Discussion, Implementation, and Demonstration of AI-guided active workflows

LaTeX using TU Darmstadt's Corporate Design

Master thesis by Bruce Lim

Date of submission: November 22, 2021

1. Review: Prof. Karsten Albe
2. Review: Prof. Matthias Scheffler
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fritz-Haber Institute

Materials and Earth
Sciences Department
Institut
Computational Materials

Declaration on the thesis according to §22 Abs. 7 and §23 Abs. 7 APB of the TU Darmstadt

I, Bruce Lim, hereby certify that I have prepared this Master Thesis without the help of third parties and only with the sources and aids indicated. All passages taken from sources have been marked as such. This work has not yet been submitted to any examination authority in the same or a similar form.

I am aware that in the case of plagiarism (§38 Abs. 2 APB), an attempt at deception has been made, which will result in the thesis being graded with 5.0 and thus one examination attempt being wasted. Final papers may be repeated only once.

In the case of the submitted thesis, the written version and the electronic version submitted for archiving coincide in accordance with §23 Abs. 7 APB.

In case of a thesis of the Department of Architecture, the submitted electronic version corresponds to the presented model and plans.

Darmstadt, November 22, 2021

B. Lim

Abbreviations

AI Artificial Intelligence

CS Compressed Sensing

DA Domain of Applicability

DFT Density Functional Theory

FHI-aims Fritz Haber Institute ’ab initio molecular simulations’

GPR Gaussian Process Regression

HTC High Throughput Computation

MD Molecular Dynamics

ML Machine Learning

NAO Numerically tabulated Atom-centered Orbitals

PBEsol Perdew-Burke-Ernzerhof functional revised for solids

SC Superconductor

sc self-consistency

SCF Self-Consistency Field

SGD Subgroup Discovery

SISSO Sure Independence and Screening Sparsifying Operator

Nomenclature

ab initio from first principles

$f(x)$ Function of x

$\hat{H}^{(m)}$ Operator set

L_0 The L0 norm is the number of non-zero elements in a vector

P Set of samples measured incoherently

ϕ Objects in the set Φ

Φ Starting point for construction of the set

r_s Radius of maximum electron density for s orbital

r_{val} Radius of maximum electron density for valence (highest occupied) orbital

σ_f Selector for the Domain of Applicability

T_c Superconducting critical temperature of material

Abstract

Materials discovery has traditionally been guided by empirical approaches, with workflows that involve much trial and error. This method has yielded great discoveries of novel materials in the past centuries with the technology available, however, with the development of computational power as well as new methods for developing predictive models for material simulation, approaches are being developed that seek to involve less trial and error, saving time and resources.

Machine Learning (ML) models for various materials properties have become increasingly popular because of their potential to rapidly screen materials at a computational cost orders of magnitude smaller than traditional *ab initio* methods. However, the current ML methods only exploit the outcome of the model developed and focus on optimizing this model, neglecting the possibility of guiding the next best experiments.

Active learning is the approach of adaptively guiding the choice of the next best experiments, allowing for the best choice of materials to test next in a materials discovery workflow. Despite the promise of active learning to accelerate materials discovery, it is currently still a field that has yet to be explored thoroughly. This provides us with the impetus to explore the active learning method and workflow to materials discovery, and develop our own active workflow and test the feasibility of this approach.

Hence, we propose to develop an active learning scheme by firstly identifying a global model for predicting the bulk modulus of selected perovskite materials, using an AI-based method known as the sure-independence screening and sparsifying operator (SISSO) method. Next, we identify significant areas of interest in our dataset using subgroup discovery (SGD) to find the domain of applicability (DA), and lastly apply the subgroups found to attempt to direct our next chosen area of study.

List of Figures

1.1 (a) Key aspects of the traditional materials experimental workflow. (b) Techniques and developments in technology that can be used to accelerate the workflow. From Ref [2] 15

1.2 Demonstration of GPR with model prediction and decision making. Actual function in cyan represents the ground truth function fitted to the data. GP predicts response for test points along with the associated uncertainties. From Ref [5] 20

2.1 Descriptor algorithm combining unified subspaces with largest correlation and residual errors Δ(or P) generated by SIS with SO to extract the best descriptor. From Ref. [19] 24

2.2 Workflow for DA identification and validation for an ML model. From Ref. [20] 25

3.1 Active Workflow developed 29

3.2 Demonstration of checking Birch-fit for $CsEuF_3$ 34

3.3 Parameters used for creedo 36

4.1 The first subgroup plotted with the probability density function to show the distribution and spread of errors between the oxide perovskites inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0. 40

4.2 Cumulative Density Function for the first subgroup of materials inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0. 41
4.3 The second subgroup plotted with the probability density function to show the distribution and spread of errors between the oxide perovskites inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0. 45
4.4 Cumulative Density Function for the second subgroup of materials discovered inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0. 46
4.5 Correlation plot showing Pearson correlation between two SISSO models as well as the outlier materials. 47
4.6 Parity plots for the two SISSO models 48
4.7 Distribution of absolute errors and RMSE for the Oxide-Sulfide dataset. The model dimension with the lowest RMSE is model 5, the Validation RMSE decreases with increasing model dimension. 50
4.8 Probability Density Function for the subgroup with Oxide-Sulfide Perovskites discovered inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0. 51
4.9 Cumulative Density Function for the second subgroup of materials discovered inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0. 53
4.10 Proportion of C atom in the subgroup 54
4.11 Distribution of C atom in the subgroup 54
4.12 Distribution of absolute errors and RMSE for the Sulfide-Halide-Oxide dataset. The model dimension with the lowest RMSE is model 5, the Validation RMSE decreases with increasing model dimension. 55
4.13 Probability Density Function for the subgroup with Sulfide-Halide-Oxide Perovskites discovered with varying coverage parameters 57

4.14 Cumulative Density Function for the second subgroup of materials discovered with varying coverage parameters 59
4.15 Proportion of C atom in the subgroup (coverage 0.3) 60
4.16 Distribution of C atom in the subgroup (coverage 0.3) 60

List of Tables

3.1 Metrics developed for evaluating active workflow 37

4.1 First subgroups discovered with varying coverage parameter 38
4.2 Second set of subgroups discovered with varying coverage parameter . . . 44
4.3 First set of subgroups discovered on the new dataset 50
4.4 First subgroups discovered on new dataset with varying coverage parameter 56

Contents

Abbreviations 3

Nomenclature 5

1 Introduction 14

1.1 Background . . . 14

1.2 Machine Learning . . . 16

1.2.1 Overview . . . 16

1.2.2 Machine Learning and Materials Science . . . 17

1.3 Active Learning . . . 18

1.3.1 Overview . . . 18

1.3.2 Bayesian Optimization . . . 18

2 Theoretical Background 21

2.1 Fritz Haber Institute ’ab initio molecular simulations’ package . . . 21

2.2 SISSO methodology . . . 22

2.2.1 Principle of SISSO . . . 23

2.2.2 Descriptor algorithm . . . 24

2.3 Domain of Applicability (DA) and Subgroup Discovery (SGD) . . . 25

2.3.1 Theory . . . 25

2.3.2 Implementation using Creedo . . . 26

3 Methodology **28**
3.1 Overview of workflow and datasets 28
3.2 FHI aims calculations 30
3.2.1 Calculation setup 30
3.2.2 Calculation procedure 33
3.3 SISSO-SGD Approach 34
3.3.1 Hierarchical SISSO (hiSISSO) approach 34
3.3.2 Subgroup Discovery using Creedo 35
3.4 Metrics for evaluating the workflow 37

