rather than using more complex models, both simplify the prediction task at hand to a binary classification problem in order to report improved results. The prediction of c/a ratio was simplified to binary classification of a new material compared to the c/a of an existing material, and the prediction of MOF porosity was simplified to three binary classification models used in sequence. Were these models to be used for the screening of potential new materials, this simplification would not have a substantial effect on the use cases of these models compared to more complex regression models.

Another common theme is that none of the models presented in this chapter were used to justify synthesis. While both of the studies here supplemented publications useful to the community [125, 144], and could be used to inspire future research, no materials resulted from the works discussed here. It is clear that for the discovery better accessibility for these models are required, and more work is required to integrate models such as these into the materials discovery process. In the following chapters this issue will be further examined, and methods to address it will be suggested.

3.4 Thesis context

In this chapter, an exploration was conducted on the use of RFs for predicting materials properties. The next chapter (Chapter 4) expands on this line of inquiry by examining how RFs (and ML methods more generally) can be employed in collaboration with experimentalists to explore potential new superconductors within the context of materials discovery.

Although the featurisation methods employed were briefly outlined, a more comprehensive investigation is warranted to justify the techniques used. Consequently, the subsequent chapter (Chapter 5) undertakes a detailed examination of these methods, and further methods to quantify model (and featurisation method) performance.

Chapter 4

Collaborative workflows for discovery of superconductors and other functional materials

This chapter explores two distinct approaches to using machine learning (ML) for material discovery. The first involves direct collaboration with experimental chemists with the goal of creating new superconductors. To achieve this, a basic random forest (RF) model was trained to predict superconducting critical temperature, T_c, at the time RFs were the most notable ML models for use in the prediction of superconductors [170]. A feedback-driven workflow was established to enable effective collaboration, focussing on areas of the chemical space that align with the chemists' interests.

The second approach involves collaborating with computational researchers to improve the accessibility of their tools. Specifically, these tools are transformed into web apps that can be easily used by people without coding experience. This chapter provides a comparative analysis of the two collaborations, examining the practicalities of translating ML models into novel materials within an academic context.

Note that, as discussed (Section 1.1), this thesis was never intended to focus on chemical synthesis, but rather to collaborate with highly specialised chemists to guide their work. With this in mind, the following sections offer valuable insights into the various ways in which ML can be used to facilitate meaningful collaborations and drive innovation in materials science.

Specifically the contributions of this chapter are as follows:

- A thorough investigation of literature surrounding the prediction of superconducting critical temperature.
- Exploring the "garbage in" method found in the literature, and questioning its underlying assumptions (namely that the Crystalographic open database does not contain any superconducting materials) and investigating the effects of those assumptions.
- Definition of a workflow to allow for collaboration and feedback with experimentalists.
- Screening of over 1 billion candidate materials for superconductivity using ML methods.
- Collaborating with experimentalists to identify areas of interest in the chemical space, helping to rationalise results from the screened materials.
- Creation of a filter to identify materials which could be potentially arc-welded.
- Inspiring the synthesis of possible candidate materials in the Sr-Cu-Sn and Sr-Cu phase fields.

4.1 Collaborating to create superconductors

This section documents the process of developing superconductors through the use of RFs to predict T_c. The initial plan was to use the best published methods to predict whether a material was superconducting and present predictions to experimental chemists for them to possibly synthesise. Feedback was incorporated to guide the creation of a workflow that was used to refine lists of candidate superconductors.

A thorough review of the relevant literature is conducted to determine a viable strategy for predicting whether a material is a superconductor. When initial results were presented, it became clear the importance of identifying candidate materials that were easy to synthesise or that were within a chemical space closely related to an experimentalist's existing work and research interests. This prompted the development of a workflow that could take into account these considerations. After several iterations and hyperparameter adjustments, the synthesis of new materials was attempted. This section will go into the details of the process, including the models that were developed, the justification for these models, and the workflow that was produced to enable successful collaboration between ML and chemistry experts.

4.1.1 What are superconductors?

Superconductors are materials that exhibit zero electrical resistance below a specific critical temperature, T_c. They have a wide range of potential applications in industries such as healthcare, transportation, and energy. There are several known mechanisms by which a material can display superconducting properties. Notable categories include Bardeen–Cooper–Schrieffer (BCS) theory superconductors, cuprate superconductors (with $YBa_2Cu_3O_7$ as a notable example), iron pnictides, iron oxyarsenides, and hydrides.

BCS theory developed in the late 1950s and explains that superconductivity arises due to the formation of electron pairs called Cooper pairs [172]. At low temperatures, the low energy of the phonons in a material allow these Cooper pairs to move through a material without resistance. BCS theory superconductors have low critical temperatures, usually below 30 K.

Cuprate superconductors are a type of high temperature superconductor that have critical temperatures above 30 K [147]. They are made of copper oxide layers and other elements. The mechanism behind Curpate superconductivity is still not fully understood.

Iron-oxi-arsenide and iron pnictide superconductors were discovered more recently than cuprate superconductors and are made of iron, oxygen, and either arsenic or nitrogen [133, 160]. The critical temperatures for these superconductors typically range from 26 K to 55 K. It is thought that iron arsenide or nitride layers play a key role in the superconductivity of these materials.

Hydrides are a new class of superconductors that were discovered in the last decade [53]. They are made of hydrogen and a metal such as lanthanum or yttrium. They can have very high critical temperatures, up to 250 K, but they require extremely high pressures to be produced. Hydrogen atoms are believed to play a crucial role in superconductivity, but the exact mechanism is still being studied.

It should be noted that while these categories represent significant advances in the field of superconductivity, it is still unclear whether they represent the full spectrum of superconducting materials or whether additional categories remain undiscovered. Additionally, both pressure and electron doping can impact the superconducting properties of materials, which are not fully addressed by the methods presented in this section. These issues will be explored further in Section 4.1.10.

4.1.2 Relevant literature

Before engaging with the literature, it is pertinent to say that the literature has changed since this work was done. Changes will be noted in this section and how such changes could have impacted this project will be noted in Section 4.1.10

A key part to understanding the nuances of the literature on using ML to predict superconductors is understanding the dataset. All studies discussed here use the superconductors dataset from the Japanese National Institute of Materials (supercon) [130]. Supercon was available online on the National Institute of Materials website until approximately mid-2021, when it was removed until it migrated to a different host in December 2022 [114].

Between these two dates the two versions of supercon were publicly available. The first was in the GitHub repository associated with Stanev et al.'s work [170]. The second was redistributed on Kaggle and associated with the work of Hamidieh [64]. Stanev et al.'s work uses a RF to predict T_c of a material in three contexts:

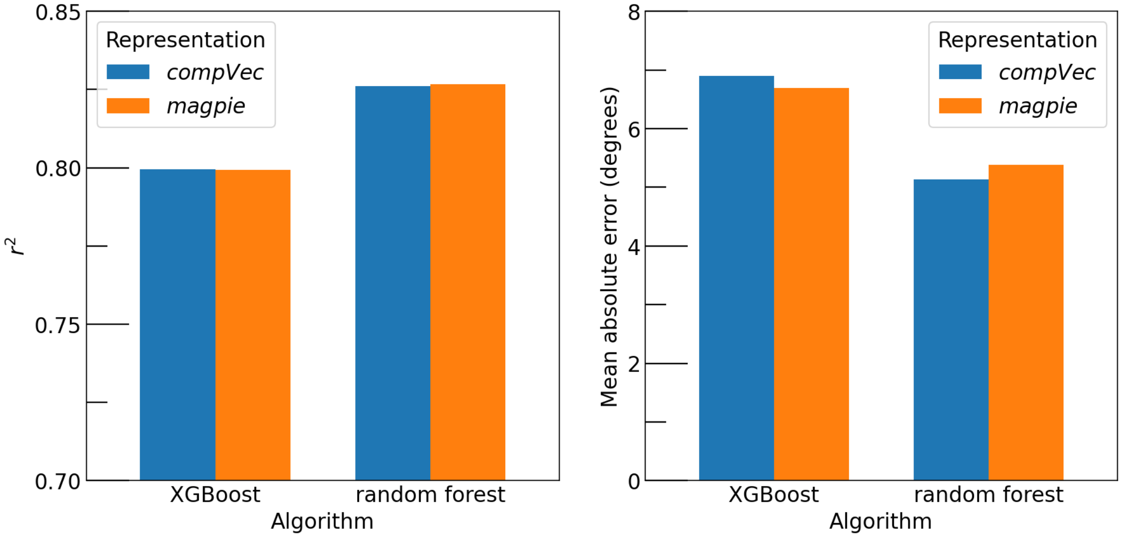
- Training a regression to predict T_c.
- Training a classification model trained to classify whether a material with have T_c greater or lower than 10K.
- Training a regression model to predict T_c for materials which have T_c greater than 10K.

Train and test sets were derived from the supercon dataset, with duplicate entries removed. Compositions were featurised using a custom composition-based feature vector (CBFV), implemented with the matminer package [194]. Due to a lack of specificity on the exact constituents of this CBFV, exact reproduction of this study is not possible, although the results reported are broadly in line with similar CBFVs and experiments (as will be seen in this section and in Section 5.2). As this dataset is only materials with a reported T_c, there is no negative data available to the model during training. That is, the model is not trained on anything that is known to not be superconducting. For example, in order for the regression model to consider a candidate material not to be a superconductor, a prediction that $T_c = 0$ K should be returned. But non-superconducting materials are not just superconductors with a T_c of 0 K, as such any model trained with such a schema will have a bias which is not indicative of the real world.

Further work used convolutional neural networks in the prediction of T_c [88]. This work predicted T_c as a regression task and whether $T_c > 10\ K$ as a classification task, convolving over a periodic table representation of the composition of materials. To address some of the issues highlighted in Stanev et al.'s work a "Garbage in" method was proposed, in which non-superconductors were considered to have a T_c of 0 K to provide more negative examples to the ML models. Non-superconductors were considered to be structures reported in the Crystallography Open Database (COD) that were not featured in the supercon dataset. It is notable that the structures reported in the COD could be superconductors that have not yet been identified or added to the supercon dataset. *i.e.*, this assumes that the COD are known negative data points, when they are just unknown data points. This work used the GitHub redistribution of supercon, and thus any repeated entries were removed.

Two further studies propose general purpose prediction of T_c and use the supercon dataset but do not remove duplicates [64, 158]. This means that the same composition with different levels of electron doping could be present in both the training and data, which is a clear source of data leakage. One of these studies used a variant of RFs and a composition vector representation, the other, Hamidieh, used extreme gradient boosting (XGBoost) with a custom CBFV [24]. Data associated with Hamidieh's study are available on Kaggle [178], which continues to lead to further confusion and publications with inflated performance due to data leakage [203]. Since the work presented in this section was carried out, Hamidieh's ML model implementation has been reproduced using Stanev et al.'s distribution of the supercon dataset [168], finding results to be marginally better than those seen using RFs. However, as will be seen, partial recreations presented here will not agree with this finding. Although recreations presented here (Figure 4.1) do not contain any hyperparameter tuning, which likely explains the difference.

It is notable that other work focused on predicting the T_c for subsets of superconductors [74]. However, as no specific subset of superconductors was highlighted at the onset of this study, these have been omitted from this review of the literature. Subsequent work has tried different ML models [57], showing some improvement over Stanev et al.'s work. Although structural information was not used in any of the experiments presented in this chapter, the performance of the ML model has been seen to improve using structural information [201], including information extracted from band structures (this will become relevant again in Chapter 7).



(a)

(b)

Figure 4.1: Two prominent algorithms used in the literature to predicting T_c. Two notable representations are also compared. (a)The r^2 correlation between predicted and and true values of T_c. (b) The mean absolute error between predicted and and true values of T_c.

4.1.3 Initial experiments

Work was done to review the work of Stanev et al., using a RF to predict the T_c for a range of superconductors. As the custom CBFV used in that work was not reproducible, a composition vector, $CompVec$ [79] and the common CBFV *magpie* [192], were used (see Section 2.2.5). $CompVec$ was found to be roughly as effective as the CBFV, which prompted a further study on the effects of material representation on ML model performance, which will be presented in Chapter 5. Both XGBoost and random forests were tested and XGBoost was found to perform slightly worse (Figure 4.1). As the goal was to minimise the number of experiments taken to successfully identify a superconductor, precision was used as the metric for success.

The garbage in method was evaluated, but rather than using data from the COD, the Inorganic Crystal Structure Database (ICSD) was used as it is a larger source of data. It was found that garbage in resulted in a slight improvement in the performance measured with a test set that had also been diluted using the same proportion of garbage in (Figure 4.3), but it was observed that the ICSD may contain superconductors that have not yet been identified, as such could not constitute garabge. As such, further experiments were carried out to see if the source of negative data for the garbage in method made a difference to the resulting precision (Section 4.1.5).

After the initial models were trained. Predictions were made from 3 lists of candidates:

- The ICSD.
- A list of 1 billion charge neutral candidates generated using combinations of elements sampled in order of the periodic table (*i.e.* first H, then He... then H and He....). Charge-balanced combinations of these elements were found using an existing tool [34] to create the candidate list.
- A list of 1 billion charge neutral candidates chosen from random combinations of between 1 and 8 elements.

Having screened these lists, it became clear that further methods were needed to limit the results to be analysed. Feedback also suggested that chemists would be more likely to synthesise something which was easy to synthesise or was adjacent to their field of interest. A filter was defined to find combinations of elements that could be smelted together in an arc welder (Section 4.1.7), and further areas of interest were defined to help filter the list of candidate predictions. To help inform chemists, different models were combined to create uncertainty estimates of the predictions (Section 4.1.8). In addition to this, a workflow was established to help communicate the results and ensure efficient work and communication (Section 4.1.4).

4.1.4 Established Workflow

Explicitly stating the steps involved with the process of using ML for materials discovery, each process can be individually optimised. For example, the workflow presented here starts (Figure 4.2) with featurisation, which garbage in is discussed in Section 4.1.5, but overall the featurisation process will be fully explored in chapter 5.

After data have been featurised, a model must be built. New ML models and evaluations of types of models are a common source for investigation [57, 38]. In this thesis, the evaluation of this step is mostly confined to Chapter 7.

After training an ML model, it should be evaluated, methods for this are introduced in Section 2.1.6, but fully explored and expanded on in Chapter 5. Once the performance

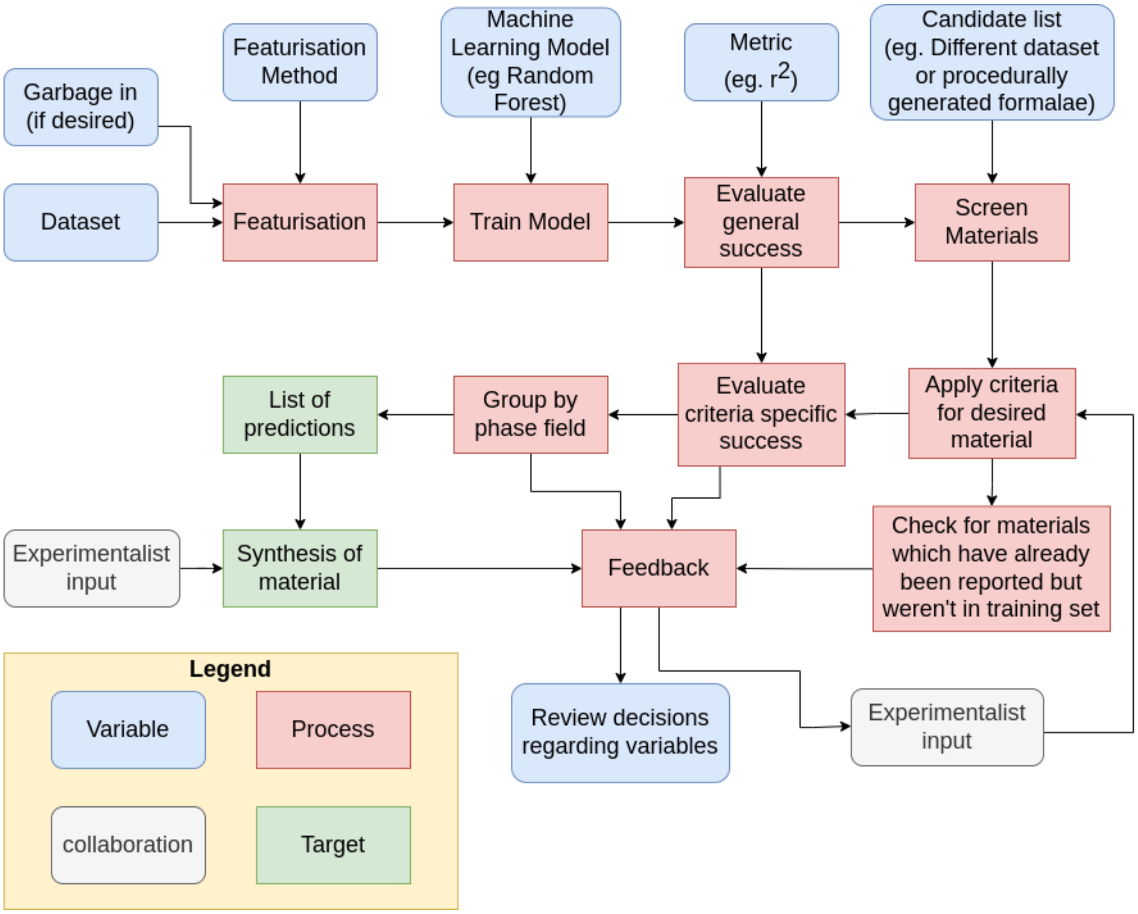

Garbage in (if desired)
Featurisation Method
Machine Learning Model (eg Random Forest)
Metric (eg. r²)
Candidate list (eg. Different dataset or procedurally generated formalae)
Dataset
Featurisation
Train Model
Evaluate general success
Screen Materials
List of predictions
Group by phase field
Evaluate criteria specific success
Apply criteria for desired material
Experimentalist input
Synthesis of material
Feedback
Check for materials which have already been reported but weren't in training set
Legend
Variable
Process
collaboration
Target
Review decisions regarding variables
Experimentalist input

Figure 4.2: A workflow for discovering new materials

of a model has been established on a general dataset, candidate lists can be screened as detailed above (Section 4.1.3). These candidate lists can then be filtered based on criteria (Section 4.1.6 and Section 4.1.7). Performance can then be reevaluated on classes of materials which are of interest, and predictions can be checked against the literature to establish if the candidate list contained materials of interest which have already been discovered. For example, in the search for superconductors described in this chapter, many of the materials flagged to be of interest, were already investigated, and were found to be superconducting but were not in the supercon dataset. Lists of candidates can be further limited by grouping them by phase field, which can result in a digestible list of predictions to be presented to experimentalists who may or may not choose to synthesise results.

Crucially, this workflow contains many areas for feedback, where decisions different choices can be made. This can be seen as a positive, as there are many opportunities to improve the process, but also as a negative, as there are many opportunities to get stuck in a feedback loop in which nothing ever gets synthesised.

4.1.5 Garbage in

It was observed that the garbage in process had a flaw in that there is an assumption that anything in the ICSD was not superconducting if it was not in the supercon dataset. To see if this impacted model performance, criteria that would preclude a material from superconductivity were established. Sets of materials in the ICSD which fit that criteria were used as the non-superconducting results for the garbage in process. Thus, the following sets were defined as sources of garbage:

1. Insulators (experimentally measured): Materials that had been experimentally measured to have a band gap greater than 1eV. Insulators are not known to be superconducting and as such provide definite negative results.

2. Insulators (DFT calculated): Materials which had been calculated to have a band gap greater than 1eV using density functional theory (DFT). Note that DFT is known to underestimate band gaps.

3. Magnets: Materials which exhibit magnetism as determined by having a Curie temperature reported for them. Magnets are not known to be superconducting and, as such, they provide a good source of negative results.

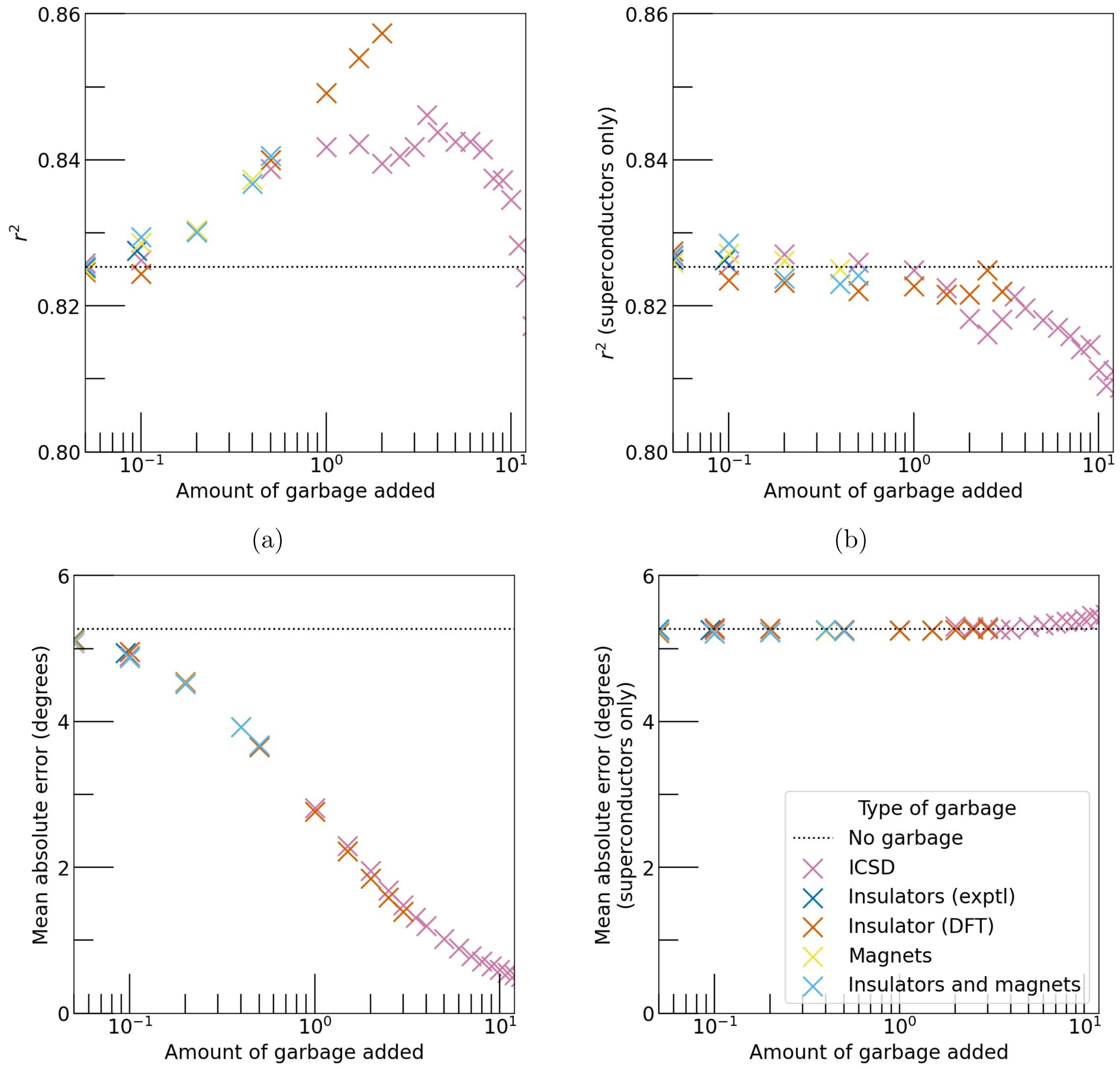
4. Insulators and magnets: the union of the sets of experimentally measured insulators and magnets described above.

5. ICSD: any member of the ICSD that is not reported in the supercon dataset

6. No garbage: No garbage in provided.

These garbage sources were combined with the supercon dataset in various proportions and the performance of a RF in prediction of whether the T_c was greater than 0 was established.

The source of garbage was found to make little difference to the performance of the models (Figure 4.3), with what impacted performance the most is the amount of garbage



(a)

(b)

(c)

(d)

Figure 4.3: How garbage in affects performance depending on different sources of garbage in. It is unclear whether garbage in should be evaluated using only the supercon test set, or on a test which also is diluted with other results. Increasing the amount of garbage in is seen to improve performance using the former schema. Using the latter schema, performance in the test set decreases when the training set contains approximately twice as much garbage as it does ground truth data points. Different sources of garbage do not seem to affect performance. Results shown here are the mean of two RF models, one trained using a *CompVec* representation and the other a *magpie*. (a) r^2 of models trained with different proportions of garbage in as measured by a test set with the same proportions of garbage in. (b) r^2 of models trained with different proportions of garbage in as measured by the supercon test. (c) Mean absolute error of models trained with different proportions of garbage in as measured by a test set with the same proportions of garbage in. (d) Mean absolute error of models trained with different proportions of garbage in as measured by the supercon test set.

present relative to the size of the supercon dataset. Depending on whether results were measured with a test set diluted by non-superconductors changed whether garbage in was beneficial. To balance observations made on test sets with and without garbage in, a ratio of 2:1 garbage to superconductors was chosen for the training set for experiments going forward. For sources of garbage which did not contain enough data points to make up that ratio, all of that source was used as garbage.

This results in 12 different RFs: one *magpie* and one *CompVec* model for each point in the above enumeration. The 12 regression models were used forward, with the ratio of agreement of these models being used as a proxy for confidence. This will be discussed in Section 4.1.8.

4.1.6 Areas of interest

Discussion in the literature largely focusses on finding high-temperature superconductors [170, 117]. That is, the objective is often seen to be to find superconducting materials with the highest T_c possible, with the ultimate goal being a material that is superconducting at room temperature.

As such tasks in the literature are modelled with this in mind. As noted earlier (Section 4.1.2), work is often divided into two tasks, prediction of T_c and prediction of whether T_c is above 10 K. Initial experiments reflected this (Section 4.1.3), focussing on the latter of these tasks. However, it became clear that what was considered to be more interesting were materials which fall into specific categories.

The following categories were identified as areas of interest:

1. The materials predicted to have $T_c > 30$ K, that do not contain cuprates, iron arsenides, or iron selenides. High-temperature superconductors that fall into this category are of interest as it would be unclear what mechanism would cause superconductivity.

2. Materials predicted to have $T_c > 20$ K, and less than 10% molar mass oxygen. These materials were considered to be intermetalics, which tend not to be superconductors at higher temperatures

3. Materials predicted to have $10 \text{ K} < T_c < 30$ K with one of chromium, manganese, iron, cerium or nickel. Superconductivity with a T_c less than 30 K would be indicative

of a superconductor of BCS theory, but the listed elements are not typical of such a material. As such, these would be of interest.

4. Materials predicted to have 1 K < T_c < 20 K no oxygen. For reasons similar to the above but with a lower threshold to exclude cuprates.

5. Materials predicted to have 1 K < T_c < 20 K no oxygen, and no iron arsenides or iron selenides. For reasons similar to the above.

It is with the last three items that the disadvantage of the approach taken here and in the literature becomes clear. As non-superconductors are represented to the model as having a T_c of 0 K, any material with a T_c of 0 K must be explicitly avoided. The nature of RFs involves taking a mean of different predictions; thus, any prediction in a constituent tree that is greater than 0 will result in a prediction from that RF of greater than 0. As such, a lower limit must be put on to filter out very low results. This was set at 1 K for categories 4 and 5, but increased to 10 K for category 3 based on the number of phase fields predicted to have a T_c between 0 K and 10 K (approximately 60,000)

4.1.7 Defining a filter for materials that can be arc welded together

The candidate lists defined in Section 4.1.3 were screened through the model and then filtered to highlight candidates that fit into the above categories. However, it was further helpful to screen through materials which are likely to be easily synthesised.

Specifically, materials that can be arc-welded were suggested. As such, the above categories were further filtered using the following criteria:

- The minimum melting point of any constituent element must be greater than 1273 K (1000 Celsius)

- The difference between melting points of the constituent elements must be less than 500 degrees.

- The material cannot contain oxygen.

By definition, none of the predictions which fit into categories 4 or 5 also passed this filter. None of the candidates were discarded for not passing this filter, but candidates that did pass this filter were presented separately for researchers who were interested.

4.1.8 Ensembles models as a proxy for uncertainty

To further aid decision making when presenting lists of candidates, an associated uncertainty was provided. As 12 RF models were presented (see Section 4.1.8), the amount of agreement between them could be used as an uncertainty measure. Specifically, agreement here is defined as the proportion of models which agree that a given candidate is predicted to fall into one of the areas of interest listed above.

It is fairly typical to use the differences between predictions of constituent trees in an RF as a proxy for the certainty of a prediction [19], however, in this case numerous different RFs were compared for three reasons:

- Candidates are grouped by phase field but screened as individual compositions: rather than finding a way to aggregate different predictions across candidates in the same phase field, one can count how many models predicted candidates in a phase field fell into one of the areas of interest. This number can then be compared to the number of predictions that considered a candidate to not fall into that area of interest.
- It would theoretically allow for differentiation between predictions based on the source of garbage that was used. In practise, this was argued against; RFs expect some deviation in predictions, and reading too much into this deviation is ill advised.
- It was requested by the collaborators.

This uncertainty was presented with the list of candidate phase fields.

4.1.9 Results

The screening process described above allowed the lists of screened candidates to be limited to a reasonable number of interesting candidates (Figure 4.4). 242 predictions from the ICSD candidate list that fell into areas of interest were further researched. Of these 226 had been found to be superconductors which had been characterised already in the literature but were not part of the supercon dataset, and 16 had been characterised and confirmed not to be superconductive. While this is encouraging because it implies that the models are accurate, it also means that no materials that have already been synthesised have potential to be resynthesised and characterised based on the findings here. Presenting the list of candidates in this way prompted synthesis investigations the Sr-Cu-Sn phase field,

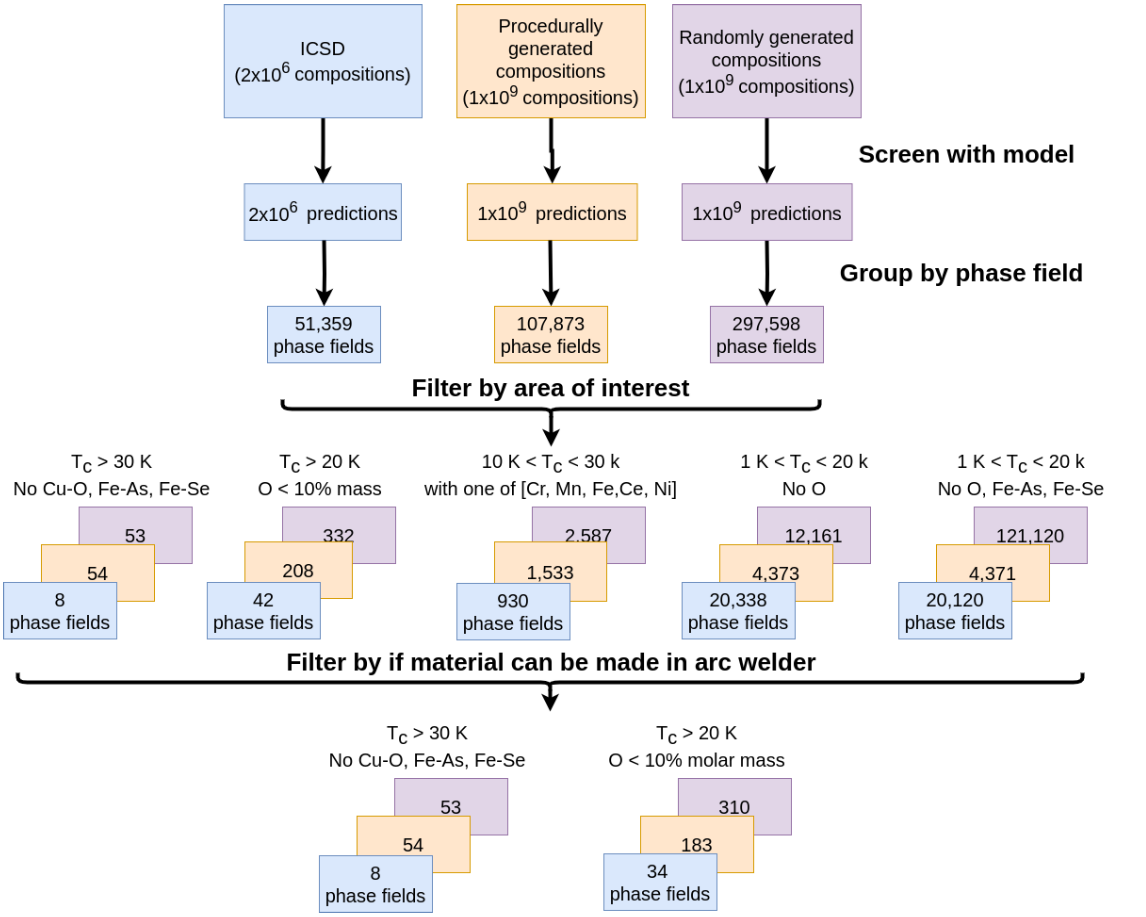


Figure 4.4: The process utilised to screen large numbers of materials.

as well as SrCu5 and SrCu. SrCuSn2 and SrCu9Sn4 have been successfully synthesised, but neither compound shows signs of superconductivity above 2 K.

4.1.10 Discussion

Defining the workflow formally proved to be a very helpful tool for collaboration and by successfully filtering the lists of screened candidates into manageable sizes. Although none of the RFs presented perform particularly better than those presented in the existing literature, they are very simple and provide a good proof of concept for the workflow presented. These results could likely be improved upon through the use of deep learning techniques or different featurisation methods. If any researcher does choose to try and improve on results seen here and wishes to communicate these results, it may be advisable to outline a workflow such as that presented here to separate model development, featurisation, evaluation, and screening of materials. However, any researcher looking to improve on results here may be better advised to rethink the tasks that a ML model is trained to do and differ more from existing literature than this work did.

The literature surrounding this topic is flawed [168] and the work presented here does little to fix it. The task of predicting new superconductors cannot be adequately modelled on the assumption that materials that are not superconducting have a T_c of 0. As such a regression model alone is not a good choice for this task. A better approach may be to train a classifier as was done to predict MOF porosity in Section 3.2, with a further category being for non-superconducting materials.

These ML models for predicting T_c may also benefit from incorporating pressure into the prediction in some way. Many of the superconductors that have been reported to have high T_c only do so in a pressurised environment. Incorporating information in some way into a model better reflects where state-of-the-art is currently with respect to discovering high-temperature superconductors.

Although the workflow presented here may be helpful, it is fairly typical of the literature not only for superconductivity prediction, but also for a variety of other materials science tasks [35]. Methods (such as those discussed in Sections 4.1.6 and 4.1.7) for filtering candidate predictions based on specific criteria are often not explicitly noted. In this collaboration explicitly noting the filtering methods used was helpful for communicative purposes, and without them attempts to synthesise materials would have been unlikely.

Although this workflow is typical (particularly at the time when this work was done), more recent literature seems to be moving away from this approach in favour of generative models [4] or Bayesian optimisation (BO) [188]. Generative models used in this setting would typically be used instead to generate structures or compositions which the model believes to be typical of a material with a certain target property. This effectively eliminates the screening step and the need to find lists of candidates. BO works by balancing the uncertainty of a prediction with the value of the prediction to explore a problem space (in this case a chemical space). BO is not suitable for all areas of chemistry; for example, it is sometimes easier to sample a phase field uniformly than it is to target a specific structure [175]. BO also sometimes requires that points be sampled from an area of a space that a chemist may know is unlikely to yield positive results. It may not be conducive to collaboration to discount the knowledge of a chemistry expert by relying solely on the output of a ML model.

The situation in which a computational researcher is not synthesising materials is common. There are broadly two ways for an experimental chemist to synthesise a screened candidate. Either that chemist chooses to take time and resources from their research schedule and synthesise the material, or the chemist is told to synthesise the candidate by a higher up (such as a supervisor). Although automated laboratories are the topic of current research, at the time of writing, collaboration is required.

In a hierarchical setting such as academia, it is completely possible that an experimentalist is told to make a material by, for example, a supervisor. At this point the person for whom candidates are screened is not the experimentalist, but the experimentalist's supervisor. Although this abstraction is not necessarily bad, it does mean that the interest of that experimentalist may not be taken into account. The creation of phase-pure novel materials is not easy, and it may require many attempts, as such enthusiasm for collaboration is important.

Where a chemist chooses to synthesise predictions, such enthusiasm is already implied. This reflection on the importance of enthusiasm for material synthesis would be true regardless of whether ML was involved in a process, but it is still important for ML practitioners to note when collaborating in this field. It is not the purview of this thesis to discuss how to inspire enthusiasm, but the observation is made that enthusiasm in collaboration must be reciprocated. That is, enthusiasm for the techniques presented is important from the computational researcher.

RFs are an algorithm that produce good results, and were state of the art for this specific task at the time this work was done. However, other areas of this thesis were seen as more pertinent and, as such, received more enthusiasm. Without enthusiasm for the work presented in this section, any collaboration based off this work was likely to fail.

4.2 Implementing a cloud platform to aid material discovery

This section contains results under peer review at Royal Society of Chemistry: Digital Discovery. Note that while credit as co-author of the system architecture is taken, only the metal organic framework (MOF) prediction tool is a production solely of this thesis. Original scripts for thermal conductivity prediction and heat capacity modelling were adapted into web applications (with varying levels of code rewriting), and no credit is claimed for element movers distance (ElMD) based applications or the lithium ionic conductivity pre-

diction applications.

4.2.1 Introduction

In recent years, substantial investment has been made in developing computational tools to expedite experimental workflows in material discovery. However, the integration of these tools into the materials discovery process remains an ongoing research field. Although code repositories like GitHub or compiled code [194] sharing can facilitate tool sharing, such approaches require technical proficiency, which can hinder experimental researchers without the necessary know-how from using the tools. Thus, reducing the technical expertise prerequisite for synthetic researchers to employ computational tools is vital to ensure their successful adoption.

Direct collaboration with computational experts offsets this need for computational expertise. Computational chemists are able to identify state-of-the-art tools and can develop bespoke applications where needed. This type of collaboration may require organisational restructuring to most effectively accommodate computational developers and trained technical users identifying separate priorities.

Three distinct paradigms for sharing computational tools may be observed; the private, shared, and cloud paradigms (fig. 4.5). Private tools are those described in the literature but not made publicly available. Shared tools are accessible when running on local hardware, and cloud tools are accessible for remote usage. Notably, both shared and cloud paradigms may or may not be open-source and/or free. This section focusses on the cloud and shared paradigms, with the aims to use cloud resources to ease collaboration between experimental and computational chemists.

Often nebulously defined, "the cloud" typically refers to the global network of computer servers on which computation. Cloud based tools (also called applications, apps, services, or micro-services) described in this section are, broadly, algorithms or functions which take user input communicated via web protocols (such as HTTP [51]), process these data, and return the output to the user in a presentable fashion (*i.e.*, in a graphical web browser). For materials scientists, examples of such tools range from state of the art ML models to predict material properties [174], and vast libraries of DFT calculations [78], to simple utilities that assert the charge neutrality of a chemical formula [72]. Web applications are a prominent example of software that promotes collaboration. The use of web browsers is ubiquitous and graphical user interfaces (GUIs) are the typical method of interacting

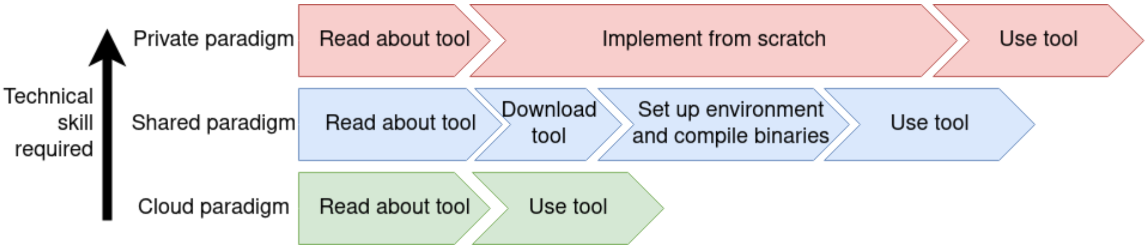


Figure 4.5: Paradigms of sharing computational tools

with software. As such, accessing computational tools through a GUI in a web browser minimises the technical expertise required to use such tools.

It is possible for web applications to be run locally (and thus fall under the private or shared paradigm), but this often demands programming and networking expertise, which may be outside the past experience of researchers with familiarity in other technical domains. Furthermore, designing bespoke applications for internal use clearly limits the audience which can interact with a tool. Locally hosting python notebooks [81] is a common solution to sharing tools within research groups; however, this still requires some technical knowledge to setup, and does not provide an accessible interface for non-technical users. Publicly accessible web applications [174, 72, 132, 32], which fall under the cloud paradigm, can be designed towards a specific use case to improve usability and allow many researchers to interact with an application.

A collection of these tools together forms a cloud platform. Cloud platforms offer computational researchers the opportunity to share their tools with a wider audience. These platforms can also include related resources, for example, both AFLOW [32] and the Materials Cloud [174] platforms host datasets, and provide a front-end to access to these datasets.

Cloud platforms offer many benefits, but they may not offer suitable extensibility for computational researchers who want to share newly developed tools. Some platforms, such as Materials Atlas [72], allow developers to upload new tools; however, developers may be reticent to launch a tool with these platforms for a number of reasons. Vendor lock-in can arise when relying on third parties to host tools, direct access to live code may be limited, use of certain code libraries or programming languages may be restricted, there may be limited control over end-of-life provisioning, and a culture of collaboration may not have been established between teams. An alternative solution is to host applications in house.

Using modern frameworks, this can be done easily and securely using while minimally increasing maintenance duties for the research team.

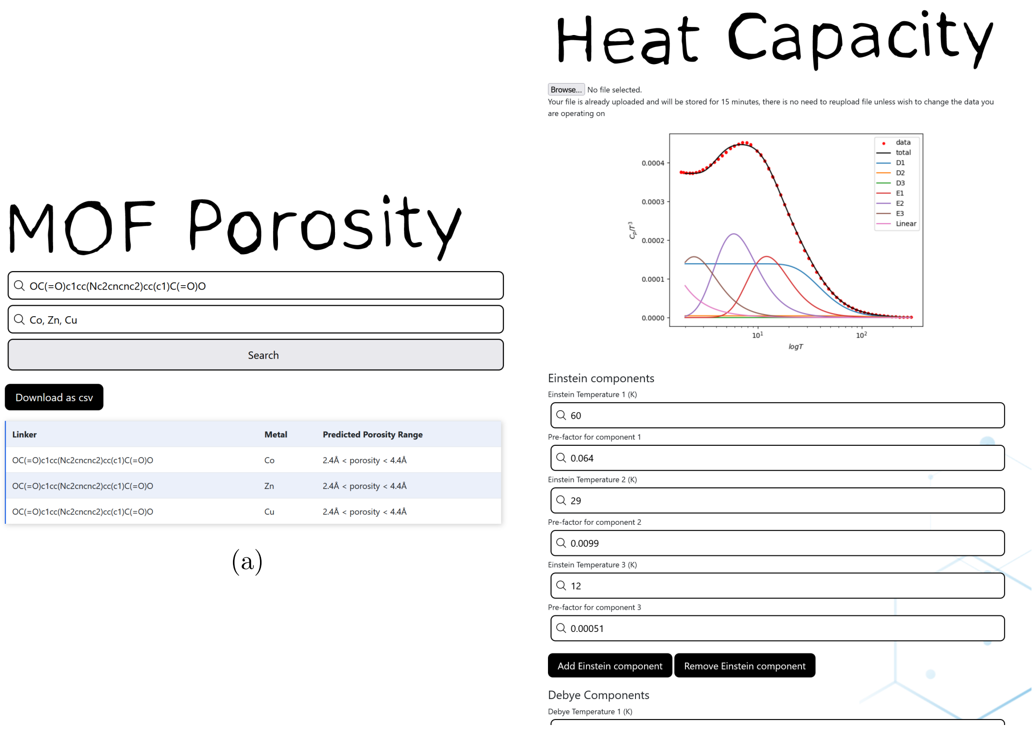
This section presents a new cloud platform, the “Liverpool Materials Discovery Server” (LMDS). Applications currently accessible on this platform will be detailed, as will the approach employed to enable researchers flexibility when deploying new applications. The aim of this platform is not only to share tools created by local researchers, but also to provide frameworks for other research groups to launch bespoke platforms while minimising the technical debt associated with such a task.

LMDS is designed to be simple and easily replicated, with an emphasis on reducing technical overhead rather than computational overhead. Source code and architectural information is provided, allowing easy adoption by other research groups to share their ML models either on their local intranet or on a public facing website. Giving each team personal ownership of their work promotes diversity in the field, and allows each group to discover their own optimal workflow as well as share their findings to the wider community. The approaches outlined in this section should allow the launch of new cloud platforms with minimal time and financial investment. In the following discussion considerations are addressed that must must be taken into account when sharing computational tools, and the role of such tools in the materials discovery workflow.

4.2.2 Available tools

The tools currently available to the public on the LMDS (lmds.liverpool.ac.uk) are as follows:

- ElMTree - An tool for finding structures most similar to an input, as measured by the element movers distance (ElMD) [66], a mathematically justified composition similarity metric
- ElM2D - A method of generating ElM2D scatter plots of compositional similarity is provided, which allows the results of querying the ElMTree application to be visualised in 2 dimensions, providing an intuitive representation of the chemical distribution.
- MOF porosity prediction - an implementation of the model presented in Section 3.2.
- Lithium - ionic conductivity prediction - An implementation of CRABNet models [186] trained for Lithium ion conductivity prediction [67].



MOF Porosity

OC(=O)c1cc(Nc2cncnc2)cc(c1)C(=O)O

Co, Zn, Cu

Linker	Metal	Predicted Porosity Range
OC(=O)c1cc(Nc2cncnc2)cc(c1)C(=O)O	Co	2.4Å < porosity < 4.4Å
OC(=O)c1cc(Nc2cncnc2)cc(c1)C(=O)O	Zn	2.4Å < porosity < 4.4Å
OC(=O)c1cc(Nc2cncnc2)cc(c1)C(=O)O	Cu	2.4Å < porosity < 4.4Å

(a)

Heat Capacity

No file selected.

Your file is already uploaded and will be stored for 15 minutes, there is no need to reupload file unless wish to change the data you are operating on

Einstein components

Einstein Temperature 1 (K)

60

Pre-factor for component 1

0.064

Einstein Temperature 2 (K)

29

Pre-factor for component 2

0.0099

Einstein Temperature 3 (K)

12

Pre-factor for component 3

0.00051

Debye Components

Debye Temperature 1 (K)

(b)

Figure 4.6: (a) A screenshot of the MOF porosity prediction tool. (b) a screenshot of the heat capacity modelling tool described.

- Thermal conductivity prediction - A front end to a RF model of thermal conductivity of thermoelectric materials, which can predict thermal conductivity from a chemical formula [31].

- A tool for modelling the thermal conductivity of structures to extract Einstein and Debye temperatures. A linear combination of Einstein and and Debye functions are plotted against observed data. Users can adjust the temperatures and prefactors associated with each function in order to extract the relevant temperatures.

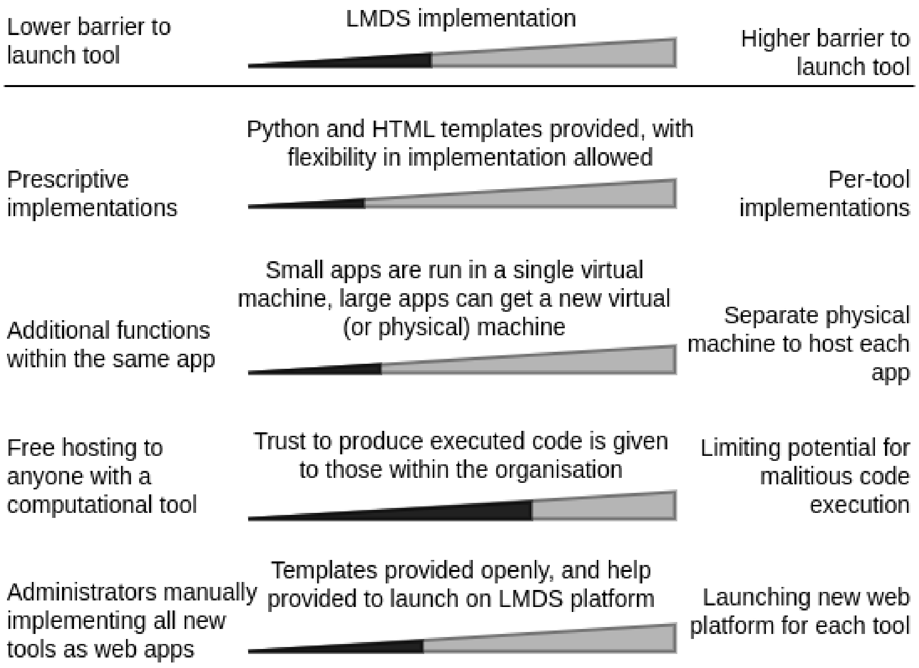

Lower barrier to launch tool
LMDS implementation
Higher barrier to launch tool
Prescriptive implementations
Python and HTML templates provided, with flexibility in implementation allowed
Per-tool implementations
Additional functions within the same app
Small apps are run in a single virtual machine, large apps can get a new virtual (or physical) machine
Separate physical machine to host each app
Free hosting to anyone with a computational tool
Trust to produce executed code is given to those within the organisation
Limiting potential for malitious code execution
Administrators manually implementing all new tools as web apps
Templates provided openly, and help provided to launch on LMDS platform
Launching new web platform for each tool

Figure 4.7: A non-exhaustive demonstration of how different computational issues may impact the ease of launching new tools to a web platform combined, with labels as to the decisions that were made on each of these issues when creating the LMDS.

4.2.3 Considerations when launching cloud platforms

Architectural Considerations

A key consideration in the LMDS was was the ease of extending the platform through publication of new tools. ML experts may have limited knowledge in computer networking, but frameworks and examples (such as those found in the associated code repositories [42]) can reduce the technical complexity of deploying new tools. New web frameworks are frequently released, and balancing the technical debt of learning how to interface and maintain each new library with networking and computational resource issues becomes non-trivial (fig. 4.7). This section outlines some of the technical design choices which were taken when designing the LMDS architecture.