4 Results and Discussion **38**
4.1 First Active Workflow cycle 38
4.1.1 First Subgroup Discovery 38
4.1.2 SISSO Cross Validation Second Model 42
4.1.3 Second Subgroup Discovery 43
4.2 Second Active Workflow cycle 49
4.2.1 Sulfide Oxide Dataset 49
4.2.2 Sulfide-Halide-Oxide Dataset 55

5 Conclusion and outlook **61**

6 Annex **66**
6.1 Code used for fitting 66

1 Introduction

1.1 Background

The discovery of novel materials has been accelerated in recent years by the increasing shift toward computational materials science and materials modeling. The advancement of computational power with High Throughput Computation (HTC) has enabled this development, which allow for increasingly computationally expensive calculations to be completed, essential to the demands of materials simulations. [1] Additionally, *ab initio* electronic structure methods have been increasingly well developed in order to be more efficient and/or more accurate for simulating new materials, with Density Functional Theory (DFT) as one common tool for electronic structure simulation, Molecular Dynamics (MD) codes and Monte Carlo simulations as nuclear dynamics methods, providing researchers with more tools to explore the diverse possible combinations of materials. [2]

The use of Artificial Intelligence (AI) in automating materials experiments is another development in the field, and is motivated by the reduction of the costs of the experiments required in the trial and error approach, in terms of time and resources, as well as for the reduction of human error in the process. The challenge is then to find a way to use AI to further improve the materials discovery workflow, which is traditionally done with Machine Learning (ML) models that aim to describe a particular material property or feature. [3]

As seen in Figure 1.1, automation, parallelization (with HTC) are key accelerators, along with ML models, data repositories automated reasoning and finally active learning.

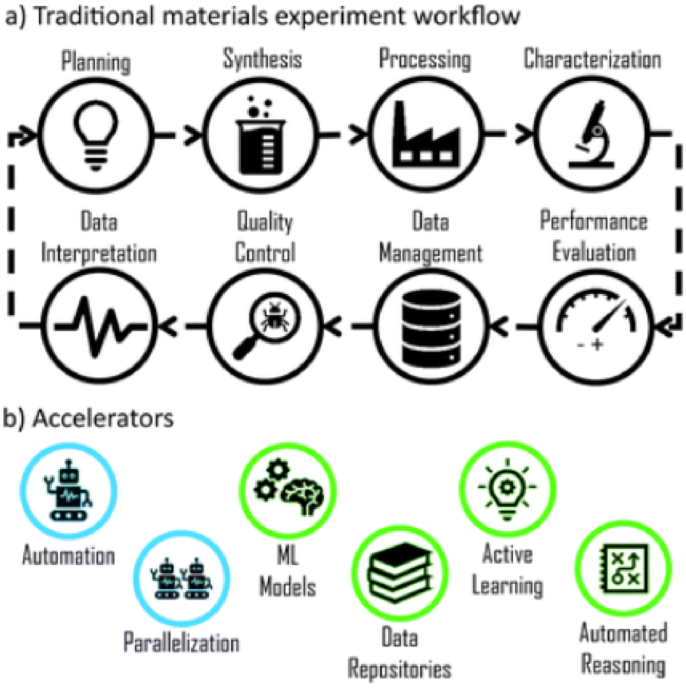


Figure 1.1: (a) Key aspects of the traditional materials experimental workflow. (b) Techniques and developments in technology that can be used to accelerate the workflow. From Ref [2]

1.2 Machine Learning

1.2.1 Overview

Machine Learning can refer to any form of neural network learning, with many approaches currently using deep neural network (DNN), which are defined as an artificial neural networks with multiple hidden layers of units between the input and output layers. Deep networks are also more complex and computationally demanding. There are more neurons and more connections between them and some neurons may even influence the environment by triggering actions. ML algorithms also have certain parameters known as hyperparameters, which control the complexity of the model and can also influence the effectiveness of the model prediction. [4]

Two widely used classes of ML approaches are supervised and unsupervised learning. Supervised learning is one branch of machine learning where the model is supplied with labeled examples of what it should learn. The critical difference is that with supervised learning, we have some notion of a label or one characteristic of each data point that is significant. In unsupervised learning, the data does not have labels. Consequently, unsupervised problems focus on discovery: looking at the raw data and seeing if it naturally falls into groups.

Supervised learning can be further split into regression or classification problems. In classification, the output of the model are discrete class labels, whereas for regression the output is a prediction in terms of continuous quantities. For the purposes of our active workflow and property simulated, regression is the kind of machine learning that is carried out.

1.2.2 Machine Learning and Materials Science

One of the grand challenges in materials science with machine learning is mainly the issue with generality and encompassing of a high dimensional search space with a high number of compounds that should be explored. This search requires inspection of crystal structure, chemistry and microstructure, with many material properties that could be dependent on the material descriptors, across different length scales. Materials scientists have so far mainly utilised machine learning to develop predictive models. [5]

In the field of materials science, depending on the problem to be simulated, the size of datasets can be limited, and hence not truly 'big'. Thus, in some instances the utility of big data for materials science applications is limited. [6]. There have been efforts to expand this data available, with the advent of data sharing and open access databases, such as the NOMAD archive which enables sharing of data on the platform in an attempt to expand the big data available to the materials science community [7].

One example of an application of building ML models in materials science is in determining the critical temperature (T_c) of Superconducting (SC) materials. In this case, there is a small subset of materials where the model learned applies, as the subset of SC materials that follow the relationship for determining the Tc are in materials space not determined by a linear function. This is also due to the fact that there are features of the phenomenon of SC that are not fully understood, such as the relation between SC and the chemical properties of materials. [8]

Nonetheless, it is still imperative to develop a model using fewer instances of training, as normally a supervised learning system has to be trained on labeled data or instances in the magnitude of thousands or more. This is where an active learning scheme can be greatly useful in order to develop a model for a desired property with less training. Subgroup discovery is also useful here in determining the relevance of the global model locally.

1.3 Active Learning

1.3.1 Overview

Active learning is a set of methods that automatically selects the next set of experiments to run from a given pool of candidates, without requiring the users input. As a field of study, it is a subset of ML, as it involves This approach leans on efficiently adding the samples needed into the training set. Uncertainties are used to push the system toward exploitation, instead of exploration. However, the distinguishing factor is where there is a necessity to continuously update the model while carrying out the experimental workflow. This is also known as a closed-loop workflow, and can eventually lead to experiments that are self-driven. [9]

Depending on the dataset size, for materials science applications it can be difficult to develop ML models without large uncertainties. One such method to tackle this problem is the approach of improving the performance of the ML model, in order to increase the size of labeled observations iteratively, and enhance the ML model in question successively. This involves the trade-off between exploration and exploitation of data, in order to guide the search for recommended samples for the experiments from a readily available pool of possibilities. Exploration is defined as using the data points outside the scope of the current training set to expand the model, while exploitation is done to add points inside the current training set, where the optimal solution is determined to be. [10]

1.3.2 Bayesian Optimization

Active learning is typically implemented through Bayesian Optimization, a strategy that finds the extrema of objective functions, which are normally more difficult or expensive to evaluate. When considering the number of function evaluations that are required, this method is one of the more efficient approaches. This is mainly thanks to the capability of

Bayesian optimization, to use some previous knowledge about the problem to aid in the sampling. [11]

Optimization is a field that is fundamental to mathematics, and here is defined in the case of maximization rather than minimization. The problem to be solved is stated by $max_x f(x)$, to find the x that maximizes the function $y = f(x)$. The efficiency in this process is using the prior beliefs in form of smoothness, which is essentially the locality of $f(x)$ in order to direct the adaptive sampling. This means that $f(x)$ is estimated normally with a surrogate model, which is commonly carried out using Gaussian process Regression (GPR). [12]

GPR is a Bayesian approach that allows for modeling of functions. Similar to how a Gaussian distribution is over a random variable, GPR is a distribution over functions with inference drawn directly in the function space. GPR extends the multivariate Gaussian distribution, with an infinite number of variables and forming a stochastic process: a process that governs the properties of the functions. [13]

The demonstration of GPR can be seen in Figure 1.2. Five data points are chosen at random to train the surrogate model with GPR, with predictions shown in red with the curve that is fitted. Next in (b) the updated surrogate model is shown, which is obtained after first iteration with the total of six data points used for training. The reduction in error bars from (b) compared to (a), is noteworthy, and the exploration strategy suggests the next data point, with the largest uncertainty for the next computation. The trained model is now used to predict the response of the data point that was left out. [5]

However, one key issue remains to be the use of Bayesian Optimization with GPR results a model which may not be generalized beyond the specific solution. Hence, our proposal is to develop an active workflow that develops a global machine learning model using the Sure-independence screening and sparsifying operator method (SISSO) method, while remaining locally relevant with a small number of local models found using subgroup discovery.

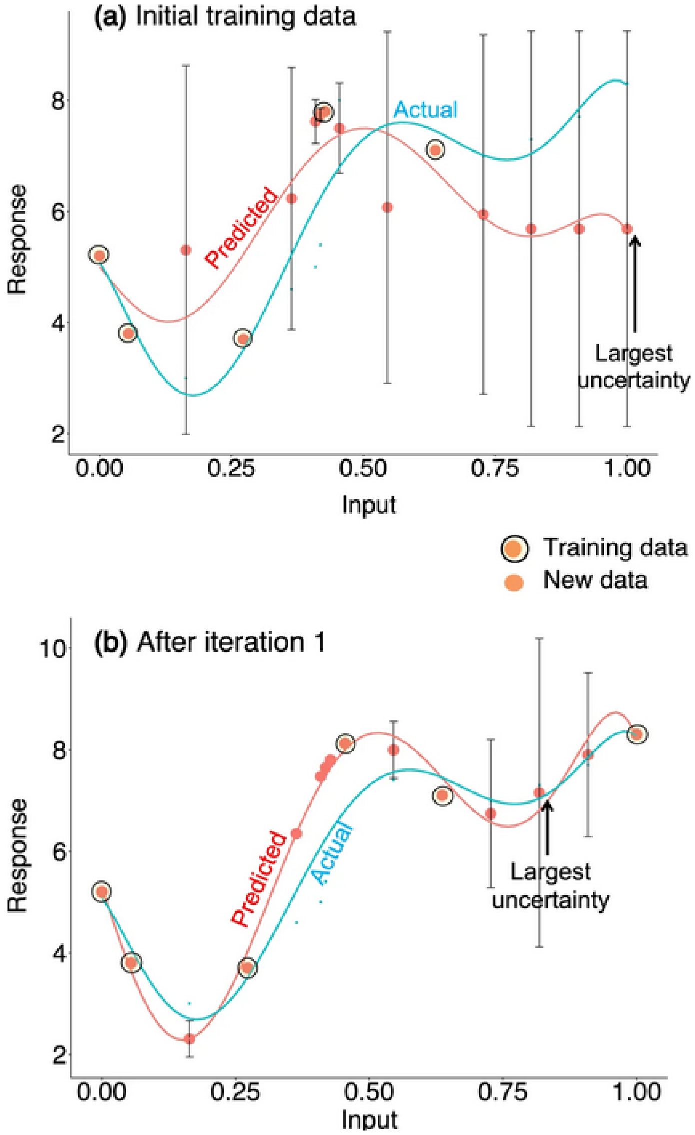


Figure 1.2: Demonstration of GPR with model prediction and decision making. Actual function in cyan represents the ground truth function fitted to the data. GP predicts response for test points along with the associated uncertainties. From Ref [5]

2 Theoretical Background

2.1 Fritz Haber Institute ’ab initio molecular simulations’ package

DFT is a powerful computational materials modeling method used to simulate and investigate the electronic structure of many-body systems, developed based on a the theory by Hohenberg and Kohn. [14], [15] The FHI-aims package is an all-electron, full potential framework, and is used in this work for the DFT calculations. FHI-aims has an efficiency close to the fast pseudopotential schemes based on plane-waves, and also being scalable with system size of N up to thousands of atoms, being developed for parallel execution of jobs to ensure efficient workflows. [16] FHI-aims is based on numerically tabulated atom-centered orbitals (NAO), which allow for a wide range of material and molecular properties to be captured from quantum-mechanical first-principles.

Basis sets here are defined as a set of functions that are used to represent the electronic wave function according to DFT. [17] In this case of FHI-aims, the basis sets are constructed following a set of principles, in order to allow for fast calculations as well as basis set convergence that is able to reach the level of meV in terms of total energy accuracy. The following principles are used to guide the basis set construction: 1) Basis sets should be made available beginning from minimal basis until the meV-level for total energy convergence. 2) Basis sets for the same elements are successively hierarchical, hence a larger basis set would contain the smaller ones, ensuring a more strict variational total energy convergence. 3) The procedure for constructing the basis with minimal human bias, hence as objectively as possible without preliminary human intuition interfering. Since all of the basis sets are strictly localized, this means that the operations scale as O(N) with a system size of N. Overall, FHI-aims provides an efficient and accurate way to carry out our DFT calculations as part of our workflow.

2.2 SISSO methodology

The increasing trend toward computational materials has led to the need for a systematic approach that allows us to identify possible materials which are functionally useful (e.g. stable, nontoxic, etc.). The difficulty in developing a systematic approach lies in the many factors to be taken into account when determining the stability and functionality of a material.

Hence, insight into these processes must be obtained through discovering structure and patterns in data that arise from functional relationships between these processes and properties. The key step in this process is identifying suitable descriptors, a set of parameters that is able to capture the main characteristics that describe a given property or function for a material. In essence, descriptors are functions of the simple parameters or features that characterize the actuating mechanisms of a material property, and allow us to construct 'maps of material properties'. This descriptor will then enable us to uncover the relationships, allowing an artificial intelligence based learning approach to then be applied to this system.