One such consideration was the level of restriction on the range of technologies that technical users may employ in their development cycle. Constraining developers to certain libraries enforces a greater degree of homogeneity in a codebase, allowing for a larger quantity of each application's code to be reused and reducing the work required to deploy new tools. However, restricting to specific libraries may impose limits on newer approaches, which could be deployed, or may simply not align with developers' personal preferences. Each of the provided tools is written in Python (when applicable) owing to its popularity, but otherwise collaborators are not constrained by which external libraries they may use. Were researchers to propose tools based in other programming languages or frameworks, existing HTML and styling are provided for reuse.

The degree of separation between applications in a cloud platform is a notable design decision. Malfunction or security compromise of one tool should not effect other tools on the platform. In the past, applications were isolated their own physical server, in a process referred to as server segmentation. Current best practise is to isolate applications virtually for efficiency purposes.

One such method to separate applications from each other is virtualisation, which allows multiple virtual "machines" (VMs) to operate on a single physical machine through the use of hardware emulation. A host operating system runs a virtualisation program (hypervisor) which manages the computational resources of each VM. Each VM runs its own operating system, which may be selected depending on the task at hand, with Linux distributions often chosen for web applications. Virtualisation allows for dynamic scheduling of resources, while ensuring that a single application's malfunction does not affect other tools on the platform. VMs are easy to deploy, and updating VMs remains similar to updating to physical machines, although VMs are accessed through a hypervisor. Each VM does carry some overhead, as each operating system needs to store its own data in memory for each application, but the capacity of modern systems means this is generally not a concern.

Where the capacity of a system is a concern, containerisation provides a similar method of isolating applications from one another, with Docker [118] and Kubernetes [92] being two popular tools for this. Containerisation comes with a lower computational overhead than VMs (in particular with regards to memory consumption) [197, 103]. However this becomes yet another technology for developers to learn when deploying new tools. As a low barrier to entry for new application deployment is a key goal for the LMDS, containerisation technology was not used.

To balance the increased memory cost of VMs over containers such as docker and the need for the isolation of apps, the LMDS hosts smaller applications (for example, the MOF prediction tool and heat capacity modelling tool) on a single VM. Meanwhile, individual

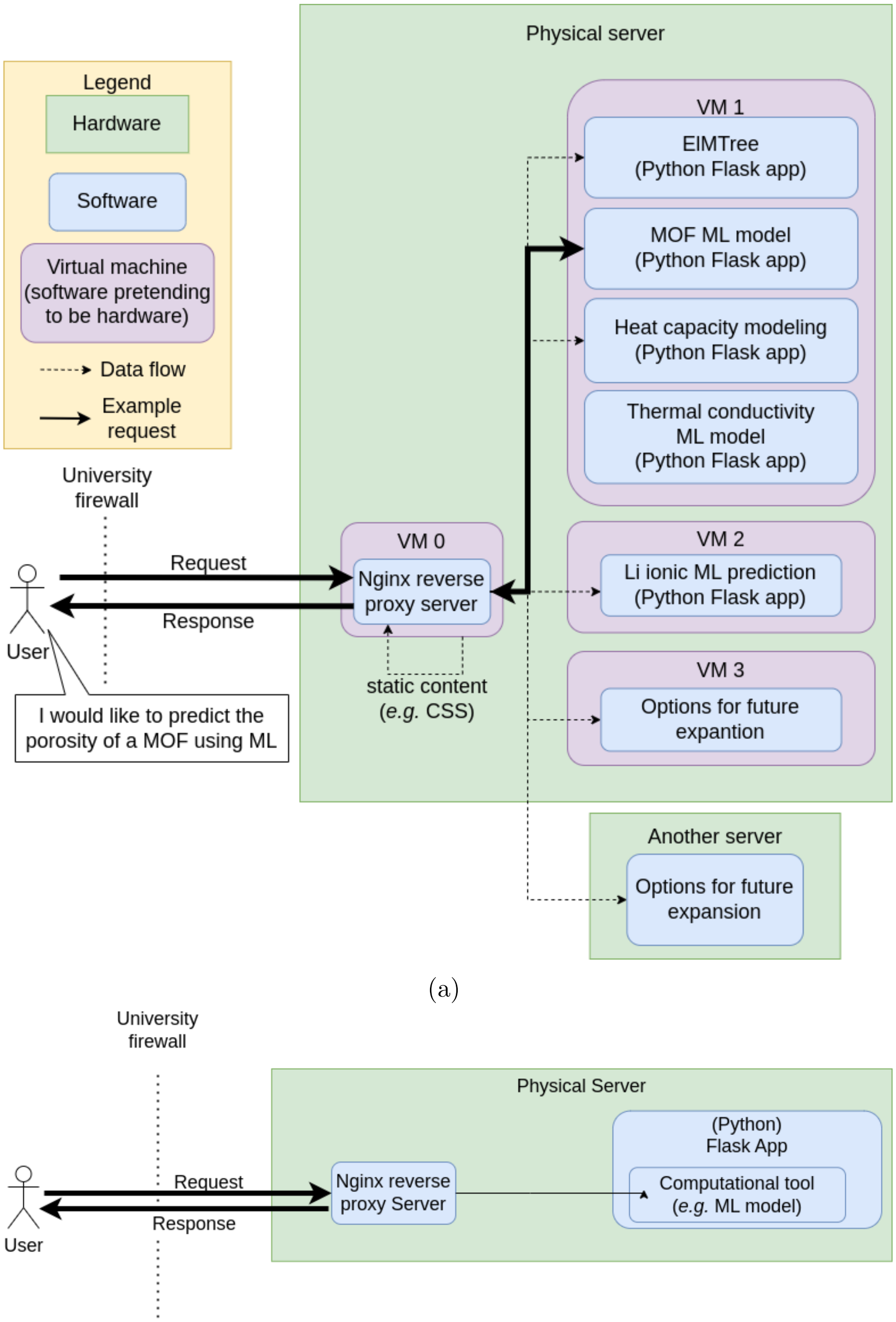
VMs are allocated to larger tools (such as ElMTree) to provide a level of isolation.

Managing these different VMs requires a reverse proxy server to direct requests with different web addresses to the correct tool without requiring a separate domain name or subdomain for each tool. To create this reverse proxy, HTML requests to each application are routed through a VM running Nginx [151], which enables each of the separate applications to be accessed through a single domain name. Internally, this server resolves each request to the internal IP address the specific application is hosted on. The reverse proxy provides some protection from direct denial of service (DDOS) attacks by enabling rate-limiting functionality. This Nginx server also encrypts HTTP traffic in HTTPS traffic [52] (fig. 4.8), which provides a security assertion to users that their data has not been seen or interfered with by any third parties. Apache HTTP server is an application historically popular for this task [173, 128], however, Nginx was selected due to its wide market adoption, strong performance [95], and simple configuration. Nginx can redirect requests to additional physical machines external to the hypervisor that the LMDS is currently hosted on, providing flexibility in future expansion (fig. 4.8a). Setup scripts have been provided to configure this proxy for new tools [42].

Research context and significance

As the user complexity and model complexity of ML projects grow, it is important that access to, and understanding of, ML tools do not become a barrier for their use. Cloud platforms such as the LMDS provide easier access to these tools, however it is on a per-practitioner basis to portray the understanding of best use and interpretation of such tools.

For example many ML models (such as that used in the thermal conductivity prediction model) take composition as input but have no mechanisms to check the chemical viability of such compositions. Without adequately communicating this limitation with the collaborators, the trust in such a model may be hindered, and the interpretation of its results will be incomplete. While explainability and uncertainty estimates in ML are active research areas [5, 105], a good understanding of the limits and correct usage of ML models by those who use them is also important. Drawing up interesting counter examples when presented with predictive models to demonstrate their limits is a valid method of testing the capabilities of models. This may not be a particularly useful test in many cases, as it is known that statistical models will underperform on chemical domains they have never been exposed to, and this may not be how the model should be used in practise. For


Legend
Hardware
Software
Virtual machine (software pretending to be hardware)
Data flow
Example request
University firewall
User
I would like to predict the porosity of a MOF using ML
Request
Response
Physical server
VM 0
Nginx reverse proxy server
static content (e.g. CSS)
VM 1
ElMTree (Python Flask app)
MOF ML model (Python Flask app)
Heat capacity modeling (Python Flask app)
Thermal conductivity ML model (Python Flask app)
VM 2
Li ionic ML prediction (Python Flask app)
VM 3
Options for future expantion
Another server
Options for future expansion
University firewall
User
Request
Response
Physical Server
Nginx reverse proxy Server
(Python) Flask App
Computational tool (e.g. ML model)

(a)

(b)

Figure 4.8: Architectures possible using provided tools (a) The architecture used in the LMDS (b) A simpler architecture using Nginx reverse proxy security certificates to serve a single AI model. This could be hosted on redundant hardware, such as an older workstation, and expanded to additional hardware when required.

example, the thermal conductivity model is trained only on thermoelectric materials; while it may be used to predict thermal conductivity for other materials, it may underperform outside the thermoelectric domain.

Similarly, it is a per-application question as to whether a tool will actually be useful in the materials discovery workflow. While the MOF porosity prediction tool may offer a variety of input options for flexible usage by experimental researchers, if no such researcher exists, the usefulness of such a tool is limited. As such, while cloud based tools do provide ease of access, it remains vital that collaborations between experimental and computational researchers involve open communication channels.

Cloud platforms are excellent suppliment to existing communications methods, and need not be prohibitively expensive to deploy. While ML models may be costly to train and require significant compute resources, after training is performed, the models can often be deployed using lower end hardware and still make inferences in a timely manner. Consequently, the LMDS platform with the architecture outlined above may be deployed on relatively inexpensive hardware. A minimum framework for the launch of platforms (fig. 4.8b) is provided, as well as how this was expanded for isolation of applications (fig. 4.8). The frameworks and implementation details provided here should provide a reasonable starting point for other researchers to share their tools with the wider community.

While the LMDS is hosted in house, the frameworks provided could be used for platforms deployed to commercial cloud providers, such as Amazon Web Services or Microsoft Azure. Third-party cloud providers alleviate concerns over server maintenance and hardware failure. However, each commercial cloud platform requires bespoke training to use, which may be a niche skill for computational materials science researchers. Concerns may be raised over vendor lock in, as such services may become more expensive or less reliable in the future. Further, the monthly billing cycle commercial cloud providers often demand is not compatible long term with the fixed consumable budgets that are typically provided as part of a research grant. Depending on funding and available hardware, a commercial cloud provider may be the best solution for rapid delivery. If the mission critical up time that is guaranteed by server hardware is not a driving design choice, then many computational research groups may find they already have the necessary resources to hand, as this architecture may be run on an underused workstation.

In house solutions, such as those presented here, may be deployed onto new or existing hardware, and tailored to suit the team's existing technical specialities, but this approach is not without disadvantages. Local hardware needs ongoing maintenance in the case of equipment failure, such as hard drives, which will have an associated cost. Networked applications operating under the framework of an institution will have to comply with the organisation's pre-existing networking and security protocols, especially if accessible from the public internet, which may introduce further tasks which must be satisfied to launch a new cloud tool. By working with the University of Liverpool's servers and storage team to test the architecture throughout the development process, it was ensured that the final product is robust and secure.

Releasing the LMDS as a simple technology stack with limited functionality means other research groups can extend this framework to rapidly prototype bespoke applications to suit their specific requirements. Cloud platforms enhance partnerships between computational and experimental research teams, and provide an additional interactive medium for accessing research.

However, monolithic platforms are by definition less integrated with experimental researchers than bespoke platforms. Enhancing the interface between technical developers and their expert users allows new tools to be integrated into materials discovery workflows. The process of constructing a new cloud platform requires technical expertise, but the barrier to entry is low enough that many computational researchers do, in fact, possess these skills. It is hoped that the tools provided here reduce this technical gap further to make the creation of cloud platforms simpler for others. Future undertakings could investigate methods of unifying multiple cloud platforms into singular portals or developing frameworks that require even less technical expertise to create new cloud applications. Centralised or monolithic systems risk excluding researchers who wish to share computational tools with the wider community if the technical or organisational processes to host such tools remain unclear.

Creating new methods to access computational tools through cloud platforms is one way to explore how computational methods may be adopted by experimental researchers. As computational methods continue to develop, so too will their place in the discovery of new materials. Future research may lead to more cloud platforms, new frameworks to ease the creation of such platforms, or focus on entirely novel collaborative techniques. While advancing the accuracy of the predictions made by ML models remains a dominant research area in this field, the concurrent development of tools which interface with these models is a crucial piece of supporting work to ensure wide and effective adoption.

Cloud platforms offer a compelling solution to lack of model accessibility, but it remains to be seen whether this accssibility will trainslate into model usage. Anecdotally tools presented here (and in the accomponaying publication [46]), are seen as helpful with the research group, however more thorough evaluation is required before this cloud platform can be considered successful. Such evaluation may include qualitative surveyance key stakeholders to the platform, quantitative investigations on telemetry taken by the platform, and/or quantitive investigations into subsequent research citing this platform. A survey of literature citing this platform could be performed, noting the ratio of papers which actually use the platform in order to make materials (rather than review papers or similar), though this method would introduce a publication bias. Overall evaluation of this platform should be encouraged but is beyond the scope of this thesis.

4.2.4 Methods

All bespoke programs presented here are implemented in Python. The MOF porosity prediction and thermal conductivity tools use random forests implemented in scikit learn [140]. MOF porosity prediction tool was featurised using mordred and RDKit [121, 150]. The Li-Ion conductivity tool uses CrabNet [186] to predict the conductivities of compositions. ElMTree uses the ElMD [66] library and a simple implementation of the list of clusters. ElM2D uses the aforementioned ElMTree application with the UMAP [116] and plotly [143] libraries. The modeling of heat capacity was carried forward using the SciPy library [183]. Each of the web applications are implemented using Flask, with the gunicorn process manager used to spawn Flask processes, examples of these implementations have been provided [164, 47]. VMs are run through VMWare vSphere ESXi 7 [185], with Nginx used to serve HTTP responses to users and route URLs to each VM [151]. Example setup scripts for gunicorn and Nginx can are available [42]. The linux utility crontab is used to to ensure regular updates are executed and to remove temporary files in relevant LMDS applications.

4.2.5 Conclusions

This section presented the LMDS, a cloud platform for experimental researchers to use in discovering new materials, available at lmds.liverpool.ac.uk. The LMDS platform allows for easy access to previously published computational models [142], as well as novel tools to help experimental researchers.

Making computational tools easily accessible is crucial to maximise their benefit. Thus, the LMDS platform was developed with the objective of simplifying the sharing of computational tools, ensuring that they are readily available to researchers with minimal computational expertise. Considerations that lead to the production of this platform are provided and key design considerations are justified.

The difficulties in applying computational methods in experimental research are discussed, as are the barriers for deployment of such methods to cloud platforms. Examples of the tools discussed in this manuscript have been provided [164, 47], as well as scripts setting these up with an Nginx reverse proxy server, and Python process manager [42], enabling other researchers to reproduce this tool chain and share their own methods, either internally or on the open internet.

Minimising organisational overhead in collaborations between computational and experimental researchers promotes the incorporation of computational methods in the synthesis of new materials. Access to state of the art computational methods, such as in ways presented here, accelerates research and improves the prediction, analysis, and realisation of new materials.

4.3 Discussion

Two methods for collaboration were seen in this chapter. Collaborations discussed with regards to the discovery of superconductors (Section 4.1) were very active, and top down collaborations. That is, multiple meetings and discussions were held, and there was a lot of direct involvement between experimental chemists and the computational work presented. Work was closely overseen, and because of this oversight an array of experts in experimental chemistry and physics were able to give invaluable input. This also presented drawbacks: work was very directed and also quite broad (as seen from the wide array of areas of interest in Section 4.1.6), these two things in combination were not very conducive to collaboration.

The LMDS platform offers two distinct collaborations. Firstly, collaboration is done with computational based researchers, offering a platform and method for easier sharing of tools. Secondly, collaboration is done between the creators of tools on the platform and anyone who chooses to use those tools.

Collaboration with computational based researchers was very organic. Research topics relating to computational tools and collaboration are similar to this thesis. As such, a culture of collaboration had already been established. This was still an active process and each tool presented required a lot of communication.

The sharing of these tools though is somewhat more passive. Once a tool is available on a cloud platform such as the LMDS, it can be accessed by collaborators with very little action required on behalf of that tools creator. This has both benefits and drawbacks as discussed (Section 4.2.3). While cloud platforms may increase the reach of research and ML models, they do not bring purpose to research that has no clear use.

4.4 Conclusion

The two collaboration methods presented here are not in competition with each other. This chapter has presented two methods that were used, and presented findings related to each of them.

Workflows were presented for the discovery of new superconductors. The literature on predicting superconductors was reviewed, with methods such as garbage in being found to be beneficial. The CBFVs used in existing work were not found to improve models and as such a composition vector was used to as a representation for materials. 12 RFs were trained and over a billion candidates were screened.

A cloud platform, the LMDS was developed, presenting 6 computational tools for use by experimental chemists. Considerations were given as to it's development, as well as scripts and architectures to encourage research groups to produce similar cloud platforms.

Collaboration is key to achievement in academia. The aggregation of knowledge in institutions instrumental to discovery across all fields. In applied ML this is no different. This chapter has discussed cooperation at length, and it is hoped that the works presented are able to help maximise the value of the scientific community.

4.5 Thesis context

This chapter explores the application of ML in the context of materials discovery, emphasising the practical considerations involved. This is both in terms of working directly with researchers to establish suitable candidates for synthesis, and making tools available researchers to empower them to screen materials directly. Understanding the realities of this process is crucial for contextualising the subsequent work.

While the upcoming chapters may diverge from materials discovery workflows, the processes and conclusions discussed in this chapter remain relevant. As a result, this chapter serves as an anchor, bridging the practical aspects of material creation with the theoretical aspects of improving and exploring ML algorithms that will be explored in the following chapters.

Chapter 5

Random projections and leave one cluster out cross validations: improving evaluations of machine learning for materials properties.

5.1 Introduction

The previous chapter focused on ways to improve collaborations with synthetic chemists, but for those chemists to want to spend time making materials which have been predicted by machine learning (ML) models, those models must be well justified and well communicated. This chapter examines justification and communication of supervised ML models with respect to two aspects, material representation, and the validity of leave one cluster out cross validation (LOCO-CV) measurements.

With the plethora of available ML models available (and the fast moving nature of the field), ML models should be justified not just on their own performance, but in comparison to competing methods. A key choice when building a ML model for predicting material properties is the choice of representation for a material. The boom in supervised machine learning (ML) research in materials science has seen a large number of possible representations for compositions suggested for use with ML algorithms (as discussed in Section 2.2.5). Representations are often tested with different ML models, different implementations of those ML models, and on different datasets. Many libraries for creating representations of materials exist [194, 32, 27] (this process is also called featurisation). Thus, composition based feature vectors (CBFVs) generated using these libraries are common. There have been some recent comparisons between these representations (some of which the chapter aims to present) [124, 45], but it is still unclear that any particular CBFV is better than alternative representations like $CompVec$ (see Sections 2.2.5 and 3.2). Because of this gap in the literature, it is pertinent not only to evaluate the efficacy of CBFVs, but also to discuss methods and best practice with which to evaluate CBFVs.

Evaluation of successful models is a problem in ML research generally [148, 110]. In materials informatics, a lack of applicable measurements can lead to scepticism from chemists when time comes to synthesise predicted materials (as discussed Chapter 4). Other ML research areas have domain specific metrics (such as the Bi-lingual evaluation understudy score in machine translation [135]) and domain specific metrics that are suited to materials science have already been discussed and proposed in this thesis (Section 3.1.2).

Similarly many domains have specific, commonly used evaluation datasets (such as the ImageNet dataset in image recognition [36]). Owing to the large range of tasks undertaken in materials science and the relative infancy of ML research in this field, such benchmarks are still being established [30, 38], though the last few years have seen an increase in their usage. This chapter explores possible benchmark tasks outside of those which are being established as the norm. Firstly, because some of the experiments done here predate the publication of such benchmarks and secondly, because the compilation of alternative benchmark tasks can be seen as a positive contribution to the community. One evaluation tool which has been proposed in materials science is LOCO-CV [117] (introduced in Section 2.1.6). Data in materials science tends to be clustered around known families of heavily studied materials, and LOCO-CV has been suggested as a technique to compare the outputs of competing ML models by measuring extrapolatory power on unseen clusters of data. LOCO-CV uses K-means clustering to exclude similar families of materials from the training set to measure the extrapolatory power of an ML algorithm (its ability to predict the performance of materials with chemistries qualitatively different from the training set). The value of such an approach can be seen in the case of predicting new classes of superconductors. One may choose to remove cuprate superconductors from the training set, and if an ML model can then successfully predict the existence of cuprate superconductors without prior knowledge of them, we can conclude that that model is likely to perform better at predicting new classes of superconductors than a model which could

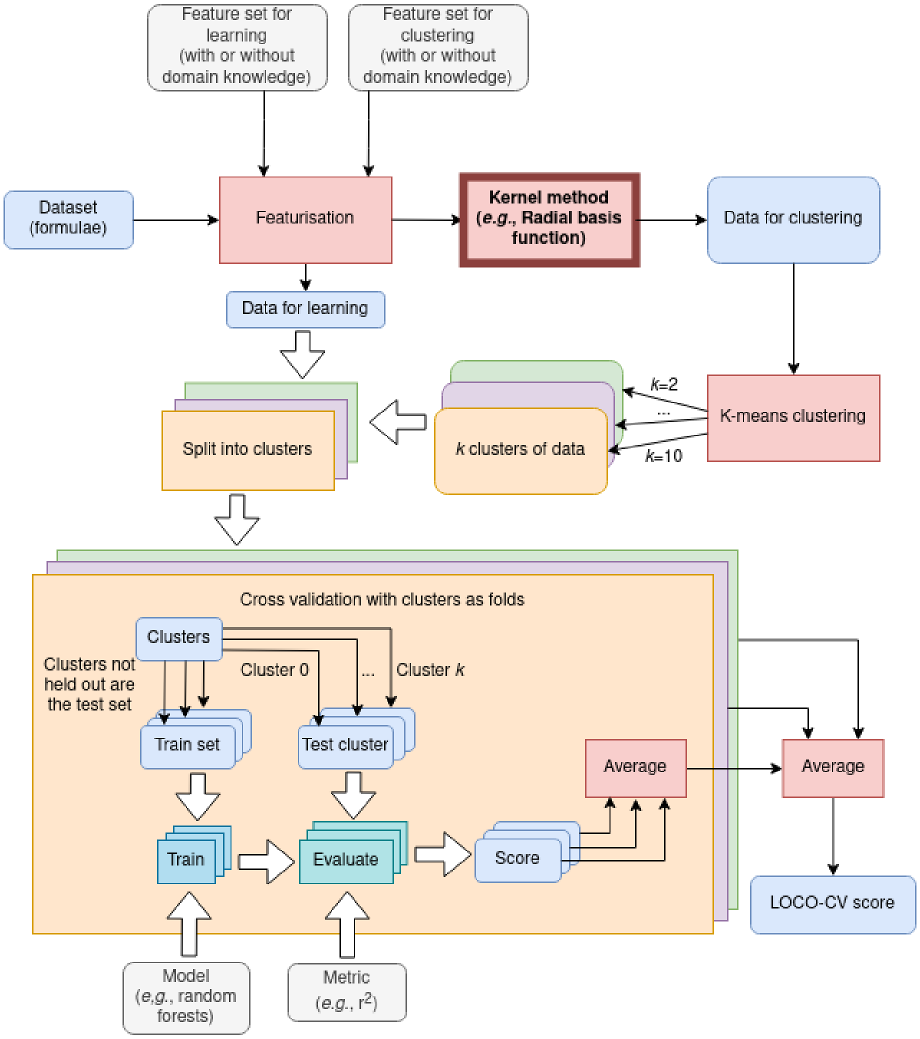

Feature set for learning (with or without domain knowledge)
Feature set for clustering (with or without domain knowledge)
Dataset (formulae)
Featurisation
Kernel method (e.g., Radial basis function)
Data for clustering
Data for learning
K-means clustering
k=2
...
k=10
k clusters of data
Split into clusters
Cross validation with clusters as folds
Clusters
Clusters not held out are the test set
Cluster 0
...
Cluster k
Train set
Test cluster
Average
Average
Train
Evaluate
Score
LOCO-CV score
Model (e,g., random forests)
Metric (e.g., r2)

Figure 5.1: A flow chart of the kernelised LOCO-CV process in a property prediction task. The novel kernel application is highlighted in a bold frame. Note that representation used for clustering is independent of that used for training the models. Consequently, kernel methods can be easily integrated into existing property prediction workflows without changes to how models are trained

not predict the existence of cuprate superconductors. LOCO-CV provides an algorithmic framework to measure the performance of models on predicting new classes of materials by defining these classes as clusters found by the K-means clustering algorithm.

Clusterings are selected using the K-means clustering algorithm [108, 171], which infers K clusters without the need for target labels (introduced in Section 2.1.3). This is done by grouping data into clusters based on their Euclidean distance to K randomly chosen "centroids". The centroids are then redefined as the mean of all points in a cluster and the data are regrouped based on these new centroids. This process is repeated until the positions of centroids (or the contents of their associated clusters) converge. K-means is quick, robust and readily implemented [140].

LOCO-CV as explored here uses K-means clustering with values of K between 2 and 10 (inclusive), taking the mean of the resulting metrics. This is the version of LOCO-CV most thoroughly explored by the authors of LOCO-CV (though they use the median rather than the mean). However, alternative methods of selecting a single value of K were suggested in that work. Namely alternatives suggested were use of X-means [141], G-means [63], or silhouette factor threshold [162] for selection of K.

Known problems with applying LOCO-CV include, non-determinism, uneven cluster sizes (as discussed in Section 3.1.5), and measured performance being highly dependant on choice of hyperparameters, which can make LOCO-CV measurements unreliable. These problems make LOCO-CV hard to be seen as valid, hard to reproduce, and hard to compare, respectively. This chapter aims to address these drawbacks of LOCO-CV, and demonstrate how the use of kernel approximation methods and varying representations of compositions can be used to improve linear separability within materials data sets and improve the general applicability of LOCO-CV in this domain.

It is unclear how the non-deterministic nature of LOCO-CV will affect the repeatability of measurements taken using this evaluation method. This chapter (Section 5.4) details experiments performed to test how repeatable LOCO-CV is, finding that while it is less repeatable than using an 80:20 train:test split to evaluate a random forest, it is the deviation between measurements made were not sufficient to substantially impact the interpretation of the results seen in this paper.

A further consideration in use of LOCO-CV is that K-means does not guarantee the size of any clusters, nor does it guarantee that clusters would be deemed chemically sensible (this is discussed further in Section 5.3.1). It has been observed that clusters taken on materials data can vary in size by multiple orders of magnitude, which hinders the application of LOCO-CV [125]. While, sizes of clusters are expected to differ in this domain (for example due to research bias in the generation of example materials), should the sizes of the clusters found in LOCO-CV differ by orders of magnitude then LOCO-CV's ability to measure extrapolatory power is hampered. Intuitively, if one of ten clusters contains 90% of the materials in the dataset, then a measurement made with this cluster left out may give a measurement of algorithmic performance given a small fraction of the available training data, rather than indicating extrapolatory power. K-means clustering by its nature can only linearly separate clusters in a given data space. Clusters that are more distinct from one another are more likely to be isolated than clusters of data points that overlap with each other. There are other clustering algorithms, such as agglomerative clustering [111] or DBSCAN [49], that could be explored for LOCO-CV applications on materials datasets. This study measures the separability of clusters of compounds in materials science datasets with K-means clustering.

One of the hyperparameters on which LOCO-CV depends is the representation of materials. Data representation can play a major role in the performance of ML algorithms; however, the optimum choice of representation is not always apparent. In materials science, it is often difficult to choose an appropriate representation due to variability in the ML task and in the nature of the chemistry, composition and structures of the materials studied. Additionally, some properties of a material, such as its crystal structure in the case of crystalline materials, may not be known until its synthesis. Accordingly, many studies derive representations from either the ratios of elements in the chemical composition, or from domain knowledge- based properties (referred to as features) of these elements, or both, in a process called "featurisation."

Given the ubiquity of featurisation methods in materials applications, it is important to evaluate the statistical advantage of specific feature sets [124]. Section 5.2.1 overviews different featurisation techniques and how their effectiveness has been previously reported. This evaluation is expanded upon in Section 5.2.

Seven representations are investigated across five case studies from the literature to explore how these representations perform in published ML tasks. These cases thus represent practical applications, rather than constructed tasks. Each of these representations is also compared to a random projection of equal size to establish the performance benefit of domain knowledge over random noise. LOCO-CV measurements is also taken for these experiments in the appendix to this chapter.

In Section 5.3 the effect of representation on measurements made with LOCO-CV is investigated. Case study datasets will then be used to evidence methods to adjust LOCO-CV to make cluster sizes more even. Kernel approximation methods, can be used to non-linearly translate data into a data space that can then be linearly separated (Figure 5.4). Such kernel approximation methods can serve as an a priori alternative to replacing euclidean distance with kernel tricks in the application of an algorithm. Kernel approximation methods, such as radial basis function (RBF) approximation, were applied to chemical datasets to improve the linear separability of data and reduce variance between cluster sizes and thus increase the validity of LOCO-CV measurements (Figures 5.6 and 5.9), thus enhancing the assessment of performance found when using different representations as well as assessment of model performance as a whole.

Experiments are then carried out to establish whether the two techniques of random projections and kernelised LOCO-CV can be used together. These find no specific advantage in using random projections with LOCO-CV, but random projections were not universally worse than other CBFVs either. These techniques can be used independently or together depending on a researcher's needs.

Having noted reproducibility concerns due to the non-deterministic nature of LOCO-CV, an investigation into the repeatability of LOCO-CV is done. A subset of the experiments presented are repeated five times, and standard deviations in results found are noted. Results are seen to be reliable.

Finally, the findings and implications of this chapter are discussed. Conclusions are drawn, use cases are suggested, and the motivations behind finding representations for materials are questioned. The methods investigated in this chapter help to improve the applicability of LOCO-CV, and help to justify choice of representation of composition for ML models.

LOCO-CV evaluation is affected by representation of a compound and, conversely, choice of compound representation is affected by the methods used to evaluate these representations. Thus, it is pertinent to investigate these two issues simultaneously even though the representation used in clustering does not need to be the same as that used to train the model (Figure 5.1). Utility of LOCO-CV measurements was improved by using kernel approximation methods to create a more separable data space. These measurements were used to evaluate featurisation methods using practical supervised ML tasks found in the literature. The key contributions and findings of this chapter are as follows:

- Comparing the influence of composition based feature vectors (CBFVs) on ML model performance in practical tasks (explained further in Section 5.2.1, before being carried out in section Section 5.2). It was found that CBFVs with engineered features (*i.e.*, imbued with domain knowledge) do see some benefit in certain tasks, particularly band gap prediction tasks. While *magpie* representations [194] were seen to outperform other CBFVs in many tasks, this finding was not universal across tasks.

- Examining the effectiveness of random projections as featurisation methods for property prediction from chemical composition. Random projections can be used as a baseline against which to justify more involved featurisation methods (explained further in Section 5.2.1 before being carried out in Section 5.2). It was found that in many tasks, CBFVs with engineered features do not perform substantially better than random projections.

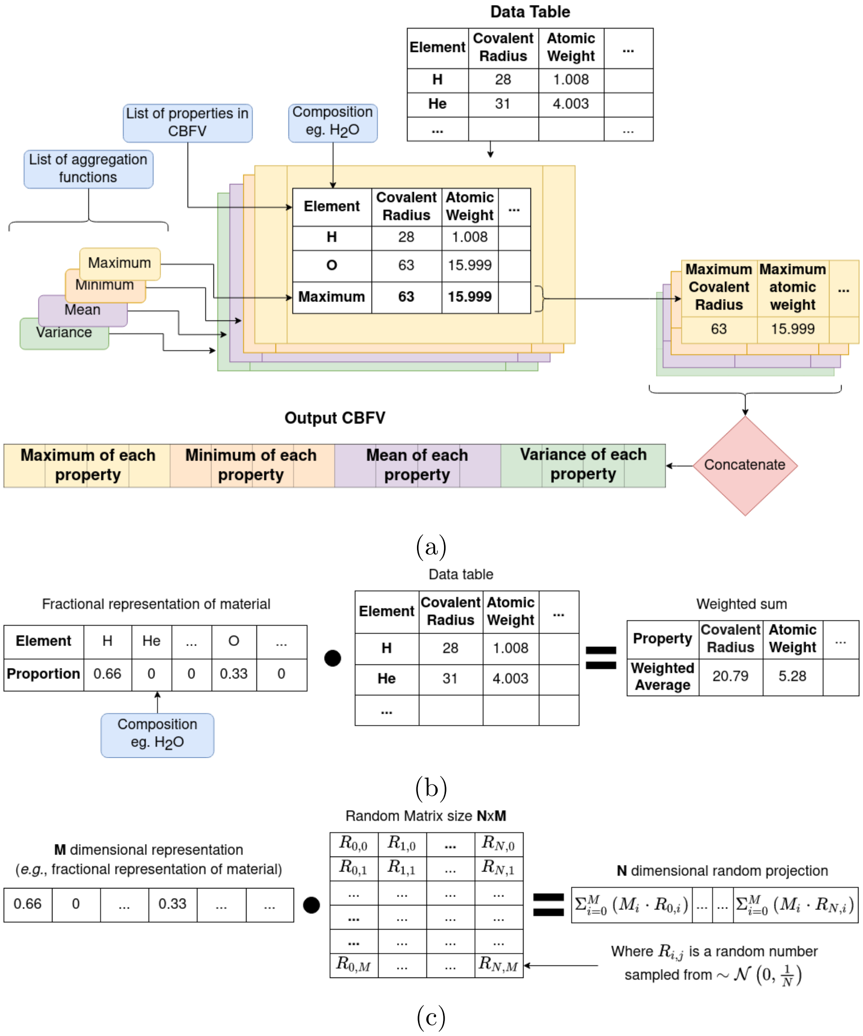
- Studying the effect of kernel approximation functions (explained further in Section 5.3) on the application of K-means clustering to materials data, and presenting a workflow to incorporate these methods into the LOCO-CV algorithm (Section 5.3). It was found that kernel approximation functions are a good way to reduce the variance between sizes of clusters found by K-means clustering on materials data. Using kernel approximation functions in the suggested workflow (kernelised LOCO-CV) results in a more robust evaluation method than LOCO-CV with no kernels.

- It can be recommended to use radial basis function (RBF) approximation when clustering for LOCO-CV, as clusterings found after application of RBF are seen to be more even in size than with no kernel method applied, and models are trained more reliably for property prediction. This helps to reduce the risk that performance differences on predicting an unseen cluster of data are caused by the training set size as opposed to the intrinsic inability of a model to perform well on that cluster of data.

- It was found that the use of RBF approximation in clustering for LOCO-CV leads to more reliable and consistent model training, compared to using LOCO-CV without any kernel approximation methods.

- Use of random projections as a baseline against which to compare engineered feature vectors is recommended. It is noted that commonly used CBFVs have little to no advantage over random projections in most tasks investigated.

- The use of random projections as a featurisation method for clustering compositions in LOCO-CV was investigated, finding that random projections have no clear advantage over other CBFVs tested here.


Data Table
List of properties in CBFV
Composition eg. H_2O
List of aggregation functions
Maximum
Minimum
Mean
Variance
Concatenate
Output CBFV
(a)
Data table
Fractional representation of material
Composition eg. H_2O
Weighted sum
(b)
Random Matrix size **N**x**M**
M dimensional representation
(*e.g.*, fractional representation of material)
N dimensional random projection
Where $R_{i,j}$ is a random number sampled from $\sim \mathcal{N}\left(0, \frac{1}{N}\right)$
(c)

Data Table

Element	Covalent Radius	Atomic Weight	...
H	28	1.008	
He	31	4.003	
...			...

Element	Covalent Radius	Atomic Weight	...
H	28	1.008	
O	63	15.999	
Maximum	63	15.999	

Maximum Covalent Radius	Maximum atomic weight	...
63	15.999	

Output CBFV

Maximum of each property	Minimum of each property	Mean of each property	Variance of each property

(a)

Fractional representation of material

Element	H	He	...	O	...
Proportion	0.66	0	0	0.33	0

●

Data table

Element	Covalent Radius	Atomic Weight	...
H	28	1.008	
He	31	4.003	
...			

=

Weighted sum

Property	Covalent Radius	Atomic Weight	...
Weighted Average	20.79	5.28	

(b)

M dimensional representation (*e.g.*, fractional representation of material)

0.66	0	...	0.33	...	...

●

Random Matrix size **N**x**M**

$R_{0,0}$	$R_{1,0}$	...	$R_{N,0}$
$R_{0,1}$	$R_{1,1}$	...	$R_{N,1}$
...	...	...	...
...	...	...	...
...	...	...	...
$R_{0,M}$	...	...	$R_{N,M}$

=

N dimensional random projection

$\Sigma_{i=0}^{M}\left(M_i \cdot R_{0,i}\right)$	...	...	$\Sigma_{i=0}^{M}\left(M_i \cdot R_{N,i}\right)$

Where $R_{i,j}$ is a random number sampled from $\sim \mathcal{N}\left(0, \frac{1}{N}\right)$

(c)

Figure 5.2: Comparison of the creation of composition based feature vectors (CBFVs) and random projections. (a) General workflow for creation of CBFV. Application of aggregation function to each property of a material will result in a fixed sized vector for each aggregation function, these are then concatenated together (merged sequentially) to form the final CBFV. Both the properties in the CBFV and the list of aggregation functions can be changed to create variants of CBFVs, which may influence algorithms that use the resulting CBFV. (b) Calculation of the weighted sum of properties of a material. This is equivalent to the matrix multiplication of the fractional representation of that material and its properties. (c) Calculation of a random projection. Using random projection to (approximately) linearly project a representation into a different number of dimensions (N). The original M dimensional representation for our purposes may be a fractional representation for the chemical composition of a material, but this technique can be used for any input data, in domains outside of chemistry.

5.2 Effect of representation on predictive ability of random forest: Case Studies

ML algorithms require a consistent definition of a data point in order to analyse trends within a dataset. For example, it would be hard to learn from a dataset in which “a data point” may refer to a phase field, a specific crystal structure, or a composition. One such algorithm is RFs, which are widely used in materials science as well as other domains[19]. They are fast to train, readily implemented [140], and see a good performance in a plethora of tasks without hyperparameter tuning. Experiments in this section will use RFs for reasons outlined above, however, good evaluation methods for fixed dimensional representations of materials are also important for the plethora of other ML algorithms that use such representations as basis for predictions. This section investigates 7 different representations across 10 different ML tasks seen in the literature. These representations are compared to random projections, a dimensionality reduction technique from computer science.

Representations, issues surrounding these representations, and random projections will all be introduced. Following this results will be discussed of all ML tasks (Section 5.2.3). The results are split by the study from which the tasks were taken and a summary of the themes seen in the results will be noted. Discussion of the implications of these results will be done in the discussion for the whole chapter (Section 5.6).

5.2.1 Composition based representations for materials property prediction.

Representation learning, and feature engineering are the two main preprocessing methods to make data more interpretable to ML algorithms. Representation learning is a fast-evolving field that uses deep learning in order to create representations, while feature engineering involves defining a set of features (or descriptors) for a data point that adequately encapsulates all information needed [13].

Representation learning (the learning of the best representation by ML rather than through experiment) is a highly active research area, with implications for transfer learning which could prove useful for materials science, where datasets are limited in size. As such, notable investigations into using deep representation learning has been done in the form of Elemnet [79], CRABnet [186], and Roost [60] which use deep fully connected neural networks, transformer networks, and graph neural networks, respectively, to learn properties from a one-hot style encoding of composition. Learnt representation is an exciting area growth area of the literature, but is not the focus of this section.

As explored (Section 2.2.5), feature engineering has been used extensively in inorganic chemistry and materials science. However, no set of features has emerged as the clearly dominant representation for a material, likely due to the variety of tasks carried out in these domains, which may require different input representations. Many of these representations use only composition-based information, as this allows screening of materials without the need for DFT calculations or synthesis, greatly reducing the costs associated with such screenings. Composition-based screening is less powerful than the incorporation of structure, as both structure and composition control properties, but more general as structural information is not required and is less widely available than composition (as structure is not known until the material is realised by synthesis, whereas compositions can be proposed without knowing structure). Composition-based feature vectors (CBFVs), which offer a list of compositional attributes of a material, and a one-hot style (also called fractional) encoding of composition [79], are widely used composition-based representations. Investigations in this chapter will focus on composition-based representation. Composition-based representations which are widely used in the literature, as will be seen from the case studies explored in this section.

Notable CBFVs including *magpie*, *Oliynyk*, and *JARVIS* [192, 131, 27] (differences between which are discussed further in Section 5.2.2) were recently investigated and found to provide benefit over one-hot style representations. This benefit was measured using neural networks predicting numerous properties, however the benefit became little to none as the dataset size increased above 1000 points [124].

This chapter furthers the investigation into the use of CBFVs by examining their applicability in five case studies. Namely, performance of examined using *Oliynyk*, *magpie*, and *JARVIS*, a variant of random projection of size 200 (discussed more in Section 5.2.1) used in a previous review on this topic [124], as well as one-hot style encodings of composition, and random linear projection of the composition. The performance of RFs using different representations are compared on ML tasks found in the literature, using the relevant datasets for each study [170, 102, 195, 35, 84].

The representations were chosen as they are commonly used, and as these are the non-structural representations investigated for their efficacy in neural networks in previous work [124]. Seeing whether previous results hold for RFs should help gauge whether these results could be used as a rule of thumb for many ML algorithms or whether these conclusions should only be applied to neural networks similar to those used in that study.

Can implementation details in CBFVs affect performance

It is common for a CBFV to be comprised of a list of elemental properties that are combined using several "aggregation functions", for example the weighted average, and standard deviation of various elemental properties in a compound (Figure 5.2a). The aggregation functions of a CBFV can vary between implementations [192, 124]. Using different numbers of aggregation functions results in representations of different lengths (Figure 5.2a), which may affect ML performance depending on the algorithm being used.

Problems associated with building statistical models using increasingly large data representations without also increasing the number of data points are well documented, often being described as the curse of dimensionality [12]. Strong correlation between different dimensions (known as co-linearity, or cross correlation between dimensions) can also impact model performance. For example, RFs are affected by co-linearity between dimensions as RF's random bagging process is unlikely to select a subset of features that include none of a set of cross corelated features. This would make the information in features with such cross-corelates more likely to be available to discriminate with at any branch in a tree, compared with those features without such cross-corelates. It is intuitive that different aggregation functions may be cross-correlated, for example the maximum atomic weight of an element in a compound is likely to correlate with the average atomic weight of an element in that compound, thus RFs may be affected by additional aggregation functions.

Without investigation, it is unclear what effect different aggregation functions will have on algorithm performance. Interrogation of the repository associated with the previous review of featurisation methods indicates use of the weighted average, sum, range, and variance of each feature [124]. This includes the features of the fractional (one-hot style) representation, which uses only the ratios of each element in a material in its definition. This implementation difference could affect the performance of a model that uses these representations, so we distinguish between the two, using "fractional" to refer to a one-hot style encoding that includes the average, sum, range, and variance of each element and "CompVec" (for composition vector) to refer to an implementation of one-hot style

Aggregation function	Na	Cl	All other columns
weighted average	0.5	0.5	0
sum	1	1	0
range	1	1	0
variance	0.0042	0.0042	0

Table 5.1: Values that would occur in each column across different aggregation functions for a composition fractional representation of NaCl. This demonstrates how the inclusion of additional aggregation functions does not add additional information for this representation. These calculations assume a representation which allows for 118 different elements, a smaller number of represented elements would result in the values in the variance columns being larger.

encoding which contains just the ratios of elements in a compound.

The nature of the fractional representation means that a given compound would contain the same representation three times, scaled by different amounts (depending on the number of elements in the compound) in a single vector (four times if elements in a compound are in equal ratios). This can be exemplified by examining a simple composition such as NaCl (Table 5.1).

This offers an opportunity to investigate how increasing dimensionality (the number of dimensions) of a representation while adding no new information affects performance. We leave the investigation of the effect of information added by different aggregation functions on different feature sets to future work. We experiment using both a (*CompVec*) one-hot style encoding as proposed for use with ElemNet [79] (with no additional aggregation functions), and the one-hot style approach used previously that includes different aggregation functions (*fractional*) [124], to see how this increase in dimensionality above will affect experiments.

While this increase in dimensionality will be seen to affect the clusterings found with K-means clusterings, for most tasks investigated there was not an appreciable difference between CompVec and fractional representations. In band gap prediction tasks fractional representation outperformed CompVec; however, in regression tasks relating to bulk metallic glass formation this trend was reversed (Figure 5.3).

Random Vectors as featurisation methods

Each elemental property (for example covalent radius) aims to bring with it some sort of information about that element. That property's inclusion in a feature set aims to improve an ML algorithm's performance in a given problem. Every feature included either means an increase to the dimensionality of a CBFV or the exclusion of an alternative feature. Though the importance of a feature to an ML model can be measured [127, 3], it is hard to take such measures of feature importance out of the context of the model that is trained with it, or the dataset that the model is derived from [97].

As it is hard to distinguish the effects of dimensionality of a representation from the effects of the information imbued in it, Murdock *et al.* introduce a set of vectors, one for each element each consisting of 200 random numbers to represent nonsensical elemental properties. From these vectors, they derive the CBFV *RANDOM_200* to represent a lower bound for feature performance. That is to say; rather than using features that would be expected to give information about an element (covalent radius, atomic number etc.), they instead assign each element a vector of random numbers. If these random numbers can result in a well-performing model then whether the chemically-derived features that are commonplace in the literature are justified can be called into question. When the aggregation function is a weighted sum (discussed further in Section 5.2.1), this has the same effect as a matrix multiplication of the one-hot style encoding of a compounds formulae, $\boldsymbol{C}$, (referred to in this paper as *CompVec*), and a random matrix, $\boldsymbol{R}$ which can be noted as $\boldsymbol{C} \cdot \boldsymbol{R}$ (Figure 5.2b). Thus, the weighted sum part of the *RANDOM_200* can be seen as a matrix multiplication of the random vectors and the fractional encoding of the composition.

This matrix multiplication is similar to that used in a random projection. Random projection is a dimensionality reduction technique that uses the observation that in high dimensions random vectors approach orthogonality [154, 83]. When the columns of $\boldsymbol{R}$ are normalised to be unit vectors, $\boldsymbol{C} \cdot \boldsymbol{R}$ becomes an approximately linear projection of $\boldsymbol{C}$. Another way to closely approximate normalisation of the columns of a random matrix, such as $\boldsymbol{R}$, is to sample the values of that matrix from a Gaussian distribution of mean 0 and variance $\frac{1}{N}$ $(\sim \mathcal{N}\left(0, \frac{1}{N}\right))$ where N is the size of the projection. This is mathematically justified by the Johnson-Lindenstrauss lemma, which states that for a set of N dimensional data points there exists a linear mapping that will embed these points into an n dimensional data space while preserving distances between data points within some error value, ϵ. This value of ϵ is shown to decrease as n increases [33]

$RANDOM\_200$ samples from $\sim \mathcal{N}(0,1)$ also included aggregation functions (namely sum, range, and variance) [124], as discussed in Section 5.2.1. It is unclear what impact this will have; however, preliminary investigations show little difference in performance between sampling from $\sim \mathcal{N}(0,1)$ and $\sim \mathcal{N}\left(0, \frac{1}{N}\right)$.

Use of random projection as an alternative to more widely used techniques is investigated by comparing each representation to a random projection of the same size (Figure 5.3). This should allows observations of improvements made by the quality of features as opposed to the quantity. *RANDOM_200* is included in this investigation. Notice the key difference between *RANDOM_200* and the random projection being that the random numbers are drawn from different distributions (as outlined above) and that *RANDOM_200* includes aggregation functions, where a random projection does not.

5.2.2 Tasks and representations investigated

This section examines ten ML tasks across five case study publications' datasets to compare the representations used in them with a non-structural CBFV examined in previous work [124], and with the composition vector ($CompVec$) suggested for use with ElemNet [79] (introduced in Section 2.2.5). Case studies have been selected to incorporate the prediction of a variety of material properties, research groups, and notable works that reflect the state-of-the-art. The original datasets are used to replicate studies, but with an 80:20 train:test split.

A consistent 80:20 train:test split is used across all datasets to enable conclusions to be drawn about which representations work better generally. This should help to establish whether previous findings (*i.e.*, that domain knowledge is more beneficial in smaller datasets and that benefit diminishes as dataset size increases over 1000) [124], hold true for RFs. LOCO-CV measurements for these experiments are are available and will be discussed in Section 5.2.3, and the clusterings found for LOCO-CV are available in the associated git repository [40].

Representations compared are:

- *Oliynyk* [131]. Originally designed for prediction of Heusler structured intermetallics [131], the *Oliynyk* feature set as implemented in previous work includes 44 features [124]. For each of these, the weighted mean, sum, range, and variance of that feature amongst the constituent elements of the compound are taken. Features include atomic weight, metal, metalloid or non-metallic properties, periodic table based properties (period, group, atomic number), various measures of radii (atomic, Miracle, covalent), electronegativity, valency features (such as the number of s, p, d, and f valence electrons), and thermal features (such as boiling point and specific heat capacity).

- *JARVIS* [27]: *JARVIS* combines structural descriptors with chemical descriptors to create "classical force-field inspired descriptors" (CFID). Structural descriptors include bond angle distributions neighbouring atomic sites, dihedral atom distributions, and radial distributions, among others. Chemical descriptors used include atomic mass and mean charge distributions. Original work generated CFIDs for tens of thousands of DFT-calculated crystal structures [27], and subsequent work adapted CFIDs for individual elements to be used in CBFVs for arbitrary compositions without known structures (*i.e.*, Figure 5.2a) [124].