This approach has been developed into a compressed-sensing formulation, coined as the sure-independence screening and sparsifying operator (SISSO) method. Compressed-sensing (CS) based approaches allow for reproduction of a high quality signal from a small set of observations, which is highly useful in the field of materials science, where the luxury of massive datasets is not always available depending on the material or process to be simulated. In essence, SISSO handles large, correlated feature spaces, converging to an optimal solution by using a combination of training sets. This approach means that SISSO can handle an immense number of candidate features in the scales of billions, and still be able to find an optimal solution when the features are correlated.

2.2.1 Principle of SISSO

The SISSO procedure begins first with feature space construction, whereby an initial set of user-defined properties are utilized in order to start constructing the feature space. Features can be categorized into those which are atomic or species specific, and those which are collective and depend on the environment of the atoms. Next, functional and algebraic (analytical) operations extend the feature space recursively. This can be described better using the operator set $\hat{H}^{(m)}$:

$$\hat{H}^{(m)} \equiv \{I, +, -, \times, /, exp, log, |-|, \surd, ^{-1}, ^{-2}, ^{-3}\}[\phi_1, \phi_2] \tag{2.1}$$

where ϕ_1 and ϕ_2 are objects in Φ (starting point for the construction of the set) and the superscript m denotes that the dimensional analysis is performed only to retain some meaningful combinations.

The features created currently are candidate descriptors which have a linear relationship with P. For every iteration, $\hat{H}^{(m)}$ operates all combinations available, which results in the feature space growing recursively.

In SISSO the problem is resolved by combining SO with Sure Independence Screening (SIS) which is effective in reducing the dimensionality for the massively high dimensional feature spaces. SIS is used as L_0 regularization is too expensive at a certain point, and SIS reduces the feature subspace to a more manageable one. [18]

Hence, SISSO is useful for our application as it reduces the extensive feature space and allows for training on a single chosen expression.

2.2.2 Descriptor algorithm

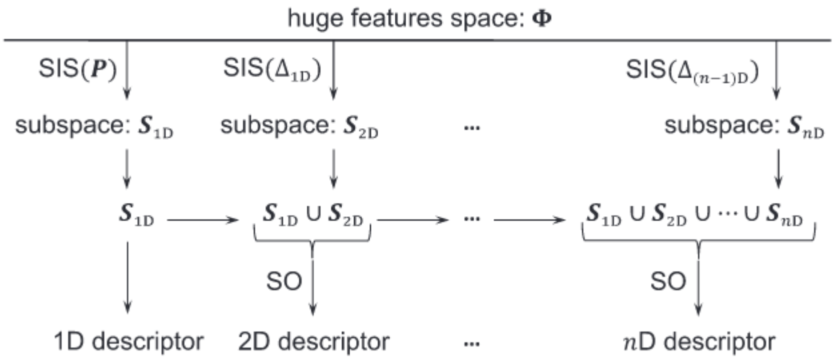


Figure 2.1: Descriptor algorithm combining unified subspaces with largest correlation and residual errors Δ(or P) generated by SIS with SO to extract the best descriptor. From Ref. [19]

As seen from Figure 2.1 , SIS is used to score each feature with a metric, retaining only top-ranked features. From the large feature space, SIS selects the subspace which contains the features with the most correlation with response P which is the target material property. The larger this subspace is, the higher the probability for it to contain the best descriptor.

2.3 Domain of Applicability (DA) and Subgroup Discovery (SGD)

2.3.1 Theory

One of the key issues with ML models applied to materials, is the difficulty in generalizing the global model learned for specific applications. Hence, conventionally, different ML models for materials are proposed to provide predictions that remain accurate for specific applications. The challenge with this approach is that the ML models are based on simplistic metrics such as model test error simply with respect to the whole class of materials. Hence, an approach that detects domains of applicability (DA) of ML models in a material class can be seen as one way to resolve this issue. [20]

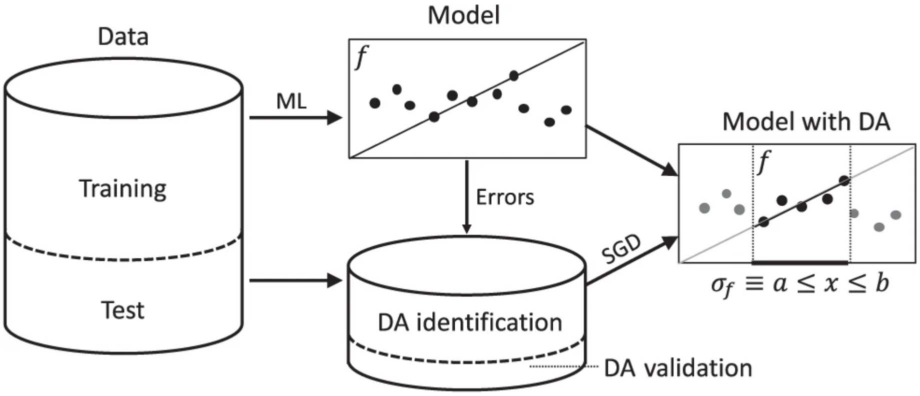


Figure 2.2: Workflow for DA identification and validation for an ML model. From Ref. [20]

The DA can be described usually by a selector (σ_f) which comprises logical conjunctions from a space of representation. The DA is normally constructed using simple representation, as complex representations may not be easily mapped back intuitively to a given material. A single representation here is defined as one that comprises of features, specifically catered to the description of subdomains of interest. As an example, the shape of the unit cell, the number of atoms in a unit cell and the unit-cell volume are features. The DA optimization and validation is carried out using the workflow seen in Figure 2.1, domains of applicability are described by a selector, which comprises logical conjunctions pertaining to a specific representation space. The selector is normally chosen through applying the subgroup discovery (SGD) method to individual machine learning model errors for a subset of a test set. The regions where the model error is significantly lower than the global average for that material class is then found. This provides a method in contrast to other methods that provide uncertainty estimates for individual data points. Lastly, an unbiased estimate of the performance of the model within the DA found is obtained on the rest of the test set that were not in the DA validation set.

SGD thus provides a flexible framework which focuses more on exploratory data analysis, discovering new insights regarding the DA in question, compared to most methods which currently use predictive or global modeling, which focuses more on developing a model that can best fit the given data and features available. In building a global model, the prediction accuracy is the key metric to optimize, which is achieved through more training data as well as optimizing hyper-parameters.

The implementation for SGD is carried out by Creedo, a software that was developed for scalable data analysis studies that are tailored by the user for specific experiments.

2.3.2 Implementation using Creedo

Creedo is a conceptual framework that allows for empirical user studies and is used here specifically in the domain of pattern discovery. It is a Java based server application, allowing users to define the settings for experimental studies and to tailor these studies for their needs. Creedo provides the framework that allows for the creation of studies that are repeatable as well as scalable, and can be implemented without use of extensive resources, compared to a study that requires supervision or manual implementation of the design. Creedo is built in a manner such that the application performs the tasks assigned, and can be implemented on any web browser. Creedo uses realKD algorithms that allows to create and execute online user studies.[21] realKD is a free open-sourced Java based library that is designed to be a new tool for data exploration, with a strong focus on developing a detailed data model, which allows for the specification of domain-dependent semantics. The model developed is also meant to capture meaningful knowledge from the data, and should be able to describe algorithms and their parameters in a user-friendly manner. In the case of this work, realKD is mainly used for discovering associations, and exceptional model patterns, or subgroups in the data.