- *magpie* [192]: While the Materials-Agnostic Platform for Informatics and Exploration (MAGPIE) is the name of a library associated with Ward *et al.*'s work, it this has become synonymous with the 115 features used in the paper and, as such, we will use *magpie* refer to the feature set. These features include 6 stoichiometric attributes which are different normalistion methods (L^P norms) of the elements present. These capture information of the ratios of the elements in a material without taking into account what the elements are, 115 elemental based attributes are used, which are derived from the minimum, maximum, range, standard deviation, mode (property of the most prevalent element) and weighted average of 23 elemental properties including atomic number, Mendeleev number, atomic weight among others. Remaining features are derived from valence orbital occupation, and ionic compound attributes (which are based on differences between electronegativity between constituent elements in a compound).

- *RANDOM_200* [124]: a random vector featurisation used by Murdock *et al.* to represent a lower bounds for performance.

- *fractional*[124]: An implementation of a one-hot style encoding of composition which includes average, sum, range, and variance of each element.

- *CompVec* a one-hot style encoding of composition as used in ElemNet [79] (containing only the proportions of each element in a composition). Differences between this and *fractional* are further discussed in Section 5.2.1.

Each of these representations are compared to a random projection of equal size. This controls for the size of a representation when investigating the advantage of the domain knowledge built into a CBFV. Several of the five case studies investigated contain multiple applications of ML within a single publication. The tasks which were recreated in this comparison (and their relevant case study references) are as follows:

- T_c: Using a regressor to predict the superconducting critical temperature (T_c) of a material (12666 data points in training set) [170].
- $T_\mathrm{c} > 10\mathrm{K}$: Classifying if the T_c of a material is greater than 10 K (12666 data points in training set) [170].
- $T_\mathrm{c}|(T_\mathrm{c} > 10\mathrm{K})$: Regressing to find T_c given $T_\mathrm{c} > 10\mathrm{K}$ K (4833 data points in training set)[170].
- HH stability: Predicting the stability of half-Heuslers (8948 data points in training set) [102].
- E_gap(oxides): Predicting the band gap of oxides found in the Computational Materials Repository database (599 data points in training set) [35].
- Glass Forming Ability (GFA): predicting the ability of a bulk metallic glass alloy (BMG) to exist in an amorphous state (5051 data points in training set) [195].
- D_max: Predicting the critical casting diameter of a BMG (4724 data points in training set) [195].
- ΔT_x: The supercooled liquid range of a BMG (495 data points in training set) [195].
- E_gap(DFT): Predicting the band gap of materials calculated using DFT (35653 data points in training set) [84]. This dataset combines data from the materials project and Duke University's AFLOW [78, 32].
- E_gap(exptl): Predicting the band gap of materials measured experimentally (1986 data points in training set) [206]. This was used in experiments as to the effect of transfer learning from DFT to experimental band gap prediction [84].

- $E_{\text{gap}}(\text{DFT}) \cup E_{\text{gap}}(\text{exptl})$: Predicting the band gap of a dataset consisting of both DFT calculated and experimentally measured band gaps (37639 data points in training set) [84].

Performance in regression tasks was measured using r^2 correlation and classification task performance is measured using accuracy. Therefore, the percentage improvement over random projections can be considered to be:

$$100\left(\frac{M(y,\hat{y})}{M(y,\hat{y}_p)} - 1\right)$$

Where y is the target label for a prediction, $\hat{y}$ is the label predicted by a model that uses a given representation, $\hat{y}_p$ is a label predicted by a model that uses a random projection of equal size to the given representation, and M is accuracy for classification tasks and r^2 for regression tasks. Measurements found using other values of M can be found in the supplementary information. To investigate repeatability of these results, a large subset of these experiments have been repeated 5 times and the standard deviations of these results calculated (Section 5.4).

5.2.3 Results

Observations for each individual case study will be considered before a summary of the results is given. For each ML task investigated attempts were made to recreate the representation used in that study, and train a RF on this representation to compare to representations listed above. When recreation proved infeasible, alternatives have been noted. Full tables of results for each case study are provided, including (LOCO-CV) and kernelised LOCO-CV measurements (Tables 5.10 to 5.12). The featurisation used in K-means clustering for LOCO-CV and kernelised LOCO-CV measurements was done using *magpie* representation, as it generally demonstrated balanced clustering across the datasets and tasks investigated here (Figure 5.7a), and resulted in more models learning trends more consistently (Figure 5.9b).

As noted, these papers were selected for interesting use of ML, not for the choice of representation which was used in each paper. Several of these case studies mention that representation could be improved through further feature selection and none make any claims that their representation is advantageous over existing other representations such

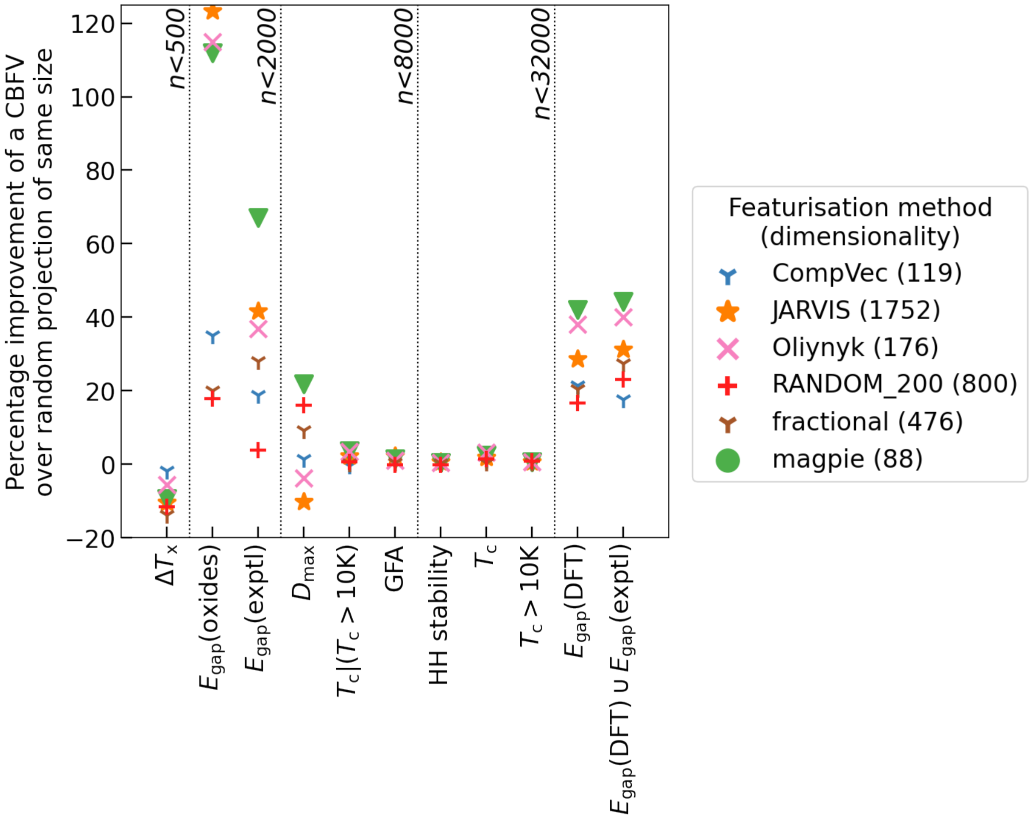


Figure 5.3: Performance of composition-based feature vectors (CBFVs) on predictive tasks compared to random projections.Random projections exhibit similar performance to CBFVs for most tasks. This is not true for band gap prediction tasks, where CBFVs with domain knowledge demonstrate marked improvement.

as those being examined here.

Machine learning modelling of superconducting critical temperature (2018)

This study uses data from the Japanese National Institute of Materials Science superconductivity dataset (total training set size 13077) [170]. They use random forests to predict superconducting critical temperature (T_c) in three contexts:

- T_c: Using a regressor to predict the superconducting critical temperature (T_c) of a material.
- $T_\mathrm{c} > 10\mathrm{K}$: Classifying if the T_c of a material is greater than 10 K.
- $T_\mathrm{c}|(T_\mathrm{c} > 10\mathrm{K})$: Regressing to find T_c given $T_\mathrm{c} > 10\mathrm{K}$.

Stanev *et al.* derive a custom CBFV from the magpie package. In recreating all three of the above tasks, their custom CBFV performs similar to the CBFVs discussed in Section 5.2.2 (tables 5.2 to 5.4). This is in line with the suggestion that a dataset of this size will see little benefit from domain knowledge. Due to limited reproducibility, results shown here are compared to their results as published, rather than as recreated.

Materials screening for the discovery of new half-Heuslers: Machine learning versus ab initio Methods

Legrain *et al.* use random forests to predict whether a half-heusler is stable or unstable using a custom made descriptor containing structural information of a compound [102]. The dataset they use contains 164 stable vs 11022 unstable half-heuslers which introduces some difficulties when applying LOCO-CV.

A dataset which is overwhelmingly one class is no longer suitable for LOCO-CV measurements as it is possible for all of the outlier class will lie in one cluster, which breaks many metric formulae which require all classes to have at least one example to avoid division by zero. For example in binary classification the specificity can be measured by

$$\text{Specificity} = \frac{tn}{N}$$

where tn is the number of true negative predictions and N is the total number of negative observations in the dataset. Where $N = 0$, even if you were to tweak the formula to stop division by zero (such as by adding a small number to the denominator), such a metric would be meaningless. In experiments ran here LOCO-CV failed due to all of the classes ending up in one cluster for all featurisation methods.

While LOCO-CV will not allow for extrapolatory measures of algorithms trained on these data, given a random split it is highly unlikely that all stable Heuslers end up in test dataset. The performance of CBFVs listed above (Section 5.2.2) was compared to the featurisation used in this case study. F_1 score and precision were considered the most important metrics for success, as the unbalanced nature of the dataset makes accuracy and recall are approximately 1 for all models measured. CBFVs with domain knowledge resulted in more precise predictions than both the structural representation used by in this paper and representations without domain knowledge (table 5.5).

This is in contrast to previous suggestions that there would be little benefit for domain knowledge in CBFVs for a dataset of this size [124], however, those findings had no stipulations on dataset balance, which likely affected results. CBFVs with domain knowledge outperforming the representation used in this case study is surprising given that CBFVs are made using no structural information, suggesting that just because a representation *should* contain more knowledge does not mean such a representation will outperform others without such information.

Data-driven discovery of photoactive quaternary oxides using first-principles machine learning

This case study predicts band gaps found in the Computational Materials Repository database, using the 799 oxides as training/test data [35]. The representation used in the paper is a CBFV of 148 features generated with matminer, most (132) of which are derived from the magpie descriptors, with the rest constituting information on the highest occupied molecular orbital and lowest unoccupied molecular orbital, norms of stoichiometric attributes, ionic properties (including maximum and average ionic character between two atoms), and an estimation of absolute position of band centre. Some of these features are repetitions of those in the magpie feature set for example the average number of s, p, d, and f valence electrons. The aggregation functions implemented included the mean mean absolute deviation and modal value for magpie descriptors as well as the mean, sum, range, and variance of magpie descriptors which are used in previous work (and the main text of this work).

The representation used in this study resulted in better predictions than those found using no domain knowledge, performing equivalently to other CBFVs with domain knowledge, and performing significantly better in LOCO-CV measurements (table 5.6). This would fit the suggestion that inclusion of domain knowledge improves performance for ML methods when dataset size is smaller than 1000. It is notable that the representation used in this study did not outperform *magpie* as implemented for this and previous work[124]. This suggests that including the aggregation functions mode and mean absolute deviation of a feature does not meaningfully impact performance.

A machine learning approach for engineering bulk metallic glass alloys

This study uses ensemble learning methods for three separate prediction tasks related to the engineering of bulk metallic glass alloys (BMG) [195]. The following are predicted:

- Glass Forming Ability (GFA): predicting BMG's ability to exist in an amorphous state.
- D_{max}: Predicting the critical casting diameter of a BMG.
- ΔT_{x}: The supercoooled liquid range of a BMG.

The work uses a CBFV derived from the magpie descriptors with a total of more than 200 features, the exact number varying depending on prediction task. This is compared to the originally proposed 145 features [192] and the variant used here with 88 features [124]. This is applied to custom datasets collected from 41 different papers and one handbook, they used subsets of these for each task as GFA, D_{max}, and ΔT_x were not available for all compounds.

In regression tasks (D_{max} and ΔT_x prediction), the custom CBFV used in this study marginally outperforms the representations being investigated here in some metrics (table tables 5.7 and 5.8). The performance difference between the representation used in this work and the other CBFVs investigated (both with and without domain knowledge) was significantly smaller in the D_{max} dataset. This fits previous findings that specialised domain knowledge becomes less important as dataset size increases [124], as the D_{max} training dataset size was almost an order of magnitude larger than that of the ΔT_x (4725 and 497 respectively). Regardless of the CBFV used all RFs failed to predict reliably in LOCO-CV (Tables 5.7 and 5.8), this may suggest that RFs should not be used for extrapolation in this task, however this is likely due to uneven cluster sizes used in the LOCO-CV process. Methods for adressing this problem are explored later in the chapter (Section 5.3). As will be seen once cluster sizes are made more even, RFs manage to perform similarly in extrapolation as they on an 80/20 train/test split. .

In recreation of the GFA classification task, the representation used in this study performed similarly to other CBFV's investigated (table 5.9). This fits with the hypothesis that for larger datasets CBFV domain knowledge becomes less important with size as the training dataset was size 5053.

Extracting knowledge from DFT: Experimental band gap predictions through ensemble learning

This work focusses on the use of neural networks to predict DFT calculated band gaps and transferring this knowledge to retrain them on a smaller set of experimental measurements, finding the transfer learning to be advantageous [84]. They use *magpie* featurisation on DFT data extracted from the Materials project and AFLOW as well as experimental data compiled in previous work [78][32][206].

As the transfer learning approach used in the case study is not applicable to RFs, in recreating this case study these are considered to be 3 separate datasets:

- $E_{\mathrm{gap}}(\mathrm{DFT})$: Predicting the band gap of materials calculated using DFT.
- $E_{\mathrm{gap}}(\mathrm{exptl})$: Predicting the band gap of materials measured experimentally.
- $E_{\mathrm{gap}}(\mathrm{DFT}) \cup E_{\mathrm{gap}}(\mathrm{exptl})$: Predicting the band gap of a dataset consisting of both DFT calculated and experimentally measured band gaps.

Experiments on which CBFV is most effective on these datasets showed that datasets $E_{\mathrm{gap}}(\mathrm{exptl})$ and $E_{\mathrm{gap}}(\mathrm{DFT}) \cup E_{\mathrm{gap}}(\mathrm{exptl})$ yielded similar results, which is logical as they are very similar datasets. In these datasets domain knowledge based CBFVs outperformed those without domain knowledge, with $JARVIS$ slightly outperforming all other CBFVs (tables 5.11 and 5.12).

The larger datasets saw the performance difference caused by different CBFVs become smaller with the range of r^2 between different CBFVs becoming 0.050 smaller (the range was 0.16, 0.15, and 0.21 in the datasets 1, 2, and 3, respectively). While a dataset size increase usually sees the benefit of domain knowledge decrease, here the decrease of that benefit is less. Here datasets of more than 35,000 compounds still showing a notable benefit to domain knowledge.

5.2.4 Summary observations of effect of representation on predictive ability of random forests

Overall, recreation of these tasks shows that, broadly, changes in CBFV made little difference to performance when compared to a random projection of the same size (Figure 5.3). Featurisation methods inspired by domain knowledge do show advantages in some datasets. These advantages seem to be task-specific as opposed to based on dataset size, specifically band gap-based tasks seem to see benefit from knowledge-based features, however most other tasks do not see noticeable improvement from this feature engineering (Figure 5.3). This could be because vast amounts of band gap data can be acquired through DFT calculations [78] and as such band gap prediction is a widely available benchmark that researchers could use when testing a newly proposed CBFV[30].

Intuition may suggest introducing more dimensions that do not contain any additional information would result in worse algorithmic performance. However, despite having 68% more dimensions, RANDOM_200 performs within 5% of the fractional representation. On large enough data sets (more than approximately 3000 data points) the random representation does not perform appreciably differently to the *magpie* representation. Notably, on tasks outside of band gap prediction there is little advantage to domain based representations over a random projection. Random projection can be seen as an alternative to CBFVs, it can be used as a comparative measure against CBFVs. If a feature set cannot appreciably outperform a random projection of the same size or smaller, then, while there may still be benefits to analysis of the feature importance of such a feature set, that feature set does not enrich the representation of a material when it comes to algorithmic performance.

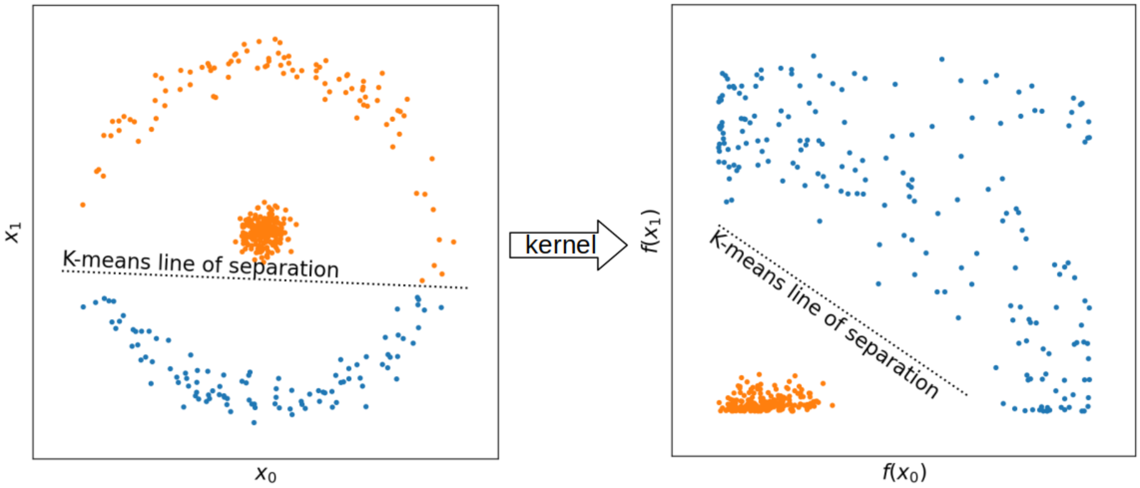


Figure 5.4: A visualisation of how application of kernel functions can affect the data in an example dataset. Here, we show the radial basis function (RBF) approximation so $f(x) = \exp(-x^2)$. There is no clear way to linearly separate classes before application of RBF approximation; however, non-linear translation of each point with the RBF approximation yields a data space through which a straight line can be drawn to separate the classes.

5.3 Improving the linear separability of chemical data spaces for more applicable measurements of extrapolatory power

As noted above, uneven cluster sizes pose problems for the LOCO-CV assessment of the extrapolatory power of ML models, but such issues with K-means clustering are not only found in materials science. K-means clustering attempts to linearly separate clusters (*i.e.*, draw a straight line between them); some clusters cannot be separated this way (Figure 5.4). In classical computer science problems, methods have been applied to datasets on which a linear discriminator (such as K-means or support vector machines) exhibits poor performance. In many cases, measuring distances between points using a non-linear function rather than the Euclidean distance can resolve these issues. Known as the "kernel trick", this can be applied to many methods but does require modification to existing algorithms. Examples of kernels used in this way are the radial basis function (RBF) and the $\tilde{\chi}^2$ kernels. In many cases, algorithms are implemented in such a way that these modifications are easy or are considered a hyperparameter of the algorithms [140].

However, graphing changes made by kernel tricks are difficult, and some kernel tricks may be more computationally expensive (or just less well optimised) than the Euclidean distance. Linear discrimination can be improved by incorporating kernels into the ML algorithms rather than Euclidean distance. However, where algorithms may be run multiple times, it can be easier or more efficient to apply an approximation of this *a priori.* Applying a non-linear transformation to every data point in a dataset can transform the data so that it is more amenable to linear discrimination (Figure 5.4). To perform this non-linear translation of data points, kernel approximation methods can be used. These methods approximate the kernel methods by transformation of the original data points using a set of basis functions, such as the radial basis function (RBF), additive χ^2, and skewed χ^2. These basis functions map the original data points to a higher-dimensional feature space, where they can be more easily separated by a linear classifier.

For an example of how kernel tricks relate to kernel approximation, consider the RBF. The RBF kernel is defined as:

$$k(x, x') = \exp(-\gamma||x - x'||^2) \tag{5.1}$$

where γ is a hyperparameter and $|| \cdot ||$ denotes the Euclidean (l_2) norm. The RBF kernel can be expensive to compute, especially for large datasets.

One RBF approximation (which is also known as random kitchen sinks) [146] approximates the RBF kernel by using n:

$$f(x) = \frac{\sqrt{2}\cos(x \cdot w + o)}{\sqrt{l}} \tag{5.2}$$

where l is the number of components, w is an $l \times d$ matrix with i.i.d. standard normal entries, b is a vector of random phases uniformly distributed in $[0, 2\pi]$, and d is the dimensionality of the input. This equation maps the input vector x to a l dimensioned feature space using the RBF kernel.

The intuition behind this approximation is that random projections can preserve pairwise distances between data points with high probability, especially in high-dimensional spaces. By randomly projecting the input data to a lower-dimensional space and applying a nonlinear function (such as cosine), the RBF kernel can be approximated in a computationally efficient way. The number of random projections required depends on the desired approximation accuracy and can be determined empirically. As with random projections, the error in this approximation is inversely proportional to the number of dimensions. In the studies presented here, 100 dimensions were always used, as this is the default hyperparameter in the code library used [140].

Having established that RBF approximations could be a suitable tool to improve linear discrimination (such as in the materials datasets explored in Section 5.2), it remains unclear whether this will result in more even clusters (as is desirable for LOCO-CV). Radial basis, additive $\tilde{\chi}^2$, and skewed $\tilde{\chi}^2$ approximations were applied across several datasets and material representations to see if these non-linear translations will reduce the cluster size unevenness found by K-means clustering. Reduced cluster size unevenness found with K-means would improve the applicability of LOCO-CV measurements, addressing one of the problems previously highlighted.

After an exploration of how success could be measured in K-means clustering (Section 5.3.1), cluster size uneveness was quantified by using the standard deviation between the cluster sizes in a single run of K-means clustering. The representations of the material and the tasks investigated were those described in Section 5.2.2. Data normalisation affects both K-means clustering and kernel approximation methods. As such, tests were carried out to determine which normalisation methods lead to the greatest reduction in standard deviation between clusters (Section 5.3.2).

With the chosen normalisation methods, the kernel approximation methods were tested for their ability to make clusters found with K-means clustering more even (Section 5.3.3). Finally, the impact of these methods are discussed when used in LOCO-CV is discussed (Section 5.3.3).

This section investigates the effect of kernel approximation methods, $\tilde{\chi}^2$, and skewed $\tilde{\chi}^2$ and RBF approximation on materials science data, specifically studying their use to improve the suitability of leave-cluster-out cross-validation (LOCO-CV) by addressing the problems of uneven cluster sizes. Kernel approximations reduced the variance of class sizes in clustering, regardless of input feature representation. This resulted in more reliable model training when using these clusterings for LOCO-CV.

5.3.1 Performance metrics in K-means clustering

Without prior knowledge of the expected clusters for each data point, the results found with K-means clustering are difficult to interpret, although expert inspection can yield insight into what different clusters can represent. Expert inspection of results may be justifiable with less than 10 clusters (each of which could have thousands of materials); however, when using K between 2 and 10 (as originally proposed [117]), the LOCO-CV algorithm presents 54 different clusters ($\sum_{n=2}^{10} n$), making such expert inspection infeasible. Thus, metrics must be used to quantify the success of a clustering.

Where target labels exist, metrics such as mutual information score, homogeneity, and completeness score can be used. Without labels, Euclidean distance-based measures such as sum-squared distance to cluster centroid or average distance between each point and the other points in its cluster can be used, however, this does not intrinsically tell us how much information is in a clustering, just how tightly packed a cluster's members are. The average distance between each point and the other points in its cluster is computationally prohibitive, so it will not be used in this study.

Euclidean distance-based measurements such as these lack comparability in our use case, as each dataset and each featurisation technique should be considered independent. Identifying trends in these measurements with different numbers of clusters and looking at the effect of kernel methods on Euclidean distance-based measurements are both valid uses. However, as Euclidean space is affected by dimensionality, it is important that conclusions into the effect of different featurisation approaches are not drawn from such measures. While noting these caveats, the mean distance from a point in a cluster to the centroid of the cluster is used as a measure of how tight the clusters are in Euclidean space, this metric is labelled the spread of cluster.

As the aim of this investigation is to improve the validity of the measures taken with LOCO-CV, specifically to address issues with vastly uneven cluster sizes, the standard deviation in cluster sizes is used as a metric for success (the unevenness in cluster sizes). Material science datasets may have uneven cluster sizes due to research bias towards exploration of promising materials, and identically sized clusters would be unexpected for materials data; identically sized clusters were, in practise, never observed in this study. Using the unevenness of the cluster sizes serves as a measure of whether the cluster sizes differ by many orders of magnitude, which would affect the validity of the measurements taken using LOCO-CV. This does not imply that more even clusters are more chemically sensible groupings of materials, just that they may be more sensible for use with LOCO-CV, as uneven cluster sizes raise questions about measurements taken with LOCO-CV (Section 3.1.5).

The ease of clustering is expected to vary between datasets. Accordingly, to appropriately compare standard deviation in cluster sizes, max-min normalisation was performed across different featurisation techniques and numbers of clusters in the same dataset. Consequently, for each dataset, the most uneven cluster size measurement found is 1 and the least uneven cluster size measurement is 0. These normalised values are used when comparing cluster size unevenness between datasets.

5.3.2 Normalising inputs for kernel approximation methods

Various normalisation methods were tested to establish which is the most appropriate. As skewed χ^2 and additive χ^2 are only well defined for a positive input, the data were scaled between 0 and 1 using min-max normalisation before use with these functions. As the RBF approximation (and K-means without kernels) can be affected by the disparity of scale between the axes, different normalisation methods were investigated. The data normalisation which most often resulted in the lowest cluster size uneveness could then be used for further experimentation into the effects of kernel approximation methods.

When performing K-means clustering with either the radial basis function (RBF) or no kernel method at all (the identity function), the following normalisation methods were considered:

- *l2*: l2 normalisation.
- *min-max -1:1*: Min-max normalisation to scale data between -1 and 1.
- *min-max 0:1*: Min-max normalisation to scale data between 0 and 1.
- *standard*: Standardisation of each dimension to mean 0 and unit variance.
- *none*: No normalisation method.

Every dataset discussed in Section 5.2.2 was normalised using each normalisation method. The normalised data were then used as input to RBF and the identity function, the resulting data were clustered using K-means clustering (K used between 2 and 10 inclusive). For each kernel, dataset, and value of K, the normalisation method which resulted in the lowest standard deviation between cluster sizes (cluster size unevenness) was recorded. Normalisation methods which most frequently resulted in the lowest cluster size uneveness were used in the results reported in the main text. For RBF no normalisation was used, and when testing without a kernel, data were scaled between -1 and 1 using Min-Max scaling (fig. 5.5)

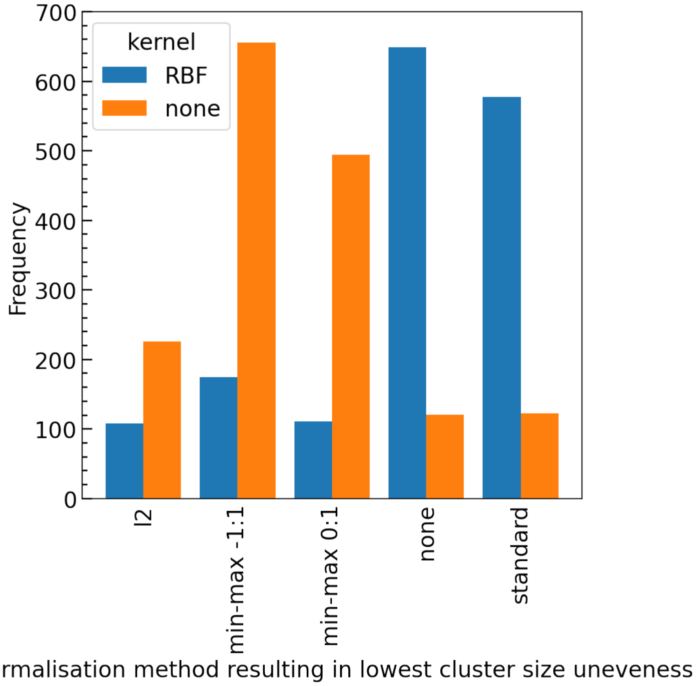
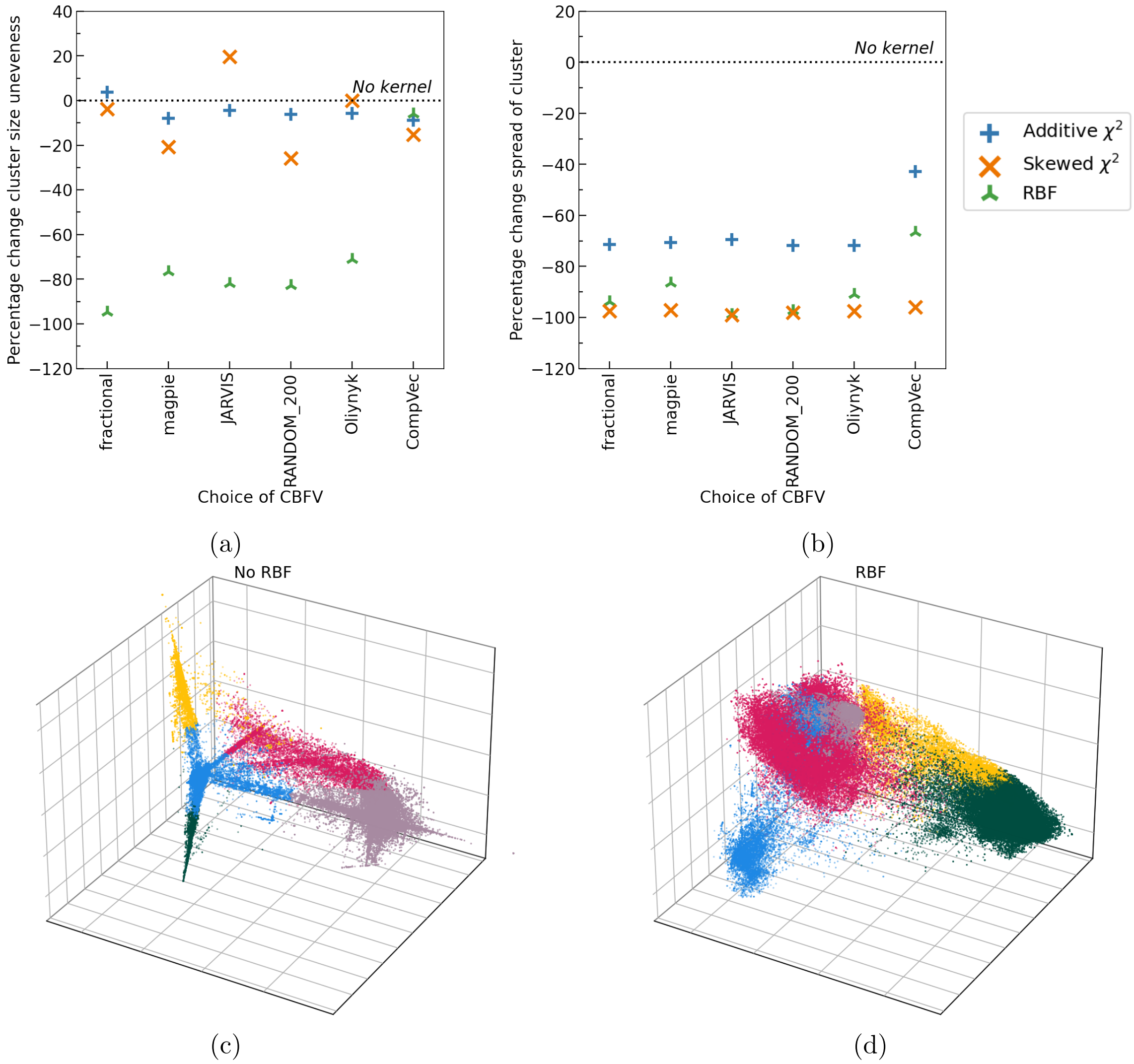


Figure 5.5: The frequency for which different normalisation methods resulted in the lowest cluster size uneveness (standard deviation in cluster size), grouped by kernel usage.

5.3.3 Do kernel methods result in more even clusters in materials science data?

The three investigated kernel approximation functions resulted in clusters of a more even size than when no kernel function being applied at all. On average, the RBF approximation resulted in the largest reduction in standard deviation of cluster size (fig. 5.6). Furthermore, note that application of any of these kernel methods generally resulted in a reduction in the distance between points in a cluster and their centroids (spread of cluster), indicating more tightly packed clusters (fig. 5.7b). On average, application of skewed $\tilde{\chi}^2$ saw the greatest reduction in the spread of the cluster. As this investigation looks to create more even cluster sizes for use with LOCO-CV, we focus on impacts of RBF, as, of the kernel methods tested, it resulted in the greatest impact on this metric as defined by the largest reduction in standard deviation of cluster size.

Before application of a kernel function, cluster sizes are more even in domain knowledge-



(a) (b)

(c) (d)

Figure 5.6: Demonstration of the effect of kernel methods on clustering of compositions in the ICSD. (a) Changes in standard deviation of cluster size found by K-means clustering of ICSD (k=5) with application of kernel methods. Most of the time, application of kernel methods reduces the variation between cluster sizes. This effect is most pronounced with the basis function (RBF) kernel. (b) Variation in cluster spread for K-means clustering of ICSD (k=5). Application of kernel methods reduces the spread in Euclidean space within a cluster. This effect is most pronounced with skewed χ^2 and RBF. (c) To visualise these results, PCA was used to generate the first three principal components of all compositions in the ICSD featurised using a CompVec. Colours correspond to clusters found by K-means (k=5) clustering on this representation. Inspection of these clusters reveals highly anisotropic clusters with no meaningful boundaries in the data to unambiguously separate clusters. (d) The first three principal components found when examining an RBF translation of the ICSD (featurised using CompVec), points are coloured according to clusters found by K-means (k=5) applied to the kernelised data. The application of an RBF (as defined in Section 5.3) to every composition vector in the ICSD (before clustering) leads to clusters that are more isotropic with more clearly resolved boundaries between clusters.

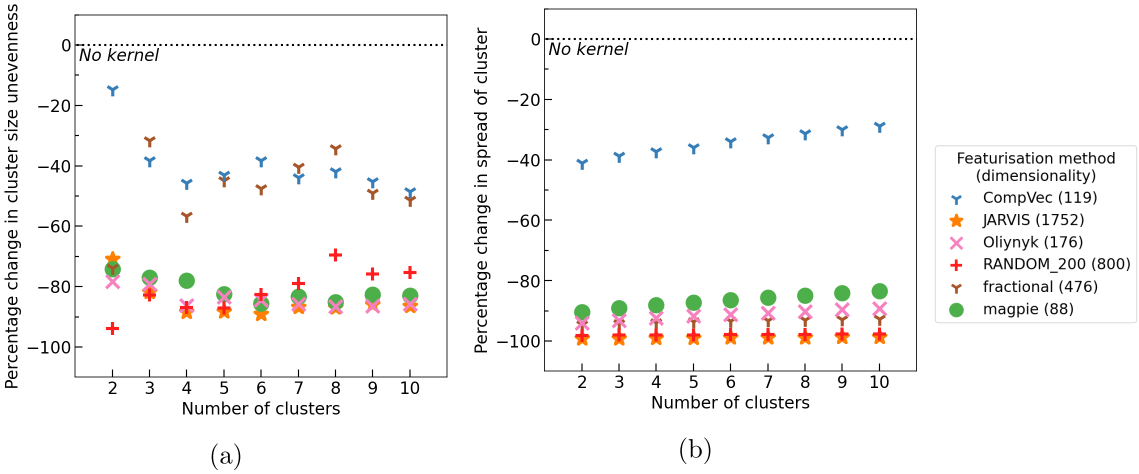
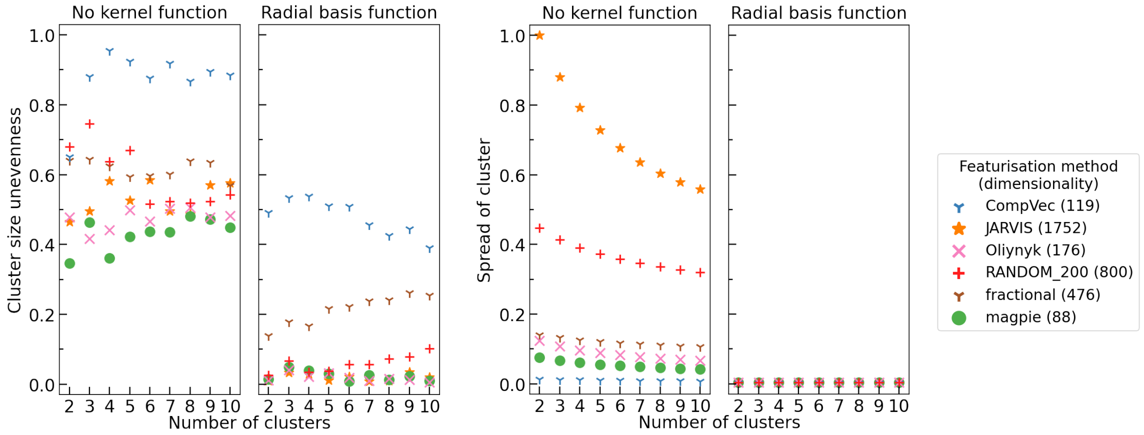


Figure 5.7: Effect of radial basis function (RBF) on standard deviation of cluster sizes (cluster size unevenness) and spread of cluster sizes. This is performed using K-means clustering with different values of k. (a) RBF leads to more evenly sized clusters for all featurisation methods and nearly all values of k. (b) RBF leads to more compact clusters (*i.e.*, smaller average Euclidean distance between points within a cluster) for all featurisation methods and all values of k

based representations, as measured by the standard deviation in cluster sizes. *CompVec* representation resulted in a larger standard deviation between the sizes of the clusters (*i.e.*, less evenly sized clusters) than all other representations investigated, likely due to the sparse nature of this representation, with the *magpie* representation resulting in the most even cluster sizes (fig. 5.8a). The two one-hot based representations, *fractional* and *CompVec*, generally did not result in as even cluster sizes as other representations. The application of *CompVec* resulted in substantially worse performance than that of *fractional* despite their very similar nature, only differing in the use of the aggregation functions (as discussed in section 5.2.1).

RBF universally resulted in more even clusters. The smallest change (as a percentage of the standard deviation in the size of the cluster prior to the application of the RBF) was seen in *fractional* and *CompVec* representations (two of the representations that resulted in the worst performance in this metric) (fig. 5.7a). However, outside these two representations, the proportional impact of RBF on this measure did not correlate with the performance of a CBFV in this measure prior to application of RBF.

Without the use of kernel functions, there is a clear correlation between the size of

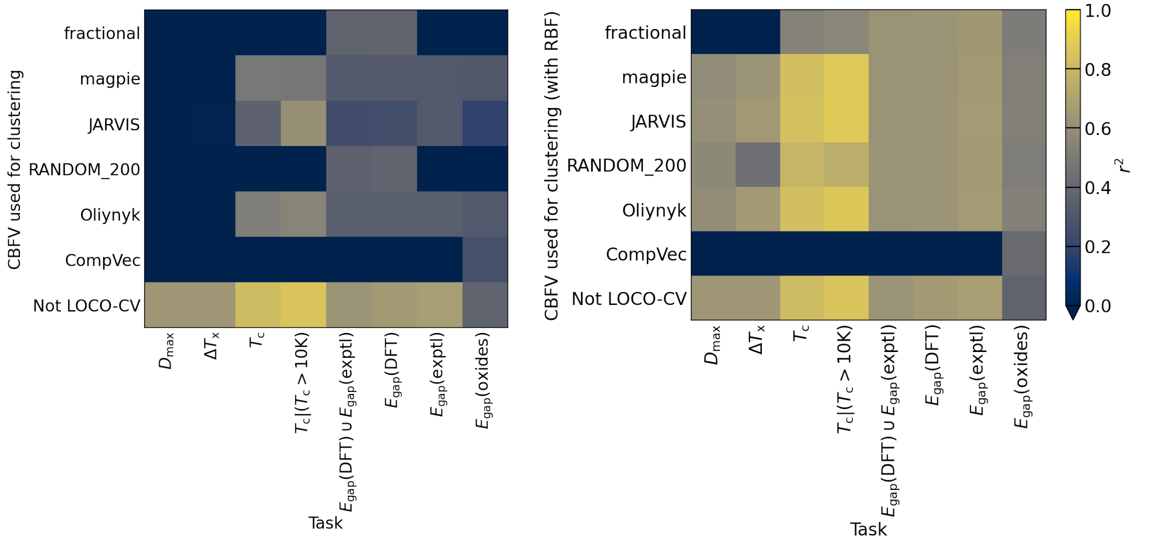


(a) (b)

Figure 5.8: Mean cluster size unevenness and spread of clusters found by K-means when clustering different representations of datasets. Measurements are normalised to between one and zero on a per dataset basis, as different datasets would be expected to cluster with different amounts of ease. The normalised values are then averaged across different datasets for each representation and value of k. (a) Clusters are generally more even in domain knowledge based representations as measured by the standard deviation in cluster sizes. (b)Without application of kernel function, spread of clusters as measured by the average distance between a point in a cluster and its centroid correlates to the size of the representation with the exception of *CompVec* which has the tightest clusters. Application of radial basis function makes this trend insignificant.

a representation and the spread of the clusters found using that representation, with the exception of *CompVec*, which saw the tightest clusters (Figure 5.8b). This trend is no longer seen after applying the RBF approximation. The application of the RBF approximation to a CBFV before K-means clustering reduced the spread of the clusters found (Figures 5.7b and 5.8b). The relative size of the change seen after applying the RBF approximation was correlated with the spread of clusters found when no kernel method was used. The higher the spread of clusters found using a CBFV without a kernel method, the larger the change seen when clustering using that CBFV and a RBF approximation.

Use of kernel methods in featurisation results in more even cluster sizes when using that featurisation for K-means clustering. As featurisation used for clustering in LOCO-CV is independent of that used for learning, incorporating these kernel approximation methods into LOCO-CV is simple and applicable regardless of the ML algorithm, the chosen metric, and the initial representation (Figure 5.1). Therefore, it is recommended to use kernel approximation methods when using clustering of K-means for LOCO-CV to address the issue of uneven cluster sizes. Addressing this issue results in models that are more reliably successful in learning trends in data using LOCO-CV (Figure 5.9).



(a)

(b)

Figure 5.9: Performance of random forests in regression tasks to compare evaluation regimens, measured using r^2. These random forests are evaluated with LOCO-CV (labelled with the CBFV used for K-means clustering), as well as a traditional 80:20 train:test split (labelled "Not LOCO-CV"). Importantly, in LOCO-CV, the representation used for K-means clustering is independent of that used for training. Accordingly, all models are trained using CompVec CBFV to remove training representation as a confounding variable. (a) Without the application of RBFs, the same random forest model which performs well in traditional 80:20 split training regimen often fails to learn trends in the data when evaluating with LOCO-CV, leading to low values of r^2. (b) Application of RBF to CBFVs before K-means clustering for LOCO-CV results in fewer models failing to learn trends in the data, leading to higher values of r^2.

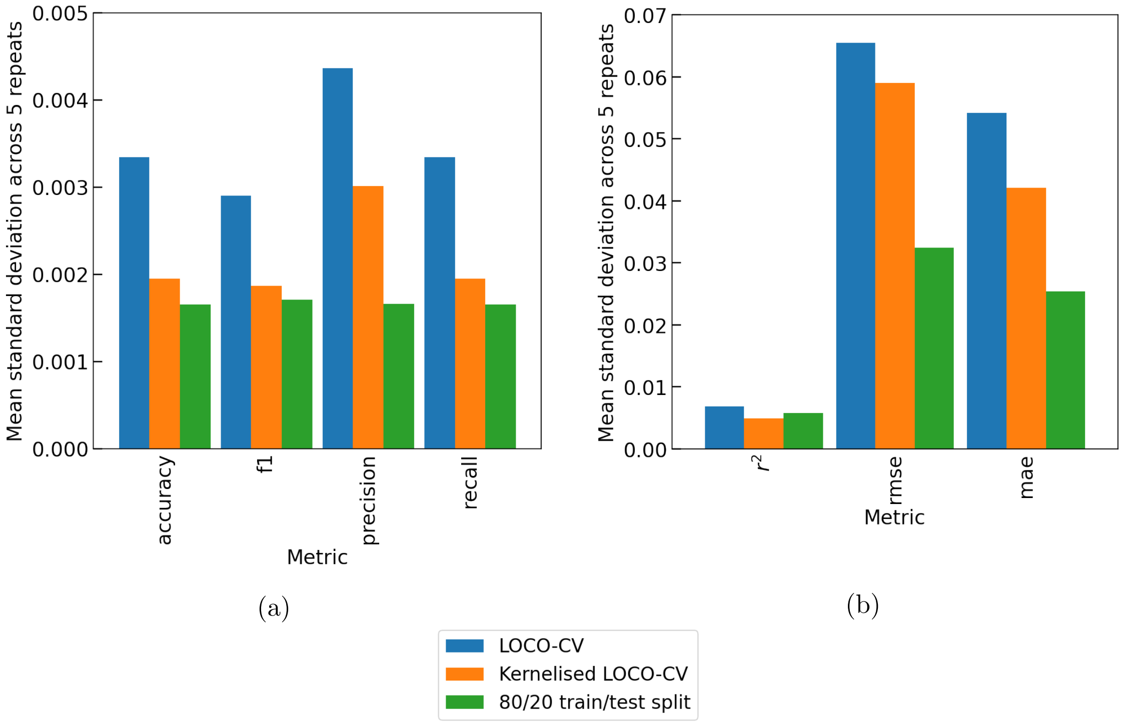
5.4 Experiments in repeatability

As the K-means clustering part of LOCO-CV (and kernelised LOCO-CV) is non-deterministic, experiments were carried out to investigate whether this would significantly impact the repeatability metrics taken using these techniques. All tasks investigated in section Section 5.2 were repeated 5 times for all representations measured that have less than 500 dimensions (since larger representations were prohibitively expensive to train multiple times). Exclusion of representations larger than 500 dimensions meant that the representations investigated for these experiments in repeatability were:

- *magpie* (88 dimensions)
- *CompVec* (119 dimensions)
- *Oliynyk* (176 dimensions)
- *Random Projection* (88 dimensions)
- *Random Projection* (119 dimensions)
- *Random Projection* (176 dimensions)

Random forests trained using these representations were evaluated with LOCO-CV, kernelised LOCO-CV and a traditional 80%/20% train/test split. By comparing the standard deviations of measurements across different repeats of a task, it is possible to compare the repeatability of LOCO-CV and kernelised LOCO-CV to that of an 80/20 80%/20% train/test split. Clustering for LOCO-CV and kernelised LOCO-CV in these experiments was performed using the representation *magpie* (as in Section 5.2).

In both the regression and the classification results, the application of the RBF approximation improved the repeatability of LOCO-CV (fig. 5.10). Although LOCO-CV and kernelised LOCO-CV are less repeatable than a 80%/20% train/test split, the decrease in reliability is small enough to not substantially impact the interpretation of the results.



(a)

(b)

Figure 5.10: The standard deviation of LOCO-CV, kernelised LOCO-CV, and 80/20 train test split scores for 5 repeats of a task. The mean of these standard deviations is taken across all tasks and all representations. Tasks tested here are all those explored in Section 5.2, and representations are those explored in Section 5.2 which are less than 500 dimensions. (a) Standard deviation of performance in classification tasks across 5 repeats. Further breakdowns of these data can be seen in the appendix to this chapter. (b) Standard deviation of performance in regression tasks across 5 repeats. Further breakdowns of these data can be seen in tables S4-S6. As r^2 is unbounded below 0, results shown here is calculated by excluding and r^2 measurement less than 0.

5.5 Clustering Random Projections with and without kernel methods

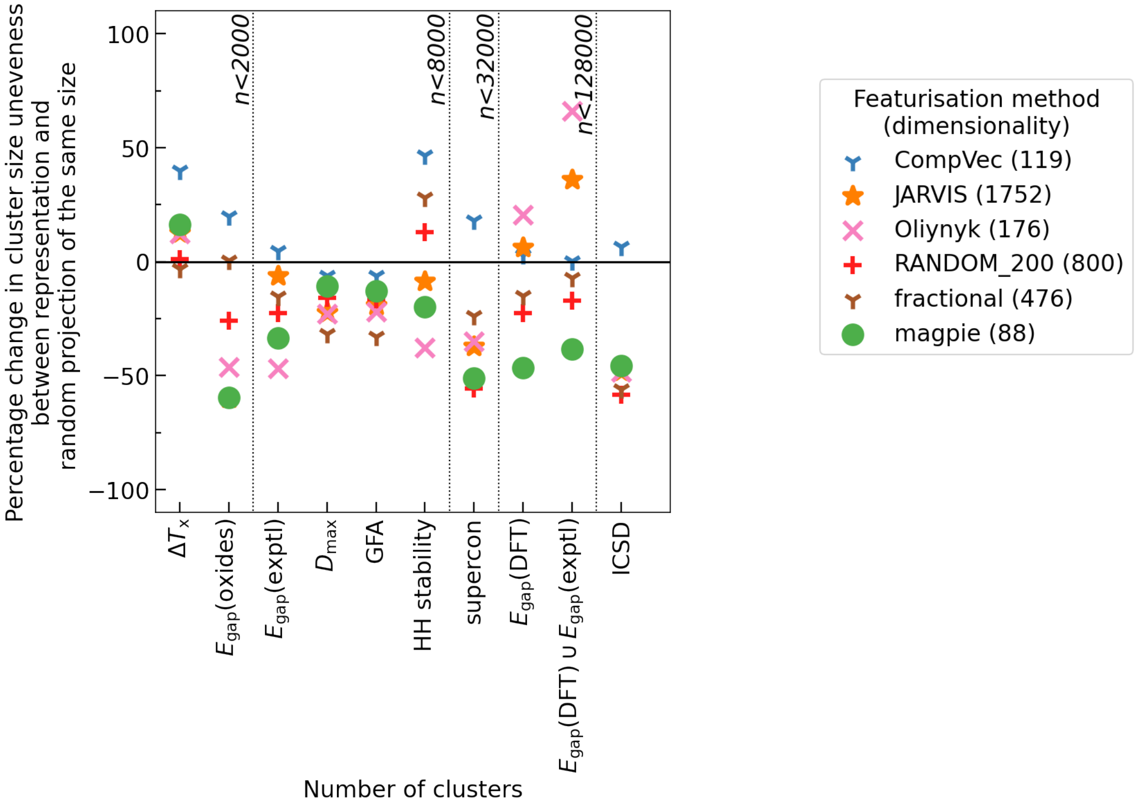
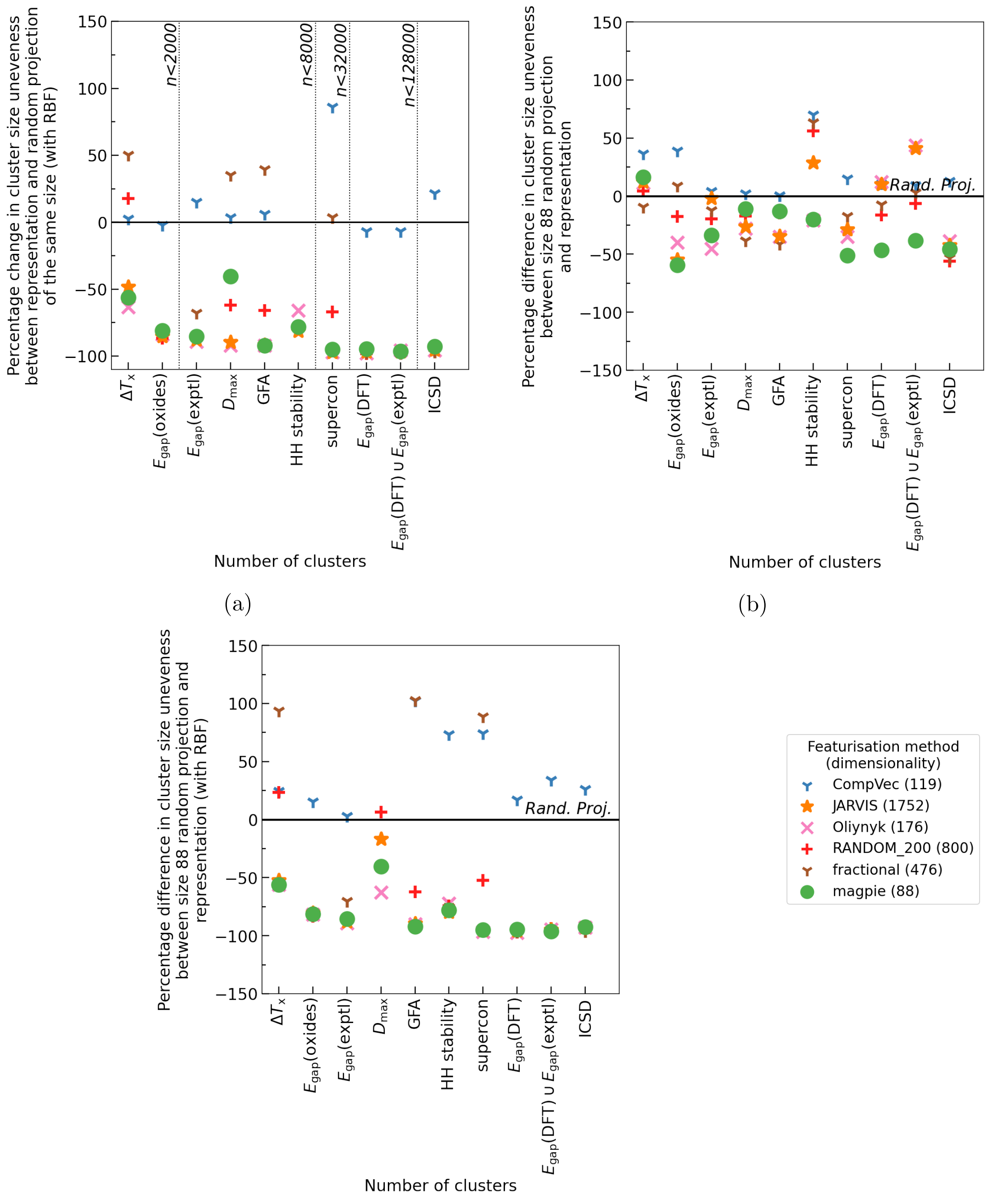


Figure 5.11: Reduction in cluster size unevenness (standard deviation in cluster size) of different CBFVs when compared to equal sized random projections of composition vectors across different datasets with no kernel applied. While random projection consistently outperformed *CompVec*, all other CBFVs form more even clusters than an equally sized random projection.

Having established that random projections perform similarly to engineered feature vectors in many tasks (Section 5.2) and that kernel approximation methods can be used to reduce cluster size variance in K-means clustering on materials datasets (Section 5.3), experiments were carried out to measure the cluster size variance of random projections of compositions both with and without application of kernel methods. As noted (Section 5.1) the RBF approximation does include a random projection, so it is unclear if it is suitable to apply this RBF approximation to a random projection.

Without application of kernel approximation, when each CBFV was compared to a random projection of equal size (Figure 5.11), using random projections of composition vectors did, more often than not, resulted in more evenly sized clusters than *CompVec*, but less evenly sized clusters than all other CBFVs investigated. However, no representation (either random projection or CBFV) universally resulted in more even clusters. Comparing the best performing size of random projections (88 dimensions) with other CBFVs without any kernel methods did narrow the differences in cluster size uneveness (Figure 5.12b), however other CBFVs still outperformed random projections in several datasets.

Radial basis, additive χ^2, and skewed χ^2 approximation functions were applied to these projections before clustering using K-means. The resulting clusters were compared to those found without any kernel methods, showing that RBF and skewed χ^2 reduced the unevenness of the cluster size (Figure 5.13). However, these results do not create a consistent pattern of either outperforming or underperforming the cluster size unevenness found by applying RBF approximation to CBFVs (Figure 5.12a). As no representation universally results in more even clusters, a variety of CBFVs and random projections should be investigated when choosing the best representation for clustering a dataset. Application of kernel approximation methods such as RBF are advantageous in this context regardless of representation.



(a)

(b)

(c)

Figure 5.12: Performance advantage of different CBFVs against *Random Projections* of composition vectors across different datasets as measured by cluster size unevenness (standard deviation in cluster size) (a) CBFVs are compared wtih *Random Projections* of equal size and a RBF kernel is applied. (b) CBFVs are compared to *Random Projection* of size 88 with no kernel applied (c) CBFVs are compared to *Random Projection* of size 88 with a RBF kernel applied

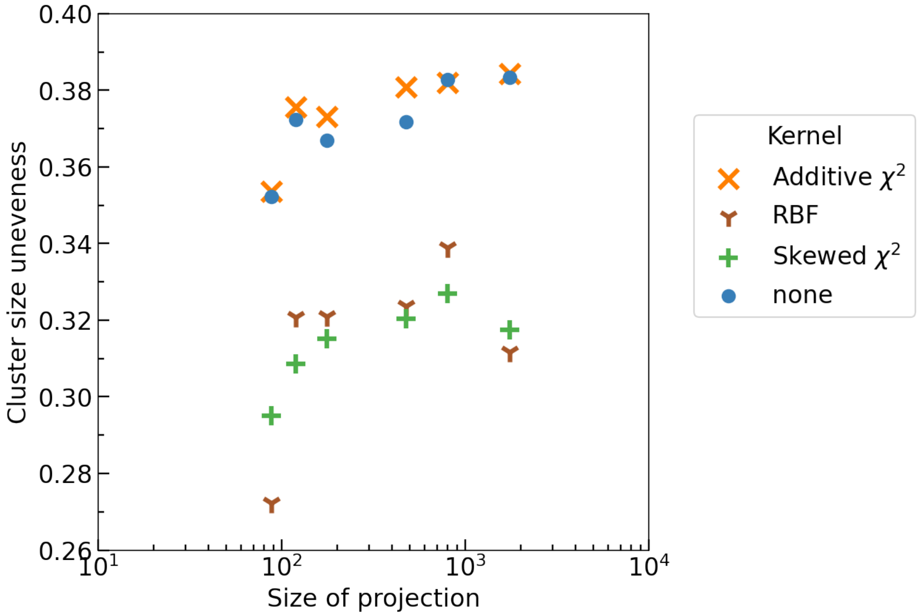


Figure 5.13: Average cluster size unevenness found using K-means clustering on datasets featurised using random projections of various sizes. Cluster size variances are normalised between 1 and 0 for each dataset (as different datasets would be expected to cluster with different amounts of ease), and then averaged for each size of random projection and each kernel. RBF and skewed χ^2 is seen to reduce cluster size uneveness, with the projections of approximately 100 dimensions performing better than larger projections.

5.6 Discussion

Recreation of the studies discussed in Section 5.2 shows that, broadly speaking, featurisation methods used in research are not necessarily advantageous over random projections, especially on tasks that are not related to band gaps. ML-led materials science research, including research presented here (Chapter 3), often aims to highlight the success of a ML model either in a materials discovery pipeline, as a proof of concept that a model can learn from a given dataset, or a proof of concept that a property can be predicted; this thesis is not outside of this criticism. As such, the exact implementation of a CBFV and its effectiveness compared to other CBFVs are often not included in the main text of a paper. Comparison studies thus facilitate evaluation of the impact of CBFVs on ML performance.

With modern libraries such as matminer [194], creating new featurisation methods and changing existing ones is straightforward. The engineered featurisation methods show no advantage over more widely used, or simpler alternatives, in the tasks considered here.

Both the findings here and in previous work suggest that for sufficiently large and balanced datasets, domain knowledge in CBFVs yields only a small advantage [124]. Promising results in representation learning could further reduce these advantages [60], which means the question as to whether these small advantages of feature-engineered CBFVs justify the difficulty in comparison between the models using them is an open one.

The choice of representation for a supervised ML algorithm may be influenced by the degree to which the goal of the algorithm is to maximise predictive accuracy for a property (*e.g.*, to screen potential candidates for synthesis), and the extent to which the goal is to gain insight into the causes of that property. Linked to this consideration is the question of whether domain knowledge features are being used as a proxy for the composition, or whether the composition is a proxy for the properties of a material which are quantified by the domain knowledge features.

For example, a model trained to predict whether a superconductor has a T_c greater than 30 K could be trained on a feature engineered CBFV and find that the number of d electrons is an important indicator for this property. A similar model could be trained using a *CompVec* representation and find that containing Cu is an important indicator for this property. Whether the number of d electrons serves as a proxy for the presence of Cu in a material or the presence of Cu in a material serves as a proxy for the number of d electrons is a matter of perspective. Bearing this difference in perspective in mind may help guide towards use of a representation which is best suited for the workflow in which a ML algorithm is being used. If ML is used to gain insight into the causes of properties and phenomena, then examining the importance of different domain knowledge areas in a CBFV for an algorithm will allow for that. This would suggest that the task becomes a matter of finding the best set of features for an element to adequately explain how a property interacts with the chemistries of a compound. At this point, experimenting with various combinations of elemental properties becomes appealing. However, to justify this approach, an adequate analysis of which properties are important is needed.

When choosing a representation to maximise predictive accuracy, domain knowledge seems to provide some advantage for some tasks examined here (particularly band gap prediction tasks). However, neither this evidence, nor that found in previous work [124], is sufficient to reject featurisation methods without domain knowledge, such as fractional encoding of composition or random projections, for more complex or parameter dependant algorithms. When using a CBFV, random projection offers a helpful baseline for performance, as it is simple to implement and works fairly well. Their single hyperparameter is the size of the projection, which allows one to draw conclusions as to the usefulness of a CBFV under investigation without introducing the size of a representation as a contributing factor for its performance.

Extrapolatory power is particularly pertinent in the field of materials discovery; therefore, previous work presented LOCO-CV as a way to estimate the extrapolatory power of a supervised ML algorithm [117]. LOCO-CV (along with many other linear algorithms such as principal component analysis), relies on linear separability in the data. This chapter has shown that, regardless of representation being used, kernel approximations such as RBF are advantageous in reducing cluster size unevenness and so should be strongly considered where such linear algorithms are applied. This reduction in cluster size unevenness tackles previously discussed caveats to LOCO-CV and results in more reliable model training (Figure 5.9).

Examination was done into the use of random projections to featurise chemical compositions to be used with kernelised LOCO-CV. As for other CBFVs examined, random projections used in conjunction with kernel methods produce more even clusters than without kernel approximation methods. However, no representation (either CBFV or random projection) consistently resulted in more even clusters than all other representations. Although most of the time CBFVs found more even clusters than random projections (with the exception of *CompVec*), these findings were not universal across datasets tested. Kernel approximation methods applied to random projections resulted in cluster sizes being even enough to be usable in the LOCO-CV algorithm without negatively impacting conclusions drawn from measurements taken using this method.

Random projections and kernelised LOCO-CV can be used together to create a generalised workflow to evaluate the extrapolatory power of a supervised ML algorithm (such as seen in Figure 5.1), which can be used regardless of the input representation to the ML algorithm in question. This can be combined with using a random projection as input representation to the ML algorithm to see a baseline measure of extrapolatory power which prospective CBFVs can be compared against to measure their usefulness.

5.6.1 Conclusion

Random projections are a generic and powerful way to featurise compositions for material property prediction. This is motivated by fundamental principles discussed in the

Johnson-Lendenstrauss lemma [33]; randomly projecting a composition vector can be used to move such vectors into a different dimensional space while preserving relationships between points in a dataset (within some error). These random projections have only a single hyperparameter (the size of the projection), which allows isolation of the relationships between the dimensionality of a representation and the predictive performance of algorithms trained using that representation. Random projections can be used as a baseline representation to examine what benefit is added by domain knowledge imbued into CBFVs.

Common CBFVs were investigated for use in ten property prediction tasks from the literature to establish what advantage domain knowledge offers in constructing such vectors. With the notable exception of band gap prediction tasks, CBFVs engineered to incorporate domain knowledge do not substantially outperform an equal-sized random projection for most prediction tasks investigated here. If the purpose of an ML model is to maximise predictive performance, the choice of using one of many complex representations (*e.g.*, CBFVs) should be justified by demonstrating an advantage over a random projection of the same size.

Kernelised LOCO-CV was presented to overcome issues with imbalanced cluster sizes that often occur when performing linear clustering on material sciences datasets. The application of kernel approximation methods, such as the RBF examined here, to data before K-means clustering leads to more even cluster sizes across many different datasets and input representations. Furthermore, using these kernel approximation clusters in LOCO-CV led to more reliable model training in the models examined here. Applying kernel approximation in LOCO-CV is independent of representations used by a supervised ML algorithm, so it is suggested that researchers looking to deploy LOCO-CV use the kernelised version presented here. Both random projections and kernelised LOCO-CV can be implemented independently or together.

More than 70 RF models were trained across ten property predictions tasks found in the materials science literature to show that random projections are a reliable baseline to use when evaluating a CBFV. More than 36,000 K-means clustering applications were evaluated, on the datasets used in these tasks and on the ICSD, and have shown that applying kernel functions to these data before K-means clustering results in more evenly sized clusters, and more reliable model training when these clusters are used in LOCO-CV. Findings presented here provide a basis for materials scientists in selecting and evaluating representations and laying out evaluation workflows.

5.7 Methods

Above experiments were implemented in Python using RF, K-means clustering and kernel method algorithms from the sci-kit learn library [140]. Hyperparameters of all sci-kit learn algorithms were set to default as of version 2.4.1, with the exception of the value of k for K-means clustering which was varied between 2 and 10 as needed for the LOCO-CV algorithm. While data standardisation was sometimes done before application of K-means clustering (as detailed in the supplementary information section S1), data standardisation was not done before application use of RFs as by their nature RFs consider dimensions independently making such standardisation redundant.

Graphs were plotted with the MatPlotLib library [73] with the exception of Figure 5.9 which was also uses the Seaborn library [196]. Featurisation was done using the utilities provided with the github associated with Murdock *et al.* [124], with the exception of *CompVec* which was implemented from scratch, and case study specific featurisations, which were obtained in supplementary information for the relevant case study. All implementations, are made available through the associated git repository as are data used in this study [40].

5.8 Thesis context

This chapter questions the assumptions about featurisation which were used in previous chapters (Chapters 3 and 4). This concludes most of the thesis' discussion about the uses of non-structural discriptors for materials properties prediction. Although non-structural descriptors are mentioned in subsequent chapters, their primary role becomes contextualising the work presented.

Similar to how this chapter discusses assumptions made in previous chapters (that featurisation methods may have a large effect on algorithmic performance), the next chapter examines assumptions and observations in this chapter. Namely, an exploration of the clustering such as that used in kerenlised LOCO-CV, examination of the shapes of clusters and discussion of methods which could be used to quantify clusters of unlabelled data.

5.9 Appendix

Table 5.2: Full table of results for the task of predicting T_c. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.83	120	11.0	5.37
CompVec	119	0.82	125	11.2	5.17
Stanev	145	0.88			
Oliynyk	176	0.83	122	11.1	5.33
fractional	476	0.82	130	11.4	5.24
RANDOM_200	800	0.83	121	11.0	5.47
JARVIS	1752	0.83	117	10.8	5.21
Random Projection	88	0.81	134	11.6	5.88
	119	0.81	132	11.5	5.86
	176	0.80	140	11.8	5.97
	476	0.81	132	11.5	5.78
	800	0.82	129	11.4	5.74
	1752	0.82	128	11.3	5.71
LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.39	199	12.7	7.89
CompVec	119	0.48	192	12.1	6.91
Oliynyk	176	0.25	204	13.0	8.18
fractional	476	0.49	180	11.9	6.87
RANDOM_200	800	0.49	177	11.9	7.28
JARVIS	1752	0.44	197	12.4	7.77
Kernelised LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.83	127	11.2	5.59
CompVec	119	0.83	123	11.1	5.32
Oliynyk	176	0.84	120	10.9	5.50
fractional	476	0.84	119	10.9	5.25
RANDOM_200	800	0.84	119	10.9	5.60
JARVIS	1752	0.85	114	10.7	5.36

Table 5.3: Full table of results for the task of predicting $T_\mathrm{c}|(T_\mathrm{c} > 10\text{ K})$. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.87	109	10.4	6.36
CompVec	119	0.86	118	10.9	6.44
Stanev	145	0.88			
Oliynyk	176	0.88	99.3	9.96	6.24
fractional	476	0.87	108	10.4	6.26
RANDOM_200	800	0.87	109	10.4	6.47
JARVIS	1752	0.88	103	10.1	6.25
Random Projection	88	0.84	134	11.6	7.05
	119	0.86	116	10.8	6.76
	176	0.85	124	11.1	6.82
	476	0.87	109	10.5	6.49
	800	0.86	113	10.6	6.66
	1752	0.86	119	10.9	6.70
LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.45	222	13.8	9.29
CompVec	119	0.47	198	12.9	8.27
Oliynyk	176	0.47	195	13.0	8.84
fractional	476	0.50	183	12.3	8.01
RANDOM_200	800	0.27	214	13.8	9.33
JARVIS	1752	0.44	197	13.1	8.87
Kernelised LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.88	105	10.2	6.42
CompVec	119	0.88	108	10.4	6.36
Oliynyk	176	0.88	103	10.1	6.35
fractional	476	0.88	103	10.1	6.22
RANDOM_200	800	0.87	109	10.4	6.57
JARVIS	1752	0.89	98.7	9.92	6.24

Table 5.4: Full table of results for the task of predicting $T_c > 10$ K. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

	80%/20% train/test split				
CBFV	dimensions	accuracy	f1	precision	recall
magpie	88	0.92	0.92	0.92	0.92
CompVec	119	0.92	0.92	0.92	0.92
Stanev	145	0.91	0.89	0.87	0.92
Oliynyk	176	0.92	0.92	0.92	0.92
fractional	476	0.92	0.92	0.92	0.92
*RANDOM*_200	800	0.92	0.92	0.92	0.92
JARVIS	1752	0.92	0.92	0.92	0.92
Random Projection	88	0.91	0.91	0.91	0.91
	119	0.91	0.91	0.91	0.91
	176	0.91	0.91	0.91	0.91
	476	0.91	0.91	0.91	0.91
	800	0.91	0.91	0.91	0.91
	1752	0.91	0.91	0.91	0.91
	LOCO-CV scores				
CBFV	dimensions	accuracy	f1	precision	recall
magpie	88	0.82	0.81	0.82	0.82
CompVec	119	0.84	0.83	0.84	0.84
Oliynyk	176	0.82	0.80	0.81	0.82
fractional	476	0.83	0.82	0.83	0.83
*RANDOM*_200	800	0.82	0.80	0.81	0.82
JARVIS	1752	1.0	1.0	1.0	1.0
	Kernelised LOCO-CV scores				
CBFV	dimensions	accuracy	f1	precision	recall
magpie	88	0.91	0.91	0.91	0.91
CompVec	119	0.91	0.91	0.91	0.91
Oliynyk	176	0.91	0.91	0.91	0.91
fractional	476	0.91	0.91	0.91	0.91
*RANDOM*_200	800	0.91	0.91	0.91	0.91
JARVIS	1752	1.0	1.0	1.0	1.0

Table 5.5: Full table of results for the task of predicting HH stability. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split					
CBFV	dimensions	accuracy	f1	precision	recall
LeGrain	51	0.99	0.99	0.99	0.99
magpie	88	1.0	0.99	1.0	1.0
CompVec	119	0.99	0.99	0.99	0.99
Oliynyk	176	0.99	0.99	0.99	0.99
fractional	476	0.99	0.99	0.99	0.99
*RANDOM*_200	800	0.99	0.99	0.99	0.99
JARVIS	1752	1.0	1.0	1.0	1.0
Random Projection	88	0.99	0.99	0.99	0.99
	119	0.99	0.98	0.99	0.99
	176	0.99	0.99	0.99	0.99
	476	0.99	0.98	0.99	0.99
	800	0.99	0.99	0.99	0.99
	1752	0.99	0.98	0.99	0.99

Table 5.6: Full table of results for the task of predicting E_{gap}(oxides). Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.71	1.57	1.25	0.934
CompVec	119	0.37	3.49	1.87	1.38
Davies	148	0.82	0.990	0.995	0.776
Oliynyk	176	0.77	1.26	1.12	0.854
fractional	476	0.45	3.05	1.75	1.32
RANDOM_200	800	0.42	3.22	1.79	1.41
JARVIS	1752	0.70	1.68	1.30	0.945
Random Projection	88	0.34	3.65	1.91	1.48
	119	0.27	4.01	2.00	1.58
	176	0.36	3.54	1.88	1.46
	476	0.37	3.46	1.86	1.42
	800	0.35	3.57	1.89	1.44
	1752	0.31	3.80	1.95	1.47

LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.49	2.29	1.47	1.16
CompVec	119	0.31	3.30	1.78	1.39
Oliynyk	176	0.53	2.05	1.40	1.10
fractional	476	0.27	3.45	1.83	1.43
RANDOM_200	800	0.23	3.51	1.85	1.47
JARVIS	1752	0.50	2.19	1.47	1.16
Davies	148	0.58	1.79	1.32	1.01

Kernelised LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.73	1.44	1.20	0.908
CompVec	119	0.52	2.56	1.59	1.17
Oliynyk	176	0.75	1.34	1.15	0.868
fractional	476	0.52	2.55	1.59	1.18
RANDOM_200	800	0.52	2.57	1.60	1.25
JARVIS	1752	0.72	1.47	1.21	0.912
Davies	148	0.76	1.25	1.11	0.838

Table 5.7: Full table of results for the task of predicting ΔT_{x}. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.61	191	13.8	10.8
CompVec	119	0.64	177	13.3	10.2
Oliynyk	176	0.60	196	14.0	10.8
Ward	213	0.68	159	12.6	9.80
fractional	476	0.58	209	14.4	11.1
RANDOM_200	800	0.59	202	14.2	11.1
JARVIS	1752	0.61	193	13.9	10.8
Random Projection	88	0.68	160	12.6	9.93
	119	0.65	172	13.1	10.3
	176	0.64	178	13.4	10.5
	476	0.67	163	12.8	10.1
	800	0.67	164	12.8	9.99
	1752	0.68	158	12.6	9.96

LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	-0.29	524	22.2	17.8
CompVec	119	-0.11	450	20.9	16.8
Oliynyk	176	-0.20	478	21.4	17.3
Ward	213	-0.020	418	19.9	16.0
fractional	476	-0.19	471	21.5	16.9
RANDOM_200	800	-0.14	454	20.8	16.9
JARVIS	1752	-0.17	464	21.1	16.8

Kernelised LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.59	212	14.4	9.99
CompVec	119	0.63	195	13.8	9.45
Oliynyk	176	0.61	202	14.1	9.94
Ward	213	0.65	184	13.4	9.29
fractional	476	0.60	208	14.3	9.92
RANDOM_200	800	0.60	212	14.4	10.2
JARVIS	1752	0.61	205	14.2	10.0

Table 5.8: Full table of results for the task of predicting $D_{\max}$. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.69	0.904	0.951	0.271
CompVec	119	0.64	1.06	1.03	0.271
Oliynyk	176	0.60	1.17	1.08	0.289
Ward	213	0.65	1.03	1.02	0.282
fractional	476	0.61	1.12	1.06	0.286
RANDOM_200	800	0.69	0.908	0.953	0.277
JARVIS	1752	0.55	1.31	1.15	0.308
Random Projection	88	0.57	1.29	1.14	0.407
	119	0.64	1.09	1.04	0.389
	176	0.62	1.13	1.06	0.377
	476	0.56	1.31	1.14	0.397
	800	0.59	1.21	1.10	0.399
	1752	0.61	1.16	1.08	0.385
LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	-15.	6.46	2.14	1.22
CompVec	119	-7.4	5.39	1.94	0.780
Oliynyk	176	-20.	8.49	2.54	1.55
Ward	213	-3.2	3.75	1.73	0.470
fractional	476	-9.1	5.21	1.92	0.758
RANDOM_200	800	-27.	10.9	2.77	1.54
JARVIS	1752	-50.	15.6	3.27	2.07
Kernelised LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.62	2.11	1.37	0.292
CompVec	119	0.59	2.47	1.44	0.276
Oliynyk	176	0.57	2.45	1.46	0.299
Ward	213	0.64	2.06	1.34	0.273
fractional	476	0.61	2.32	1.40	0.285
RANDOM_200	800	0.57	2.54	1.47	0.308
JARVIS	1752	0.57	2.50	1.47	0.311

Table 5.9: Full table of results for the task of predicting GFA. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split					
CBFV	dimensions	accuracy	f1	precision	recall
magpie	88	0.88	0.88	0.88	0.88
CompVec	119	0.88	0.88	0.88	0.88
Oliynyk	176	0.88	0.88	0.88	0.88
Ward	213	0.89	0.89	0.89	0.89
fractional	476	0.87	0.87	0.87	0.87
*RANDOM*_200	800	0.87	0.87	0.87	0.87
JARVIS	1752	0.89	0.89	0.89	0.89
Random Projection	119	0.87	0.86	0.87	0.87
	176	0.87	0.87	0.87	0.87
	476	0.87	0.87	0.87	0.87
	800	0.87	0.87	0.87	0.87
	1752	0.87	0.87	0.87	0.87
LOCO-CV scores					
CBFV	dimensions	accuracy	f1	precision	recall
magpie	88	0.64	0.64	0.70	0.64
CompVec	119	0.73	0.72	0.74	0.73
Oliynyk	176	0.65	0.66	0.71	0.65
Ward	213	0.74	0.74	0.77	0.74
fractional	476	0.66	0.66	0.72	0.66
*RANDOM*_200	800	0.63	0.61	0.70	0.63
JARVIS	1752	0.56	0.57	0.71	0.56
Kernelised LOCO-CV scores					
CBFV	dimensions	accuracy	f1	precision	recall
magpie	88	0.88	0.88	0.88	0.88
CompVec	119	0.88	0.88	0.88	0.88
Oliynyk	176	0.88	0.88	0.88	0.88
Ward	213	0.88	0.88	0.88	0.88
fractional	476	0.87	0.87	0.87	0.87
*RANDOM*_200	800	0.87	0.87	0.87	0.87
JARVIS	1752	0.88	0.88	0.88	0.88

Table 5.10: Full table of results for the task of predicting E_{gap}(exptl). Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.85	0.394	0.628	0.433
CompVec	119	0.68	0.829	0.910	0.558
Oliynyk	176	0.85	0.397	0.630	0.422
fractional	476	0.75	0.633	0.796	0.513
RANDOM_200	800	0.63	0.947	0.973	0.575
JARVIS	1752	0.85	0.394	0.628	0.421
Random Projection	88	0.51	1.27	1.13	0.680
	119	0.57	1.11	1.05	0.647
	176	0.62	0.986	0.993	0.639
	476	0.59	1.06	1.03	0.623
	800	0.61	1.00	1.00	0.619
	1752	0.60	1.04	1.02	0.623

LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.52	0.982	0.978	0.721
CompVec	119	0.32	1.40	1.17	0.814
Oliynyk	176	0.60	0.810	0.892	0.673
fractional	476	0.35	1.33	1.14	0.807
RANDOM_200	800	0.38	1.29	1.12	0.828
JARVIS	1752	0.56	0.899	0.937	0.687

Kernelised LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.81	0.420	0.645	0.434
CompVec	119	0.66	0.765	0.871	0.535
Oliynyk	176	0.81	0.416	0.641	0.424
fractional	476	0.71	0.648	0.802	0.501
RANDOM_200	800	0.64	0.825	0.904	0.566
JARVIS	1752	0.82	0.395	0.626	0.418

Table 5.11: Full table of results for the task of predicting E_{gap}(DFT). Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.77	0.621	0.788	0.523
CompVec	119	0.66	0.922	0.960	0.663
Oliynyk	176	0.78	0.605	0.778	0.513
fractional	476	0.71	0.790	0.889	0.552
RANDOM_200	800	0.70	0.819	0.905	0.616
JARVIS	1752	0.79	0.572	0.756	0.502
Random Projection	88	0.54	1.23	1.11	0.841
	119	0.54	1.23	1.11	0.839
	176	0.56	1.18	1.09	0.819
	476	0.59	1.11	1.05	0.796
	800	0.60	1.09	1.04	0.790
	1752	0.61	1.04	1.02	0.769

LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.54	1.19	1.09	0.833
CompVec	119	0.32	1.77	1.32	0.988
Oliynyk	176	0.57	1.12	1.05	0.803
fractional	476	0.40	1.56	1.24	0.922
RANDOM_200	800	0.42	1.51	1.22	0.953
JARVIS	1752	0.58	1.08	1.03	0.795

Kernelised LOCO-CV scores					
CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.77	0.608	0.779	0.533
CompVec	119	0.63	0.982	0.991	0.686
Oliynyk	176	0.78	0.584	0.764	0.520
fractional	476	0.73	0.708	0.841	0.561
RANDOM_200	800	0.71	0.763	0.873	0.634
JARVIS	1752	0.79	0.556	0.745	0.510

Table 5.12: Full table of results for the task of predicting $E_{\mathrm{gap}}(\mathrm{DFT}) \cup E_{\mathrm{gap}}(\mathrm{exptl})$. Clusterings for LOCO-CV were done with *magpie* featurisation, and kernelised LOCO-CV was *magpie* featurisation with RBF kernel.

80%/20% train/test split

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.77	0.602	0.776	0.524
CompVec	119	0.63	0.955	0.977	0.673
Oliynyk	176	0.78	0.581	0.762	0.513
fractional	476	0.74	0.679	0.824	0.551
RANDOM_200	800	0.73	0.728	0.853	0.614
JARVIS	1752	0.79	0.555	0.745	0.504
Random Projection	88	0.54	1.20	1.10	0.834
	119	0.54	1.20	1.10	0.834
	176	0.56	1.15	1.07	0.812
	476	0.58	1.08	1.04	0.788
	800	0.59	1.07	1.03	0.779
	1752	0.60	1.03	1.02	0.765

LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.53	1.20	1.09	0.840
CompVec	119	0.32	1.76	1.32	0.986
Oliynyk	176	0.56	1.13	1.06	0.805
fractional	476	0.39	1.56	1.24	0.925
RANDOM_200	800	0.42	1.50	1.22	0.950
JARVIS	1752	0.58	1.09	1.04	0.797

Kernelised LOCO-CV scores

CBFV	dimensions	r^2	mse	rmse	mae
magpie	88	0.76	0.613	0.783	0.537
CompVec	119	0.62	0.981	0.990	0.686
Oliynyk	176	0.77	0.592	0.769	0.524
fractional	476	0.72	0.721	0.849	0.567
RANDOM_200	800	0.70	0.775	0.880	0.635
JARVIS	1752	0.78	0.567	0.753	0.515

task	CBFV	dimensions	accuracy		f1		precision		recall	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
$T_{\mathrm{c}} > 10\mathrm{K}$	*magpie*	88	0.92	0.0013	0.92	0.0013	0.92	0.0012	0.92	0.0013
	CompVec	119	0.92	0.0019	0.92	0.0019	0.92	0.0019	0.92	0.0019
	Oliynyk	176	0.92	0.0011	0.92	0.0011	0.92	0.0011	0.92	0.0011
	Random Projection	88	0.91	0.0016	0.91	0.0016	0.91	0.0016	0.91	0.0016
		119	0.91	0.000 67	0.91	0.000 68	0.91	0.0007	0.91	0.000 67
		176	0.91	0.0012	0.91	0.0012	0.91	0.0012	0.91	0.0012
GFA	*magpie*	88	0.88	0.0028	0.88	0.0029	0.88	0.0027	0.88	0.0028
	CompVec	119	0.88	0.0049	0.88	0.005	0.88	0.0049	0.88	0.0049
	Oliynyk	176	0.88	0.003	0.88	0.003	0.88	0.003	0.88	0.003
	Random Projection	88	0.87	0.0033	0.87	0.0034	0.87	0.0033	0.87	0.0033
		119	0.87	0.004	0.87	0.0042	0.87	0.0039	0.87	0.004
		176	0.87	0.0017	0.87	0.0017	0.87	0.0018	0.87	0.0017
HH stability	*magpie*	88	1.0	0.0	0.99	0.0	1.0	0.0	1.0	0.0
	CompVec	119	0.99	0.000 24	0.99	0.000 41	0.99	0.000 24	0.99	0.000 24
	Oliynyk	176	0.99	0.000 45	0.99	0.000 58	0.99	0.000 44	0.99	0.000 45
	Random Projection	88	0.99	0.0	0.99	0.0	0.99	0.0	0.99	0.0
		119	0.99	0.0	0.98	0.0	0.99	0.0	0.99	0.0
		176	0.99	0.000 24	0.99	0.000 49	0.99	0.000 24	0.99	0.000 24

Table 5.13: The mean and standard deviation of various metrics of classification tasks across 5 repeats measured using an 80/20 train/test fit. Note that for the HH stability task, the highly unbalanced nature of the dataset results in unusually repeatable and high performing results.

task	CBFV	dimsensions	accuracy		f1		precision		recall	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
$T_c > 10$K	*magpie*	88	0.82	0.000 92	0.81	0.0016	0.82	0.0015	0.82	0.000 92
	CompVec	119	0.84	0.000 45	0.83	0.000 24	0.83	0.000 19	0.84	0.000 45
	Oliynyk	176	0.82	0.0016	0.80	0.0020	0.82	0.0024	0.82	0.0016
	Random Projection	88	0.64	0.0014	0.53	0.0022	0.64	0.012	0.64	0.0014
		119	0.64	0.0012	0.53	0.0014	0.65	0.012	0.64	0.0012
		176	0.64	0.0012	0.53	0.0015	0.65	0.014	0.64	0.0012
GFA	*magpie*	88	0.64	0.011	0.64	0.0081	0.70	0.0046	0.64	0.011
	CompVec	119	0.72	0.0017	0.72	0.0018	0.75	0.0033	0.72	0.0017
	Oliynyk	176	0.65	0.0069	0.66	0.0046	0.71	0.0032	0.65	0.0069
	Random Projection	88	0.53	0.0083	0.50	0.0064	0.61	0.0027	0.53	0.0083
		119	0.53	0.011	0.49	0.0091	0.62	0.010	0.53	0.011
		176	0.52	0.012	0.49	0.010	0.61	0.0057	0.52	0.012
HH stability	*magpie*	88	0.98	0.000 41	0.98	0.000 36	0.97	0.000 37	0.98	0.000 41
	CompVec	119	0.97	0.000 45	0.97	0.000 38	0.97	0.000 78	0.97	0.000 45
	Oliynyk	176	0.98	0.000 39	0.97	0.000 34	0.97	0.000 50	0.98	0.000 39
	Random Projection	88	0.97	0.000 55	0.96	0.000 70	0.95	0.0019	0.97	0.000 55
		119	0.97	0.000 48	0.96	0.000 67	0.95	0.0015	0.97	0.000 48
		176	0.97	0.000 42	0.96	0.000 57	0.96	0.0017	0.97	0.000 42

Table 5.14: The mean and standard deviation of various metrics of classification tasks across 5 repeats measured using LOCO-CV without any kernels

task	CBFV	dimensions	accuracy		f1		precision		recall	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
$T_c > 10$K	*magpie*	88	0.91	0.000 43	0.91	0.000 43	0.91	0.000 43	0.91	0.000 43
	CompVec	119	0.91	0.000 47	0.91	0.000 47	0.91	0.000 48	0.91	0.000 47
	Oliynyk	176	0.91	0.000 59	0.91	0.000 60	0.91	0.000 61	0.91	0.000 59
	Random Projection	88	0.68	0.0019	0.58	0.0028	0.74	0.0097	0.68	0.0019
		119	0.68	0.0017	0.58	0.0027	0.74	0.0087	0.68	0.0017
		176	0.68	0.0017	0.58	0.0027	0.74	0.0066	0.68	0.0017
GFA	*magpie*	88	0.88	0.000 58	0.87	0.000 60	0.88	0.000 57	0.88	0.000 58
	CompVec	119	0.88	0.0011	0.88	0.0011	0.88	0.0011	0.88	0.0011
	Oliynyk	176	0.88	0.000 72	0.88	0.000 72	0.88	0.000 66	0.88	0.000 72
	Random Projection	88	0.55	0.0054	0.51	0.0039	0.61	0.0056	0.55	0.0054
		119	0.54	0.010	0.51	0.0080	0.61	0.0056	0.54	0.010
		176	0.53	0.0065	0.51	0.0050	0.61	0.0049	0.53	0.0065
HH stability	*magpie*	88	0.98	0.000 29	0.98	0.000 34	0.98	0.000 30	0.98	0.000 29
	CompVec	119	0.97	0.000 32	0.97	0.000 34	0.97	0.000 49	0.97	0.000 32
	Oliynyk	176	0.98	0.000 43	0.98	0.000 37	0.98	0.000 46	0.98	0.000 43
	Random Projection	88	0.97	0.000 88	0.96	0.0012	0.96	0.0027	0.97	0.000 88
		119	0.97	0.000 74	0.95	0.0011	0.95	0.0022	0.97	0.000 74
		176	0.97	0.000 80	0.96	0.0012	0.95	0.0029	0.97	0.000 80

Table 5.15: The mean and standard deviation of various metrics of classification tasks across 5 repeats measured using kernelised LOCO-CV (using radial basis function kernel)

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
D_{max}	*magpie*	88	0.68	0.020	0.97	0.031	0.27	0.0057
	CompVec	119	0.65	0.014	1.0	0.020	0.27	0.0032
	Oliynyk	176	0.61	0.018	1.1	0.025	0.29	0.0039
	Random Projection	88	0.56	0.029	1.1	0.037	0.40	0.0069
		119	0.63	0.012	1.0	0.017	0.38	0.0057
		176	0.61	0.019	1.1	0.027	0.39	0.0060
E_{gap}(DFT)∪ E_{gap}(exptl)	*magpie*	88	0.77	0.000 93	0.77	0.0016	0.52	0.000 96
	CompVec	119	0.63	0.0011	0.98	0.0015	0.68	0.0015
	Oliynyk	176	0.78	0.0012	0.76	0.0021	0.51	0.000 70
	Random Projection	88	0.54	0.0012	1.1	0.0014	0.83	0.0010
		119	0.54	0.0012	1.1	0.0014	0.83	0.0012
		176	0.56	0.0023	1.1	0.0027	0.81	0.0021
E_{gap}(DFT)	*magpie*	88	0.77	0.0012	0.79	0.0021	0.52	0.000 76
	CompVec	119	0.66	0.0018	0.96	0.0025	0.66	0.0016
	Oliynyk	176	0.78	0.0013	0.78	0.0022	0.51	0.000 58
	Random Projection	88	0.54	0.000 93	1.1	0.0011	0.84	0.0010
		119	0.54	0.0015	1.1	0.0018	0.84	0.0019
		176	0.56	0.0027	1.1	0.0034	0.82	0.0016
E_{gap}(exptl)	*magpie*	88	0.84	0.0032	0.63	0.0065	0.43	0.0023
	CompVec	119	0.68	0.0052	0.91	0.0074	0.56	0.0044
	Oliynyk	176	0.84	0.0035	0.64	0.0069	0.43	0.0021
	Random Projection	88	0.51	0.0060	1.1	0.0068	0.68	0.0045
		119	0.58	0.0069	1.0	0.0085	0.64	0.0075
		176	0.61	0.0059	1.0	0.0076	0.65	0.0036

This table is continued on the next page

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
E_{gap}(oxides)	*magpie*	88	0.71	0.0054	1.3	0.012	0.94	0.0081
	CompVec	119	0.36	0.019	1.9	0.027	1.4	0.022
	Oliynyk	176	0.76	0.0051	1.1	0.012	0.86	0.012
	Random Projection	88	0.35	0.015	1.9	0.021	1.5	0.012
		119	0.27	0.011	2.0	0.014	1.6	0.016
		176	0.35	0.0099	1.9	0.014	1.5	0.019
$T_{\mathrm{c}}\|(T_{\mathrm{c}} > 10\mathrm{K})$	*magpie*	88	0.87	0.000 84	10	0.034	6.3	0.039
	CompVec	119	0.86	0.0016	11	0.061	6.4	0.045
	Oliynyk	176	0.88	0.000 34	10	0.014	6.2	0.028
	Random Projection	88	0.84	0.0018	12	0.064	7.0	0.050
		119	0.86	0.0010	11	0.040	6.8	0.012
		176	0.85	0.0016	11	0.061	6.8	0.041
T_{c}	*magpie*	88	0.83	0.0013	11	0.041	5.4	0.018
	CompVec	119	0.82	0.000 79	11	0.025	5.2	0.015
	Oliynyk	176	0.83	0.000 47	11	0.015	5.3	0.021
	Random Projection	88	0.81	0.000 97	12	0.030	5.9	0.024
		119	0.81	0.0013	12	0.040	5.9	0.019
		176	0.80	0.0018	12	0.055	5.9	0.018
ΔT_{x}	*magpie*	88	0.60	0.0049	14	0.086	11	0.040
	CompVec	119	0.65	0.011	13	0.20	10	0.19
	Oliynyk	176	0.60	0.0058	14	0.10	11	0.044

This table is continued on the next page

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
	Random Projection	88	0.67	0.0060	13	0.12	9.9	0.14
		119	0.67	0.0069	13	0.13	10	0.18
		176	0.65	0.0063	13	0.12	10	0.12

Table 5.16: The mean ($\bar{x}$) and standard deviation (σ) of r^2, mean squared error (mse), root mean squared error (rmse) and mean absolute error (mae) of regression tasks across 5 repeats measured using an 80/20 train/test split. Unlike Tables 5.17 and 5.18, none of the r^2 values found using this method were less than 0.

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
$D_{\max}$	*magpie*	88	−15		2.1	0.018	1.2	0.015
	CompVec	119	−7.4		2.0	0.027	0.78	0.013
	Oliynyk	176	−21		2.6	0.039	1.6	0.021
	Random Projection	88	−510		4.6	0.13	3.7	0.12
		119	−210		4.2	0.21	3.4	0.2
		176	−240		4.2	0.18	3.4	0.18
E_{gap}(DFT)∪ E_{gap}(exptl)	*magpie*	88	0.53	0.000 36	1.1	0.000 48	0.84	0.000 48
	CompVec	119	0.38	0.000 46	1.3	0.000 46	0.93	0.0007
	Oliynyk	176	0.56	0.000 65	1.1	0.000 86	0.8	0.000 35
	Random Projection	88	−0.12		1.5	0.0032	1.2	0.0023
		119	−0.026		1.5	0.0039	1.2	0.0031
		176	−0.05		1.5	0.0024	1.2	0.0013
E_{gap}(DFT)	*magpie*	88	0.54	0.000 61	1.1	0.000 84	0.83	0.000 87
	CompVec	119	0.38	0.000 36	1.3	0.000 33	0.93	0.000 57
	Oliynyk	176	0.57	0.000 65	1.1	0.000 92	0.8	0.001
	Random Projection	88	−0.13		1.5	0.004	1.2	0.0027
		119	−0.022		1.5	0.0059	1.2	0.0044
		176	−0.061		1.5	0.0054	1.2	0.0037
E_{gap}(exptl)	*magpie*	88	0.52	0.0045	0.98	0.0034	0.72	0.0027
	CompVec	119	0.28	0.0033	1.2	0.000 64	0.79	0.0014
	Oliynyk	176	0.6	0.0032	0.89	0.0024	0.67	0.0016
	Random Projection	88	−0.6		1.4	0.008	1.1	0.0057
		119	−0.54		1.5	0.0076	1.2	0.0055
		176	−1.2		1.6	0.034	1.2	0.034

This table is continued on the next page

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
E_{gap}(oxides)	*magpie*	88	0.49	0.007	1.5	0.0058	1.2	0.0052
	CompVec	119	0.3	0.0056	1.8	0.0057	1.4	0.0054
	Oliynyk	176	0.53	0.0046	1.4	0.004	1.1	0.0027
	Random Projection	88	0.22	0.018	2.0	0.021	1.6	0.019
		119	0.19	0.011	2.1	0.011	1.7	0.01
		176	0.26	0.0041	2.0	0.0051	1.6	0.0064
$T_{\mathrm{c}}\|(T_{\mathrm{c}} > 10\mathrm{K})$	*magpie*	88	0.45	0.026	14	0.048	9.3	0.053
	CompVec	119	0.45	0.025	13	0.087	8.3	0.07
	Oliynyk	176	0.48	0.014	13	0.043	8.8	0.03
	Random Projection	88	−16		21	0.29	17	0.28
		119	−21		23	0.24	19	0.15
		176	−32		22	0.29	19	0.3
T_{c}	*magpie*	88	0.39	0.0059	13	0.089	7.9	0.054
	CompVec	119	0.48	0.0033	12	0.046	6.9	0.039
	Oliynyk	176	0.23	0.0096	13	0.07	8.2	0.038
	Random Projection	88	−1.3		17	0.18	13	0.12
		119	−0.97		16	0.14	13	0.12
		176	−0.99		17	0.11	13	0.07
ΔT_{x}	*magpie*	88	−0.31		22	0.12	18	0.1
	CompVec	119	−0.092		21	0.15	17	0.12
	Oliynyk	176	−0.19		21	0.13	17	0.098

This table is continued on the next page

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
	Random Projection	88	−1.7		27	0.21	22	0.16
		119	−0.52		23	0.12	18	0.063
		176	−0.66		23	0.17	19	0.14

Table 5.17: The mean ($\bar{x}$) and standard deviation (σ) of r^2, mean squared error (mse), root mean squared error (rmse) and mean absolute error (mae) of regression tasks across 5 repeats measured using LOCO-CV. As r^2 has no lower bound, standard deviations of r^2 were not included when calculating the standard deviation, where none of the repeats found an $r^2 > 0$, no standard deviation has been reported.

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
D_{max}	*magpie*	88	0.63	0.012	1.4	0.021	0.29	0.000 81
	CompVec	119	0.6	0.0075	1.4	0.0078	0.27	0.0021
	Oliynyk	176	0.58	0.01	1.4	0.015	0.29	0.000 75
	Random Projection	88	−120		4.6	0.17	3.5	0.1
		119	−61		4.1	0.18	3.1	0.099
		176	−57		4.0	0.066	3.1	0.03
E_{gap}(DFT)∪ E_{gap}(exptl)	*magpie*	88	0.77	0.000 75	0.78	0.0012	0.54	0.000 41
	CompVec	119	0.71	0.001	0.87	0.0014	0.58	0.000 49
	Oliynyk	176	0.77	0.000 44	0.77	0.000 69	0.52	0.000 39
	Random Projection	88	0.045	0.013	1.4	0.0084	1.1	0.0075
		119	0.1	0.0083	1.4	0.0085	1.1	0.0069
		176	0.083	0.0081	1.4	0.0056	1.1	0.0059
E_{gap}(DFT)	*magpie*	88	0.77	0.000 13	0.78	0.0002	0.53	0.000 22
	CompVec	119	0.72	0.000 34	0.87	0.000 49	0.57	0.000 21
	Oliynyk	176	0.78	0.000 25	0.76	0.000 42	0.52	0.000 34
	Random Projection	88	0.04	0.0032	1.4	0.0074	1.1	0.0071
		119	0.11	0.01	1.4	0.0084	1.1	0.0082
		176	0.083	0.0087	1.4	0.0069	1.1	0.0069
E_{gap}(exptl)	*magpie*	88	0.81	0.0014	0.65	0.0025	0.43	0.000 97
	CompVec	119	0.69	0.0022	0.83	0.0022	0.51	0.000 69
	Oliynyk	176	0.81	0.000 78	0.64	0.0019	0.42	0.000 94
	Random Projection	88	−0.38		1.4	0.0054	1.1	0.0062
		119	−0.43		1.5	0.01	1.1	0.01
		176	−0.6		1.5	0.013	1.1	0.014

This table is continued on the next page

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
E_{gap}(oxides)	*magpie*	88	0.72	0.004	1.2	0.0067	0.91	0.0054
	CompVec	119	0.51	0.0032	1.6	0.0049	1.2	0.0038
	Oliynyk	176	0.75	0.0024	1.2	0.0046	0.87	0.0039
	Random Projection	88	0.24	0.011	2.0	0.021	1.6	0.022
		119	0.21	0.012	2.0	0.022	1.6	0.018
		176	0.26	0.015	2.0	0.024	1.6	0.02
$T_{\mathrm{c}} \| (T_{\mathrm{c}} > 10\mathrm{K})$	*magpie*	88	0.88	0.0003	10	0.012	6.4	0.0091
	CompVec	119	0.87	0.0009	10	0.034	6.4	0.021
	Oliynyk	176	0.88	0.000 76	10	0.03	6.3	0.013
	Random Projection	88	−16		21	0.12	17	0.074
		119	−22		24	0.41	20	0.37
		176	−39		23	0.35	19	0.26
T_{c}	*magpie*	88	0.83	0.000 83	11	0.028	5.6	0.013
	CompVec	119	0.83	0.000 62	11	0.021	5.3	0.011
	Oliynyk	176	0.84	0.000 77	11	0.026	5.5	0.0092
	Random Projection	88	−1.4		17	0.11	13	0.08
		119	−0.56		15	0.07	12	0.054
		176	−0.92		18	0.14	13	0.077
ΔT_{x}	*magpie*	88	0.6	0.009	14	0.11	9.9	0.057
	CompVec	119	0.64	0.011	14	0.11	9.3	0.057
	Oliynyk	176	0.62	0.011	14	0.14	9.9	0.057

This table is continued on the next page

task	CBFV	dims.	r^2		rmse		mae	
			$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
	Random Projection	88	−1.5		27	0.18	22	0.18
		119	−0.5		23	0.22	18	0.21
		176	−0.68		23	0.11	19	0.13

Table 5.18: The mean ($\bar{x}$) and standard deviation (σ) of r^2, mean squared error (mse), root mean squared error (rmse) and mean absolute error (mae) of regression tasks across 5 repeats measured using LOCO-CV with radial basis function kernel. As r^2 has no lower bound, values of r^2 lower than 0 were excluded when calculating σ. Where none of the repeats found an $r^2 > 0$, no σ has been reported.