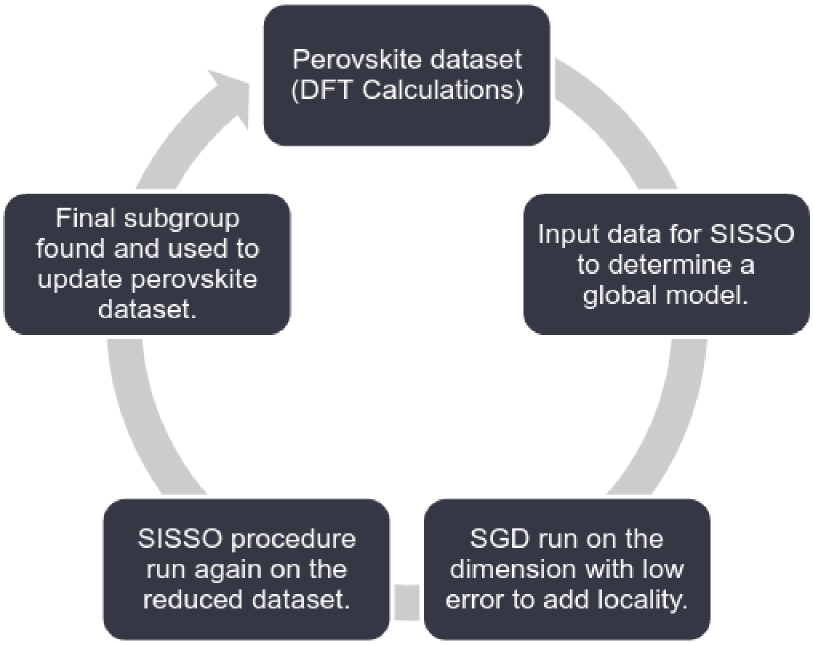
The motivations for having creedo run on a web application are firstly, to enable scalable studies, as any computer with access to the server has the potential to be used as a terminal, and secondly, to ensure that all definitions and statistics can be controlled from one central location.

3 Methodology

3.1 Overview of workflow and datasets

To develop the active workflow, the first step was to consider the data readily available for training of a global model. For the purpose of our training, an oxide perovskite dataset of 504 oxide perovskites was chosen. The data consists of 504 ABO_3 perovskites, that are evaluated with DFT-PBEsol (Perdew-Burke-Ernzerhof functional revised for solids) using the FHI-aims full potential all-electron code. For the selection of evaluated materials, the A and B atoms were chosen based on the oxide perovskites available in literature (cite here). The atomic (primary) features of the isolated atoms (A and B) were calculated with DFT-PBEsol. These features include the radius of maximum electron density for s (r_s) and valence (highest occupied) (r_{val}) orbitals, the energy of highest occupied and lowest unoccupied states, the electron affinity (EA) and ionization potential (IP), the nuclear charges (Z) and the ionic charge of A (c_A). These features are used by SISSO to develop a global predictive model for the bulk modulus of the perovskitee.

As seen in Figure 3.1, the workflow begins with the training of a global SISSO model, based on the oxide perovskite dataset. This global model is then used to perform SGD to find the domains of applicability. The subgroup discovered is then used to reduce the dataset, whereby we remove the materials in question with low error, and are left with the remaining oxide perovskites. A further second global model is found using SISSO again,



Perovskite dataset (DFT Calculations)
Input data for SISSO to determine a global model.
SGD run on the dimension with low error to add locality.
SISSO procedure run again on the reduced dataset.
Final subgroup found and used to update perovskite dataset.

Figure 3.1: Active Workflow developed

and the procedure is repeated to obtain further SGD. The first subgroup found is then applied to a new dataset, to show us which materials we can search for which would be of interest in terms of the property bulk modulus.

3.2 FHI aims calculations

3.2.1 Calculation setup

To further develop the dataset and to develop our active learning framework, 50 halide perovskites are considered, with the A and B atoms chosen based on charge balance. This dataset of halide perovskites serves as a basis to be explored upon by the active learning framework, where the promising candidates for the next group of materials are identified from this list.

The first step in the FHI aims DFT calculations is the use of two different input files, control.in and a geometry.in file. The geometry.in file provides the information that is relevant for the atomic structure in any given calculation, which includes the atomic positions as well as the description of particulars of every element that is contained in control.in file. The lattice vectors are also defined here for periodic calculations, which apply in this calculation for the perovskite solid crystal structures. The remaining information that is given here is only done so if it is directly linked to an atom, with parameters including the initial spin moment, initial charge, relaxation constraint etc. This can be better seen in Listing 3.1 of the geometry.in file.

For the purposes of preparing the geometry.in file, the AFLOW (automatic flow) package was utilized, which is a software framework that allows for high throughput calculations of crystal structure properties of various materials. [22] Specifically, AFLOW-Xtal finder, [23] was used to set up the input geometries in order to further automatize the process. The relaxation in this case was done with parametric constraints to ensure homogeneity in the perovskite cubic structures simulated [24].

An example of a geometry.in file for the halide perovskite $AgCoF_3$ generated by AFLOW-Xtal finder for an FHI aims calculation can be seen in the Listing 3.1.

```
# AgCoF/AB3C_cP5_221_a_c_b.ABC params=3.86 SG=221 [ANRL doi: 10.1016/j.commatsci.2017.01.017 (part 1), doi: 10.1016/j.commatsci.2018.10.043 (part 2)]
# AFLOW::AIMS BEGIN
lattice_vector    3.86000000000000   0.00000000000000   0.00000000000000
lattice_vector    0.00000000000000   3.86000000000000   0.00000000000000
lattice_vector    0.00000000000000   0.00000000000000   3.86000000000000
atom_frac    0.00000000000000    0.00000000000000    0.00000000000000   Ag
atom_frac    0.50000000000000    0.50000000000000    0.50000000000000   Co
atom_frac    0.00000000000000    0.50000000000000    0.50000000000000   F
atom_frac    0.50000000000000    0.00000000000000    0.50000000000000   F
atom_frac    0.50000000000000    0.50000000000000    0.00000000000000   F
# format: symmetry_n_params [n n_lv n_fracpos]
symmetry_n_params 1 1 0
symmetry_params a
symmetry_lv a , 0 , 0
symmetry_lv 0 , a , 0
symmetry_lv 0 , 0 , a
symmetry_frac 0 , 0 , 0
symmetry_frac 0.5 , 0.5 , 0.5
symmetry_frac 0 , 0.5 , 0.5
symmetry_frac 0.5 , 0 , 0.5
symmetry_frac 0.5 , 0.5 , 0
# AFLOW::AIMS END
```

Listing 3.1: Example of set-up for geometry.in file created by AFLOW XTAL with parametric constraints

The control.in file, which can be seen in listing 3.2, is then required for all information that is runtime-specific for the calculations. In general, there is a segment of the file that is more general, while additional more specific information related to the atoms is added on.

```
#Physical Settings
xc                              pbesol
spin                            none
relativistic                    atomic_zora scalar
basis_threshold                 1E-5
override_illconditioning        .true.

# SCF settings
sc_accuracy_forces      1E-5
sc_iter_limit           900
sc_accuracy_rho         1E-5
sc_accuracy_eev         1E-3
sc_accuracy_etot        1E-6
sc_init_iter            10

#k-point grid and relaxation
k_grid                  5 5 5
relax_geometry          trm 1E-2
relax_unit_cell         full
```