Chapter 6

Mathematically quantifying isotropy

This section contains results under peer review for Applied Intelligence [44]

6.1 Introduction

Clustering algorithms have become a vital tool in materials science for tasks such as machine learning evaluation [45, 159] and data exploration [204]. In the field of materials science, datasets can often be high-dimensional and lack target labels, making the task of clustering data a challenging one. The appropriate representation of a material is often unclear (Chapter 5) [45, 124], and manual evaluation of identified clusters is infeasible due to the size of the datasets [69, 38, 78].

The previous chaptered saw use of K-means clustering on materials data investigated. As part of this analysis attempts were made to examine defference in clustering before and after kernel approximation application. In visualisation of these results, it was noticed that the shape of clusters after kernel approximation application was more isotropic with no kernel approximation applied (Figures 5.6c and 5.6d). However it was unclear whether this difference could be quantified. As such an investigation followed into measuring the isotropy of a set of clusters, which is presented in this chapter. While somewhat tangetial to the initial research question, it is hoped that this investigation remains interesting.

Recent publications in materials science have introduced new clustering techniques that can work in a supervised or semi-supervised manner [138]. Although new techniques show promise and are seeing uptake [159, 137, 122, 14], like many other clustering algorithms [108, 49, 191], these techniques rely on distance similarity metrics (usually Euclidean distance) in order to cluster the data. Thus, these algorithms depend on the representation of the materials. But it is unclear which material representation is most appropriate; thus, evaluating the results of clustering is important, regardless of which clustering algorithm is used. As datasets can be too large to evaluate manually, semi-automatic, or automatic metrics must be used to quantify characteristics of clusters and the success of clustering. This chapter discusses and presents such metrics.

Existing metrics for clustering unlabelled data, such as silhouette or Folkes–Mallows scores, focus on quantifying the compactness of individual clusters, and the separation of clusters from each other [107]. One aspect of sets of clusters, which remains difficult to quantify, is their average shape. This chapter focusses on the isotropy of a cluster: Do the points in a cluster form a round shape, or are they a "spikier" shape. Although other shape markers, such as squareness or hexagonality, may be given more importance, isotropy could still be a relevant factor to some researchers.

Isotropy of data representations has been associated with improved performance in downstream machine learning tasks [123], and anisotropic clusters of data may be indicative of representations dominated by specific features or correlations of features. Where representation is unclear, such as in materials science, exploring the effects of representation on the isotropy of clusters can be informative to a researcher.

The shape of a cluster can be visually observed by projection of data onto a lower dimensional space [56, 180], but this projection excludes significant amounts of information, and observations about these projections are subjective. For example, previous work used visual inspection of Principal Component Analysis (PCA) to project representations of the Inorganic Crystal Structure Database (ICSD) in 3 dimensions and qualitatively described changes in cluster shape upon application of different kernel approximation methods [45]. Although qualitative observation can be helpful, there is an unmet need for robust metrics to quantify the isotropy of clusters and understand the underlying structure of the data. As will be seen, such numerical analysis can highlight non-intuitive results, which are present at higher dimensions.

The chapter expands on existing methods to quantify cluster shape by measuring the isotropy ("roundness") or anisotropy ("spikiness"). The robustness of existing metrics is examined, and the metrics are then expanded upon from use on one cluster to use on a set of clusters. Metrics of this kind are commonly used in three dimensions in the field of medical imaging to identify the diffusion of water in the brain [48]. In higher dimensions, similar metrics arise in the context of data science to draw conclusions about the shape of point clouds [123]. This chapter analyses of existing methods in high dimensions using random matrix theory and describes an alternative implementation for an existing derivation of isotropy. The concept of measuring isotropy is extended to examine sets of clusters, and example uses of this extension are provided in the material science and data science domains.

After a brief introduction to metrics for unlabelled clustering and metrics for isotropy, an existing derivation is expanded to define a new isotropy metric. The usefulness of this new isotropy metric is demonstrated for quantifying the clustering of the ICSD, which is one of the foundational datasets in inorganic chemistry and materials science. These metrics of isotropy are used to quantify the shape of clusters within the ICSD when using different representations and non-linear (kernel-approximation) transformations.

Although this technique was developed in a material science context, it has a broader applicability wherever the shape of sets of clusters must be quantified. For example, analysing learnt embeddings is a common task in machine learning [66, 177, 106] and often relies on low dimensional visualisation methods [56, 180]. As such, the usefulness of this metric is further demonstrated by using it to quantify differences in learnt embeddings of images of digits using the Modified National Institute of Standards and Technology (MNIST) dataset, a foundational data science dataset [99].

To examine the difference between the metrics for isotropy used, clusters of random points are generated in various dimensionalities and their isotropy is measured. Using mathematical tools from the field of random matrix theory [113] an explaination is offered for the behaviour observed in the existing metrics.

In the field of materials science an “anisotropic material” is often used to describe materials with elongated conventional unit cells. However, conventional unit cells are decided with some level of human judgment (Section 2.2.2). Having an objective mathematical measure for this would be beneficial. As such, isotropy metrics presented here were used on unit cells from Ruddlesden-Popper and Kagome type structures to see if they could objectively measure what would be considered an isotropic material. However, this measurement is found to be highly correlated with $\frac{c}{a}$ ratio of the unit cell and, as such, depends on the definition of the unit cell. Thus, when quantifying the isotropy of a material (rather than that of a cluster of materials), the measures investigated here do not offer more objective measures than existing methods.

Finally, time complexity of the proposed metrics is examined, and the advantages and disadvantages of each metric are discussed. Concluding remarks are then made.

The specific contributions of this paper are as follows:

- Exploring how metrics used for measuring isotropy in 3 dimensions [10] generalise to higher dimensions.
- Providing a new implementation for an isotropy measure based on an existing mathematical derivation (Section 6.3.1).
- Proposing adaptions to the measures of isotropy for single clusters such that one can measure the average isotropy across a set of clusters (Section 6.3.2).
- Highlighting the need for analysis of representation when clustering datasets relating to materials (Section 6.4.1).
- Demonstrating analysis of isotropy in a supervised learning context using a foundational data science dataset (Section 6.4.2).
- Examining the robustness of the metrics under random noise perturbations (Section 6.4.3).
- Using random matrix theory to prove that the measurements of isotropy are related to the dimensionality of data, especially if the data are noisy (Section 6.4.3).

6.2 Metrics for unsupervised clustering

While supervised clustering tasks allow the use of metrics such as the adjusted mutual information score [182], Folkes–Mallows score [54], or homogeneity and completeness scores [157], the selection for unsupervised clustering is more sparse. Unsupervised metrics must rely on features present in the data, and thus are dependent on data representation. Due to this dependency, such metrics can be referred to as “internal clustering validation measures” [107]. Internal clustering validation measures aim to quantify the quality of a set of clusters in an abstract sense, by focussing on either the compactness of each cluster or the separation between clusters. To contextualise the new implementations presented here, examples of prominent unsupervised clustering metrics are outlined in this section.

Using distance metrics such as the Euclidean distance, one can compute the average distance between each point and every other point in its cluster. This can become computationally expensive because the calculation of the pairwise distance matrix scales with the square of the number of points (Table 6.1). Thus, the average distance between a point in a cluster and its centroid can be used instead, which scales linearly with the number of points in the cluster. These computations provide a measure of how tightly packed a cluster is in the space distance is being measured over (for example, the Euclidean space). This is useful for numerically comparing clusters and clusterings of points which exist in the same space, such as comparing clusters found using different clustering algorithms. The representation of data, and any transformations will affect the distance measurements, so, this use of distance metric based quantification of clusters is inadequate for making comparisons between sets of clusters found on different representations of data.

The silhouette score uses distance measurements to provide a number bounded between -1 and 1 to measure how well a point is clustered [162]. Where 1 is considered a well clustered point (*i.e.*, according to this metric the point is in the correct cluster) and -1 is considered a poorly clustered point (*i.e.*, according to this metric this point should be in a different cluster). It is calculated by comparing the mean distance between a point and other members of that point's cluster, to the mean distance of that point and all members of its next closest cluster (the cluster who's members are on average closest to that point). By calculating the mean silhouette score for all points one can obtain a score for the quality of a set clusters. This score can be compared to silhouette scores found using alternative representations of data points or kernel transformations.

The Davies–Bouldin index provides a lower bounded metric of clusters without requiring a pairwise distance between points in the dataset [96]. By comparing distances of points in a cluster to its centroid and distances between cluster centroids, a score is calculated with a minimum of 0, where lower scores indicate better clusterings.

The Calinski–Harabasz measure (or variance ratio criteron) [21] is a metric that considers both the dispersion and the separation of the clusters. It is calculated by using the sum of square distances between a point in a cluster and its centroid and the sum of square distances of the cluster's centroid from the mean data point in the dataset. In other words, it compares the dispersion within clusters to the dispersion of centroids in the representation space. Unlike the Davies–Bouldin index, a higher Calinski–Harabasz measure indicates a more separated set of clusters.

Although evaluation based on dispersion and cluster compactness provides a metric of how “good” an application of clustering is, it does not provide information about the clusters themselves. It can be relevant to the use case of clustering algorithms to have evenly sized clusters (*i.e.* clusters should contain approximately the same number of data points) [117]. In this case, the variance between cluster size has been used as a metric [45].

Another property which has been difficult to reason with about clusters is their shape. We present a novel application of isotropy metrics in a clustering context.

Table 6.1: A summary of metrics for unsupervised learning. Included are descriptions, optimal values where applicable, whether their output has upper bounds and/or lower bounds ($L.B.$), and approximations of time complexity. Where $\mathcal{D}$ is a set of clusters, $|\mathcal{C}|$ is the size of a cluster, $\mathcal{E}$ is the set of all points in all clusters of $\mathcal{D}$ (thus $|\mathcal{E}| = \Sigma_{\mathcal{C}\in\mathcal{D}}|\mathcal{C}|$), n is the number of dimensions in which the cluster exists and r is the number of random vectors used for I_{rnd}.

Metric	Description	Optimal value	Bounded	Complexity
Mean distance to centroid	Measures compactness of clusters	Min	$L.B. = 0$	$O(\lvert\mathcal{E}\rvert)$
Mean distance between points in cluster	Measures compactness of clusters	Min	$L.B. = 0$	$O\left(\lvert\mathcal{E}\rvert\lvert\mathcal{C}\rvert\right)$
Silhouette score	Measures how close a point is to other points in its cluster compared to points in other clusters	Max	$-1, 1$	$O\left(\lvert\mathcal{E}\rvert^2\right)$
Davies–Bouldin index	Ratio of within cluster distances to between cluster distances	Min	$L.B. = 0$	$O(\lvert\mathcal{E}\rvert)$ [*]
Calinski–Harabasz measure	Ratio of between and within cluster dispersion, weighted by the size of the cluster	Max	$L.B. = 0$	$O(\lvert\mathcal{E}\rvert)$
Cluster size variance	Measures how evenly sized clusters are	N/A [†]	$L.B. = 0$	$O(\lvert\mathcal{D}\rvert)$
Fractional isotropy	Measures the shape of clusters	N/A [‡]	0, 1	$O(\lvert\mathcal{E}\rvert n^2)$ [§]
Isotropy (EigenVec)	Measures the shape of clusters	N/A [¶]	$0, 1$	$O(\lvert\mathcal{E}\rvert n^2)$ [§]
Isotropy (random)	Measures the shape of clusters	N/A [¶]	$0, 1$	$O(\lvert\mathcal{E}\rvert r)$

[*] Assuming $|D| > |D|^2$ otherwise $O(|D|^2)$.

[†] Lower indicates more evenly sized clusters.

[‡] Lower indicates more isotropic.

[¶] Higher indicates more isotropic.

[§] Assuming $n < |\mathcal{C}|$ else $O(|\mathcal{E}||\mathcal{C}|n)$.

6.3 Metrics for Isotropy

As discussed above, the shape of the clusters is often an important consideration. A cluster with an isotropic shape, where the distribution of points is roughly equal in all directions, can be preferable to a cluster with a highly elongated shape [123]. Elongated clusters could be indicative of outliers: In clustering methods that only consider single linkage when creating clusters (*e.g*, hierarchical agglomerative clustering [111] or iterative label spreading [138]), long chains of outliers can be grouped together [111]. This sections discusses possible pitfalls when attempting to measure isotropy, and explore various metrics for measuring the isotropy of clusters, providing a quantitative way to evaluate and compare different cluster shapes.

One common pitfall when analysing the isotropy of a cluster of points is that it can be highly subjective as to whether a cluster is anisotropic or isotropic. For example, a spiral cluster (Figure 6.1a) may appear anisotropic if the spirals are loose. However, as spirals become closer together, or longer it becomes more subjective as to whether this cluster can be considered anisotropic or if the spirals have collapsed into a single isotropic cluster. Similarly, an L-shaped cluster (Figure 6.1d) may seem anisotropic, but could arguably be two isotropic clusters which have been wrongly grouped. Reducing complex correlations to single numbers will necessarily remove some information. Isotropy may be a secondary consideration compared to other markers of shape (for example, spiral, square, or hexagonal). Nevertheless, when isotropy is a primary concern for a researcher, having the ability to quantify isotropy is useful for analytical and descriptive purposes.

Although metrics such as kurtosis or variance may measure the spread of a cluster in individual dimensions, anisotropy may occur between dimensions (Figure 6.1d) thus, a more complex statistical analysis must be used. Two important properties that a measure of isotropy must have are *(a)* being invariant against uniform scaling and *(b)* being invariant against linear isometries. In other words, applying linear transformations such as translations, reflections, rotations or uniform scaling to a cluster of points should not change its isotropy measurement. Confusingly, in the field of probability theory, functions that satisfy the property of being invariant under linear isometries are sometimes referred to as being "isotropic measures" or "isotropic processes" [16], not to be confused with the measures for isotropy or metrics for isotropy, which are explored in this paper.

A simple way to incorporate invariance upon linear isometries as a metric for isotropy is to base that metric on the principal components of a cluster. While the eigenvectors that

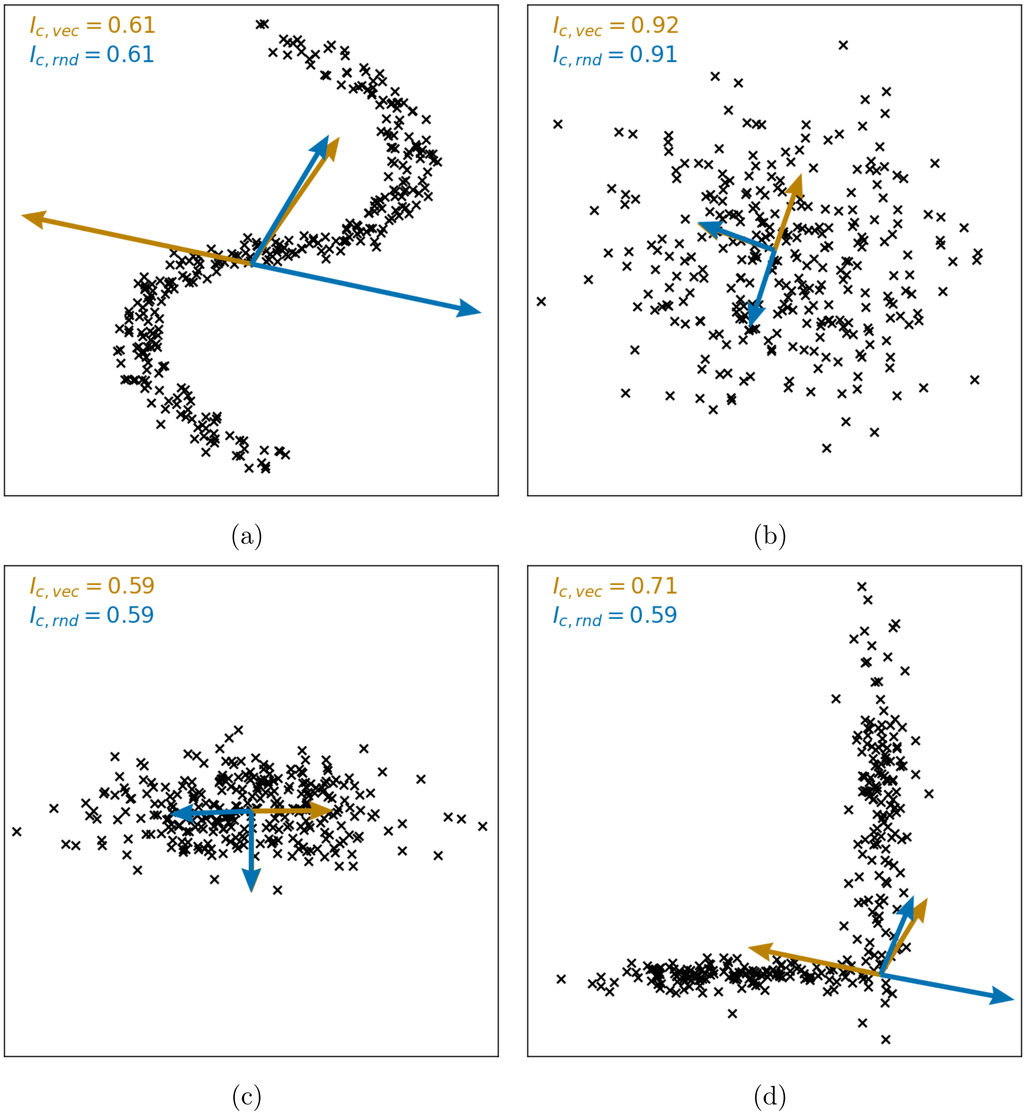


Figure 6.1: Examples of two dimensional clusters of points, labelled with measurements for $I_{c,\text{vec}}$ and $I_{c,\text{rnd}}$. Unit vectors $\mathbf{a}$ which resulted in $min(Z(\mathbf{a}))$ and $max(Z(\mathbf{a}))$ are shown, and colour coded according to the metric for which they have been used. Each cluster consists of 300 points. As was the case for most other low dimensional experiments (Section 6.4.3), in all the examples shown here, $I_{c,\text{rnd}}$ is seen to be lower than $I_{c,\text{vec}}$ and thus is more accurate in these cases (Equation 6.11) (a) An s-shaped, or spiral cluster (b) A cluster picked from a two dimensional Gaussian distribution (c) A cluster picked from a two dimensional Gaussian distribution where the Y axis has a lower standard deviation than the X axis (d) A reverse L shaped cluster.

make up these principal components may change with rotation, their relationship with the points in the cluster and the set of eigenvalues (Λ) associated with each eigenvector will not change. Using the variance of these eigenvalues would be a simple proxy of isotropy. If the variance between eigenvalues is large, there are large differences between the eigenvalues, then there are principal components that are more significant, and thus a cluster will become anisotropic. If the variance between the eigenvalues is small, then the eigenvalues are similar, and the cluster extends evenly in each of the principal axes. If the eigenvalues are normalised before measuring the variance (Var), then this proxy is invariant upon uniform scaling and invariant upon linear isometries. These normalised eigenvalues are usually denoted as the set λ:

$$\lambda_i = \frac{\Lambda_i}{\sum_{j=0}^{n} \Lambda_j} \tag{6.1}$$

Where Λ are the eigenvalues of the principal components of a cluster. As uniform scaling of points in the cluster will equally scale $\sum_{j=0}^{n} \Lambda_j$, $\mathrm{Var}(\lambda)$ is a simple measure for isotropy which is invariant upon linear isometries and upon uniform scaling.

While this means that $\mathrm{Var}(\lambda)$ is theoretically bounded between 0 and 0.25 (Theorem 1), $\mathrm{Var}(\lambda)$ is usually very small and does not use the whole range (Tables 6.2 and 6.3). Consequently, $\mathrm{Var}(\lambda)$ is not a common measure of isotropy and will not be discussed here.

Similar measures for isotropy based on eigenvalues of the principal components are widely used in the field of diffusion tensor imaging. In diffusion tensor imaging, isotropy can be used as a proxy for the flow of water in the brain or spinal chord [48]. While many metrics for isotropy have been proposed in the medical imaging field [2], the most widely used of these is fractional anisotropy (FA) [10]. FA is defined as the square root of the variance of the normalised eigenvalues of a covariance matrix, divided by the expected value of the square of the normalised eigenvalues, given by Equation 6.3. Due to its application in magnetic resonance imaging, this is often defined in three dimensions:

$$\mathrm{FA}(\lambda) = \sqrt{\frac{3}{2}} \sqrt{\frac{\left(\lambda_1 - \hat{\lambda}\right)^2 + \left(\lambda_2 - \hat{\lambda}\right)^2 + \left(\lambda_3 - \hat{\lambda}\right)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \tag{6.2}$$

$$= \sqrt{\frac{\mathrm{Var}(\lambda)}{\mathrm{E}\left(\lambda^2\right)}} = \sqrt{1 - \frac{E(\lambda)^2}{E(\lambda^2)}} \tag{6.3}$$

Where $\hat{\lambda}$ is the mean of λ ($\hat{\lambda} = (\sum_{i=0}^{n} \lambda)/n$). FA is bounded between 0 and 1 with 1 indicating a highly anisotropic cluster and 0 indicating a highly isotropic cluster. In medical imaging λ is usually the set of normalised eigenvalues for the principal components of a diffusion tensor. This diffusion tensor is a small voxel of a larger medical image that allows mapping of water diffusion in parts of the brain [48]. We are unaware of any higher-dimensional applications of this metric or applications on larger sets of points. This chapter investigates the use of FA to quantify anisotropy in larger point clouds and in higher dimensions. As will be seen shortly, special considerations are needed when using FA in higher dimensions.

Although FA is popular in medical imaging, an alternative approach to measuring the isotropy of a cluster was proposed in the natural language processing domain [123]. This research aimed to quantify the changes to high-dimensional word embeddings. This research used a previously defined function to quantify the cosine similarity between a vector and a cluster of points [123]:

$$Z(\mathbf{a}) = \sum_{\mathbf{d} \in \mathcal{C}} \exp\left(\mathbf{a}^{\intercal}\mathbf{d}\right) \tag{6.4}$$

In an isotropic cluster, $\mathcal{C}$, of data points, $\mathbf{d}$, the value of $Z(\mathbf{a})$ should be approximately constant with any unit vector $\mathbf{a}$. The ratio between the largest and smallest values of $Z(\mathbf{a})$ for a cluster $\mathcal{C}$ can be used to define an isotropy measurement, I_c, with a range between 0 and 1. The true ratio of largest to smallest values of $Z(\mathbf{a})$, would be calculated using every $\mathbf{a}$ on the unit sphere [123]:

$$\frac{\min_{|\mathbf{a}|=1} Z(\mathbf{a})}{\max_{|\mathbf{a}|=1} Z(\mathbf{a})} \tag{6.5}$$

However, this definition is not invariant under linear isometries or uniform scaling: moving a cluster away from the origin will result in a smaller measure for isoptropy (Theorem 2). In order to make this definition invariant upon linear isometries and uniform scaling, we adjust the $Z(\mathbf{a})$ function used in previous work:

$$Z'(\mathbf{a}) = \sum_{\mathbf{d} \in \mathcal{C}} \exp\left(\mathbf{a}^{\intercal}\left(\frac{\mathbf{d} - \hat{\mathbf{d}}}{\mu}\right)\right) \tag{6.6}$$

$\hat{\mathbf{d}}$ is the centroid (or mean) of $\mathcal{C}$ and μ is the mean magnitude of $\mathbf{d} - \hat{\mathbf{d}}$:

$$\mu = \frac{\sum_{\mathbf{d} \in \mathcal{C}} |\mathbf{d} - \hat{\mathbf{d}}|}{|\mathcal{C}|} \tag{6.7}$$

This allows the adjustment definition of a measure, I_c, for the isotropy of a cluster, which is bounded between 0 and 1, and invariant upon linear isometries and uniform scaling:

$$I_{c,\text{true}} = \frac{\min_{|\mathbf{a}|=1} Z'(\mathbf{a})}{\max_{|\mathbf{a}|=1} Z'(\mathbf{a})} \tag{6.8}$$

While linear isometries applied to $\mathcal{C}$ may change $Z'(\mathbf{a})$ for a single value of $\mathbf{a}$, intuitively it will not change the value of $I_{c,\text{true}}$. Note that for $I_{c,\text{true}}$ (and its non-invariant counterpart in Equation 6.5) an isotropic cluster will result in a measurement close to 1 and an anisotropic cluster will be close to 0. This is the opposite to FA which is 1 for anisotropic clusters and 0 for isotropic clusters.

As the set of vectors on the unit sphere is infinite, previous work [123] approximated $I_{c,\text{true}}$ by measuring Z for the set of eigenvectors found in PCA. As an alternative approximation will later be proposed (Section 6.3.1), for clarity, this implementation of I_c is labelled as $I_{c,\text{vec}}$. $I_{c,\text{vec}}$ (adjusted for invariance under scaling and linear isometries) is thus defined by:

$$I_{c,\text{vec}}(C) \approx \frac{\min_{\mathbf{a} \in A} Z'(\mathbf{a})}{\max_{\mathbf{a} \in A} Z'(\mathbf{a})} \tag{6.9}$$

where $\mathbf{a}$ is the set of eigenvectors found by applying SVD to $\mathcal{C}^\intercal\mathcal{C}$. Readers familiar with PCA will note that this is the same process by which PCA is calculated (although PCA here will be applied to the cluster). Thus, $\mathbf{a}$ is the set of eigenvectors that are the principal components of the cluster.

6.3.1 Alternative interpretation of isotropy definition

Much like FA, $I_{c,\text{vec}}$ assumes that isotropy originates from the principal axis of a cluster. As seen empirically, this is often a valid assumption (Tables 6.2 and 6.3 and Figure 6.5b), but it is possible to think of clusters for which this is clearly not the case (Figure 6.1d).

The task of finding the set $\mathcal{B}$ for which $Z'(\mathbf{a})$ is measured can thus be approached as an optimisation task. The set of points on the unit sphere is of infinite size, thus, a subset of unit vectors $\mathcal{B}$ must be defined for which to calculate $\forall_{\mathbf{a}\in\mathcal{B}} Z'(\mathbf{a})$:

$$I_{c|\mathcal{B}}(\mathcal{C}) \approx \frac{\min_{\mathbf{b}\in\mathcal{B}} Z'(\mathbf{b})}{\max_{\mathbf{b}\in\mathcal{B}} Z'(\mathbf{b})} \tag{6.10}$$

A good approximation of $I_{c,\text{true}}$ must not incur excess computation and must be accurate. The computational complexity can be estimated theoretically and stated using the "big O notation" (examples in Table 6.1) or can be measured experimentally (Figure 6.5a). Assessing which set $\mathcal{B}$ provides the most accurate $I_{c|\mathcal{B}}$ is also straightforward. Given our definition of $I_{c,\text{true}}$ (6.8), $I_{c|\mathcal{B}}$ will always be an upper bound for the true value of $I_{c,\text{true}}$ (Theorem 3):

$$\forall_{c,\mathcal{B}} : I_{c,\text{true}} \leq I_{c|\mathcal{B}} \tag{6.11}$$

From this it can be concluded that, given two sets of vectors to constitute $\mathcal{B}$, the one for which $I_{c|\mathcal{B}}$ is smaller will be the more accurate of the two. Thus, this task is framed as finding a set $\mathcal{B}$, which will result in the smallest value of $I_{c|\mathcal{B}}$, while taking into account the incurred computational costs.

As a possible solution, using a random set of unit vectors, r, to define the set $\mathcal{B}$ is proposed. This solution is labelled $I_{c,\text{rnd}}$, which can be defined as:

$$I_{c,\text{rnd}}(\mathcal{C}) \approx \frac{\min_{\mathbf{b}\in r} Z(\mathbf{b})}{\max_{\mathbf{b}\in r} Z(\mathbf{b})} \tag{6.12}$$

Here, r is a random set of unit vectors.

While $I_{c,\text{rnd}}$ is non-deterministic, the random set r can be chosen *a priori* and used to calculate $I_{c,\text{rnd}}$ for multiple clusters with the same dimensionality. As in calculations of I_g, I_c must be calculated for many different clusters (Equations 6.14 and 6.15). This means that in many circumstances it is more computationally efficient to calculate $I_{g,\text{rnd}}$ than $I_{g,\text{vec}}$ when $\mathcal{B}$ is pre-calculated (Table 6.1, Figure 6.5a).

The stochastic nature of this equation means that found values of $I_{c,\text{rnd}}$ will slightly differ for different values of r. However, since the number of random unit vectors sampled is a hyper parameter of $I_{c,\text{rnd}}$, the ability of $I_{c,\text{rnd}}$ to approximate $I_{c,\text{true}}$ can be improved by sampling more random unit vectors. As per Theorem 4, we can find:

$$\lim_{|r|\to\infty} I_{c,\text{rnd}} = I_{c,\text{true}} \tag{6.13}$$

$I_{g,\text{rnd}}$ to $I_{g,\text{vec}}$ are compared in two different contexts, to show an example use in the materials science domain, and a basic data science example to show the broader applicability.

6.3.2 Isotropy of sets of clusters

Both the internal clustering validation metrics and the measures of isotropy explored here rely on features of data to produce a numeric measurement. To adapt the isotropy measurements into an internal clustering validation metric, this section extends these measurements to be defined globally for a set of clusters, $\mathcal{G}$, rather than a single cluster of points. To adapt $I_{c,\text{vec}}$ and $I_{c,\text{rnd}}$ to estimate average $I_{c,\text{true}}$ for a $\mathcal{G}$, a weighted sum of isotropy for each cluster can be taken to establish a measure for a global set of clusters, I_g (weighted by the number of data points in a cluster). Thus, $I_{g,\text{vec}}$ can be defined as:

$$I_{g,\text{vec}}(\mathcal{G}) \approx \frac{1}{|E|}\sum_{\mathcal{C}\in\mathcal{G}} |\mathcal{C}| I_{c,\text{vec}}(\mathcal{C}) \tag{6.14}$$

Where E is the set of all points in the dataset ($|\mathcal{E}| = \sum_{C\in D} |\mathcal{C}|$, where $|\mathcal{C}|$ is the number of points in $\mathcal{C}$).

Similarly, $I_{g,\text{rnd}}$ can be defined:

$$I_{g,\text{rnd}}(\mathcal{G}) \approx \frac{1}{|\mathcal{E}|}\sum_{\mathcal{C}\in\mathcal{G}} |\mathcal{C}| I_{c,\text{rnd}}(\mathcal{C}) \tag{6.15}$$

Both $I_{c,\text{rnd}}$ and $I_{c,\text{vec}}$ are bounded between 0 and 1 where 0 represents a set of clusters that are anistropic, and 1 represents a set of clusters that are isotropic.

FA can also be adapted for a set of clusters $\mathcal{G}$. Taking the weighted sum of FA measurements for each cluster allows us to define FA_g:

$$\text{FA}_g(\mathcal{G}) = \frac{1}{|\mathcal{E}|}\sum_{\mathcal{C}\in\mathcal{G}} |\mathcal{C}|\text{FA}(\mathcal{C}) \tag{6.16}$$

FA_g is bounded between 0 and 1, with 0 representing a highly isotropic set of clusters and 1 representing a highly anisotropic set of clusters.

6.4 Results

6.4.1 Use of isotropy measurements in the context of materials science

To investigate the behaviour of FA_g, $I_{g,\text{vec}}$, and $I_{g,\text{rnd}}$, two potential use cases for these measures are examined. This is done using a canonical crystal structure dataset (*e.g.*, the ICSD), and later in a more general context using a canonical data science dataset (Section 6.4.2).

This exploration extends the previous work (Chapter 5) [45], which applied K-means clustering to prominent representations of the ICSD before and after the application of radial basis function (RBF) approximation [146]. This work qualitatively observed that clusters of chemical compositions in the ICSD were more isotropic after application of RBF approximation. This observation was made by visually inspecting 3 dimensional PCA projections of these high-dimensional representations (Figure 5.6). These PCA projections inherently remove some of the information present in higher dimensions. Using the measures for isotropy of a set of clusters presented here, it is possible to quantify the changes in cluster shape and ensure that all dimensions of a representation are considered.

This work represents the ICSD using two popular composition based representations explored in Chapter 5: a fractional composition vector ($CompVec$) encoding [79] and the *magpie* composition based feature vector [192]. $CompVec$ is an n-hot style encoding of composition where each entry in the vector corresponds to an element, the value of the entry represents the molar proportion of that element in a material. Thus, $CompVec$ is a sparse representation, with each entry being between 0 and 1, and the sum of all entries being equal to 1. *magpie* is a feature engineered vector, using features such as the covalent radius, electronegativity, or Mendeleev number observed for each element of a composition. The features are then aggregated using weighted mean, sum, variance, and range. As such, *magpie* is a dense representation, with the range of features varying significantly (Section 6.4.1), and many features being highly correlated [45].

Min-Max scaling was used for both representations, to transform the feature values to be between -1 and 1 before these data were clustered using K-means clustering [108] (values of K set to 5 and 10). The RBF approximation was then applied *a priori* to the data being clustered with K-means clustering (values of K set at 5 and 10).

While RBF and RBF approximation are distinct techniques (Chapter 5), for brevity, (and in line with the previous chapter) RBF will be used in figures and that it is an approximation is mentioned in the text and captions. Changes observed in the previous chapter are measured to validate visual findings and observe differences between measures of isotropy, FA_g, $I_{g,\mathrm{rnd}}$ and $I_{g,\mathrm{vec}}$.

Visual inspection of PCA projections shows that applying an RBF approximation [146] to the ICSD creates more isotropic clusters (Figure 6.2). However, due to the limitations in the visualisation, it is difficult to determine which representation leads to the largest changes upon application of the RBF approximation. The use of internal clustering validation metrics, including the isotropy measures developed here, allows quantification changes in the clusters (Table 6.1).

Examining how metrics change with RBF approximation also allows reflection on which metric may be most useful depending on the required application. For example, cluster size variance was used in previous work, where more evenly sized clusters were sought to reduce data imbalance for training machine learning models [45]. However, more evenly sized clusters are not necessarily well separated, thus other applications seeking to measure the distinctness of clusters may favour different internal cluster validation methods such as the Davies–Bouldin or silhouette scores. In the task of quantifying the anisotropy (*i.e.*, spikeyness) of clusters, no existing internal validation methods were suitable. Thus, the methods presented here are pertinent in this case.

As was initially expected, applying RBF approximation has a consistent effect on most existing measurements in both investigated representations. The only internal cluster validation metrics in which $CompVec$ and *magpie* exhibited divergent behaviour with the RBF approximation were the mean Euclidean distance to the centroid, and measures for isotropy FA_g, $I_{g,\mathrm{rnd}}$ and $I_{g,\mathrm{vec}}$ (Table 6.2). An explanation is offered for each of these. The measures for isotropy introduced here (FA_g, $I_{g,\mathrm{rnd}}$ and $I_{g,\mathrm{vec}}$) display unique behaviour from other internal cluster validation metrics. I_g were the only metrics for which applying the RBF approximation *magpie* and $CompVec$ representations offered divergent behaviour regardless of whether measurements were taken after normalising *magpie*.

While there are apparent divergent behaviours in the mean Euclidean distance between a point in a cluster and its centroid, this divergence can be explained by a lack of normalisation in the *magpie* feature vector. The formula of the RBF approximation is examined in order to demonstrate this.

Apparent divergent behaviour in mean Euclidean distance to centroid

When measuring the mean distance between a point in a cluster and its centroid after the RBF approximation, a *magpie* representation without normalisation exhibits behaviour opposite to that of $CompVec$. This behaviour can be explained by examining the formula for the radial basis function approximation:

$$f(x) = \frac{\sqrt{2}\cos(x \cdot w + o)}{\sqrt{l}} \tag{6.17}$$

l is length of vector. w and o are random weights, where w is mean 0 variance $\sqrt{2\gamma}$ and o is uniform between 0 and 2 π.

The input to this function is randomly projected and translated and then put through a cosine function before being scaled relative to the length of the input.

In high dimensions, this random projection approaches a linear projection [154, 83]. This linear projection maps into a data space in which distance relationships between points are preserved (with some error margin which scales inversely to the number of dimensions). If the inputs to this function are not normalised, there could be large differences between the scale of different axes in the new data space, which will be present after random projection. However, as the cosine function is bounded [-1,1], these large differences in scale of dimensions will be removed, akin to normalisation (Table 6.2).

Unlike $CompVec$, *magpie* is not a normalised representation. Min-Max scaling was performed on both representations before application of K-means clustering, however metrics were measured with this scaling (as this scaling would effect these measurements). For example, one of the features used in *magpie* is the sum of the melting temperatures of the constituent elements, which ranges from 10 Kelvin to 43 million Kelvin. This large range means that the magnitude of *magpie* feature vectors are on average very large (Table 6.2), which results in large mean distances between a point in a cluster and that clusters centroid. After the RBF approximation, the mean magnitude of the representation decreases, so in turn the mean distance from the centroid; this is in keeping with the RBF approximation having a bounded output domain (Equation 6.17).

As all values in a $CompVec$ representation range from 0 to 1, the mean magnitude is smaller (0.66). Applying an RBF approximation results in a mean magnitude which is larger, and in turn a larger mean Euclidean distance between a point and its centroid. Thus, the divergence of behaviour between representations on application of an RBF ap-

Table 6.2: Analysis of entire ICSD in *CompVec* and *magpie* representations before and after RBF approximation. Values here are means of those found with K-means clustering aplied with $K = 5$ and $K = 10$

Metric	CompVec		magpie		magpie (normalised)	
	No RBF	RBF	No RBF	RBF	No RBF	RBF
l^2-norm of representation	0.660	1.01	8.21×10^5	1.00	7.59	1.00
mean distance to centroid	0.397	0.52	2.33×10^5	0.987	2.91	6.98
silhouette	0.247	0.177	0.559	6.42×10^{-3}	-2.06×10^{-2}	6.34×10^{-3}
Davies–Bouldin	1.91	2.58	0.515	8.33	7.45	8.33
Calinski–Harabasz	2.09×10^4	1.88×10^4	7.11×10^5	8.53×10^2	2.97×10^3	8.53×10^2
cluster size variance	9.26×10^8	6.33×10^8	6.34×10^8	3.15×10^5	6.34×10^8	3.15×10^5
fractional anisotropy	0.855	0.870	0.994	0.182	0.951	0.182
Var(λ)	3.33×10^{-4}	4.96×10^{-4}	1.08×10^{-2}	3.54×10^{-6}	1.24×10^{-3}	3.53×10^{-6}
$I_{g,\mathrm{vec}}$	0.942	0.923	0.180	0.993	0.894	0.993
$I_{g,\mathrm{rnd}}$	0.992	0.988	0.682	0.999	0.981	0.999

proximation is due to changes in the magnitude of a representation that this function entails.

Divergent behaviour between representations for measurements of isotropy

Applying an RBF approximation to all chemical compositions of the ICSD changes measurements of anisotropy, depending on the representation of those compositions (Table 6.2). The *magpie* representation is measured to be more isotropic after application of an RBF approximation, whereas the *CompVec* is more anisotropic. This difference in behaviour is due to differences in the sparsity of the representations. While *magpie* is a dense representation, *CompVec* is a sparse representation, which has important consequences, as discussed below.

Unlike the mean distance to the centroid, the divergence in the effect of the RBF approximation on isotropy cannot be explained by the bounded nature of the function (Equation 6.17). Even when measuring $I_{g,\text{rnd}}$ and $I_{g,\text{vec}}$ of this clustering when *magpie* is normalised (with Min-Max scaling), the isotropy increases after the RBF approximation, whereas isotropy decreases in a $CompVec$ representation (Table 6.2).

As $CompVec$ is a sparse representation, the projection seen in the RBF approximation will result in a sparse output. However, as this is over fewer dimensions, non-zero values in single axis have a greater effect on the isotropy of a cluster, leading to more anisotropic clusters.

As *magpie* is a dense representation, the opposite is true. Variations in any one dimension will be diluted, leading to more isotropic clusters.

This behaviour is not necessarily obvious when looking at the transformation performed to both representations (Equation 6.17). In this case, the measures for isotropy $I_{g,\text{vec}}$, $I_{g,\text{rnd}}$, and FA_g are valuable to measure an effect and help identify changes that may not be intuitive.

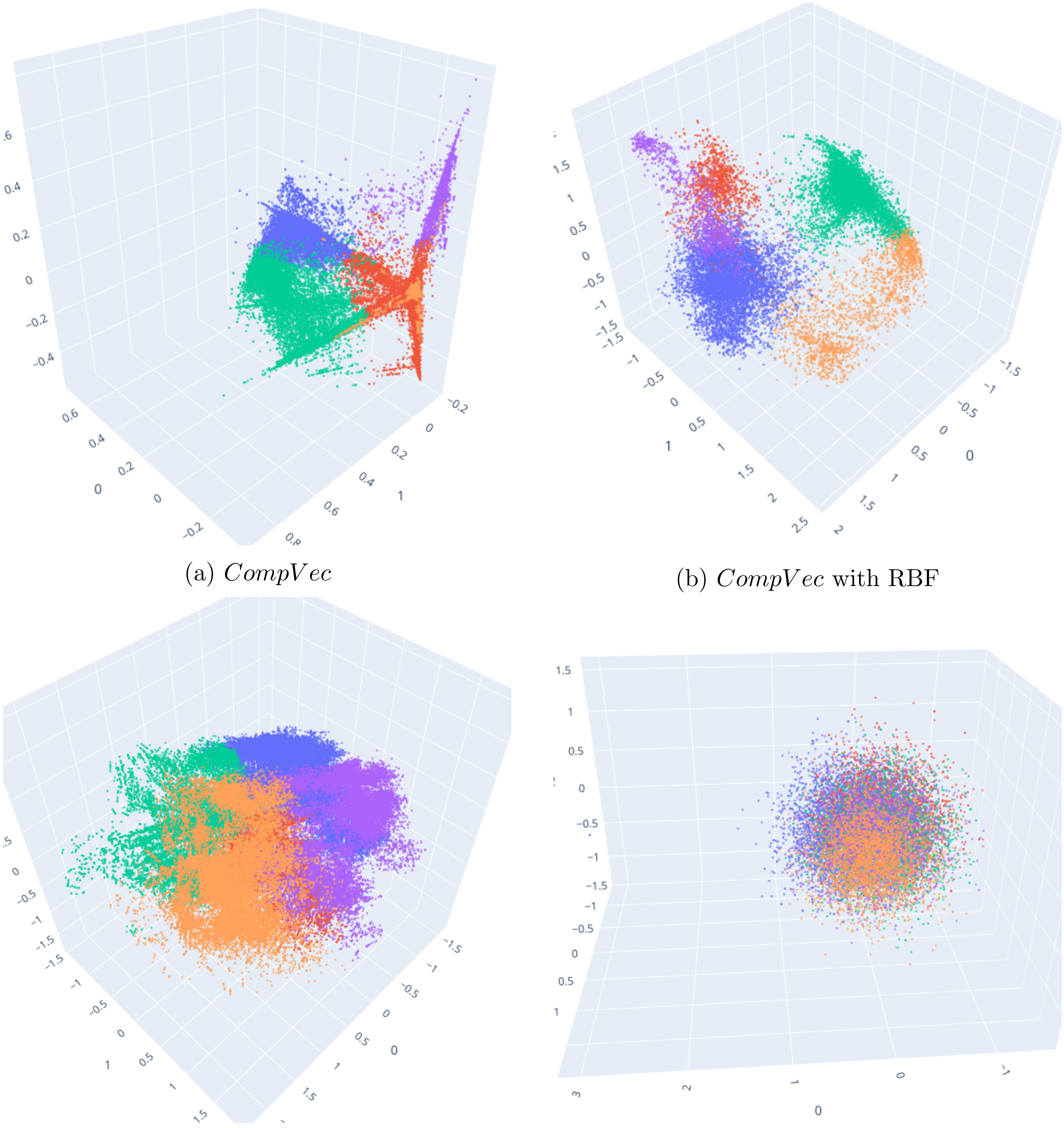
6.4.2 Example basic use of cluster isotropy measurements for data science application

Examining learnt embeddings (also called learnt representations) is common in understanding how deep learning algorithms interpret input data [177]. However, this is often qualitative analysis, based on 2D or 3D projections of the embeddings, which by their nature remove some of the information present in higher dimensions.

Cluster isotropy measurements can be used to quantify differences between embeddings without dimensionality reduction. This can be used in conjunction with qualitative analysis to judge the differences between embeddings produced by deep learning models.

In order to demonstrate the broader applicability of measures of cluster isotropy, this section presents quantitative analysis of learnt embeddings. Two models were trained on a basic computer vision dataset, the Modified National Institute of Standards and Technology dataset (MNIST) [99]. The MNIST is a widely used dataset of 70,000, 28x28 pixel images, each depicting a handwritten number between 0 and 9. A common ML introductory task is to create a model that can correctly classify these images.

Two types of models were trained to create low-dimensional embeddings MNIST im-



(a) *CompVec*

(b) *CompVec* with RBF

(c) *Magpie*

(d) *Magpie* with RBF

Figure 6.2: 3 dimensional PCA representations of the ICSD (a random 20% subsample is used for visual clarity) with clusters found with K-means clustering. Coloured according to clusters found using K-means clustering on that data with $K = 5$ (*i.e.*, the same point may not be the same colour between subfigures) (a) Data was represented using $CompVec$ with no RBF application. (b) Data was represented using $CompVec$ with RBF application. (c) Data was represented using magpie with no RBF application. (d) Data was represented using magpie with RBF application.