Listing 3.2: FHI aims calculations set-up

The Self-Consistency Field (SCF) cycles are controlled using the SCF settings, these are important in determining the convergence criterion for the calculations. These settings will now be detailed. The sc accuracy forces set the convergence criterion for the self-consistency cycle, which are normally based on energy derivatives. The sc accuracy force is usually a small positive real number against which the maximum difference of atomic forces is checked between the successive SCF iterating. The sc accuracy rho setting provides the criterion for convergence of the self-consistency cycle and is based on charge density. Thus, it is the most important criterion for convergence. Normally, this will take a value of $1E^{-5}$ for a geometry.in file that contains between 6 to 60 atoms. The sc accuracy eev is similar to sc accuracy rho, however it instead provides the convergence criterion based on eigenvalues, while sc accuracy etot provides the criterion based on total energy. The Harris-Foulkes form of this functional is used for FHI-aims as the total energy. Lastly, the sc iteration limit sets the maximum number of SCF cycles that can occur before the calculation is abandoned.

The choice of the basis set is also contained in the control.in file, and is essential as it affects both the efficiency as well as the accuracy of a calculation. The xc keyword specifies the exchange-correlation approach that is used for the self-consistent DFT. The values for the k-pint grid are crucial for sampling the Brillouin zone accurately, and are necessary for periodic calculations.

The last important setting to describe is the use of species-specific information, that are provided preconstructed as default definitions for different species, found in a subdirectory in FHI aims. There are four different levels of these defaults, from light to intermediate, to tight and finally, really tight. Light relaxations are mainly for quick prerelaxations, with few convergence errors arising. Intermediate is only present for some elements but can be useful for large calculations that are computationally expensive. The tight default is highly useful to provide meV level accurate energy differences for more accurate calculations. Really tight settings are similar to tight, however there are some numerical aspects such as the grids or Hartree potential that are strongly overconverged here.

3.2.2 Calculation procedure

The FHI-aims calculations that were carried out in our workflow, were first done with a ’light’ relaxation, followed by a final ’tight’ relaxation. The Birch-"murn.py" script (Annex 6.1), provided in the "utilities" folder from FHI-aims is used to fit the Birch-Murnaghan equation of state to a series of volume/energy calculated values and thus obtain the bulk modulus. For the volume/energy values, the equilibrium volume of the structure, obtained after relaxing the structure, and other values not too far from it (+/- 1% volume) were used.

As seen in the figure 3.2, the calculation for $CsEuF_3$, the Burch fit can be plotted using the tools provided, and the calculations were done in a manner to ensure the Burch fit is

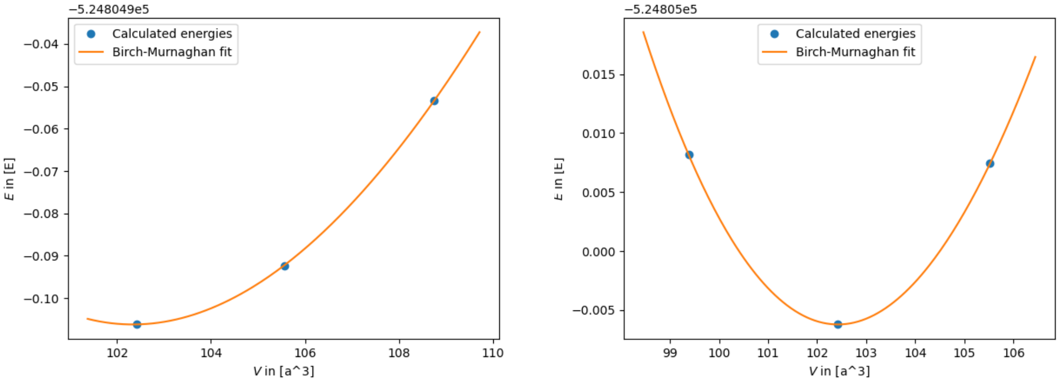


Figure 3.2: Demonstration of checking Birch-fit for $CsEuF_3$

appropriately done about the minimum (+/- 1$\%$ volume) as seen on the figure on the right.

3.3 SISSO-SGD Approach

3.3.1 Hierarchical SISSO (hiSISSO) approach

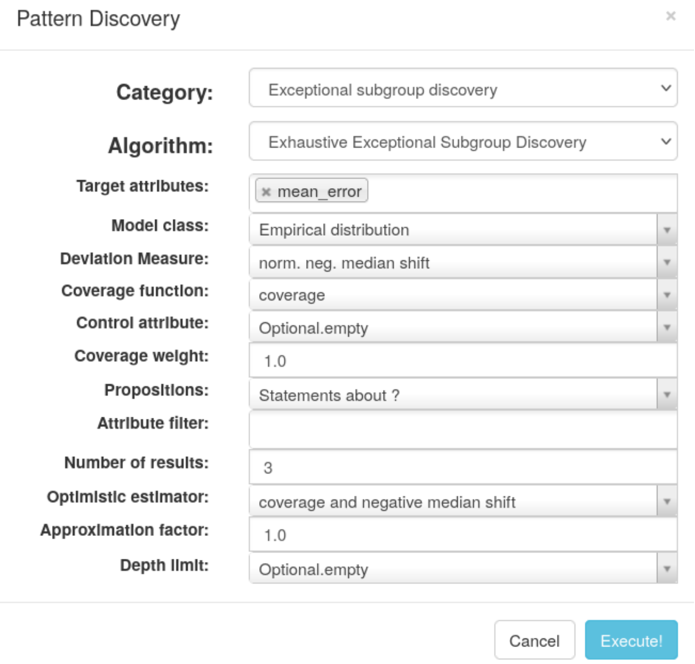
The approach adopted here combines domain of applicability with SISSO, in order to develop a global model (based on SISSO), that is still locally relevant. In other words, a model is developed that functions globally, with a small number of local models that are relevant in certain DAs. If the employed model is particularly applicable in a specific subdomain of the materials class, and if that subdomain has a simple and interpretable shape that permits to generate new materials from it, this allows one to directly focus the screening there. The procedure will now be detailed based on the different steps involved, beginning with the SISSO calculations. As mentioned previously, the primary features used are as follows: the radius of maximum electron density for s (r_s) ad p (r_p) orbitals, the energy of highest occupied and lowest unoccupied states, the electron affinity (EA) and ionization potential (IP), the nuclear charges (Z) and the ionic charge of A (c_A).

3.3.2 Subgroup Discovery using Creedo

The model that was developed on the oxide perovskite dataset uses an advanced form of SISSO, known as hierarchical SISSO (hiSISSO), which further exploits the relationship between material properties, with the goal of improving descriptor identification. Firstly, the descriptors for a simple property are identified, then the descriptors are used as primary features to address a related property, which is more complex in nature. The hiSISSO approach works for the situation whereby the relationship between the primary features and the material property is too complex.