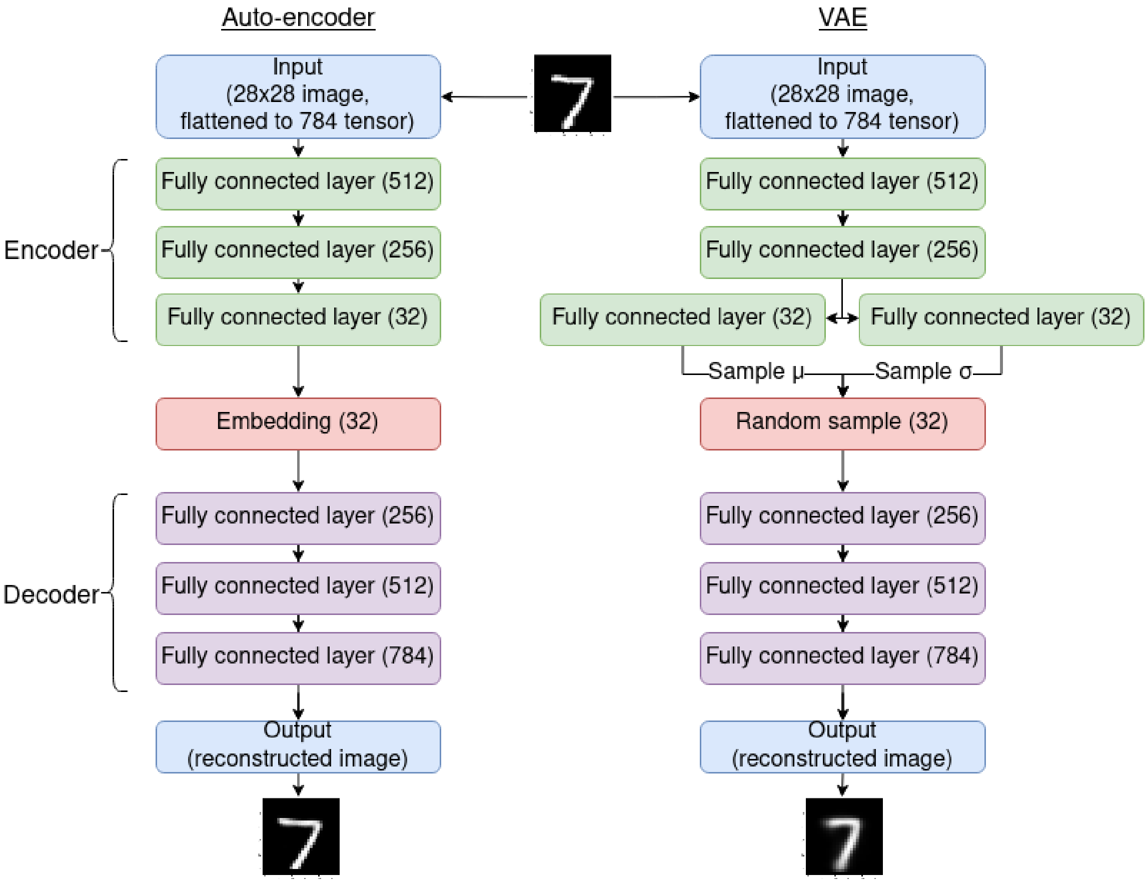
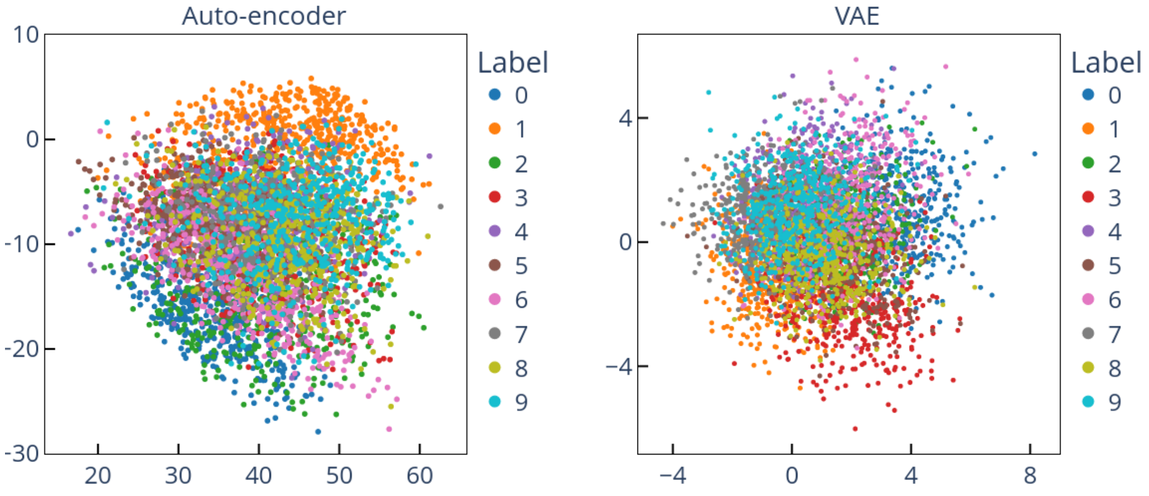

Auto-encoder
VAE
Input
(28x28 image,
flattened to 784 tensor)
Input
(28x28 image,
flattened to 784 tensor)
Encoder
Fully connected layer (512)
Fully connected layer (256)
Fully connected layer (32)
Fully connected layer (512)
Fully connected layer (256)
Fully connected layer (32)
Fully connected layer (32)
Sample μ
Sample σ
Embedding (32)
Random sample (32)
Decoder
Fully connected layer (256)
Fully connected layer (512)
Fully connected layer (784)
Fully connected layer (256)
Fully connected layer (512)
Fully connected layer (784)
Output
(reconstructed image)
Output
(reconstructed image)

Figure 6.3: Layout of auto-encoder and variational auto-encoder (VAE) used in the example discussed in 6.4.2. Embeddings shown in red here are those seen plotted in Figure 6.4 and measured in Table 6.3.

ages, and compare their embeddings of unseen test data using two different models. The exact mechanics of these models is not vital to understand the novel analysis of isotropy of clusters which is presented here. This section does not aim to present a novel model; instead, it used isotropy of clusters to analyse models' outputs in an interesting way. Regardless; a brief overview is of these models given.

Auto-encoders are neural network models that learn a representation of a dataset by passing each dataset through a neural network with an information bottleneck before trying to reconstruct the input data point [90]. The part of the model up to and including this bottleneck is called the encoder, and the part after the bottleneck is called the decoder



(a)

(b)

Figure 6.4: PCA transformations of latent space embeddings of MNIST training set. A random 50% of the training set is shown for visual clarity. (a) Autoencoder embedding (b) Variational autoencoder embedding

(as they, respectively, encode and decode a learnt representation). Once trained, output of the bottleneck layer can be used as a lower-dimensional representation of the input (also referred to as a learnt or latent space). Feeding random noise to a trained auto-encoder should generate an output which is similar to existing data points, but is completely fictional. However, in practise, when used in this generative fashion, auto-encoders can give outputs identical to received training data.

Variational auto-encoders (VAEs) aim to be a more useful generative model by introducing randomness into the encoder [176]. This forces the model to learn the distribution of points in a dataset, rather than the points themselves. As a result, a VAE tends to generate a data point which is harder to map on directly onto a data point in the training set. In order to stop overfitting when learning the distribution of datapoints, a further term, Kullback–Leibler divergence (KL divergence), is introduced into the loss function [94]. The KL divergence measures the distance between the learnt distribution and either the true distribution or the a distribution chosen a priori (often the normal distribution).

An auto-encoder and a VAE (Figure 6.3), were trained for 100 epochs on the MNIST dataset. Each features 3 fully connected layers in both the encoder and the decoder, however, with rectified linear units (ReLUs) after each layer to provide non-linearity (with the exception of the final layer which is followed by a sigmoid function). The KL divergence in the VAE was measured between the learnt distribution and a normal distribution of mean 0 and standard deviation 1.

The encoder of each network produces embeddings of size 32 for an input image. The latent space of these training set can be projected into 2 dimensions using principal component analysis (PCA) to allow for inspection (Figure 6.4).

On visual inspection, it is difficult to discern differences between these latent spaces. Measurements of FA_g, $I_{g,\mathrm{vec}}$, and $I_{g,\mathrm{rnd}}$ allows the quantitative observation that images of the same label form more isotropic clusters when embedded with the VAE than when embedded with the auto-encoder (Table 6.3). All three metrics for isotropy of clusters conclude that VAE's result in more isotropic clusters. FA_g of the VAE's is measured as almost half that of the AutoEncoder, suggesting a very large change in isotropy. $I_{g,\mathrm{rnd}}$, and $I_{g,\mathrm{vec}}$ suggest a much smaller (but consistent) change in isotropy. However, as will be discussed (Section 6.5) this seems to be indicative of how these metrics perform, with changes of $\sim$0.001 being notable despite only being a very small part of the domain of $I_{g,\mathrm{rnd}}$ and $I_{g,\mathrm{vec}}$ (which is 0, 1).

Conclusions as to the difference between these latent spaces can also be drawn from other internal cluster validation metrics. The Calinski–Harabasz measure suggests that clusters in the VAE's embeddings are more dispersed than those of the Auto-Encoder. This is in line with the silhouette score, which suggests that points embedded with VAE's are closer to points in other clusters than those embedded with auto-encoders. Davies–Bouldin also suggests more poorly separated clusters in the VAE embeddings.

Existing internal cluster validation metrics all suggest a worse separation between classes in VAE embeddings compared to Auto-Encoder embeddings. This makes sense when considering how VAEs work; Gaussian noise is inherent to the model. For generative models such as these, isotropy in the embeddings has been linked to good generative performance [123]. Thus, measurements of isotropy for sets of clusters are pertinent.

Using existing internal cluster validation metrics may suggest to a researcher that VAE's are worse for class separation than Auto-Encoders. Measuring the isotropy of embeddings (and knowing that isotropy has been associated with good generative performance) gives a more nuanced picture. When observed with other internal clustering validation measures, metrics for the isotropy of clusters allows someone developing models such as these to gain

Table 6.3: Metric measurements of the MNIST test set embedded with auto-encoder and VAE

Metric	Auto-encoder	VAE
$I_{g,\text{vec}}$	0.991	0.998
$I_{g,\text{rnd}}$	0.942	0.984
$\text{Var}(\lambda)$	4.70e-3	2.76e-4
fractional anisotropy	0.901	0.451
Calinski–Harabasz	531	161
Davies–Bouldin	2.57	7.03
Silhouette	0.091	0.011

an intuitive understanding of the latent spaces produced by them.

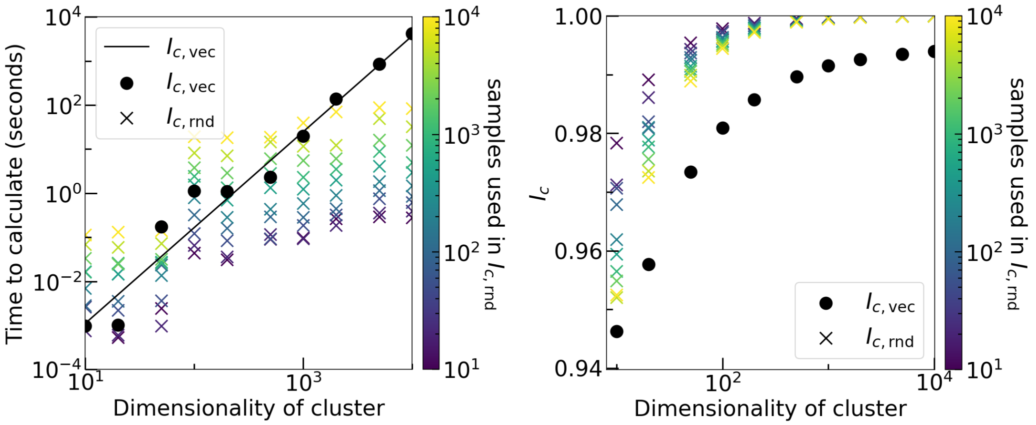
6.4.3 Examining differences between measurements of isotropy using random Gaussian point clouds

In experiments presented thus far, isotropy of sets of clusters have been measured in two different situations (*i.e.*, Auto-encoder vs VAE, RBF vs. no RBF). In both experiments, all measures of isotropy have been consistent as to which of the two situations results in more isotropic clusters (Tables 6.2 and 6.3) (note that a FA is expected to have an inverse relationship to I_c). However, it has not been clear which approximation of I_c was more accurate and if there were any advantages or disadvantages to approximating I_c as opposed to using FA.

To further explore the differences between the approximations of I_c and FA, the isotropy of randomly generated clusters of different dimensionalities were measured. Measurements of $I_{c,\text{vec}}$, $I_{c,\text{rnd}}$, and FA were taken to explore how the accuracy and time complexity of each algorithm varies with dimensionality.

Clusters of 100 points were generated, with each point having coordinates sampled from a Gaussian distribution of mean 0 and standard deviation 1. Clusters were generated between 10 and 10,000 dimensions and the $I_{c,\text{vec}}$, $I_{c,\text{rnd}}$, and FA were measured, with the number of random unit vectors used in I_{c_rnd} varying between 10 and 10,000. For each dimensionality, 10 different clusters were measured; the mean results are reported here.

The computational times of $I_{c,\text{vec}}$ and FA are approximately exponential to the dimensionality of the cluster (Figure 6.5a). As the computational time of $I_{c,\text{rnd}}$ varies with the number of random unit vectors used, $I_{c,\text{rnd}}$ often performs faster (particularly for high-

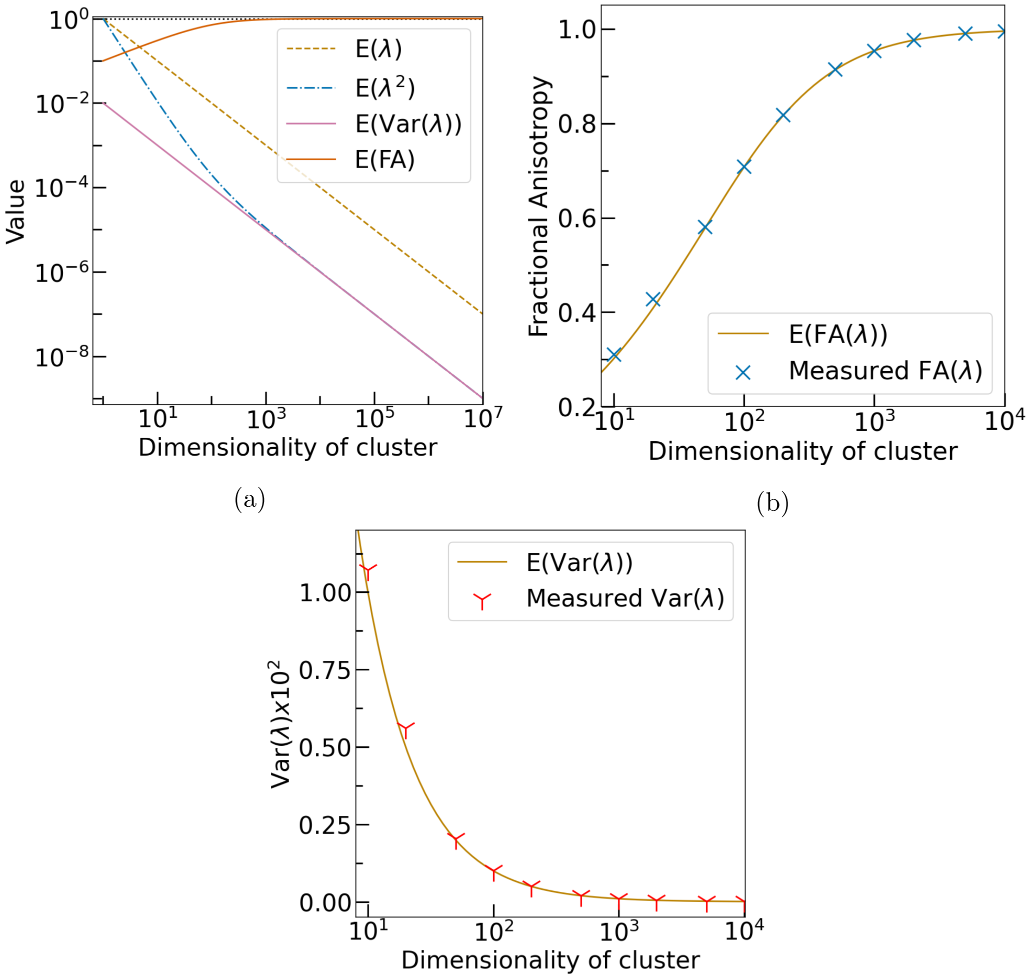


(a) (b)

Figure 6.5: Measurements of I_c using $I_{c,\text{vec}}$ and I_{c_rnd}, across a random Gaussian cluster of different dimensions. This was repeated for 10 different random clusters of 100 points, with the mean results shown here (a) In higher ($> 10^3$) dimensions, $I_{c,\text{vec}}$ becomes very expensive to compute, the computational complexity $I_{c,\text{rnd}}$ scales with the number of unit samples taken. (b) When the cluster is 10 or fewer dimensions, $I_{c,\text{rnd}}$ is a more accurate, with $I_{c,\text{vec}}$ performing better in higher dimensions. As expected (Equation 6.13), $I_{c,\text{rnd}}$ measurements are more accurate when more unit samples are used, though this effect is less noticeable in higher dimensions. Regardless of the number of unit samples used, across all dimensions all measurements of $I_{c,\text{vec}}$ and $I_{c,\text{rnd}}$ are within 10% of each other.

dimensional clusters). This is generally in line with the complexity calculated for these algorithms (Table 6.1).

In line with expectations (Equation 6.13), as more random samples are used, $I_{c,\text{rnd}}$ becomes more accurate (Figure 6.5b). Even with the highest number of unit samples, $I_{c,\text{rnd}}$ was never measured as being more accurate than $I_{c,\text{vec}}$ for these Gaussian clusters. Both $I_{c,\text{rnd}}$ and $I_{c,\text{vec}}$ measurements increased slightly with dimensionality. This is inverse to the relationship found with FA (Figure 6.6b), which would imply that in high dimensions Gaussian clusters are highly anisotropic. The reason for FA's observed increase in this experiment can be found in the field of random matrix theory, and are explored below (Section 6.4.3)



(a)

(b)

(c)

Figure 6.6: (a) The expected value of fractional anisotropy (FA) for a cluster of 100 points varying in dimensions, where the coordinates of each point are sampled from the normal distribution $\sim \mathcal{N}(0, 1)$. This $E(\text{FA})$ is calculated using the Marchenko–Pastur distribution. Examining how $E(\text{FA})$ and $E(\text{Var}(\lambda))$ vary with dimensionality demonstrates that clusters of different dimensionality cannot be compared using these measures. When comparing high dimensional clusters, FA and Var(λ) may lead to counterintuitive results, particularly when data are noisy. (b) Measurements of fractional isotropy (FA) compared to the expected value as modelled with the Marchenko–Pastur distribution. Random matrix theory successfully describes the behaviour of this measure. Points represent the mean FA of 10 different random clusters of 100 points. (c) Measurements of Var(λ) compared to the expected value as modelled with the Marchenko–Pastur distribution. Points represent the mean Var(λ) of 10 different random clusters of 100 points.

Why does fractional anisotropy of a Gaussian cluster increase with dimensionality?

Gaussian clusters are observed to have a high FA (and thus be considered very anisotropic) in higher dimensions (Figure 6.6b). If FA is highly correlated with with dimensionality, does this mean FA is not applicable in higher dimensions? In order to answer this, how the expected value of FA for a Gaussian cluster varies with dimensionality of that cluster was calculated.

To examine the expected value of FA of a point cloud with coordinates sampled from the Gaussian distribution, one can consider this n dimensional point cloud of T points to be represented as a matrix size $n \times T$. The eigenvalues for the principal components of this random matrix can be described using a probability density function. Assuming the variance of the Gaussian distribution, σ^2, to be finite, the probability density function of the eigenvalues, Λ, is given by the Marchenko–Pastur distribution [113]:

$$\mathrm{PDF}_{\mu,\sigma^2}(\Lambda_x) = \frac{1}{2\pi\sigma^2\Lambda}\sqrt{(\Lambda_{\max} - \Lambda)(\Lambda - \Lambda_{\min})} \tag{6.18}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are respectively the largest and smallest possible eigenvalues of the distribution, given by:

$$\Lambda_{\max,\min} \approx \sigma^2 \left(1 \pm \sqrt{\frac{T}{n}}\right)^2 + \mu \tag{6.19}$$

Where μ is the mean of the distribution and σ^2 is the variance. To solve this for the parameters in the experiment presented here ($\mu = 0, T = 100, \sigma = 1$) it can be said:

$$\Lambda_{\max,\min} = \left(1 \pm \frac{10}{\sqrt{n}}\right)^2 \tag{6.20}$$

In order to find the expected value for the E(FA), we can calculate E(Λ^2), and E(Λ)^2, the Marchenko–Pastur distribution can be integrated:

.

$$E(\Lambda) = \int_{\Lambda_{\min}}^{\Lambda_{\max}} \Lambda \mathrm{PDF}_{\mu,\sigma^2}(\Lambda) d\Lambda \tag{6.21}$$

Thus, in this case:

$$\begin{aligned} E(\Lambda) &= \int_{\Lambda_{\min}}^{\Lambda_{\max}} \Lambda \frac{1}{2\pi\sigma^2\Lambda} \sqrt{(\Lambda_{\max} - \Lambda)(\Lambda - \Lambda_{\min})} d\Lambda & (6.22) \\ &= \int_{\Lambda_{\min}}^{\Lambda_{\max}} \frac{1}{2\pi} \sqrt{(\Lambda_{\max} - \Lambda)(\Lambda - \Lambda_{\min})} d\Lambda & (6.23) \end{aligned}$$

Similarly it can be said:

$$E(\Lambda^2) = \int_{\Lambda_{\min}}^{\Lambda_{\max}} \frac{\Lambda}{2\pi} \sqrt{(\Lambda_{\max} - \Lambda)(\Lambda - \Lambda_{\min})} d\Lambda \tag{6.24}$$

FA is calculated with the normalised eigenvalues, λ (Equation 6.1), and so:

$$E(\lambda) = \frac{E(\Lambda)}{T} \tag{6.25}$$

and

$$E(\lambda^2) = \frac{E(\Lambda^2)}{T^2} \tag{6.26}$$

Note that when substituting these into the equation for FA(λ) (Equation 6.3), the denominators cancel out, thus $E(\mathrm{FA}(\lambda))$ is equal to $E(\mathrm{FA}(\Lambda))$:

$$E(\mathrm{FA}(\lambda)) = \sqrt{1 - \frac{E(\lambda)^2}{E(\lambda^2)}} \tag{6.27}$$

$$= \sqrt{1 - \frac{\left(E(\Lambda)/T\right)^2}{E(\Lambda^2)/T^2}} \tag{6.28}$$

$$= \sqrt{1 - \frac{E(\Lambda)^2/T^2}{E(\Lambda^2)/T^2}} \tag{6.29}$$

$$= \sqrt{1 - \frac{E(\Lambda)^2}{E(\Lambda^2)}} \tag{6.30}$$

$$= E(\mathrm{FA}(\Lambda)) \tag{6.31}$$

Plotting E(λ) against E(λ^2) shows that they converge in high dimensions, and thus FA converges to 1 (Figure 6.6a). The expectation of FA to measure Gaussian clusters as being anisotropic in high dimensions was reflected in our measurements (Figure 6.6b). A similar process can be followed for Var(λ), showing that Var(λ) trends towards 0 in high dimensions for Gaussian clusters (Figure 6.6c).

The implication of this proof is that comparing FAs for clusters that are not represented in the same number of dimensions may not be applicable. This is particularly true where clusters are noisy and thus will have Eigenvectors expected to more closely follow the Marchenko–Pastur distribution. As will be discussed (Section 6.5) this does not mean that FA is not a useful tool, but that this result should be taken into consideration, for example, by avoiding the use of noisy data or checking conclusions drawn from use of FA agree with those that would be drawn from the use of $I_{c,\mathrm{vec}}$ or $I_{c,\mathrm{rnd}}$.

6.4.4 Using isotropy measures to quantify the shape of materials

Description of crystals as being "anisotropic" or "isotropic" is common in materials science, this usually refers to the shape of the crystal lattice, with a large $\frac{c}{a}$ ratio being that of an anisotropic material. But this $\frac{c}{a}$ ratio is that of the conventional unit cell, which is not a mathematically sound concept, and is somewhat down to judgement. The $\frac{c}{a}$ of the Niggli reduced cell (see Section 2.2.2) could be used, but the origin choice of the Niggli reduced cell is not fixed, and the cell is sensitive to small changes in angle (*i.e.*, small changes in

Table 6.4: Pearson's correlation coefficients between $\frac{c}{a}$ of the cell and the isotropy of the fractional coordinates of the atomic sites, as measured with metrics presented here. Although all coordinates are between 0 and 1 several of the mothods here still identify trends in line with use of $\frac{c}{a}$. Note that I_{rnd} and I_{vec} are expected to decrease as anisotropy increases, so a negative corrolation is in line with expectations.

	Conventional unit cell	Scaled unit cell	Niggli reduced cell
FA	0.13	0.22	0.14
Var(λ)	0.14	0.30	0.14
I_{rnd}	-0.23	-0.20	-0.23
I_{vec}	0.00	-0.15	0.00

α, β, and γ lead to discontinuous jumps between spacegroups) [156].

As such, a mathematical measurement of the isotropy of a crystal based on the atomic sites within that crystal would be beneficial. The rest of this chapter, incidentally, concerns measures of isotropy, and so an investigation was undertaken to establish the usefulness I_{rnd}, I_{vec}, FA, and Var(λ) in determining the isotropy of a material based on the locations of atomic sites.

A preferable quality for a measurement for a material's isotropy would be invariance to different unit cells. If a supercell of an existing conventional unit cell was made (*e.g.* two unit cells being joined together), the isotropy of the material has not changed, as such the measurement for that isotropy should not change. This is different from the invariance to uniform scaling discussed previously (Section 6.3), as a super cell may not be uniform. This problem applies to use of $\frac{c}{a}$ as a proxy for isotropy. This may be alleviated by applying Niggli reduction to unit cells, but, as mentioned, Niggli reduced cells are instable to small changes in angles of the cell. In the investigation of measuring isotropy using atomic sites, two possible coordinate systems can be used: The cartesian coordinates (*i.e.*, absolute coordinates) or the fractional coordinates (*i.e.*, express coordinates a fraction of the size of the unit cell). Intuitively using fractional coordinates will make it hard to distinguish between isotropic and anisotropic materials; however fractional coordinates may be more likely to result in stability across different supercells. Both approaches are investigated.

The isotropy of two notably anisotropic structure types were investigated: the Ruddlesden-Popper structures and the Kagome structures. Trends are examined and then these measurements are compared to the PbFCl structure types discussed in Chapter 3. I_{rnd}, I_{vec},

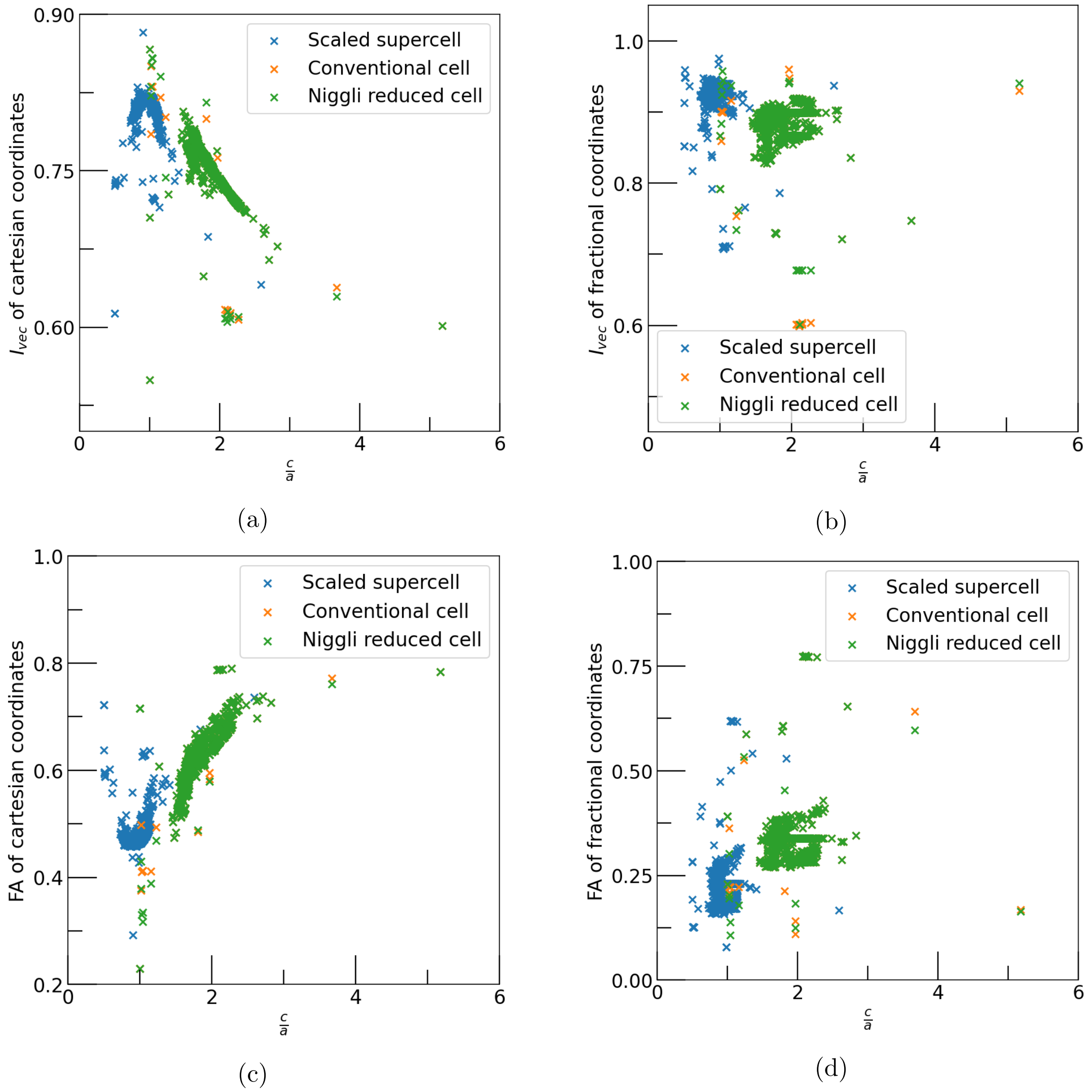
Table 6.5: Pearson's correlation coefficients between $\frac{c}{a}$ of the cell and the isotropy of the Cartesian coordinates of the atomic sites, as measured with metrics presented here. Note that I_{rnd} and I_{vec} are expected to decrease as anisotropy increases, so a negative correlation is in line with expectations.

	Conventional unit cell	Scaled unit cell	Niggli reduced cell
FA	0.81	0.17	0.81
Var(λ)	0.81	0.27	0.81
I_{rnd}	-0.26	0.19	-0.23
I_{vec}	-0.76	-0.17	-0.76

FA, and Var(λ) are compared to $\frac{c}{a}$ ratios for these materials to compare methods proposed here to more traditional proxies for the isotropy of a material. Niggli reduced and conventional unit cells are compared as well as a super cell scaled from twice the a axis for the conventional unit cell.

Applying measures of isotropy to the fractional and Cartesian coordinates of the atomic sites is somewhat correlated with the use of $\frac{c}{a}$ as a proxy of crystal isotropy (Tables 6.4 and 6.5). Using Cartesian atomic coordinates for these measures correlates more strongly with $\frac{c}{a}$, than with fractional coordinates, when considering conventional, scaled, and Niggli-reduced cells (Figure 6.7).

However, the use of Cartesian coordinates also results in decreased resilience to changes in unit cell (Figure 6.8). The more traditional measure for isotropy, $\frac{c}{a}$, by definition changes with the unit cell used. However, these changes are predictable to when compared to the change in the unit cell (in this case, doubling the a value, halves the $\frac{c}{a}$ value).



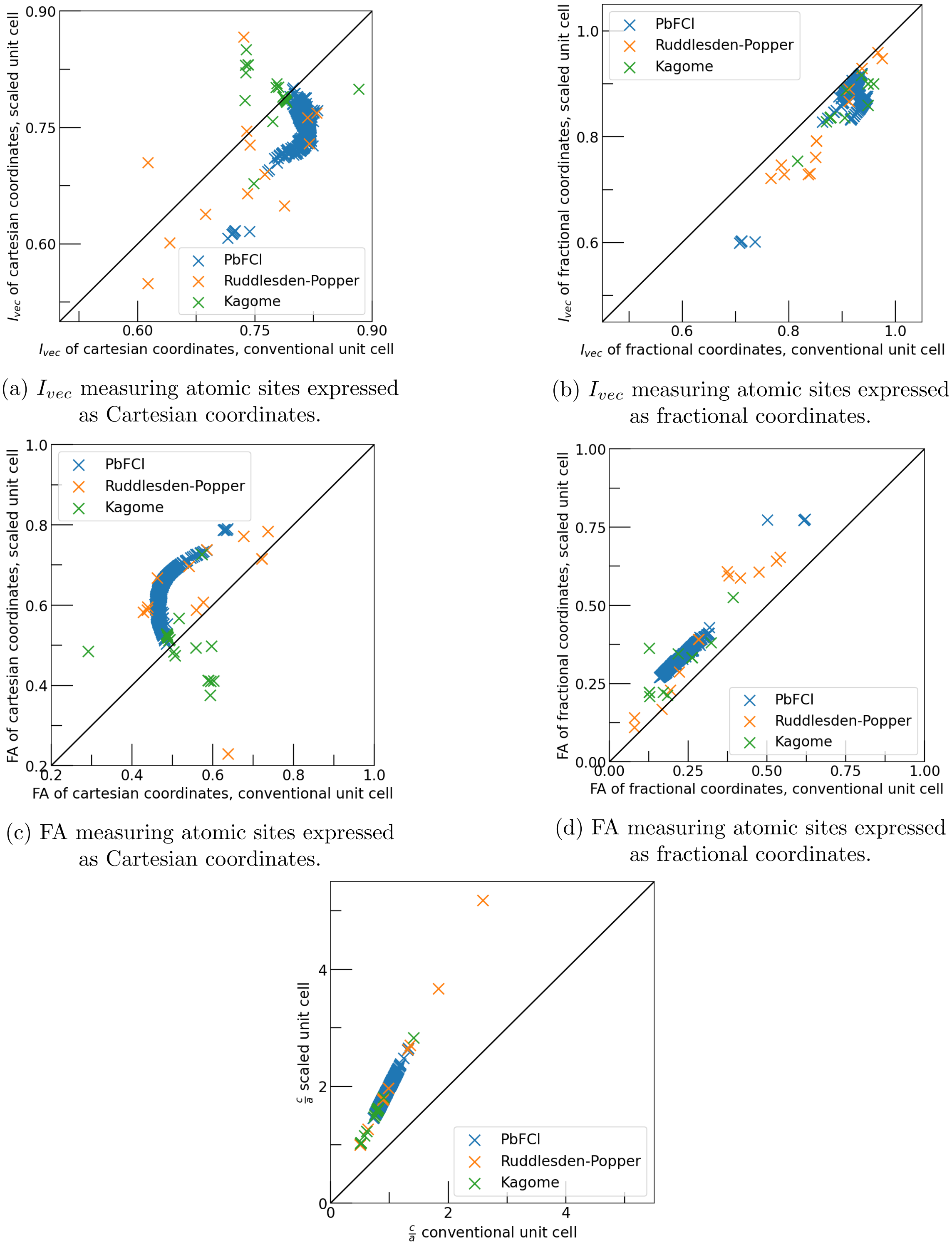
(a)

(b)

(c)

(d)

Figure 6.7: Using different measures of isotropy to quantify Crystal structure, using different atomic site coordinate systems. Use of Cartesian coordinates results in a stronger correlation. Measurements were done on conventional unit cells, Niggli reduced cells and a supercell of two conventional unit cells (scaling the a axis by 2) (a) I_{vec} measuring atomic sites expressed as Cartesian coordinates. (b) I_{vec} measuring atomic sites expressed as fractional coordinates. (c) FA measuring atomic sites expressed as Cartesian coordinates. (d) FA measuring atomic sites expressed as fractional coordinates.



(a) I_{vec} measuring atomic sites expressed as Cartesian coordinates.

(b) I_{vec} measuring atomic sites expressed as fractional coordinates.

(c) FA measuring atomic sites expressed as Cartesian coordinates.

(d) FA measuring atomic sites expressed as fractional coordinates.

(e) $\frac{c}{a}$ halves as the width of the unit cell doubles

Figure 6.8: Differences between conventional measurements of isotropy of materials when measuring a conventional unit cell and a supercell of two conventional unit cells (scaling the a axis by 2). An ideal measure for the isotropy a crystal would see no change, as indicated by the diagonal line.

6.5 Discussion

Analysis of isotropy of sets of clusters has used in two different settings to provide quantitative evidence that enhances observations of low-dimensional PCA projections. The study of isotropy of a set of clusters has highlighted non-intuitive results (*e.g.*, that RBF approximations will result in changes in isotropy that depend on the sparsity of an input representation) and helped interpretation of learnt embeddings, where low-dimensional projections were unclear.

In real-world examples seen here, measurements for isotropy have agreed in all instances. All measurements for isotropy investigated have a bounded output domain and are invariant under linear isometries as well as under uniform scaling. Thus, any of $\mathrm{Var}(\lambda)$, FA_g, $I_{g,\mathrm{vec}}$ or $I_{g,\mathrm{rnd}}$ can prove useful and will likely lead to similar conclusions for the data.

One way in which these measures differed is in the use of their output domain. As previously noted while $\mathrm{Var}(\lambda)$ has a theoretical output domain of 0, 0.25, in practise the highest $\mathrm{Var}(\lambda)$ observed was 0.033 in synthetic data (Figure 6.6c) or 0.011 in real data. As such and due to its lack of appearance in the literature, $\mathrm{Var}(\lambda)$ not the focus of this study. $I_{g,\mathrm{rnd}}$ and $I_{g,\mathrm{vec}}$ also use a small amount of their output domain (0, 1) in the examples measured; in most cases seen $I_c > 0.9$.

In contrast to $I_{c,\mathrm{rnd}}$, $I_{c,\mathrm{vec}}$ and $\mathrm{Var}(\lambda)$, FA is observed across most of its output domain (Table 6.3 and Figure 6.6). However, when measuring a Gaussian cluster, the nature of random matrices means that high-dimensional measurements of FA will seem anisotropic (Section 6.4.3). While this does not universally preclude the applicability of FA in high-dimensional clusters, where data are noisy, this may provide a measurement that does not align with intuitive understandings of isotropy.

One intuition about isotropy, which is common in the existing literature, is the use of principal components to identify the axis of anisotropy [123, 10]. The implementation introduced here $I_{c,\mathrm{rnd}}$ does not follow that intuition, yet still performs well as a measure of isotropy. In Gaussian clusters $I_{c,\mathrm{rnd}}$ is a less accurate approximation of $I_{c,\mathrm{true}}$ than $I_{c,\mathrm{vec}}$. However, in both real-world and synthetic settings, examples have been observed in which $I_{c,\mathrm{rnd}}$ is more accurate than $I_{c,\mathrm{vec}}$ (Figure 6.1d and Table 6.3). While differences between sizes of principal components can be indicative of anisotropy, a lack of data between these principal components can also contribute to anisotropy (*e.g.*, Figure 6.1d). By prioritising measurements on principal axes, existing measures deprioritise the latter of these causes of anisotropy, which can lead to unintuitive results.

Some aspects of whether a cluster of points is isotropic can be subjective (as discussed in the Section 6.1). This chapter does not argue that any of the metrics for isotropy examined here are the best, instead examining their differences and observing cases, which may give unexpected results (*e.g.*, FA being sensitive to noise in high dimensions). Describing the shape of a cluster in a single number will by necessity remove detail and is no substitute for visual inspection. This is even more the case when trying to describe sets of clusters.

However, when working with unlabelled data (as is often the case in materials science), visual inspection may prove impractical. Clusters may be difficult to visually separate, or there may be too many to feasibly inspect each of them. Thus, while simplistic, the tools explored here may be helpful in a number of settings.

In materials science, the introduction of new clustering techniques [138], provides exciting ways to apply clustering algorithms in ways which suit the unique aspects of materials science data. However, if these clustering algorithms are dependent on distance metrics such as the Euclidean distance, then they are also dependent on materials representation. As there is no definitive representation of a material, it is important to have metrics to analyse the resulting clusterings. Where no target labels exist, the metrics available for such analysis are limited. The isotropy of clusters is linked to downstream performance benefits [123], and is particularly pertinent for analysing different representations. Dominant features or correlations of features may not be apparent in lower dimensions (Figure 6.2) and identifying how such correlations may affect clustering is important when drawing conclusions from data or choosing representations of data.

Experiments indicate that these isotropy measures can be used to measure the isotropy of a crystal (Tables 6.4 and 6.5 and Figure 6.7). Which of $\frac{c}{a}$ or the measurements presented here are a better proxy for this intuition of isotropy is not seen from these experiments. As that is a qualitative judgement; it should be done by a researcher with a stronger background in chemistry than this thesis tries to appeal to. What can be said is that all of the issues raised with use of shape of the unit cell for a proxy of crystal isotropy, also apply to the measures of isotropy discussed here. As such, no clear benefit is derived from the using these isotropy measurements on crystal structures. As these measures are more numerically complex than the use of unit cell parameters, there is no reason to recommend the use of these measures for this purpose.

6.6 Conclusion

Analysing sets of clusters is a common task both in the materials science and broader data science domains. While metrics (referred to as internal cluster validation metrics) exist to analyse the compactness and separability of unlabelled clusters, analysis of the shape of clusters has till now been qualitative.

This research offers a thorough exploration of metrics for isotropy (*i.e.* spikiness) of a cluster of points. One such metric, FA was demonstrated on higher-dimensional data. Through use of theorems from the field of random matrix theory [113], it is demonstrated that in high dimensions this measure is susceptible to giving more anisotropic results than expected for noisy data.

A separate measure for isotropy of a point cloud, $I_{c,\text{vec}}$, was examined. An alternative implementation, $I_{c,\text{rnd}}$, was proposed based on the existing derivation of $I_{c,\text{vec}}$. The differences between $I_{c,\text{vec}}$ and $I_{c,\text{rnd}}$ were discussed. Neither $I_{c,\text{rnd}}$, nor $I_{c,\text{vec}}$ was seen to be universally more accurate or faster to compute. However, for high-dimensional data, $I_{c,\text{rnd}}$ is orders of magnitude faster to compute than $I_{c,\text{vec}}$

$I_{c,\text{rnd}}$, $I_{c,\text{vec}}$, FA and a basic proxy for isotropy, $\text{Var}(\lambda)$, were generalised to measure isotropy unlabelled clusters of data. This chapter demonstrates two real-world applications of these generalisations: One in the materials informatics domain and one in a broader data science domain.

In materials science, understanding the output to clustering is particularly important. Datasets are often heterogeneous, and clustering algorithms tend to perform poorly. The usefulness of isotropy measures for clusters of data is demonstrated by exploring clusters found in the ICSD, a canonical materials science dataset. Previous research qualitatively described these clusters using low-dimensional visualisations; we quantify these findings by measuring isotropy numerically.

Broader data science applicability of these measures are demonstrated on by analysing learnt representations of a fundamental data science dataset (MNIST). Isotropy measures for sets of clusters allow for more thorough exploration and description of these representations than would otherwise be possible.

Internal cluster validation measures are helpful tools for chemists and data scientists to understand and quantify unlabelled sets of clusters. Isotropy is pertinent to machine learning for materials science as appropriate material representation is often unclear, and anisotropy in a cluster can be indicative of dominant features in a representation of a material.

This study does not set out to posit isotropy as being a “good” thing in data. Instead it simply investigates how it may be measured. As isotropy is not necessarily a “good” thing, the question emerges as to whether these tools have any purpose. The examples (Section 6.4) shown here demonstrate two promising uses for these measures (and one use case which was not promising, Section 6.4.4).

Measurements for isotropy were used to explore phenomenas seen in the previous chapter, and help explain that behaviour (Section 6.4.1). This shows that these tools can be used to help better explore data.

These measurement were also seen to be helpful for quantify observations in learnt latent spaces (Section 6.4.2). While it is not suggested that isotropy measurements are used in place of visual inspection, these tools are none the less helpful to provide quantitive measurements to qualitative observations.

Implementations of these metrics are provided in the associate code repository [43]. The metrics for isotropy presented here are a helpful addition to existing metrics, which allow researchers to richly explore their datasets.

6.7 Thesis context

The exploration of clustering metrics presented in this chapter is a slight divergence from the property prediction discussed previously. Instead, this chapter levarages the context of materials science to explore new ideas about quantifying clustering.

Clustering is a widely used technique, foundational to data science. This chapter has used materials science as a setting to explore clustering. It is fascinating that the context of materials science allows for further exploration of such concepts, both as presented here and more widely in the literature [138].

The final experimental chapter takes the relationship explored in this chapter and reverses it. Examining band structures from a data science perspective by discussing the data structures through which band structures are can be represented. In exploring the interaction between data science and materials science, these two chapters mark the thematic culmination of the thesis.

6.8 Appendix

Theorem 1. *For a finite sized set of real numbers between 0 and 1, λ, the variance of λ is bounded between 0 and 0.25.*

Proof of 1. Let $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_n$ be a set of real numbers such that $0 \leq \lambda_i \leq 1$ for all i. Let $\bar{\lambda} = \frac{1}{n}\sum_{i=1}^{n} \lambda_i$ be the mean of the set λ. Then, we have:

$$\mathrm{Var}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\lambda_i - \bar{\lambda}\right)^2 \tag{6.32}$$

As for all values of i, $\left(\lambda_i - \bar{\lambda}\right)^2 \geq 0$ the lower bound for $\mathrm{Var}(\lambda)$ is 0.

To prove the upper bound of $\mathrm{Var}(\lambda)$, observe that as $\lambda_i \leq 1$ it can be said that:

$$\sum_{i=1}^{n}\lambda_i^2 \leq \sum_{i=1}^{n}\lambda_i \tag{6.33}$$

As $n\bar{\lambda} = \sum_{i=1}^{n}\lambda_i$:

$$\sum_{i=1}^{n}\lambda_i^2 \leq n\bar{\lambda} \tag{6.34}$$

And so:

$$\begin{align} n \cdot \mathrm{Var}(\lambda) &= \sum_{i=1}^{n}\left(\lambda_i - \bar{\lambda}\right)^2 \tag{6.35}\\ &= \sum_{i=1}^{n}\left(\lambda_i^2 - 2\lambda_i\bar{\lambda} + \bar{\lambda}^2\right) \tag{6.36}\\ &= \sum_{i=1}^{n}\lambda_i^2 - 2\bar{\lambda}\sum_{i=1}^{n}\lambda_i + n\bar{\lambda}^2 \tag{6.37}\\ &= \sum_{i=1}^{n}\lambda_i^2 - 2\bar{\lambda}\cdot n\bar{\lambda} + n\bar{\lambda}^2 \tag{6.38}\\ &= \sum_{i=1}^{n}\lambda_i^2 - n\bar{\lambda}^2 \tag{6.39} \end{align}$$

Thus, as per Equation 6.34:

$$n \cdot \mathrm{Var}(\lambda) \leq n\bar{\lambda} - n\bar{\lambda}^2 \tag{6.40}$$

Thus:

$$\mathrm{Var}(\lambda) \leq \bar{\lambda} - \bar{\lambda}^2 \tag{6.41}$$

The maximum value of $\bar{\lambda} - \bar{\lambda}^2$ occurs when $\bar{\lambda} = 0.5$ and so

$$\mathrm{Var}(\lambda) \leq 0.5 - 0.5^2 \tag{6.42}$$

$$\leq 0.25 \tag{6.43}$$

Thus, the variance λ is bounded [0, 0.25].

□

Theorem 2. *The ratio of minimal and maximal values Z(*$\mathbf{a}$*) for any unit vector* $\mathbf{a}$*:*

$$\frac{\min_{|\mathbf{a}|=1} Z(\mathbf{a})}{\max_{|\mathbf{a}|=1} Z(\mathbf{a})} \tag{6.44}$$

is not invariant to uniform scaling or linear isometries.

Proof of 2. Consider a cluster of points, $\mathcal{C}$, let $\mathcal{C}' = \alpha\mathcal{C} + \beta$ where α and β are a scalar and a translation vector, respectively. Were Equation 6.44 invariant under uniform scaling and linear isomoteries, then the ratio for $\mathcal{C}'$ would be the same as that of $\mathcal{C}$. The value of $Z(\mathbf{a})$ for $\mathcal{C}'$ would be:

$$\sum_{\mathbf{d}\in\mathcal{C}} \exp\left(\mathbf{a}^{\intercal}(\alpha\mathbf{d} + \beta)\right) \tag{6.45}$$

Neither α nor β can be factored out. Thus, the value of Equation 6.44 would change. Therefore, Equation 6.44 is not invariant under uniform scaling or linear isometries. □

Theorem 3. *Any* $I_{c|\mathcal{B}}$ *is an upper bound for* $I_{c,\text{true}}$*:*

$$\forall_{c,\mathcal{B}} : I_{c,\text{true}} \leq I_{c|\mathcal{B}} \tag{6.46}$$

Proof of 3. Consider the formula for $I_{c,\text{true}}$:

$$I_{c,\text{true}} = \frac{\min_{|\mathbf{a}|=1} Z'(\mathbf{a})}{\max_{|\mathbf{a}|=1} Z'(\mathbf{a})} \tag{6.47}$$

and the formula for $I_{c|\mathcal{B}}$:

$$I_{c|\mathcal{B}}(\mathcal{C}) \approx \frac{\min_{\mathbf{b}\in\mathcal{B}} Z'(\mathbf{b})}{\max_{\mathbf{b}\in\mathcal{B}} Z'(\mathbf{b})} \tag{6.48}$$

where $\mathcal{B}$ is a set of unit vectors. $\mathcal{B}$ is a subset of the set of all unit vectors, $|\mathbf{a}| = 1$. For it to be the case that $I_{c|\mathcal{B}} > I_{c,\text{true}}$, one or both of the following must be true:

$$\min_{|\mathbf{a}|=1} Z'(\mathbf{a}) > \min_{\mathbf{b}\in\mathcal{B}} Z'(\mathbf{b}) \tag{6.49}$$

and/or:

$$\max_{|\mathbf{a}|=1} Z'(\mathbf{a}) < \max_{\mathbf{b}\in\mathcal{B}} Z'(\mathbf{b}) \tag{6.50}$$

However, as $\forall_{\mathbf{b}\in\mathcal{B}} : |\mathbf{b}| = 1$ neither of these statements can be true. Thus

$$\forall_{c,\mathcal{B}} : I_{c,\text{true}} \leq I_{c|\mathcal{B}} \tag{6.51}$$

□

Theorem 4. *As the size of the set of random unit vectors used approaches infinity,* $I_{c,\text{rnd}}$ *approaches* $I_{c,\text{true}}$*:*

$$\lim_{|r|\to\infty} I_{c,\text{rnd}} = I_{c,\text{true}} \tag{6.52}$$

Proof of 4. As the number of unit vectors in r approaches infinity, the probability that a random unit vector is in r approaches:

Let $\mathbf{b}$ be a unit vector. The probability of choosing $\mathbf{b}$ from the uniform distribution of the set of all unit vectors is:

$$P(\mathbf{b}) = \frac{1}{||\mathbf{a} = 1||} \tag{6.53}$$

where $||\mathbf{a} = 1||$ is the cardinality of the set of all unit vectors. If r is sampled uniformly from the set $|\mathbf{a}| = 1$ then:

$$P(\mathbf{b} \in r) = \frac{|r|}{||\mathbf{a} = 1||} \tag{6.54}$$

Therefore:

$$\lim_{|r| \to \infty} P(\mathbf{b} \in r) \to 1 \tag{6.55}$$

and

$$\lim_{|r| \to \infty} r \to |\mathbf{a}| = 1 \tag{6.56}$$

Consequently:

$$\lim_{|r| \to \infty} I_{c,\text{rnd}} = I_{c,\text{true}} \tag{6.57}$$

□

Chapter 7

Machine learning with electronic band structures

7.1 Introduction

Most of the machine learning (ML) presented so far in this thesis has focused on composition-based prediction. As discussed (Section 2.2.5), by definition, composition-based representations cannot encapsulate all the properties of a material, as materials with the same composition may have different structures and thus different properties. The electronic band structure (EBS) of a material offers information that theoretically encompasses all of the electronic properties of materials, by representing the ways in which electrons can exist within that material (see Section 2.2.4). The EBS of a material can be approximated using density functional theory (DFT), and although this is computationally expensive, the cost of DFT is generally much lower than that of synthesis [23]. As such, if the properties of the materials can be estimated from a DFT-calculated EBS, it would be possible to more efficiently screen materials using DFT and ML than by using synthesis alone.