In this workflow, the hiSISSO approach is applied in order to identify descriptors for bulk modulus (B_0) in the oxide perovskite dataset, with the lattice constant (a_0) used as a simple property. These properties are related since they both depend on bond strength. [25] After discovering the global model using hiSISSO, the model with the dataset is then used in the subgroup discovery process. This is carried out using the creedo java application which can be run on any web browser. The parameters that can be adjusted for the subgroup discovery will now be discussed, in order to have a better understanding of the active workflow.

The different features in creedo are seen in figure 3.3, and aid in understanding the process of using creedo. Target attributes are the list of attributes for which patterns should show special characteristics. The model class is the type of data summary according to which population deviation is measured, in our case we choose an empirical distribution. The deviation measure is the function for measuring the deviation of the local population to be sought for, from the global population. Normalized negative median shift is chosen, in order to search for areas in the data that have a low error. The control attribute is that for



Pattern Discovery

Category: Exceptional subgroup discovery

Algorithm: Exhaustive Exceptional Subgroup Discovery

Target attributes: mean_error

Model class: Empirical distribution

Deviation Measure: norm. neg. median shift

Coverage function: coverage

Control attribute: Optional.empty

Coverage weight: 1.0

Propositions: Statements about ?

Attribute filter:

Number of results: 3

Optimistic estimator: coverage and negative median shift

Approximation factor: 1.0

Depth limit: Optional.empty

Figure 3.3: Parameters used for creedo

which the subgroup should show an identical distribution just as in the global population. The coverage weight is an important term that is the power of the coverage factor in the optimization function. The number of results allows us to determine the number of optimal subgroups to be found. The optimistic estimator is the function that is used for pruning the search space, and the approximation factor guarantees to return the solution with optimization value that is at least x times the optimal value. Lastly, the depth limit represents the maximum depth in the refinement tree to be expanded by the algorithm.

3.4 Metrics for evaluating the workflow

Having gone through the methods used to build the steps of the active workflow, metrics to evaluate our workflow have been developed in order to better define some key charateristics that will indicate whether to keep going on with the training, or if it is better to change approach.

Metric	Condition
Size of the subgroup	<10% of the dataset
Distribution of materials with low and high error	Is there a clear separation ?
Correlation between the errors from the models	High correlation (>0.7)

Table 3.1: Metrics developed for evaluating active workflow

The first metric is the size of the subgroup, as a local model found has to be of a certain size in order to be significant. The next is the separation in the distribution of materials with low and high error in the prediction. Lastly, in order to determine if the two models are in good agreement, the correlation factor between the two global SISSO models developed must also be taken into consideration. Through the development of the active framework, these metrics are useful as benchmarks to guide further work on our cycles of the workflow.

4 Results and Discussion

4.1 First Active Workflow cycle

4.1.1 First Subgroup Discovery

Coverage 0.3		
Property	Condition	Value
Electron Affinity of A atom	>=	0.735817 eV
Lower Unoccupied Molecular Orbital of B atom	>=	-4.68356 eV
Radius of B atom (r_{val})	<=	0.975250 Å
Coverage 0.5		
Electron Affinity of A atom	>	0 eV
Lower Unoccupied Molecular Orbital of B atom	>=	-4.68356 eV
Radius of B atom (r_{val})	<=	0.975250 Å
Coverage 1.0		
Electron Affinity of A atom	>	0 eV
Radius of B atom (r_{val})	<=	0.975250 Å

Table 4.1: First subgroups discovered with varying coverage parameter

In this section, the initial results of the first iteration of the active workflow are entailed, along with the conclusions that can be drawn in order to enhance the workflow. The results of the subgroup discovery analysis are discussed first, followed by the SISSO models, and then further discussion is made to improve the active workflow.

The subgroups found are described in Table 4.1: these conditions, namely the restriction of the radius of B atom (r_{val}), limit us to mainly the transition metals in the periodic table. The physical significance of this is mainly indicating the better performance of the global model found by SISSO in predicting the bulk modulus for oxide perovskites where the B element is a transition metal.

The probability density function of the subgroups is plotted in Figure 4.1, in order to help us visualize the distribution of materials in the subgroup as well as the size of the subgroup. In the best case, we would expect a subgroup with the low error materials completely separated from those with high error, leaving a subgroup where the predictive model for bulk modulus has performed best. This would be shown by the histogram being completely orange in the areas with low error.

In Figure 4.1, the effect of the coverage parameter on the distribution of high and low error materials is also studied. Here we observe that the subgroups discovered still contain a proportion of materials that have high error, although the search was towards low error materials. Hence, the subgroup discovered is not the best-case scenario expected, however, it is still useful in providing us with a subset of materials that have lower error in the prediction of the bulk modulus, which is according to the metric of subgroup size developed, useful for our workflow (greater than >10% of the dataset). In this case, we do not observe much difference even with a change of coverage from 0.3 to 1.0. To further investigate our findings, the cumulative density function is then plotted.

The cumulative density function is plotted in Figure 4.2, to observe the separation of the dataset between those in the subgroup and outside of it, from the distinction between the orange and blue lines. In the ideal situation, it would be expected that the orange line (materials in the subgroup) would reach 1.0 before the line in blue (materials outside the subgroup) increases, indicating. It is also expected that one would see minimal overlap between the two cumulative error distributions. Although it is also important to keep the

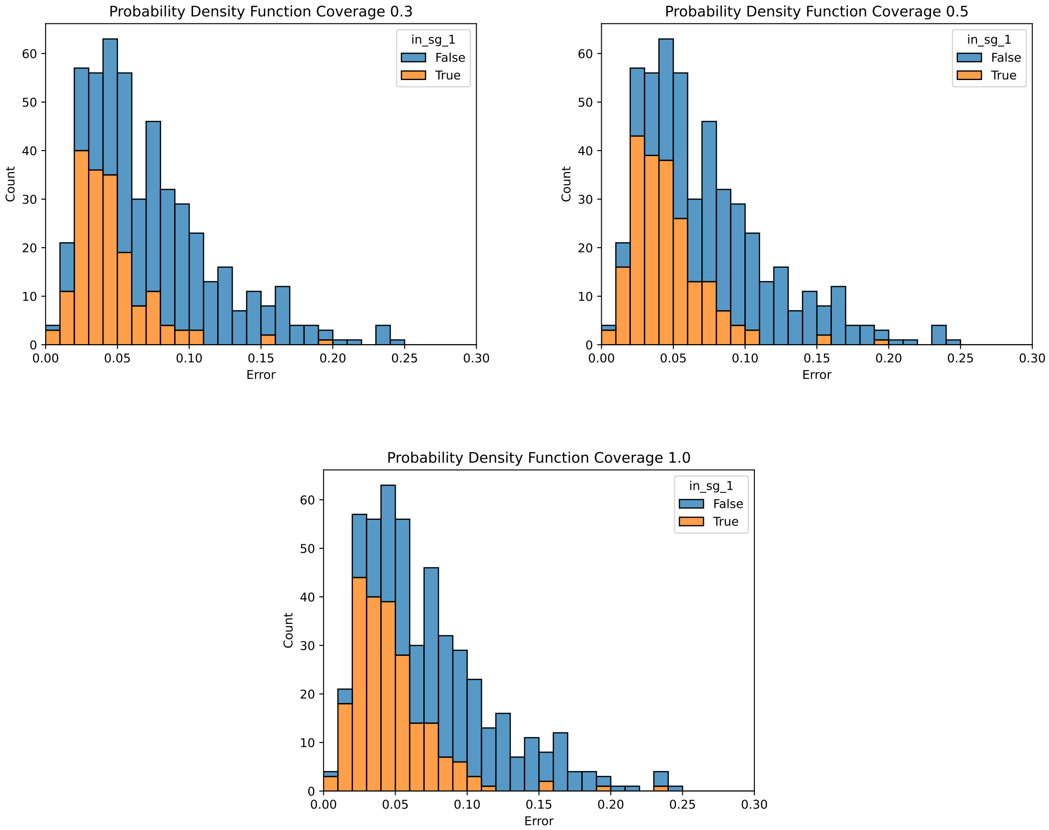


Figure 4.1: The first subgroup plotted with the probability density function to show the distribution and spread of errors between the oxide perovskites inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0.