Interpreting an EBS by humans requires expertise [167] and is time consuming, making it impractical to perform at scale. Thus, being able to algorithmically progress from an EBS to electronic properties would be advantageous for mass screening of DFT-calculated materials.

Large repositories of EBS data exist [78], which could provide ample training data for ML algorithms. Using one such repository, the Materials Project, as a proof of concept, this chapter will detail using ML to predict the electronic band gap given an EBS. This is used as a proof-of-concept for the algorithmic approaches examined. As the band gap can already be algorithmically extracted from an EBS, this means there is a target value for all materials for which an EBS is available.

While predicting band gaps demonstrates the ability of ML algorithms to interpret EBS data, it is desirable for these algorithms to predict properties that cannot be easily calculated using existing methods. However, there is a lack of data connecting EBS repositories to experimentally measured properties. This chapter will detail a method for connecting these repositories to a dataset of experimentally measured resistivities. The resulting dataset of EBSs and resistivities is then used to train ML algorithms to predict the resistivity of a material from its EBS.

Specifically, the contributions of this chapter are as follows:

- Creation of a dataset of relating EBS data to resistivity.
- Exploration of the caveats surrounding using machine with EBS data.
- Suggesting numerous paradigms through which EBS data can considered. This allows easy adaptation of ML algorithms intended for those paradigms.
- Novel implementation of deep learning algorithms across datasets of EBS data.
- Creation of two RNNs to interpret EBS data.
- Use of a set transformer model in two different arrangements to interpret EBS data.
- Improving performance in predicting resistivity, and band gap by several orders of magnitude compared to a RF.

7.2 Related Literature

Despite the large existing repositories of EBS data ([78, 32, 15]), the literature surrounding machine learning from band structure data is sparse. A prominent example of learning from EBS data is the use of the density of states (DOS) as a tool for machine learning [93, 91, 22]. This has been enhanced by including information about how DOS changes at different points in the unit cell (specifically, each high symmetry point) [76, 87].

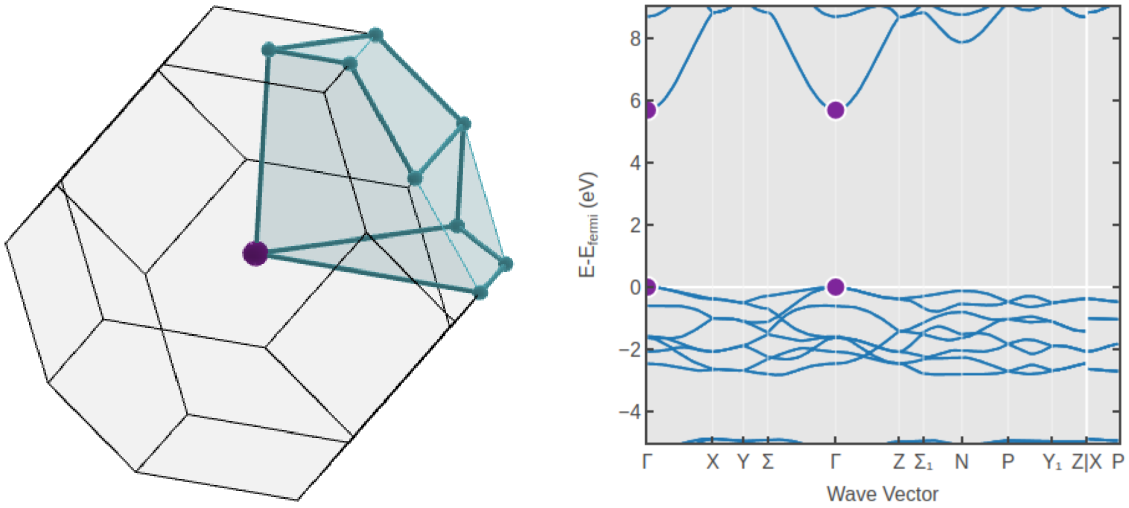
However, neither of these approaches based on DOS directly learn from EBS data. One approach which learns directly from band structure data is TBHCNN [190], which uses a convolutional network in order to parameterise tight-binding Hamiltonians. This is done by repeatedly using learnt parameters of a tight-binding Hamiltonian to reconstruct an EBS, and using the reconstructed EBS to further train the model. TBHCNN is not generalisable across EBS data, and needs to be re-trained on each band structure. This means that many of the caveats related to the heterogeneous nature of EBS datasets (which will be further discussed Section 7.3) are avoided, as only one EBS is considered at a time.

Other models take a similar approach, using Markov random fields to reconstruct EBS data, considering only one EBS at a time [200]. While these models do not serve as an example of the supervised property prediction discussed in this chapter, it is interesting to note that machine learning could be used to improve the data which are learnt in experiments presented here.

7.3 The nature of the Materials Project electronic band structure data

While DFT calculations can be done for any point in the reciprocal lattice, convention is often to perform calculations along lines of symmetry in the lattice (Figure 7.1a). This collection of lines, sometimes called a K-path, can be laid out in two dimensions with the X axis being a point (K-point) along the K-path and the Y axis being the energy level (Figure 7.1b). This is usually normalised to the Fermi level E_{fermi}, or the level to which energy states are occupied by electrons at 0 K. While EBSs can be plotted, they can equally be considered a matrix (Figure 7.2) with each row representing an electronic band and each column representing a K-point. EBS data in the Materials Project do not sample a uniform number of bands, ranging from 8 to over 1000, nor do they sample the same number of k-points per high-symmetry line. The number of high-symmetry lines varies depending on a materials space group, and the K-path between those lines may also vary.

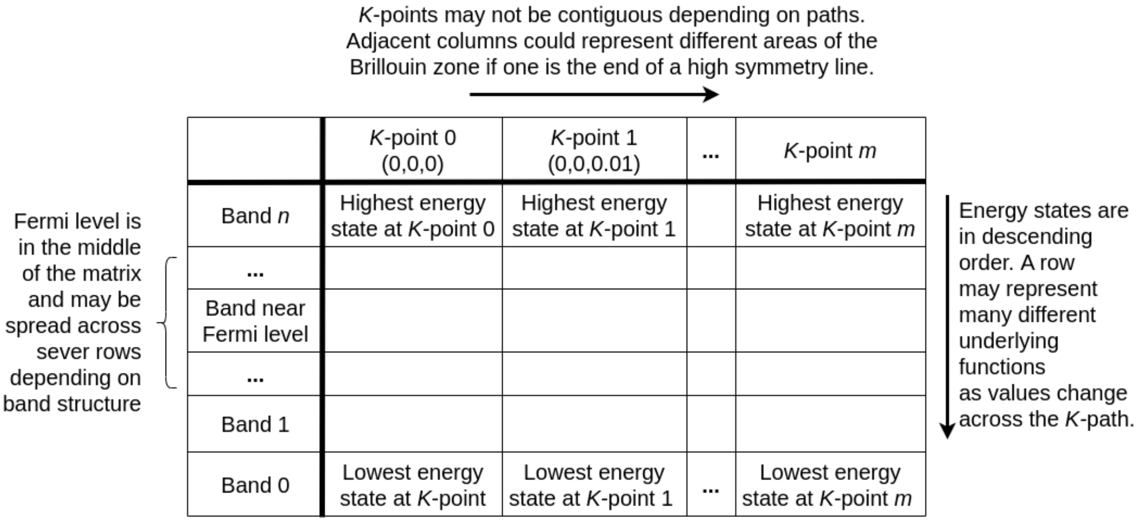
Another caveat of these data is that the matrix of data for each point does not align precisely with band theory. According to Bloch's theorem, these are discrete functions, which interact to give the properties of a material. However, when represented as a matrix, energy states at a given K-point are arranged in strict descending order (Figure 7.2). Thus, it is unclear which energy states belong to each function, and sorting energy states into those which are generated by a given function is non-trivial (machine learning has recently been used for this task, as discussed in Section 7.2). It is unclear what effect this will have



(a)

(b)

Figure 7.1: Diagrams of SiO$_2$ generated with the materials project [78]. (a) The first Brillouin zone shows a tetragonal structurre, the high symmetry path highlated is the K-path along which the band structure is calculated. (b) The electronic band structure, with K-path being represented across the x-axis, and energy states being seen along the y-axis. Note that the final line in the K-path $(X \to P)$ is disconnected from the line before it in this case.


K-points may not be contiguous depending on paths.
Adjacent columns could represent different areas of the
Brillouin zone if one is the end of a high symmetry line.
Fermi level is in the middle of the matrix and may be spread across sever rows depending on band structure
Energy states are in descending order. A row may represent many different underlying functions as values change across the *K*-path.

	K-point 0 (0,0,0)	*K*-point 1 (0,0,0.01)	...	*K*-point *m*
Band *n*	Highest energy state at *K*-point 0	Highest energy state at *K*-point 1		Highest energy state at *K*-point *m*
...				
Band near Fermi level				
...				
Band 1				
Band 0	Lowest energy state at *K*-point	Lowest energy state at *K*-point 1	...	Lowest energy state at *K*-point *m*

Figure 7.2: EBS data can be represented in a matrix with one axis representing energy and the other representing K-points. This has several caveats such as that a single row might not be representative of a single function, and adjacent columns are not necessarily contiguous.

on machine learning algorithms, but effects can be seen when operating strictly across columns of the EBS matrix. For example, obtaining the maximum value of each row in the EBS matrix will not result in the maximum value for each function.

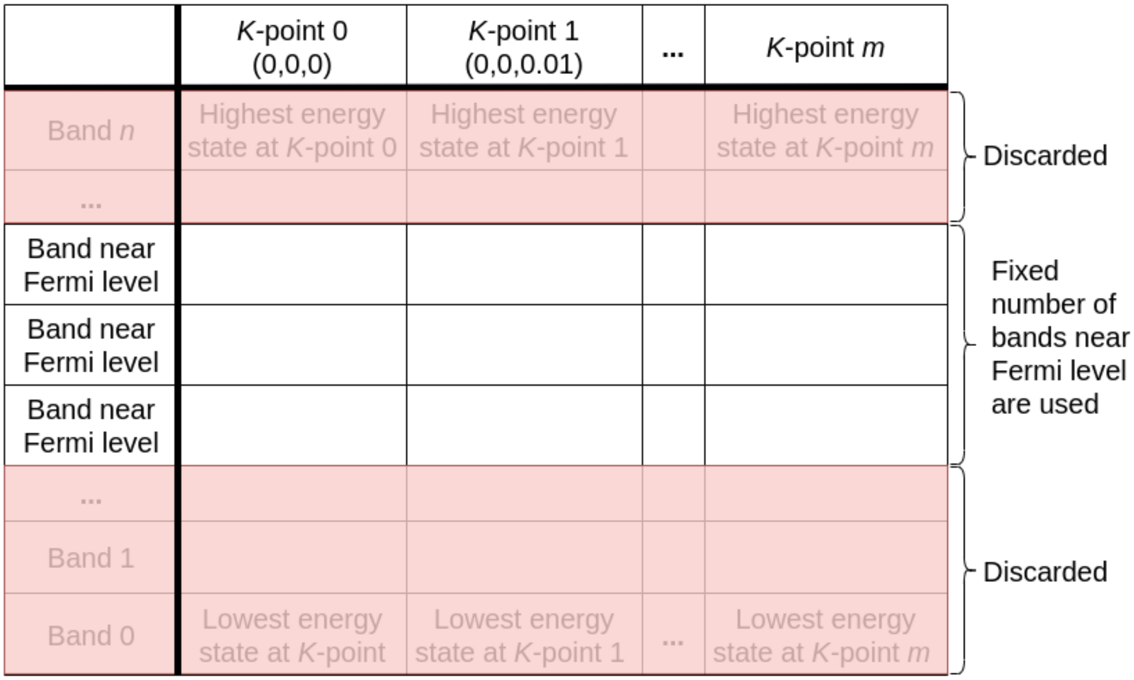
Overall, while EBSs represent the underlying physics in the same way, each data point can differ wildly and is not necessarily reflective of other discrete underlying functions described in Bloch's theorem. This makes them very interesting candidates for ML, as it limits the models that can be used on the data directly without any preprocessing. In introducing this limit, ML practitioners are forced to consider which paradigms and model classes are suited to these data. But with the literature here being so sparse, a broad horizon of research opportunities awaits.

The limitations imposed by the varying axes of EBSs can, of course, be overcome through preprocessing rather than through use of appropriate models. As mentioned (Section 7.2) there is precedent for this in the literature [161]. Examples of preprocessing steps which might make these data easier to process include:

1. Cropping to bands near the fermi surface: Only considering a fixed number of bands near the fermi surface fixes one axes of the matrix, however, discards information far

above and below the fermi surface which may influence the properties of a material. As bands closest to the fermi surface are likely to have the greatest impact on a material's properties, it is unclear what impact this will have (Figure 7.3).

2. Interpolation of K-points in the EBS: To deal with unevenly sampled K-points, or different numbers of K-points, basic statistical or unsupervised learning methods such as linear interpolation and K-nearest neighbours (K_{nn}) can be used. The interpolation of K points has previously been used to extract information from band structures [161], and more recently, more complex methods of interpolation have been suggested for these data [190, 200], so there is precedent for this in the literature. This interpolation can be used to enforce a set number of K-points per line.



	K-point 0 (0,0,0)	*K*-point 1 (0,0,0.01)	...	*K*-point *m*
Band *n*	Highest energy state at *K*-point 0	Highest energy state at *K*-point 1		Highest energy state at *K*-point *m*
...				
Band near Fermi level				
Band near Fermi level				
Band near Fermi level				
...				
Band 1				
Band 0	Lowest energy state at *K*-point	Lowest energy state at *K*-point 1	...	Lowest energy state at *K*-point *m*

Figure 7.3: Cropping to a fixed number of bands near the fermi surface means that there is a consistant number of bands in all structures in the dataset. This also means the dataset is limitted by the band structure with the least band data available.

By enforcing across the entired dataset a fixed number of bands, and a fixed number of K-points using the above processes, each EBS can be represented using a fixed sized vector. By using a fixed sized input vector many algorithms can now be used, which were previously inappropriate, such as random forests (RFs), logistic regressions and Ridge regressions.

However, this process also removes a lot of data from the EBS. It is both more interesting and perhaps more suitable to ask how do we choose algorithms to fit these data rather than how do we adapt these data to fit readily implemented algorithms. Whether this is a more suitable approach is subject to investigations such as those presented in this chapter.

The problem of extracting properties from band structures can be considered as pattern recognition, a field in which deep learning is known to be very effective. There are also various deep learning architectures that would be appropriate for these data. As such, deep learning is a prime choice for use in this case.

The classes of machine learning algorithms investigated depend on the paradigm through which one views EBS data. This chapter proposes the following paradigms are proposed to be appropriate:

1. An EBS is a sequence of bands, each of which is a sequence of K-points: A hierarchical sequence processing task, analogous to a sequence of words making up a sentence and a sequence of sentences making up a paragraph.

2. An EBS of K-points each of which has sequence of possible energy states: Similar to the above but the order of hierarchy is switched.

3. An EBS is a set of k-points which consist of sets of energy states: Similar to a hierarchical sequence, but the idea of bands existing in an order is removed.

4. An EBS is set of fixed-sized bands: By forcing a consistent number of K-points across EBSs of all materials, one can consider a variable number of band structures, this is analogous to the colour channels in a picture, which are not in a fixed order and variations both between and within channels make a define features.

5. An EBS is set of K-points each with a fixed number of energy states: As above but considering position in the reciprocal space as the channel rather than bands.

6. An EBS is an ordered list of fixed-sized bands.

7. An EBS is an ordered list of K-points, each with a fixed number of energy states.

8. An EBS is a 4 dimensional point cloud: Three dimensions represent the position in K-space with a fourth representing the energy at which an electron can exist. This

has the added benefit of explicitly noting that the order of the rows in the EBS matrix is not indicative of the discrete functions that dictate materials properties.

9. An EBS is a 3+N dimensional point cloud: By cropping data to the N bands nearest the Fermi level, one can consider a much smaller point cloud (in a higher-dimensional space).

This list is non-exhaustive and each of the above paradigms has its own advantages and disadvantages. This chapter will investigate paradigms 1, 6, 8, and 9.

7.4 Creation of dataset

Peer reviewed experimental data were collected from the Materials Platform for Data Science (MPDS) [15]. The resistivity measurements were extracted, and the results were then aligned with materials from the Materials Project for which EBS data were available. As resistivity is affected by temperature, it is important to consider the temperature at which measurements were taken. Measurements between 280 K and 315 K were considered. The measurement closest to 298 K was considered if multiple measurements were available, as 298 K was the modal temperature for resistivity measurements in the MPDS (Figure 7.5a). When multiple measurements of the same material at the same temperature were available, the mean was taken.

The resistivity in this data set varies by approximately 25 orders of magnitude (Figure 7.5b). As will be seen, this is a problem when trying to accurately predict resistivity (Section 7.6), and will lead to problems with numerical stability. (Section 7.7) Entries in the Materials Project and MPDS datasets with the same composition and space group were considered to be the same material, allowing for the alignment of 2970 materials, which were split into train/test/validation sets (of size 2227, 593, and 150 respectively) (Figure 7.4a). A drawback of this method of alignment is that just because two entries are in the same space group and have the same formula does not mean that they are the same material. However, as this is exploratory work, this is considered to be a sufficient approximation.

There remains a large repository of EBS data in the Materials Project repository (approximately 65,000 entries). As noted (Section 7.1) the prediction of the band gap can be used as a proof of concept for these data. It would be completely possible to use these

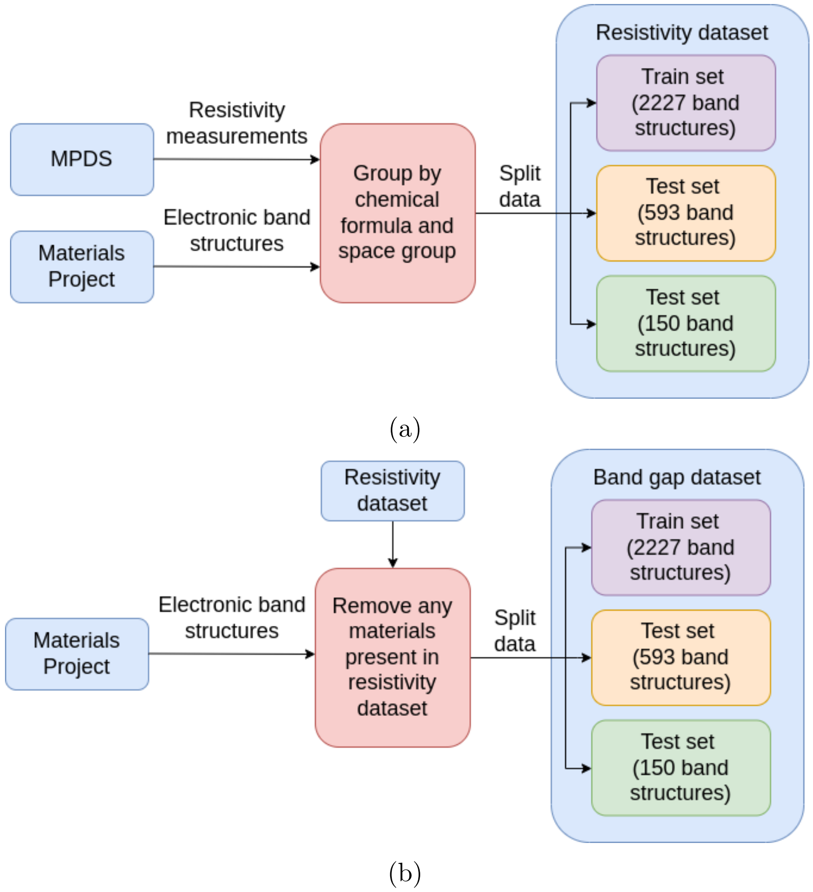
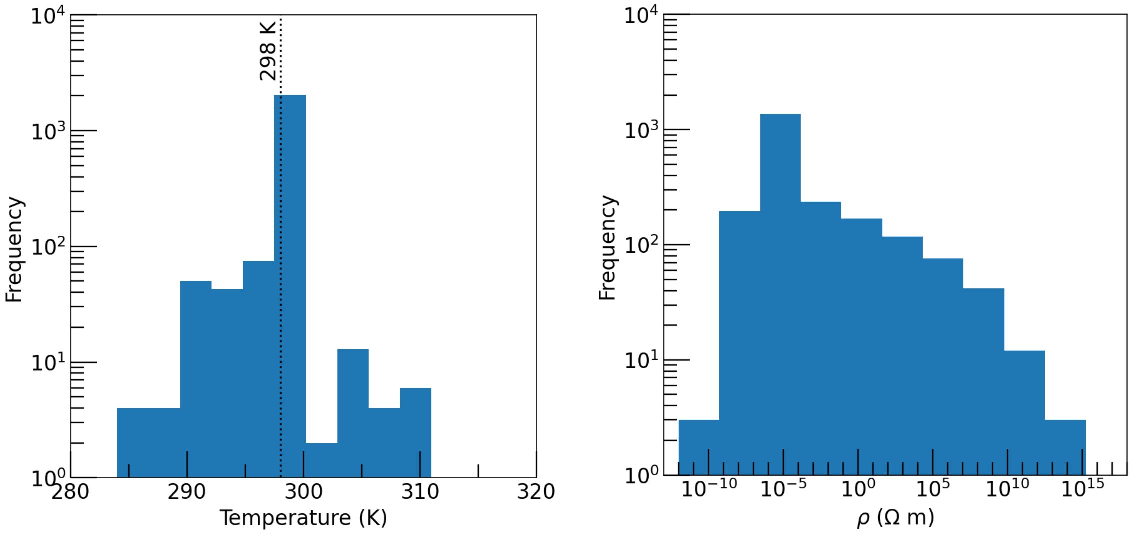

Resistivity dataset
MPDS
Resistivity measurements
Materials Project
Electronic band structures
Group by chemical formula and space group
Split data
Train set (2227 band structures)
Test set (593 band structures)
Test set (150 band structures)
(a)
Resistivity dataset
Band gap dataset
Materials Project
Electronic band structures
Remove any materials present in resistivity dataset
Split data
Train set (2227 band structures)
Test set (593 band structures)
Test set (150 band structures)
(b)

Figure 7.4: (a) Creation of the resistivity electronic band structure (EBS) dataset used in this chapter. (b) Creation of the band gap EBS dataset used in this chapter.



(a) (b)

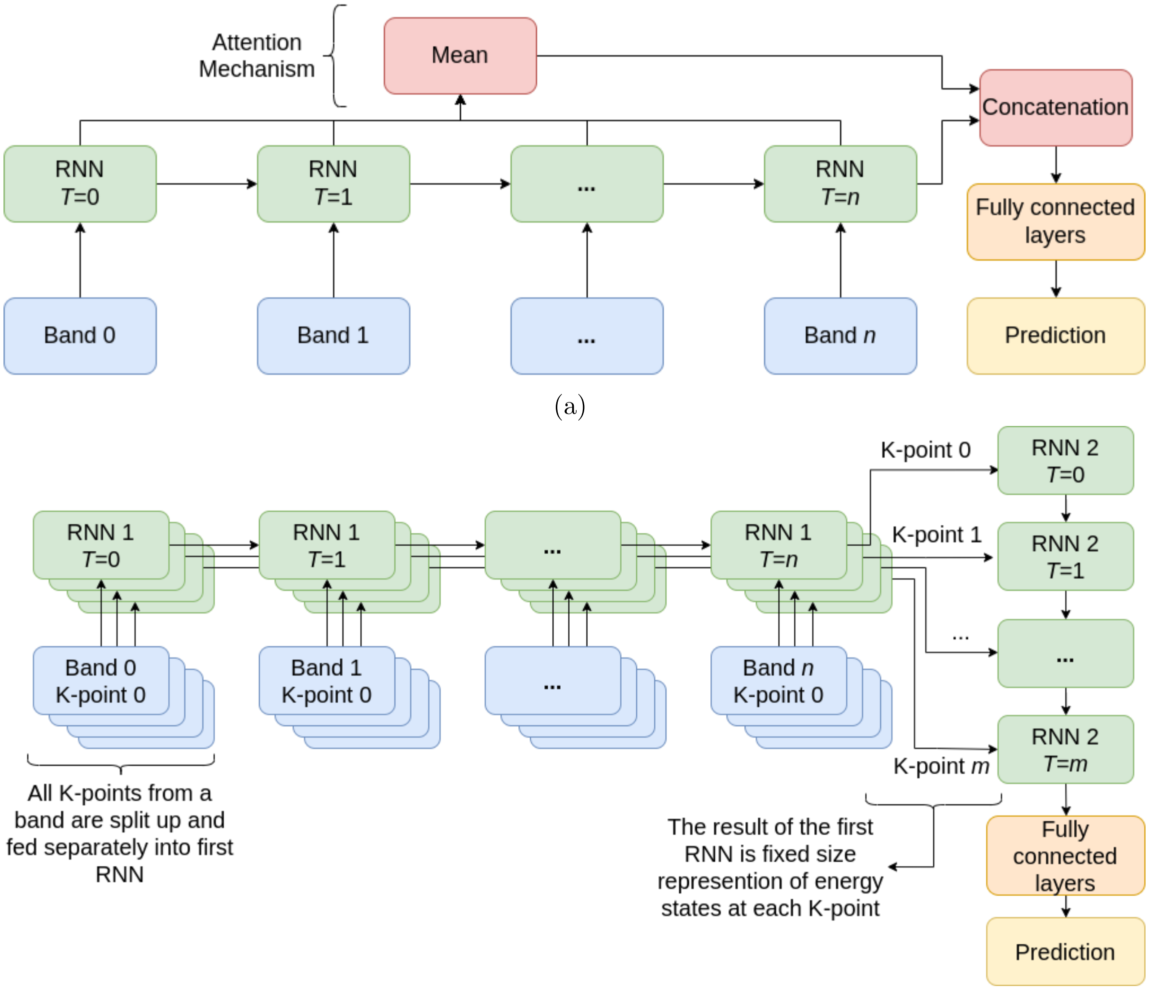
Figure 7.5: Distribution of aspects of the resistivity training set (a) Temperatures at which measurements were taken. (b) Resistivity measurements. The scale of resistivity measurements vary widely.

data as a pre-training task and use transfer learning to then adjust model weights to predict resistivity. Although this will be suggested in future work (Section 7.7), this transfer learning is beyond the scope of this chapter.

Instead of doing transfer learning with these data, the work presented in this chapter will use a subset of the EBS data in the Materials Project to as a proof of concept of whether it is possible to train deep learning algorithms using limited data on a task which is known to be algorithmically possible (*i.e.*, extracting band gap from EBS data). As such, a data set was constructed from EBS data from the Materials Project repository with associated band gaps. The size of this dataset was the same as the resistivity EBS dataset (size 2227, 593, and 150 train, test, and validation sets, respectively). Any EBS that was present in the resistivity dataset was also excluded so that future work can use this dataset as pre-training if needed (Section 7.4).

7.5 Models investigated

Four types of models were investigated on these data:



(a)

(b)

Figure 7.6: Recurrent neural network (RNN) based models for prediction, operating over time steps (T), on a band structure with n bands, each band containing m K-points. (a) RNN 1D shown with the attention mechanism. As input at each time step must be of fixed size, this means m must be fixed for the entire dataset. (b) RNN 2D architecture, for visual clarity the netowrk shown here without an attention mechanism, though similar to (a), an attention mechanism was added in some experiments. The second RNN allows m to vary between points in the dataset.

- As a baseline a RF was used, with *magpie*, *jarvis*, *Oliynyk* and *fractional* representations, as well as random projections of size 25, 50, 100, and 200 (for more information on these representations and random projections, see Chapter 5)
- A recurrent neural network (RNN), which iterates over a variable number of fixed sized bands.
- A hierarchical RNN in which one RNN iterates over the energy states for a variable number of bands, then a second RNN iterates over the K-points. This would allow for the number of K-points sampled to vary between each EBS, but for a fair comparison with the above, it was kept constant in all experiments seen here.
- A set transformer architecture taking in a set of 4D points, where the first 3 dimensions represent a position in reciprocal space, and the final dimension represents an energy level at that position.
- A set transformer architecture taking in a set of 11D points, where the first 3 dimensions represent a position in reciprocal space and the final 8 dimensions represent the 8 energy levels closest to the Fermi level at which electrons can exist.

The deep learning architectures investigated will now be briefly explained before results are conveyed.

7.5.1 Recurrent Neural Networks for band structures

Recurrent neural networks operate in sequences in time steps, each time step taking as input an item in the sequence and the output of the previous time step (Figure 7.6a) [71], this allows the input to be of variable length, with the output of the final time step being used for some predictive purpose, such as classification. As basic RNN's involve backpropagation through many time steps, the gradient of loss with respect to the weights at earlier time steps becomes very small. This is called the vanishing gradient problem and is addressed by a recurrent architecture called long-short-term memory cells (LSTMs) [70], which carry two different states between time steps, one of which is only combined in an additive manner. The result is an RNN architecture that can be trained on longer sequences.

One limiting factor for training RNNs can be the size of the associated computational graph. The memory requirements grow with respect to sequence length, and as such long sequences can become expensive to process. This can be adjusted by reducing batch size, or reducing the size of the network; however, both of these have other impacts on learning.

RNNs can be applied hierarchically, with outputs from one RNN being used as input for another RNN. This further adds to computational complexity. This means that the size of a network may need to be reduced in order to maintain compliance with any memory constraints.

Attention mechanisms can also be introduced to emphasise earlier time steps [9]. An example of such attention mechanisms, additive attention takes the sum, or mean of the output of all time steps, and uses this in conjunction with the output to the last time step to make predictions.

As noted (Section 7.3), an EBS can be considered to be a sequence of bands and a sequence of k-points. Using preprocessing steps such as linear interpolation or cropping of bands, either, or both, it is possible to force all EBS structures in the datasets to have the same size. Without interpolation, the memory costs of both hierarchical and non-hierarchical hierarchical processing across K-points were unfeasible. As such, linear interpolation was used to normalise the number of k-points to various values ranging from 50 to 200 k-points per band.

This allowed for two architectures to be investigated; both were investigated with various interpolation densities, with various LSTM layer sizes (32, 64, and 128) and numbers of LSTM layers (between 1 and 4). Both architectures were investigated with and without an additive attention mechanism.

The first investigated architecture (dubbed "1D RNN") took each band as a time step and had a fixed number of K-points per band (Figure 7.6a). The output from the final time step (and the attention mechanism, if used), was used as input to a fully connected network which outputs a single value, which was then compared to a target value (either band gap or resistivity), to generate a loss, which was then backpropagated to train the network. The fully connected network had between 1 and 4 layers of sizes 32, 64 or 128.

The second investigated architecture (dubbed "2D RNN") featured two different RNNs, the first of which iterated over the energy levels at a single K-point, the second of which iterated over the the outputs from the first RNN across all K-points (Figure 7.6b). The output to the second RNN was then used as an input (sometimes including addative attention) to a fully connnected network. Again, the fully connected network was investigated with layer sizes of 32, 64 and 128 and varying between 1 and 4 layers.

For both 1D RNN and 2D RNN dropout and layer normalisation, common techniques to improve performance were also tested. A hyperparameter search was done using Bayesean optimisation, across the following hyperparameters:

- **Attention mechanism:** true, false
- **Batch size:** Multiples of 8 between 32 and 256
- **Dropout [169]:** 0.3, 0.4, 0.5, None
- **Number of points that each band resampled to**(interpolation done with linear interpolation)**:** 10, 50, 100, 500
- **Layer normalisation [8] between layers:** Either applied or not applied
- **Learning rate:** A uniform distribution between $1{\times}10^{-07}$ and 0.1
- **Fully connected layer size:** 32, 64, 128, 512
- **Number of fully connected layers:** 1-4
- **LSTM size:** 32, 64, 128, 512
- **Number of LSTM layers:** 1-4
- **Model type:** RNN 1D or RNN 2D (for RNN 2D both LSTMs were the same size and same number of layers)
- **Optimiser:** adam [85], adamw [109] or simple gradient descent [100].

RNN hyperparameters were optimised for the band gap, the best performing hyperparameters of the RNN 1D and RNN 2D models hyperparameters were then used for networks trained on the resistivity dataset. The exception to this was the optimiser for the RNN 1D model, which was switched to AdamW (with the same learning rate), as simple gradient descent was found to result in floating-point errors for the resistivity data.

7.5.2 Set transformers for band structures

Transformers are widely used class of deep learning models [104, 89], most commonly in NLP settings, though they have also been seen to perform well in materials settings [205, 202]. The central mechanism of transformer architectures is self-attention heads. Inputs

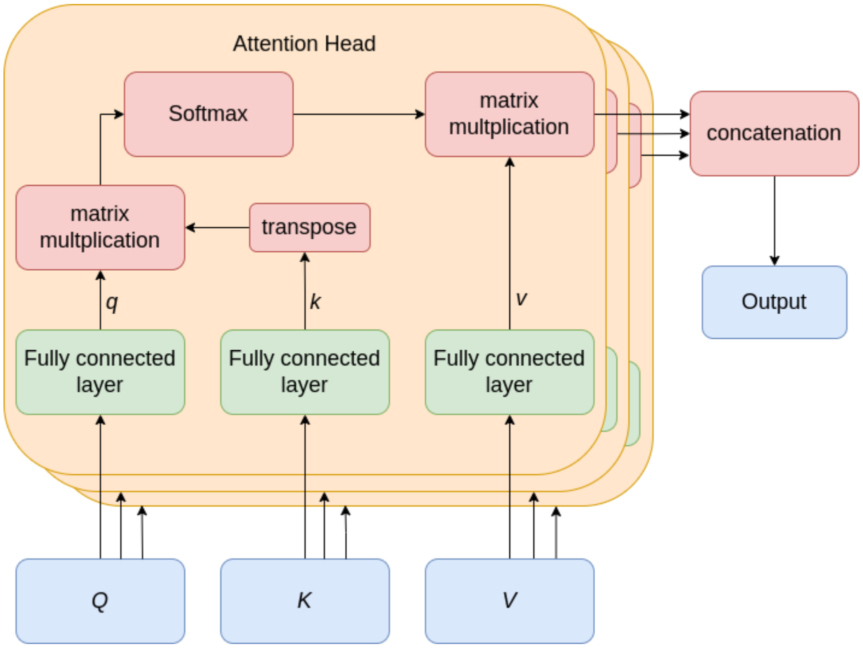


Figure 7.7: The multi-head attention block (MAB)

(Q, K, and V) are passed through three separate, fully connected layers to produce three matrices: query (q), key (k), and value (v), these are combined with the following function:

$$\text{Softmax}(qk^{\intercal})v \tag{7.1}$$

Note that often $Q = K = V$, which allow each part of the sequence to align itself with other parts of the sequence in a learnt way, rather than just through proximity in the sequence order. Note that the activation function is not by definition a Softmax function [101, 181], but for the purposes of this chapter it always will be. This process is often run multiple times in different "heads" (Figure 7.7) and the output is then concatenated and used later on in the network. Thus, a multi-head attention block (MAB) can be defined as:

$$\text{MAB}(Q, K, V) = \text{Softmax}(q_0k_0^{\intercal})v_0 +\!\!+ \text{Softmax}(q_1k_1^{\intercal})v_1 +\!\!+ \ldots +\!\!+ \text{Softmax}(q_ik_i^{\intercal})v_i \tag{7.2}$$

Where $+\!\!+$ is a concatenation operator, i is the number of heads. How the network uses MABs is network-specific, with most architectures having an encoder-decoder architecture [181, 104, 89] (see Section 6.4.2 for more on encoder-decoder architectures).

A key drawback of these networks is that their memory requirements scale in a quadratic relationship with the length of the input sequence (since each element of the sequence must be compared to every other element of the sequence). This means that transformers can be very costly to train, requiring a lot of memory. Various methods have been suggested to combat this [189, 25, 112]. One such method to reduce the quadratic relationship between sequence length and memory requirements involves using a fixed-sized random projection of v and k in the self-attention process. This is not directly relevant to this chapter, but is of tangential interest to the thesis (Chapter 5).

Transformer architectures also often include a positional encoding module on the input, which assigns unique vectors to each position in the input sequence. Set transformer architecture [101] adapts the transformer architecture by removing this module. This has the effect of making the model invariant to the order input, thus suitable for processing sets.

The Set Transformer also differs from other transformer architects by introducing an alignment a learnt set of weights (Figure 7.8), which has the effect of reducing memory constraints to be a function of the size of those weights (and thus a hyperparameter of the model). In total the set transformer is made up of three main blocks, which are labelled SAB (self-attention block), ISAB (ambiguous abbreviation [101]), and PMA (pooling by multi-head attention)

SAB is considered to be MAB where $Q = K = V$, thus:

$$\text{SAB}(X) = \text{MAB}(X, X, X) \tag{7.3}$$

ISAB (Figure 7.8) introduces a learnt set of weights, I of size $l \times h$ (l and h are hyperparameters), consists of two MAB, one of which takes in I as the query, resulting in an h dimensional output for each member of the input set:

$$\text{ISAB}(X) = MAB(X, H, H) \tag{7.4}$$

$$\text{where} \quad H = MAB(I, X, X) \tag{7.5}$$

Unlike SABs which scale with quadraticly in memory complexity with the size of the input set, ISABs scale inline with h. *i.e.* they have memory complexing $\mathcal{O}(nh)$, rather than $\mathcal{O}(n^2)$ where n is the size of the set.

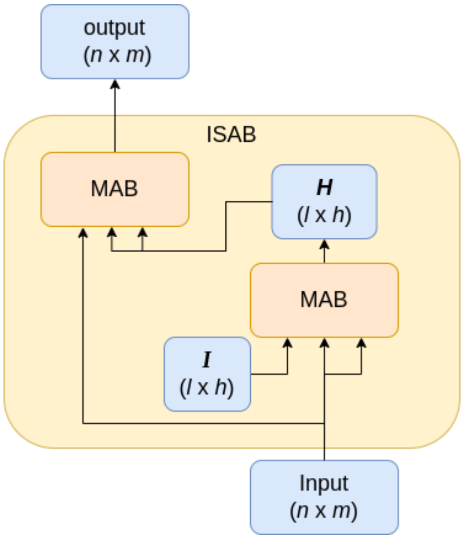


Figure 7.8: The ISAB applies MABs in combination with a learnt matrix I such that the memory constrains scale with the size of I (a hyperparamter) rather than scaling quadratically with the length of the sequence. The sizes of matrices are shown in brackets.

PMA blocks act as the intermediary between the encoder and the decoder, and takes a variable sized input and transforms it into a fixed-size output using a learnable seed vector S, (note that S can be replaced with a set of seed vectors if multiple outputs are required but that is not the case here.). PMA is defined as:

$$\mathrm{PMA}(X) = \mathrm{MAB}(S, X, X) \tag{7.6}$$

By having an encoder made out of ISABs, and decoder made out of SABs, the scaling of memory complexity of the network becomes controllable by hyperparameters (the size of l, the size of S and the number of IMAB and SAB blocks). This allows the algorithm to run with less memory (though in practise sampling to 100 points per band was still required for feasible training for band structures).

The set transformer network used for the experiments here (Table 7.1) consisted of an encoder of two ISAB blocks and a PMA, connected to a decoder of two SAB blocks and a final linear layer to achieve the required output dimension (in our case 1 as we are predicting a single property). Two inputs to this set transformer were tested:

- A 4 dimensional set where 3 of the dimensions where (fractional) K-point coordinates,

Table 7.1: The full set transformer architecture [101]. Here for all experiments h was set 128, m is the number of points in the set and n is the dimensionality of the set, which was either 4 or 11 depending on the specific run (labelled as "4D set transformer" and "11D set transformer" respectively)

Block Name	Output Size
Input	$n \times m$
ISAB	$n \times h$
ISAB	$n \times h$
PMA	h
SAB	h
SAB	h
Fully Connected Layer	1

and the final dimension was an energy level at which an electron could exist.

- An 11-dimensional set where 3 of the dimensions where (fractional) K-point coordinates, and the remaining 8 dimensions represented the 8 energy levels closest to the Fermi surfice at that location.

These two set transformers will be labelled "4D set transformer"and "11D set transformer" respectively and, other than the input dimension, no hyperparameters were changed between these networks.

Although these models could train on band structure data without interpolation of K-points, in order to work with memory and time constraints, an interpolation of 100 K-points per band was used. By the definition outlined above, 11D set transformer considers only the 8 bands closest to the fermi surface. Although a 4D set transformer could theoretically work on an unlimited number of bands, a maximum of 50 bands per EBS was considered to meet memory constraints.

Table 7.2: Results of the different models on predicting band gap (E_{gap}), resistivity (ρ), and $\log_{10}(\rho)$ from EBS data

Model	Model information		Mean Squared Error		
	Type	Value	E_{gap} (eV)	ρ (Ωm)	$\log_{10}(\rho)$
Random Forest	CBFV	*fractional*	1.17	6.97×10^{24}	4.96
		jarvis	0.957	1.41×10^{25}	3.94
		magpie	1.07	2.20×10^{25}	4.12
		Oliynyk	1.00	1.44×10^{26}	3.79
	Random projection size	25	1.26	7.95×10^{26}	6.84
		50	1.21	6.81×10^{24}	6.43
		100	1.16	6.98×10^{26}	6.01
		200	1.16	1.45×10^{26}	5.86
4D set transformer	Number of paramaters	423,000	0.0223	2.35×10^{22}	0.141
11D set transformer		426,000	0.0225	2.35×10^{22}	0.158
RNN 1D		20, 000	0.005 41	3.61×10^{21} †	0.0428
RNN 2D		63,600	0.005 43	6.39×10^{21}	0.0526

†Using AdamW optimiser rather than simple gradient descent (see Section 7.5.1)

Table 7.3: Comparison of statistics regarding the values in the resistivity dataset (ρ) and the logarithm of those values (ρ). By comparing the mean squared error (MSE) of each value in the dataset to the mean (μ) of the dataset it is possible to gauge the MSE that would be produced if a model trained on this dataset only predicted the mean.

Metric	ρ	$\log_{10}\rho$
μ	8.24×10^{11}	-3.75
Standard Deviation	3.67×10^{13}	3.76
Median	4.15×10^{-6}	-5.37
Minimum	1.00×10^{-12}	-12.0
Maximum	1.73×10^{15}	15.2
MSE comparing all entries to μ	1.34×10^{27}	14.1

7.6 Results

Table 7.4: Result of the Bayesian optimisation hyperparameter sweep for RNN models

Paramter	RNN 1D	RNN 2D
Dropout	None	None
Layer Normalisation	None	None
Fully Connected Layer Size	64	128
No. Fully Connected Layers	4	1
LSTM hidden size	32	32
No. LSTM layers	1	4
Attention Mechanism	None	None
Optimiser	Simple gradient descent	Simple gradient descent
Learning Rate	0.0013	0.026
Batch Size	224	256
No. points per band	50	100

All models managed to predict both band gap and resistivity with some degree of accuracy. Resistivity predictions were universally very poor, though they were better than a model which just guesses the mean of the dataset all the time (Table 7.3). Owing to this poor performance, all models were retrained to predict the (base 10) logarithm of resistivity ($\log_{10}\rho$). $\log_{10}\rho$ was found to be more suited when considering the distribution of target values in the dataset (Figure 7.5b), as such it resulted in more accurate (and likely more useful) predictions.

Deep learning models based on EBS data universally outperformed RFs based on compositional data (Table 7.2). 4D and 11D set transformer models performed very similarly. 11D set transformer performed slightly better on band gap and slightly worse on $\log_{10}\rho$ prediction.

RNNs universally outperformed set transformers by an order of magnitude in every task, with RNN 1D marginally outperforming RNN 2D. The hyperparameter optimisation for the RNN based models did not yield conclusive results (Figure 7.9). There were no clear trends to indicate whether the wider or deeper models performed better. Simple gradient descent was unrelliable but did manage to yield the best results (Table 7.4), while Adam and AdamW were less likely to fail to converge and less likely to cause floating point overflow issues (as will be discussed in Section 7.7 this was a problem for resistivity prediction).

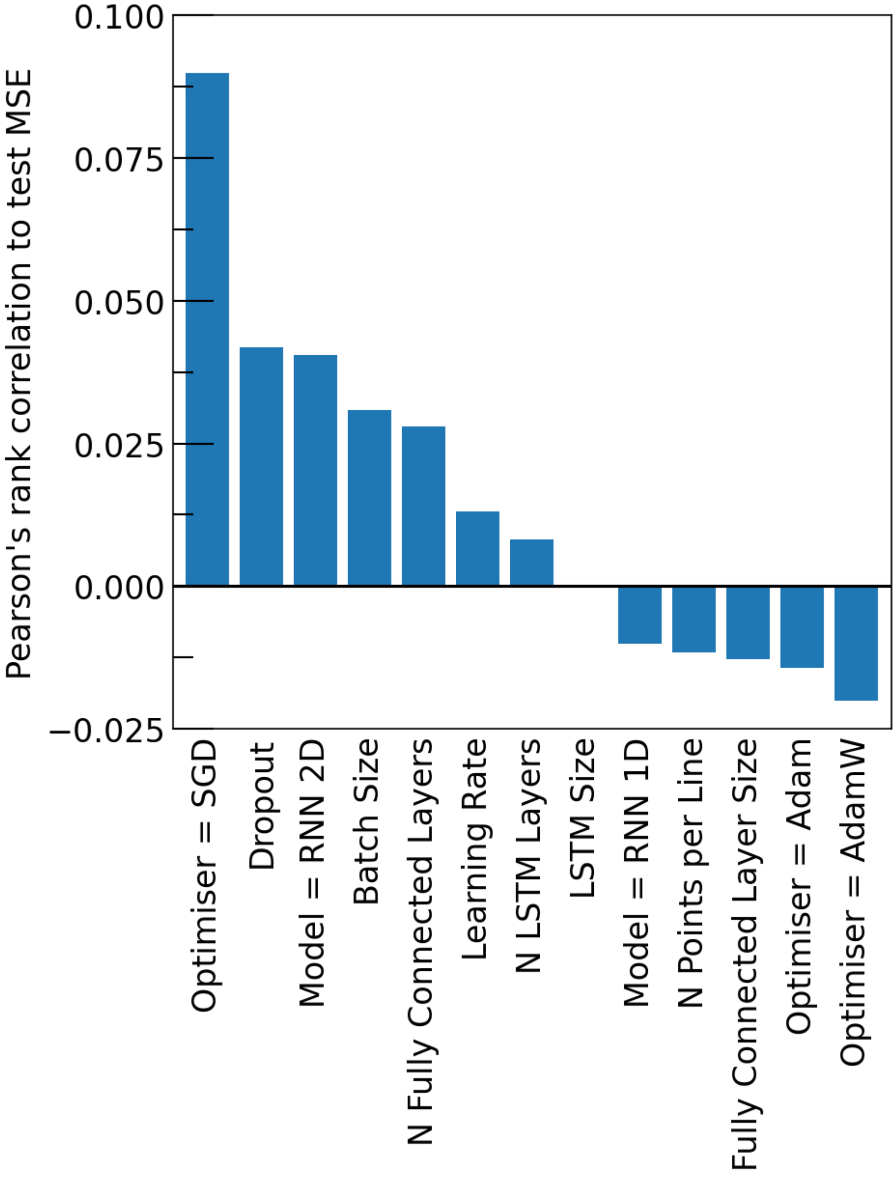


Figure 7.9: Pearson's rank correlation to mean squared error (MSE) on test set in the RNN hyperparamter sweep. There is little correlation between MSE and the hyperparameters chosen. While using simple gradient descent (SGD) to optimise weights did on average result in a worse performance, the best performing models still used SGD (Table 7.4).

Investigations presented here were done using the dataset laid out previously (Section 7.4) in order to collate band structures with resistivity and band gap without data leakage between band gap and resisistivity datasets (Figure 7.4). A downside of this is that there are no easy comparisons to be made between the performance of models presented here, and models seen in the literature.

With the caveat of different using datasets noted, models from the litterature can compared to those seen here. The most apt comparison for the task of band gap prediction is the mp_gap matbench dataset [38], which extracts band gap from the Materials Project [78], similar to the dataset start presented here, and uses crystal structure as an input. Techniques presented here outperform the top three models on this benchmark by an order of magnitude (Table 7.5). More notable still is that the mp_gap dataset features 106,113 entries, compared to the 2227 structures in the dataset used in this study. With more structures one would expect the models presented here to continue to outperform the state of the art on the mp_gap dataset.

Unfortunately for prediction of ρ and $\log_{10} \rho$ there are no clear comparisons to be made. While other datasets featuring ρ exist [126, 152, 67], most of these (and all of these which have associeted ML models) are specifically for thermoelectric materials [65, 6] and thus do not present an apt opportunity for comparison to this study. Similarly other studies which learn from band structure [91, 190], do not perform any supervised ML (only self-supervised or unsupervised), and thus comparison of the efficacy of prior techniques to those presented here are limited

Table 7.5: A comparison of models presented here to methods and results from the literature for predicting band gap on DFT data. Noted are the datasets being used for training/test, whether the model uses any sort of structure (including crystal structure or band structure), whether the model uses electronic band structure (EBS) in any way, and reported root mean squared error (RMSE).

Dataset	Model	Uses structure	Uses EBS	RMSE (eV)
Resistivity dataset (Section 7.4)	4D set transformer (Section 7.5.2)	Yes	Yes	0.149
	11D set transformer (Section 7.5.2)			0.150
	RNN1D (Section 7.5.1)			0.0736
	RNN2D (Section 7.5.1)			0.0737
	Random forest	No	No	0.978
Matbench mp_gap [38]	coGN [163]	Yes	No	0.340
	coNGN [163]			0.427
	ALIGNN [26]			0.464

7.7 Discussion

While EBS data has interesting qualities that make it difficult to use machine learning with without pre-processing methods to make EBS data more uniform across a dataset (Section 7.3). Despite this there are several deep learning architectures (such as those explored in this chapter) which manage the complexities of these data with minimal or no preprocessing.

RNNs were expected to outperform set transformers on band gap prediction tasks as the structure of RNNs lends itself to comparison of adjacent bands. It was expected that the set transformers would better incorperate interactions between bands which are further away from eachother, resulting in better performance in resistivity prediction. However, this was not the case, instead model performance had an inverse correlation with the number of paramaters that a model had (Table 7.2). This would imply that there was insuficient data to train the larger models.

Overall, deep learning methods show potential at interpretting EBS data, far outperforming RFs on these data. This may be seen as unsurprising; EBS data contains structural information, whereas RFs trained here used only composition based information. RFs have no mechinism for dealing with variable sized input data, but RFs could be trained on features extracted from band structures [76].

Perhaps where deep learning methods show the most potential over methods such as RFs is in use of transfer learning. Large repositories of EBS data exist, which could be used as a pre-training task before training a model on a smaller dataset relating band structures to experimentally measured properties. This could be done by predicting a known property (such as band gap) or by reconstructing (such as with an autoencoder; see Section 6.4.2). The trained models could then be used on the smaller dataset to fine-tune the weights for the experimentally measured property. By using transfer learning in this way future work could overcome the limitations of the smaller datasets present in materials science.

The resistivity dataset explored here is an interesting example of such dataset. Direct prediction of resistivity was found to be innapropriate for these data. Models trained to predict resistivity performed better than just guessing the mean of the dataset (Table 7.3), but the results were not good enough to be meaningfully useful. Predicting $\log_{10} \rho$ was more appropriate to the dataset (Figure 7.5b), and resulted in both better results and more numerically stable models. A specific (and interesting) instability noted when training deep learning models on these data is that floating points would often overflow on the resistivity data. As models were trained using 32 bit floating points [75], the exponent was represented using just 8 bits, when considering the bias (allowing for negative exponents), this leaves a maximum exponent value of 128. As batch sizes found to be optimal were above 200 (Table 7.4), considering the size of the mean squared errors (Table 7.3 and Table 7.2), such overflows are not unsurprising.

This could have been alleviated by using 64 bit floating points, which have an 11 bit exponent and thus have a maximum exponent value of 1024 ([75]). However, being orders of magnitude is the source of the majority of the mean squared error between predictions and true values. This means that the majority of the loss is being represented by only a small minority of the data representation (either 8 of 32 bits or 11 of 64 bits depending on floating point format). While the backpropogation process may move this loss from the exponent of the mean squared error to the mantissa of the weights (depending on activation functions used), the loss is still coarse in information (when compared to the if most of the information was carried in the mantissa). This was addressed by taking the logarithm of the resistivity as the target value and resulted in better learning.

Considerations such as these must be made when using deep learning; however, deep learning arcitectures' ability to process EBS data is still exciting. By using paradigms analogous to other popular fields of ML (Section 7.3), such as natural language processing (NLP) or image recognition, the adaptation of existing algorithms becomes possible. These algorithms are known to be effective in those fields, reducing exploratory investigations needed to build deep learning algorithms for these data. While the experiments shown here demonstrate algorithms developed for machine translation (RNNs) and point cloud classification (set transformers), other potentially interesting classes of algorithms exist. Future work may, for example, choose to look at convolutional networks, taking inspiration from convolutional architectures used for image recognition [99] (there is evidence such an approach would be applicable on these data [190]), or point cloud networks that explicitly use distance for efficiency [145]. However, the assumptions underlying the algorithms in other fields may not be applicable to EBS data.

Not only this but different fields of ML research are at different levels of maturity, for example image recognition and NLP have received far more attention than point cloud recognition, and as such are likely to have more refined algorithms that are ready to adopt. That is to say, just because an algorithm developed in a particular field of ML research is seen to have the most effect, it does not mean that a paradigm which draws analogy to that field is "right" paradigm with which to consider ML on EBS data. There is no definitive "right" paradigm, and EBS data can and should be considered an entity unto itself. EBS data are compelling and thought provoking enough to merit their own research, and the author anticipates this research eagerly.

7.8 Conclusion

This chapter has presented a novel approach to using deep learning to interpret EBS data. A dataset mapping EBS data to experimental resistivity measurements was constructed, and a second dataset mapping EBS data to the band gap was built. Eight RFs and four different deep learning architectures were built to demonstrate different ways these data could be interpreted using neural architectures. Two transformer-based models performed well, but seemed to be limited by data. Two RNN based models performed the best, managing to predict both resistivity and band gap with orders of magnitude more accuracy than both set transformer models and the baseline RF models.

EBS datasets are non-trivial, each EBS may have different numbers of bands, numbers (or density) of K-points, and different K-paths through the Brillouin zone. The approaches in this chapter are the first to use deep learning directly on entire datasets of band structures. Although this makes these approaches notable, better approaches no doubt exist. Much space for future work has been highlighted. It is hoped that future research will continue to explore EBS data as an avenue for creating accurate predictions of electronic properties.

7.9 Thesis context

This final experimental chapter shows an example of property prediction using structural (EBS) data. Linking back to prevous chapters which examined non-structural based prediction (Chapters 3 to 5), this chapter shows how structural information can improve the predictive ability of algorithms, and allow for better utilisation of deep learning approaches.

As noted (Section 6.7), this chapter also marks a thematic reversal on the previous chapter. Where Chapter 6 examined data science from a materials context, this chapter examines materials from a data science context by discussing the data structures which could be used to represent band structures.

Such thematic analysis, will be further explored in the subsequent chapter. The final chapter will give an overview of the thesis, discuss potential for future work, and make concluding remarks.

Chapter 8

Discussion and conclusion

The conclusion offered in this chapter aims to provide a comprehensive overview of the thesis. This will begin with a succinct summary of key findings and results in each chapters as well sallient points which were observed. For each chapter, a critical reflection will be offered, assessing the extent to which the research question outlined in the introductory chapter has been effectively addressed. There will be discussion on alternative approaches or methodological adjustments that could be considered should the research be repeated.

Following this overview, a discussion will be given. The thematic progression of the thesis will be discussed and how perspectives changed over the production of the research presented. The wider literature will be discussed, with reflections as to how well this research is situated within this research landscape.

This discussion will lead into the future work section, where potential directions for future investigation will be outlined, enabling subsequent researchers to build on the ideas and experiments established in each individual chapter.

Finally, concluding remarks will be provided. These reflections will discuss the approach taken in the thesis and honest thoughts as to the extent to which the project has been a success.

8.1 Overview

The introductory chapter of this thesis sets out the research question “How can ML be used effectively to enhance materials discovery?” Deffining the specific focus to be “the methods and rationale behind ML” both in terms of formulating models and tasks to make precise predictions, but also in creating frameworks by which those models can be used to discover materials. The research question concludes by stating "this thesis aims to explore how ML can be used more effectively to drive material discovery. Through a critical evaluation of existing models and the development of novel approaches, this thesis seeks to contribute to the broader goal of advancing materials science through the application of ML."

Chapter 2 delves into essential background knowledge required to understand the thesis. The following chapter (Chapter 3) serves as an introductory exploration, showcasing two examples of using random forests to predict material properties, specifically focussing on MOF porosity and unit cell characteristics. However, it is noteworthy that the exploration of the "unit cell" concept remains limited both in scope and in depth; if this were repeated, further experiments would be taken to fully explore this concept, perhaps seeing how Niggli reduction or various supercells affect results. Notable contributions of the chpater include observation of the importance of task selection for ML, as well as the introduction of the r^2_{comp} metric for substitution studies. Although this chapter contributes to the peer-reviewed literature on ML-based materials property prediction, it falls short in terms of the evaluation of existing models or the actual creation of materials. Nonetheless, it makes progress in developing novel approaches, aligning with the research question's objectives.

Chapter 4 focusses on material discovery workflows, encompassing a review of the literature on superconducting critical temperature and the development of collaborative workflows. Existing methods from the literature were examined for their assumptions, and over a billion compositions were screened as candidate superconductors. Predictions were filtered into specific areas of interest as well as materials thought to be easy to synthesise, enabling the large number of candidates screened to be presented to collaborators such that predictions were feasible to manually interrogate. Although none of the materials synthesised thus far were found to be superconducting, investgations are ongoing and the chapter aligns well with the research question. To enhance future investigations, it is recommended to shift the focus from predicting superconducting critical temperatures to predicting whether a material exhibits superconductivity and, if so, in which temperature range. Additionally, the inclusion of synthesis pressure as a parameter in the models is suggested, as pressure can impact superconductivity and aligns with current research trends.

The latter half of the chapter details the creation of the Liverpool materials discovery server (LMDS), discussing structural decisions and highlighting specific applications developped for the platform. This project aims to facilitate the sharing of computational apps among researchers, not only within the LMDS but also through similar platforms which this research hopes to inspire. Overall, this chapter aligns closely with the research question and is expected to yield further results in future endeavors.

Chapter 5 explores kernelised Leave one cluster cut Cross-validation (LOCO-CV) and use of random projections to represent materials. It is notable that the observation regarding the limited impact of feature-engineered composition based feature vectors (CBFVs) on various tasks reported in the literature [124] after observation for as part of this thesis (though before the research presented here was written). This both the timeliness of this work, but also the encouraging progression and self-evaluation of present in literature.

Novely in this chapter is seen in the use of random projections and the observation that the existing review on this topic overlooked the Johnson-Lindenstrauss lemma. One salient point in this chapter is that the "correct" representation of materials remains subjective. Specifically, it was noted that "whether domain knowledge features are being used as a proxy for the composition, or whether the composition is a proxy for the properties of a material which are quantified by the domain knowledge features." This simultaniously undermines and reinforces the importance of the research presented. Initially, investigations were conceived as noting that domain knowledge based CBFVs seem of little use. However, this observation emphasises that while CBFVs may not improve the performance of an ML algorithm, does not mean they "lack use".

The discussion on kernelised LOCO-CV acknowledges the limitations of LOCO-CV and proposes kernel approximation methods as a potential remedy. In retospect application kernels within the K-means algorithm would have improved clarity over using kernel approximation methods to process data *a priori* of the application of K-means clustering. It is uncertain whether this modification have changed the outcome of the investigation. Overall, the chapter successfully provides a comprehensive review of the literature, aligning well with the research question and yielding satisfactory results.

Chapter 6, discusses the methods for measuring the isotropy of clusters. While this is intriguing both within the themes of the thesis and in its own right, measurements for isotropy lack substantial impact in relation to the research question. The measurement of cluster shapes in high dimensions does not neatly align with materials science workflows, and the use of isotropy measures for materials characterisation proved unsuccessful (Section 6.4.4). Although the tools and observations presented are interesting and the chapter delves into intriguing mathematical concepts such as random matrix theory, the concept of measuring the shapes of cluster appears somewhat tangential. Although minimal changes would be suggested if this topic were revisited, the notion of measuring cluster shapes remains somewhat frivolous within the context of the research question.

Finally, Chapter 7 discusses electronic band structure (EBS) and offers an exciting approach that views EBS through data structure paradigms and suggests algorithms based on those data structures. The effectiveness of the Set Transformer and recurrant neural network (RNN) algorithms is notable, but the more intriguing aspect lies in the consideration of electronic band structures in terms of their underlying data representations. If revisited, it would be preferable to use band-gap prediction as a pre-training task for resistivity prediction, instead of solely using the band gap as a sanity check for whether deep learning could be conducted on such small data sets. This novel approach is the first deep learning approach on EBS datasets and aligns with the development of innovative methods for material discovery outlined in the research scope.

8.2 Discussion

The above section takes the reader from introductory experiments (Chapter 3) which introduce new tools for evaluating models to how such experiments may be adapted to work within the academic context (Chapter 4). The context and assumptions underlying the work are then questioned, by asking why feature engineered CBFVs are used (Chapter 5). The same chapter which questions CBFVs also proposed kernelised LOCO-CV, a new tool for evaluating models. Then discussion of measurements for isotropy allows for evaluation of the methods being used to evaluate models before the thesis is regrounded in the task of materials property prediction in a new (electronic structure based) context.

Through reading (and writing) of the above several themes emerge:

- ML models can be used for materials property prediction, and this can aid materials discovery. While there is some deviation, the thesis starts and ends, very intentionally, with material property prediction.

- The intersection of data science and materials science presents unique opportunities and requirements specifically regarding evaluation. From r^2_{comp} to kernelised LOCO-CV, this thesis emphasises the importance of using appropriate measures for success, which are grounded in the way models will be used.

- Questioning of assumptions in and iterative improvement. This being a core tennet of the scientific process, it is unsurprising, but fitting, that this is seen here.

- The interdisciplinary perspective yields interesting insights for both computer science and materials science.

Themes of this nature are subjective, and the above list is not meant to be deffinitive, nor is it declarative of prior intention to consider these themes.

Reflection on the patterns which emerge from years of research also offers the opportunity to reflect on the perspective taken throughout the research. Initially an eagerness was taken to conduct experiments and explore the power of ML on materials data. This eagerness is reflected in the emphasis on tangeable, physicical, results set out in the research question. However, the more projects that were undertaken, the more assumptions that underpin the results were questioned and the more motivation behind the use of ML was questioned. Why should a talented experimental chemist take time out of their day to persue research interests of an ML algorithm and its engineer? Of course, the reasons for undertaking research are complex, and often the research interests presented here align well with those of experimental chemists. Goals of learning and discovery are common among academics, regardless of discipline.

Overall, the thesis, as presented here, is reflective of trends seen in the literature. Data are used to predict properties using off-the-shelf algorithms, and simple featurisation [170] (Chapter 3), work is done to make those tools more accessible and useful in experimental workflows [72, 78] (Chapter 4). The assumptions are questioned [124], and more comprehensive evaluation standards are suggested [187, 39] (Chapter 5). Meanwhile, the materials science context inspires new algorithms [138] (Chapter 6) and bespoke deep learning methods start being used to enhace materials properties prediction [60, 186] (Chapter 7).

8.3 Future work

While future work may look to automate other parts of the materials discovery workflow, as this thesis does not consider synthesis, it seems inappropriate to make suggestions of that nature here. Instead, the suggestions made in this section will relate directly to work done in this thesis.

Future work should follow the trends in the literature. That is to say, future work involving materials property prediction would be better focused either on development of bespoke algorithms suited to the data at hand or on creation of materials and ease of access for tools (or both). It should be noted that better workflows may not look like those developed here. Recent trends in the literature towards generative models [4] and Bayesian optimisation [188] would suggest that there are alternatives to the screeening process suggested here (Chapter 4).

Kernelised LOCO-CV could further be explored through the use of kernel methods within the K-means clustering algorithm rather than as an *a priori* approximation. Methods of implementing similar repeated word removed the extrapolatory power of ML models in ways which are applicable to more computationally expensive models may also be useful, particularly as trends continue towards larger deep learning models.

Band structure data remains a large underutilised source of data for ML models. There are many suggestions for paradigms through which to explore such data in this thesis (Chapter 7). Specifically, transfer learning would be an interesting experiment to do. The conclusions in this thesis noted that larger models tended to perform worse than expected and noted that only a small subset of available data was used for the experiments presented here. Once models are trained on all available EBS data (while avoiding data leakage), it will be easier to observe whether larger models are appropriate for these data. Further work on linking specific experimentally measured electronic properties to available band structure data would also help. Although this thesis presented resistivity as an example, superconductivity or other exceptional electronic properties would also serve as prime candidates for such models.

8.4 Concluding remarks

This work has covered a fast-evolving field (almost too fast). However, the speed at which the literature in this space moves makes the intersection between materials science and data science an exciting and deeply fascinating field to be a part of. It is encouraging to note that flaws in the literature observed when starting this thesis have largely been addressed, and the standard of current work in the field is (subjectively) very high.

The work presented in this thesis aims to meet that standard. Overall, while some of the research present here could be improved, this thesis has aimed to be honest about that and seek out underexplored niches, rather than being focussing on topping benchmarks. While judgement of this thesis is left to the reader, it is with both pride and excitement for the future of the field that this work is concluded.

Bibliography

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] Andrew L Alexander, Jee Eun Lee, Mariana Lazar, and Aaron S Field. Diffusion tensor imaging of the brain. *Neurotherapeutics*, 4:316–329, 2007.

[3] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010.

[4] Michael Alverson, Sterling Baird, Ryan Murdock, (Enoch) Sin-Hang Ho, Jeremy Johnson, and Taylor Sparks. Generative adversarial networks and diffusion models in material discovery. *ChemRxiv*, 2023.

[5] Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, and Peter M. Atkinson. Explainable artificial intelligence: An analytical review. *WIREs Data Mining and Knowledge Discovery*, 11:5, 2021.

[6] Luis M. Antunes, Vikram, Jose J. Plata, Anthony V. Powell, Keith T. Butler, and Ricardo Grau-Crespo. *Machine Learning Approaches for Accelerating the Discovery of Thermoelectric Materials*, chapter 1, pages 1–32.

[7] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[8] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] Peter J Basser and Carlo Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *Journal of Magnetic Resonance B*, 111:209–219, 1996.

[11] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[12] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828, 2013.

[14] Ninad Bhat, Amanda S. Barnard, and Nick Birbilis. Unsupervised machine learning discovers classes in aluminium alloys. *Royal Society Open Science*, 10(2):220360, 2023.

[15] Evgeny Blokhin and Pierre Villars. *The PAULING FILE Project and Materials Platform for Data Science: From Big Data Toward Materials Genome*, pages 1–26. The Pauling File Project, 10 2018.

[16] Krzysztof Bogdan, Tomasz Grzywny, and Michał Ryznar. Barriers, exit time and survival probability for unimodal lévy processes. *Probability Theory and Related Fields*, 162(1):155–198, 06 2015.

[17] Peter G Boyd, Arunraj Chidambaram, Enrique García-Díez, Christopher P Ireland, Thomas D Daff, Richard Bounds, Andrzej Gładysiak, Pascal Schouwink, Seyed Mohamad Moosavi, M Mercedes Maroto-Valer, et al. Data-driven design of metal–organic frameworks for wet flue gas co2 capture. *Nature*, 576(7786):253–256, 2019.

[18] Peter G Boyd and Tom K Woo. A generalized method for constructing hypothetical nanoporous materials of any net topology from graph theory. *CrystEngComm*, 18(21):3777–3792, 2016.

[19] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[20] Mikhail G. Brik, Andrzej Suchocki, and Agata Kamińska. Lattice parameters and stability of the spinel compounds in relation to the ionic radii and electronegativities of constituting chemical elements. *Inorganic Chemistry*, 53(10):5088–5099, may 2014.

[21] T. Caliñski and J. Harabasz. A Dendrite Method For Cluster Analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[22] Anand Chandrasekaran, Deepak Kamal, Rohit Batra, Chiho Kim, Lihua Chen, and Rampi Ramprasad. Solving the electronic structure problem with machine learning. *npj Computational Materials*, 5(1):1–7, dec 2019.

[23] Chi Chen, Yunxing Zuo, Weike Ye, Xiangguo Li, Zhi Deng, and Shyue Ping Ong. A critical review of machine learning of energy materials. *Advanced Energy Materials*, 10(8):1903242, 2020.

[24] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[25] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[26] Kamal Choudhary and Brian DeCost. Atomistic Line Graph Neural Network for improved materials property predictions. *npj Computational Materials*, 7(1):1–8, dec 2021.

[27] Kamal Choudhary, Brian DeCost, and Francesca Tavazza. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Phys. Rev. Materials*, 2:083801, Aug 2018.

[28] Iek Heng Chu, Sayan Roychowdhury, Daehui Han, Anubhav Jain, and Shyue Ping Ong. Predicting the volumes of crystals. *Computational Materials Science*, 146:184–192, apr 2018.

[29] Yongchul G Chung, Emmanuel Haldoupis, Benjamin J Bucior, Maciej Haranczyk, Seulchan Lee, Hongda Zhang, Konstantinos D Vogiatzis, Marija Milisavljevic, Sanliang Ling, Jeffrey S Camp, et al. Advances, updates, and analytics for the computation-ready, experimental metal–organic framework database: Core mof 2019. *Journal of Chemical & Engineering Data*, 64(12):5985–5998, 2019.

[30] Conrad L. Clement, Steven K. Kauwe, and Taylor D. Sparks. Benchmark AFLOW Data Sets for Machine Learning. *Integrating Materials and Manufacturing Innovation*, 9(2):153–156, jun 2020.

[31] Christopher M. Collins, Luke M. Daniels, Quinn Gibson, Michael W. Gaultois, Michael Moran, Richard Feetham, Michael J. Pitcher, Matthew S. Dyer, Charlene Delacotte, Marco Zanella, Claire A. Murray, Gyorgyi Glodan, Olivier Pérez, Denis Pelloquin, Troy D. Manning, Jonathan Alaria, George R. Darling, John B. Claridge, and Matthew J. Rosseinsky. Discovery of a Low Thermal Conductivity Oxide Guided by Probe Structure Prediction and Machine Learning. *Angewandte Chemie International Edition*, 60(30):16457–16465, jul 2021.

[32] Stefano Curtarolo, Wahyu Setyawan, Gus L.W. Hart, Michal Jahnatek, Roman V. Chepulskii, Richard H. Taylor, Shidong Wang, Junkai Xue, Kesong Yang, Ohad Levy, Michael J. Mehl, Harold T. Stokes, Denis O. Demchenko, and Dane Morgan. Aflow: An automatic framework for high-throughput materials discovery. *Computational Materials Science*, 58:218–226, 2012.

[33] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, January 2003.

[34] Daniel Davies, Keith Butler, Adam Jackson, Jonathan Skelton, Kazuki Morita, and Aron Walsh. SMACT: Semiconducting Materials by Analogy and Chemical Theory. *Journal of Open Source Software*, 4(38):1361, jun 2019.

[35] Daniel W. Davies, Keith T. Butler, and Aron Walsh. Data-Driven Discovery of Photoactive Quaternary Oxides Using First-Principles Machine Learning. *Chemistry of Materials*, 31(18):7221–7230, sep 2019.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[37] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[38] Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, 2020.

[39] Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):1–10, September 2020.

[40] Samantha Durdy. Github code repository. `https://github.com/lrcfmd/KernelisedLOCO-CV`. Accessed: 2022-03-07.

[41] Samantha Durdy. Git repository for use of random forests for predicting of unit cell properties. `https://github.com/lrcfmd/RandomForestsForPredictingUnitCellProperties`, 2020.

[42] Samantha Durdy. Lmds server setup tools. `https://github.com/lrcfmd/LMDS_helper_scripts`, 2022.

[43] Samantha Durdy. Python implementations of isotropy measurements. `https://github.com/lrcfmd/Isotropy_measurements.git`, 2023.

[44] Samantha Durdy, Michael W Gaultois, Vladimir Gusev, Danushka Bollegala, and Matthew J Rosseinsky. Metrics for quantifying isotropy in high dimensional unsupervised clustering tasks in a materials context. *arXiv preprint arXiv:2305.16372*, 2023.

[45] Samantha Durdy, Michael W. Gaultois, Vladimir V. Gusev, Danushka Bollegala, and Matthew J. Rosseinsky. Random projections and kernelised leave one cluster out cross validation: universal baselines and evaluation tools for supervised machine learning of material properties. *Digital Discovery*, 1:763–778, 2022.

[46] Samantha Durdy, Cameron J. Hargreaves, Mark Dennison, Benjamin Wagg, Michael Moran, Jon A. Newnham, Michael W. Gaultois, Matthew J. Rosseinsky, and Matthew S. Dyer. The liverpool materials discovery server: a suite of computational tools for the collaborative discovery of materials. *Digital Discovery*, 2:1601–1611, 2023.

[47] Samantha Durdy, Cameron J. Hargreaves, and Jon A. Newnham. Lmds heat capacity modeling tool. `https://github.com/lrcfmd/LMDS_heat_capacity_modelling`, 2022.

[48] Benjamin M Ellingson and Julien Cohen-Adad. *Chapter 3.1 - Diffusion-Weighted Imaging of the Spinal Cord*. Academic Press, 2014.

[49] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96. AAAI Press, 1996.

[50] George S Fanourgakis, Konstantinos Gkagkas, Emmanuel Tylianakis, and George E Froudakis. A universal machine learning algorithm for large-scale screening of materials. *Journal of the American Chemical Society*, 142(8):3814–3822, 2020.

[51] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Rfc 2616, hypertext transfer protocol – http/1.1, 1999.

[52] Roy T. Fielding, Mark Nottingham, and Julian Reschke. HTTP Semantics. RFC 9110, June 2022.

[53] José A. Flores-Livas, Lilia Boeri, Antonio Sanna, Gianni Profeta, Ryotaro Arita, and Mikhail Eremets. A perspective on conventional high-temperature superconductors at high pressure: Methods and materials. *Physics Reports*, 856:1–78, 2020. A perspective on conventional high-temperature superconductors at high pressure: Methods and materials.

[54] E B Fowlkes and C L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[55] Ralph Freund, Orysia Zaremba, Giel Arnauts, Rob Ameloot, Grigorii Skorupskii, Mircea Dincă, Anastasiya Bavykina, Jorge Gascon, Aleksander Ejsmont, Joanna Goscianska, Markus Kalmutzki, Ulrich Lächelt, Evelyn Ploetz, Christian S. Diercks, and Stefan Wuttke. The current status of mof and cof applications. *Angewandte Chemie International Edition*, 60(45):23975–24001, 2021.

[56] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[57] Paulino José García-Nieto, Esperanza García-Gonzalo, and José Pablo Paredes-Sánchez. Prediction of the critical temperature of a superconductor by using the woa/mars, ridge, lasso and elastic-net machine learning techniques. *Neural Computing and Applications*, 33(24):17131–17145, 2021.

[58] Benjamin A Goldstein, Eric C Polley, and Farren B. S. Briggs. Random forests for genetic association studies. *Statistical Applications in Genetics and Molecular Biology*, 10(1), 2011.

[59] Diego A Gómez-Gualdrón, Yamil J Colón, Xu Zhang, Timothy C Wang, Yu-Sheng Chen, Joseph T Hupp, Taner Yildirim, Omar K Farha, Jian Zhang, and Randall Q Snurr. Evaluating topologically diverse metal–organic frameworks for cryo-adsorbed hydrogen storage. *Energy & Environmental Science*, 9(10):3279–3289, 2016.

[60] Rhys E.A. Goodall and Alpha A Lee. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nature Communications*, 11(1), 2020.

[61] R. W Grosse-Kunstleve, N. K Sauter, and P. D Adams. Numerically stable algorithms for the computation of reduced unit cells. *Acta Crystallographica Section A: Foundations of Crystallography*, 60(Pt 1):1–6, Jan 2004. PMID: 14691322.

[62] S. R. Hall, F. H. Allen, and I. D. Brown. The crystallographic information file (CIF): a new standard archive file for crystallography. *Acta Crystallographica Section A*, 47(6):655–685, Nov 1991.

[63] Greg Hamerly and Charles Elkan. Learning the k in k-means. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information*

Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada], pages 281–288. MIT Press, 2003.

[64] Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, 2018.

[65] Guangshuai Han, Yixuan Sun, Yining Feng, Guang Lin, and Na Lu. Machine learning regression guided thermoelectric materials discovery - a review. *ES Materials & Manufacturing*, 14:20–35, 2021.

[66] Cameron J. Hargreaves, Matthew S. Dyer, Michael W. Gaultois, Vitaliy A. Kurlin, and Matthew J. Rosseinsky. The Earth Mover's Distance as a Metric for the Space of Inorganic Compositions. *Chemistry of Materials*, 32(24):10610–10620, dec 2020.

[67] Cameron J Hargreaves, Michael W Gaultois, Luke M Daniels, Emma J Watts, Vitaliy A Kurlin, Michael Moran, Yun Dang, Rhun Morris, Alexandra Morscher, Kate Thompson, et al. A database of experimentally measured lithium solid electrolyte conductivities evaluated with machine learning. *npj Computational Materials*, 9(1):9, 2023.

[68] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[69] Mariette Hellenbrandt. The inorganic crystal structure database (ICSD) - Present and future. In *Crystallography Reviews*, volume 10, pages 17–22. Taylor & Francis, jan 2004.

[70] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[71] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[72] Jianjun Hu, Stanislav Stefanov, Yuqi Song, Sadman Sadeed Omee, Steph Yves Louis, Edirisuriya M.D. Siriwardane, Yong Zhao, and Lai Wei. MaterialsAtlas.org: a materials informatics web app platform for materials discovery and survey of state-of-the-art. *npj Computational Materials*, 8(1):1–12, dec 2022.

[73] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[74] Michael J. Hutcheon, Alice M. Shipley, and Richard J. Needs. Predicting novel superconducting hydrides using machine learning approaches. *Phys. Rev. B*, 101:144505, Apr 2020.

[75] IEEE Computer Society. Ieee standard for floating-point arithmetic, 1985. IEEE Std 754-1985.

[76] Olexandr Isayev, Denis Fourches, Eugene N. Muratov, Corey Oses, Kevin Rasch, Alexander Tropsha, and Stefano Curtarolo. Materials cartography: Representing and mining materials space using structural and electronic fingerprints. *Chemistry of Materials*, 27(3):735–743, 2015.

[77] Kevin Maik Jablonka, Daniele Ongari, Seyed Mohamad Moosavi, and Berend Smit. Big-data science in porous materials: Materials genomics and machine learning. *Chemical reviews*, 120:8066–8129, 8 2020.

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

[79] Dipendra Jha, Logan Ward, Arindam Paul, Wei keng Liao, Alok Choudhary, Chris Wolverton, and Ankit Agrawal. ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition. *Scientific Reports*, 8(1):1–13, dec 2018.

[80] R. O. Jones. Density functional theory: Its origins, rise to prominence, and future. *Rev. Mod. Phys.*, 87:897–923, Aug 2015.

[81] Project Jupyter. Jupyterlab and jupyter notebook. `https://jupyter.org/`, 2015.

[82] Nal Kalchbrenner. *Encoder-Decoder Neural Networks.* PhD thesis, University of Oxford, 2017.

[83] Samuel Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. *IEEE International Conference on Neural Networks - Conference Proceedings*, 1:413–418, 1998.

[84] Steven K Kauwe, Taylor Welker, and Taylor D Sparks. Extracting Knowledge from DFT: Experimental Band Gap Predictions Through Ensemble Learning. *Integrating Materials and Manufacturing Innovation*, 9(3):213–220, 2020.

[85] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[86] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[87] Nikolaj Rørbæk Knøsgaard and Kristian Sommer Thygesen. Representing individual electronic states for machine learning gw band structures of 2d materials. *Nature Communications*, 13(1):468, 2022.

[88] Tomohiko Konno, Hodaka Kurokawa, Fuyuki Nabeshima, Yuki Sakishita, Ryo Ogawa, Iwao Hosako, and Atsutaka Maeda. Deep learning model for finding new superconductors. *Physical Review B*, 103(1):14509, jan 2021.

[89] Evans Kotei and Ramkumar Thirunavukarasu. A systematic review of transformer-based pre-trained language models through self-supervised learning. *Information*, 14(3), 2023.

[90] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Aiche Journal*, 37:233–243, 1991.

[91] Martin Kuban, Santiago Rigamonti, Markus Scheidgen, and Claudia Draxl. Density-of-states similarity descriptor for unsupervised learning from materials data. *Scientific Data*, 9(1):646, 2022.

[92] Kubernetes Manual. `https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/`, 2017. [Online; accessed 19-Aug-2022].

[93] H J Kulik, T Hammerschmidt, J Schmidt, S Botti, M A L Marques, M Boley, M Scheffler, M Todorović, P Rinke, C Oses, A Smolyanyuk, S Curtarolo, A Tkatchenko, A P Bartók, S Manzhos, M Ihara, T Carrington, J Behler, O Isayev, M Veit, A Grisafi, J Nigam, M Ceriotti, K T Schütt, J Westermayr, M Gastegger, R J Maurer, B Kalita, K Burke, R Nagai, R Akashi, O Sugino, J Hermann, F Noé, S Pilati, C Draxl, M Kuban, S Rigamonti, M Scheidgen, M Esters, D Hicks, C Toher, P V Balachandran, I Tamblyn, S Whitelam, C Bellinger, and L M Ghiringhelli. Roadmap on machine learning in electronic structure. *Electronic Structure*, 4:023004, 6 2022.

[94] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[95] Douglas Kunda, Sipiwe Chihana, and Muwanei Sinyinda. Web server performance of apache and nginx: A systematic literature review. *Computer Engineering and Intelligent Systems*, 8:43–52, 2017.

[96] Davies D L and Bouldin D W. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[97] SciKit Learn. Scikit learn feature selection. `https://scikit-learn.org/stable/modules/feature_selection.html`. Accessed: 2022-03-07.

[98] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[99] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs*, 2, 2010.

[100] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28, 1988.

[101] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings*

of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 09–15 Jun 2019.

[102] Fleur Legrain, Jesús Carrete, Ambroise Van Roekeghem, Georg K.H. Madsen, and Natalio Mingo. Materials Screening for the Discovery of New Half-Heuslers: Machine Learning versus ab Initio Methods. *Journal of Physical Chemistry B*, 122(2):625–632, jan 2018.

[103] Zheng Li, Maria Kihl, Qinghua Lu, and Jens A. Andersson. *Performance overhead comparison between hypervisor and container based virtualization*, pages 955–962. Institute of Electrical and Electronics Engineers Inc., may 2017.

[104] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 3:111–132, 2021.

[105] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2021.

[106] Xuebo Liu, Derek Wong, Yang Liu, Lidia Chao, Tong Xiao, and Jingbo Zhu. Shared-private bilingual word embeddings for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 01 2019.

[107] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 911–916, 2010.

[108] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[109] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[110] Farhad Maleki, Katie Ovens, Rajiv Gupta, Caroline Reinhold, Alan Spatz, and Reza Forghani. Generalizability of machine learning models: Quantitative evaluation of three methodological pitfalls. *Radiology: Artificial Intelligence*, 5(1):e220028, 2023.

[111] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[112] Pedro Henrique Martins, Zita Marinho, and Andre Martins. ∞-former: Infinite memory transformer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5468–5485, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[113] V A Marčenko and L A Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, apr 1967.

[114] Materials Database Group. Mdr supercon datasheet readme, 2022.

[115] Andrea Mauri, Viviana Consonni, Manuela Pavan, and Roberto Todeschini. Dragon software: An easy approach to molecular descriptor calculations. *MATCH Communications in Mathematical and in Computer Chemistry*, 56:237–248, 01 2006.

[116] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

[117] Bryce Meredig, Erin Antono, Carena Church, Maxwell Hutchinson, Julia Ling, Sean Paradiso, Ben Blaiszik, Ian Foster, Brenna Gibbons, Jason Hattrick-Simpers, Apurva Mehta, and Logan Ward. Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. *Molecular Systems Design and Engineering*, 3(5):819–825, oct 2018.

[118] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[119] Peyman Z Moghadam, Aurelia Li, Seth B Wiggin, Andi Tao, Andrew GP Maloney, Peter A Wood, Suzanna C Ward, and David Fairen-Jimenez. Development of a cambridge structural database subset: a collection of metal–organic frameworks for past, present, and future. *Chemistry of Materials*, 29(7):2618–2625, 2017.

[120] Seyed Mohamad Moosavi, Aditya Nandy, Kevin Maik Jablonka, Daniele Ongari, Jon Paul Janet, Peter G Boyd, Yongjin Lee, Berend Smit, and Heather J Kulik. Understanding the diversity of the metal-organic framework ecosystem. *Nature communications*, 11(1):1–10, 2020.

[121] Hirotomo Moriwaki, Yushi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: A molecular descriptor calculator. *Journal of Cheminformatics*, 10, 02 2018.

[122] Benyamin Motevalli, Amanda J Parker, Baichuan Sun, and Amanda S Barnard. The representative structure of graphene oxide nanoflakes from machine learning. *Nano Futures*, 3:045001, 12 2019.

[123] Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

[124] Ryan J Murdock, Steven K Kauwe, Anthony Yu Tung Wang, and Taylor D. Sparks. Is Domain Knowledge Necessary for Machine Learning Materials Properties? *Integrating Materials and Manufacturing Innovation*, 9(3):221–227, 2020.

[125] Philip A. E. Murgatroyd, Kieran Routledge, Samantha Durdy, Michael W. Gaultois, T. Wesley Surta, Matthew S. Dyer, John B. Claridge, Stanislav N. Savvin, Denis Pelloquin, Sylvie Hébert, and Jonathan Alaria. Chemically Controllable Magnetic Transition Temperature and Magneto-Elastic Coupling in MnZnSb Compounds. *Advanced Functional Materials*, page 2100108, feb 2021.

[126] Gyoung S. Na and Hyunju Chang. A public database of thermoelectric materials and system-identified material representation for data-driven discovery. *npj Computational Materials*, 8(1):214, 2022.

[127] S. Nembrini, I. König, and Marvin N. Wright. The revival of the gini importance? *Bioinformatics*, 34:3711 – 3718, 2018.

[128] Netcraft. Web server survey, 2022. Accessed: 2022-12-22.

[129] Noel M. O'Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(1):33, Oct 2011.

[130] National Institute of Materials Science. Supercon.

[131] Anton O. Oliynyk, Erin Antono, Taylor D. Sparks, Leila Ghadbeigi, Michael W. Gaultois, Bryce Meredig, and Arthur Mar. High-Throughput Machine-Learning-Driven Synthesis of Full-Heusler Compounds. *Chemistry of Materials*, 28(20):7324–7331, oct 2016.

[132] Jordan O'Mara, Bryce Meredig, and Kyle Michel. Materials Data Infrastructure: A Case Study of the Citrination Platform to Examine Data Import, Storage, and Access. *JOM*, 68(8):2031–2034, aug 2016.

[133] Tadashi C Ozawa and Susan M Kauzlarich. Chemistry of layered d-metal pnictide oxides and their potential as candidates for new superconductors. *Science and Technology of Advanced Materials*, 9(3):033003, 2008.

[134] K. Cenzual P. Villars. Pearson's crystal data: Crystal structure database for inorganic compounds (on dvd), release 2019/20. In *Pearson's Crystal Data: Crystal Structure Database for Inorganic Compounds (on DVD), Release 2019/20.* ASM International®, 2019.

[135] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, jul 2002. Association for Computational Linguistics.

[136] Maryam Pardakhti, Ehsan Moharreri, David Wanik, Steven L Suib, and Ranjan Srivastava. Machine learning using combined structural and chemical descriptors for prediction of methane adsorption performance of metal organic frameworks (mofs). *ACS combinatorial science*, 19(10):640–645, 2017.

[137] A. J. Parker, G. Opletal, and A. S. Barnard. Classification of platinum nanoparticle catalysts using machine learning. *Journal of Applied Physics*, 128:014301, 7 2020.

[138] Amanda J Parker and Amanda S Barnard. Selecting appropriate clustering methods for materials science applications of machine learning. *Advanced Theory and Simulations*, 2:1900145, 2019.

[139] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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

[141] D. Pelleg and A. Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *IICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, 1(1):727–734, 2000.

[142] Rémi Pétuya, Samantha Durdy, Dmytro Antypov, Michael W. Gaultois, Neil G. Berry, George R. Darling, Alexandros P. Katsoulidis, Matthew S. Dyer, and Matthew J. Rosseinsky. Machine-Learning Prediction of Metal–Organic Framework Guest Accessibility from Linker and Metal Chemistry. *Angewandte Chemie International Edition*, 61(9):e202114573, feb 2022.

[143] Plotly. The interactive graphing library for python (includes plotly express). `https://github.com/plotly/plotly.py`, 2023.

[144] Rémi Pétuya, Samantha Durdy, Dmytro Antypov, Michael W. Gaultois, Neil G. Berry, George R. Darling, Alexandros P. Katsoulidis, Matthew S. Dyer, and Matthew J. Rosseinsky. Machine-learning prediction of metal–organic framework guest accessibility from linker and metal chemistry. *Angewandte Chemie International Edition*, 61:e202114573, 2 2022.

[145] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[146] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems*, 21, 2008.

[147] Md Atikur Rahman, Md Zahidur Rahaman, and Md Nurush Samsuddoha. A review on cuprate based superconducting materials including characteristics and applications. *American Journal of Physics and Applications*, 3(2):39–56, 2015.

[148] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[149] Clemens Rauer and Tristan Bereau. Hydration free energies from kernel-based machine learning: Compound-database bias. *Journal of Chemical Physics*, 153(1):014101, jul 2020.

[150] RDKit Development Team. Rdkit: Open-source cheminformatics. `https://www.rdkit.org`, accessed February 20, 2023.

[151] Will Reese. Nginx: The high-performance web server and reverse proxy. *Linux J.*, 2008(173), sep 2008.

[152] Francesco Ricci, Wei Chen, Umut Aydemir, G. Jeffrey Snyder, Gian-Marco Rignanese, Anubhav Jain, and Geoffroy Hautier. An ab initio electronic transport database for inorganic materials. *Scientific Data*, 4(1):170085, 2017.

[153] Janosh Riebesell. *Probabilistic Data-Driven Discovery of Thermoelectric Materials*. PhD thesis, University of Cambridge, 2019.

[154] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics 1989 61:4*, 61(4):241–254, aug 1989.

[155] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, 71:804–818, 2015.

[156] Jakob Ropers, Marco M. Mosca, Olga Anosova, and Vitaliy Kurlin. Introduction to invariant-based machine learning for periodic crystals. *Acta Crystallographica Section A*, 77(a2):C671, Aug 2021.

[157] Andrew Rosenberg and Julia Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic, jun 2007. Association for Computational Linguistics.

[158] B. Roter and S. V. Dordevic. Predicting new superconductors and their critical temperatures using machine learning. *Physica C: Superconductivity and its Applications*, 575:1353689, aug 2020.

[159] B. Roter, N. Ninkovic, and S.V. Dordevic. Clustering superconductors using unsupervised machine learning. *Physica C: Superconductivity and its Applications*, 598:1354078, 2022.

[160] Marianne Rotter, Marcus Tegel, and Dirk Johrendt. Superconductivity at 38 k in the iron arsenide $(\mathrm{ba}_{1-x}\mathrm{k}_x)\mathrm{fe}_2\mathrm{as}_2$. *Phys. Rev. Lett.*, 101:107006, Sep 2008.

[161] P. M.C. Rourke and S. R. Julian. Numerical extraction of de Haas-van Alphen frequencies from calculated band energies. *Computer Physics Communications*, 183(2):324–332, feb 2012.

[162] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C):53–65, nov 1987.

[163] Robin Ruff, Patrick Reiser, Jan Stühmer, and Pascal Friederich. Connectivity optimized nested graph networks for crystal structures, 2023.

[164] Cameron Hargreaves Samantha Durdy. Lmds mof porosity prediction tool. `https://github.com/lrcfmd/LMDS_MOF_Porosity_Tool`, 2022.

[165] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[166] Jonathan Schmidt, Mário R.G. Marques, Silvana Botti, and Miguel A.L. Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1–36, dec 2019.

[167] Steven H. Simon. *Chapter 18: Semiconductor Devices*, pages 197–203. Oxford University Press, Oxford, 1st edition, 2013.

[168] Natalia Sizochenko and Markus Hofmann. Predictive Modeling of Critical Temperatures in Superconducting Materials. *Molecules 2021, Vol. 26, Page 8*, 26(1):8, dec 2020.

[169] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[170] Valentin Stanev, Corey Oses, A. Gilad Kusne, Efrain Rodriguez, Johnpierre Paglione, Stefano Curtarolo, and Ichiro Takeuchi. Machine learning modeling of superconducting critical temperature. *npj Computational Materials*, 4(1):1–14, dec 2018.

[171] Douglas Steinley and Michael J Brusco. Initializing k-means batch clustering: A critical evaluation of several techniques. *Journal of Classification*, 24:99–121, 2007.

[172] H Suhl, BT Matthias, and LR Walker. Bardeen-cooper-schrieffer theory of superconductivity in the case of overlapping bands. *Physical Review Letters*, 3(12):552, 1959.

[173] W3 Techs Web Technology Surveys. Web server market share. `https://w3techs.com/technologies/history_overview/web_server/ms/y`. Accessed: 2022-12-22.

[174] Leopold Talirz, Snehal Kumbhar, Elsa Passaro, Aliaksandr V. Yakutovich, Valeria Granata, Fernando Gargiulo, Marco Borelli, Martin Uhrin, Sebastiaan P. Huber, Spyros Zoupanos, Carl S. Adorf, Casper Welzel Andersen, Ole Schütt, Carlo A. Pignedoli, Daniele Passerone, Joost VandeVondele, Thomas C. Schulthess, Berend Smit, Giovanni Pizzi, and Nicola Marzari. Materials Cloud, a platform for open computational science. *Scientific Data*, 7(1):1–12, dec 2020.

[175] Adam M. Tollitt, Rebecca Vismara, Luke M. Daniels, Dmytro Antypov, Michael W. Gaultois, Alexandros P. Katsoulidis, and Matthew J. Rosseinsky. High-throughput discovery of a rhombohedral twelve-connected zirconium-based metal-organic framework with ordered terephthalate and fumarate linkers. *Angewandte Chemie*, 133(52):27145–27152, November 2021.

[176] Michael Tschannen, Olivier Frederic Bachem, and Mario Lučić. Recent advances in autoencoder-based representation learning. In *Bayesian Deep Learning Workshop, NeurIPS*, 2018.

[177] Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98, 2019.

[178] Bojan Tunguz, 2020.

[179] Hristos Tyralis, Georgia Papacharalampous, and Andreas Langousis. A brief review of random forests for water scientists and practitioners and their recent history in water resources. *Water*, 11(5), 2019.

[180] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[181] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009, 2017.

[182] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010.

[183] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[184] Izhar Wallach and Abraham Heifets. Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization. *Journal of Chemical Information and Modeling*, 58(5):916–932, may 2018.

[185] Brian Walters. Vmware virtual platform. *Linux Journal*, 1999(63es):6, jul 1999.

[186] Anthony Yu-Tung Wang, Steven K. Kauwe, Ryan J. Murdock, and Taylor D. Sparks. Compositionally restricted attention-based network for materials property predictions. *Npj Computational Materials*, 7(1):77, 2021.

[187] Anthony Yu-Tung Wang, Ryan J. Murdock, Steven K. Kauwe, Anton O. Oliynyk, Aleksander Gurlo, Jakoah Brgoch, Kristin A. Persson, and Taylor D. Sparks. Ma-

chine learning for materials scientists: An introductory guide toward best practices. *Chemistry of Materials*, 32(12):4954–4965, 2020.

[188] Ke Wang and Alexander W Dowling. Bayesian optimization for chemical products and functional materials. *Current Opinion in Chemical Engineering*, 36:100728, 2022.

[189] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. cite arxiv:2006.04768.

[190] Zifeng Wang, Shizhuo Ye, Hao Wang, Jin He, Qijun Huang, and Sheng Chang. Machine learning method for tight-binding hamiltonian parameterization from ab-initio band structure. *npj Computational Materials*, 7(1):11, 2021.

[191] Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[192] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1):1–7, aug 2016.

[193] Logan Ward, Muratahan Aykol, Ben Blaiszik, Ian Foster, Bryce Meredig, James Saal, and Santosh Suram. Strategies for accelerating the adoption of materials informatics. *MRS Bulletin*, 43(9):683–689, sep 2018.

[194] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils E.R. Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, Kyle Chard, Mark Asta, Kristin A. Persson, G. Jeffrey Snyder, Ian Foster, and Anubhav Jain. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, September 2018.

[195] Logan Ward, Stephanie C. O'Keeffe, Joseph Stevick, Glenton R. Jelbert, Muratahan Aykol, and Chris Wolverton. A machine learning approach for engineering bulk metallic glass alloys. *Acta Materialia*, 159:102–111, oct 2018.

[196] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[197] Junzo Watada, Arunava Roy, Ruturaj Kadikar, Hoang Pham, and Bing Xu. Emerging Trends, Techniques and Open Issues of Containerization: A Review, 2019.

[198] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[199] Christopher E Wilmer, Michael Leaf, Chang Yeon Lee, Omar K Farha, Brad G Hauser, Joseph T Hupp, and Randall Q Snurr. Large-scale screening of hypothetical metal–organic frameworks. *Nature chemistry*, 4(2):83–89, 2012.

[200] R. Patrick Xian, Vincent Stimper, Marios Zacharias, Maciej Dendzik, Shuo Dong, Samuel Beaulieu, Bernhard Schölkopf, Martin Wolf, Laurenz Rettig, Christian Carbogno, Stefan Bauer, and Ralph Ernstorfer. A machine learning route between band mapping and band structure. *Nature Computational Science*, 3(1):101–114, 2023.

[201] S. R. Xie, Y. Quan, A. C. Hire, B. Deng, J. M. DeStefano, I. Salinas, U. S. Shah, L. Fanfarillo, J. Lim, J. Kim, G. R. Stewart, J. J. Hamlin, P. J. Hirschfeld, and R. G. Hennig. Machine learning of superconducting critical temperature from eliashberg theory. *npj Computational Materials*, 8(1):14, Jan 2022.

[202] Lijuan Yang, Chao Jin, Guanghui Yang, Zhitong Bing, Liang Huang, Yuzhen Niu, and Lei Yang. Transformer-based deep learning method for optimizing admet properties of lead compounds. *Phys. Chem. Chem. Phys.*, 25:2377–2385, 2023.

[203] Jiahao Yu, Yongman Zhao, Rongshun Pan, Xue Zhou, and Zikai Wei. Prediction of the critical temperature of superconductors based on two-layer feature selection and the optuna-stacking ensemble learning model. *ACS Omega*, 8(3):3078–3090, 2023.

[204] Ying Zhang, Xingfeng He, Zhiqian Chen, Qiang Bai, Adelaide M. Nolan, Charles A. Roberts, Debasish Banerjee, Tomoya Matsunaga, Yifei Mo, and Chen Ling. Unsupervised discovery of solid-state lithium ion conductors. *Nature Communications*, 10(1):5260, Nov 2019.

[205] Yun Zhang, Ling Wang, Xinqiao Wang, Chengyun Zhang, Jiamin Ge, Jing Tang, An Su, and Hongliang Duan. Data augmentation and transfer learning strategies for reaction prediction in low chemical data regimes. *Org. Chem. Front.*, 8:1415–1423, 2021.

[206] Ya Zhuo, Aria Mansouri Tehrani, and Jakoah Brgoch. Predicting the band gaps of inorganic solids by machine learning. *The Journal of Physical Chemistry Letters*, 9(7):1668–1673, 2018. PMID: 29532658.