size of the subgroups in mind, a small subgroup may improve this separation due to the small sample size. In this case, it is observed that there is an increase up to 1.0, however there is a more of an asymptotic approach of the graph to 1.0. This could indicate the limit of the model to completely separate low and high error materials. It also noteworthy that between the coverage of 0.3 to 1.0, there is no significant change in the separation.

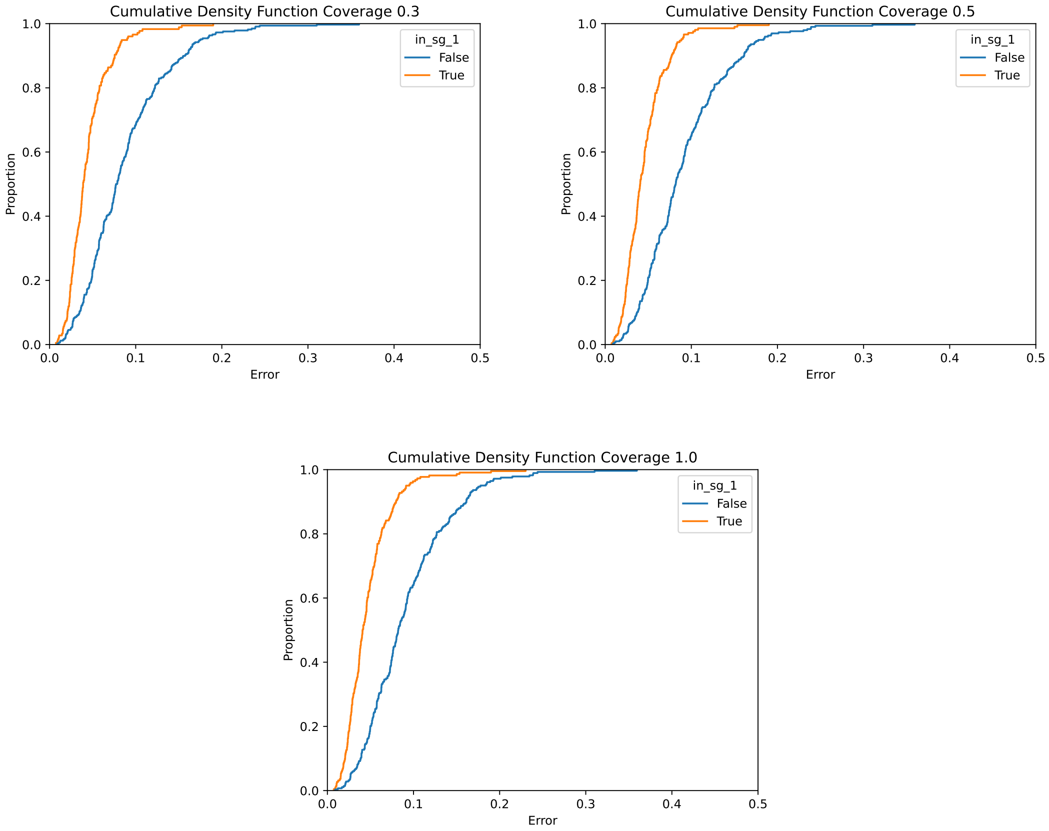


Figure 4.2: Cumulative Density Function for the first subgroup of materials inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0.

This indicates the stability of the subgroup found despite the change in coverage.

4.1.2 SISSO Cross Validation Second Model

The results from the first round of SGD, using negative median shift as the metric for searching (i.e. searching for materials with the lowest error), were that all of the subgroups discovered also had some groups with high errors in them, indicates that the high error samples are not fully removed. This could point to an inherent problem with focusing on low errors, meaning that every model will have some distribution of errors, and which samples have a low error may just be random and not systematic (an assumption we make when using SGD).

Another possibility is symmetric wells in the input feature space. Due to the nature of the study where the variables are the choice of elements, and the fact that across the periodic table there are symmetries in elemental features, this can result in ’symmetry wells’ where the material feature in question is the same. Thus, this could also impact the training of the global model by SISSO, and result in subgroups found which contain compounds with low and high error, however having the same atomic features that match the conditions given by SGD.

Hence, the next step taken was to remove the low error materials discovered, to see if other small local models exist that would need to be patched together. If another significant local model is found, this indicates to us that this subgroup should be studied in more detail, and that another cycle of the workflow (SISSO-SGD) should be possible.

Each sample (data point / material) is 5 times used to train a SISSO model (5 models in total) and also evaluated 5 times as test material. The absolute mean of the 5 prediction errors is taken, as well as evaluate the standard error, to see if the error estimate via its mean is reliable enough. This operation gives a distribution of mean (over the 5 iterations) errors, one entry for each material. This then becomes the new population for a second subgroup discovery.

This strategy allows the SISSO model to adapt to the specific training set (i.e., the descriptor can change) in order to avoid a specific bias due to the initial selection of training set. By using the Cross-Validation splits we can then ascertain the spread of the prediction errors using an ensemble of models and hopefully average out the random effects. The test error here is used instead of the training error, as the evaluation of the SISSO global model on unseen data is what we would like to investigate here.

4.1.3 Second Subgroup Discovery

Using the global model found with the lowest error, subgroup discovery is run again using Creedo, with the intention of seeking additional local models. The subgroups discovered are as follows, according to the change of coverage parameter. The second set of subgroups found can be described by the conditions shown in Table 4.2.

The conditions seen in Table 4.2 for the different subgroups in this case involve more parameters than the previous iteration of SGD. Noteworthy is the restriction of the elemental charge of the A atom, which for our dataset, limits us mainly to the elements Li, Be, Na, Mg, K, and Ca for the A atom (groups I and II in the periodic table). Interestingly, for the compounds containing the A atom belong to group I and II, the B atom for the perovskites in our dataset are also mainly transition metal elements.

Coverage 0.3		
Property	Condition	Value
Elemental charge of A atom	<	29
Electron affinity of B atom	>=	-0.793269 eV
Elemental charge of B atom	<=	46.5
Radius of B atom: Brs	>	1.43540 Å
Predicted lattice parameter from hiSISSO rung 1	<= AND >=	3.76401Å 4.00535Å
Coverage 0.5		
Ionization Potential of A atom	<=	7.36587 eV
Radius of A atom: Ars	<=	2.2645 Å
Radius of A atom: Arval	>=	0.763350 Å
Electron affinity of B atom	>=	-0.793269 eV
Ionic Potential of B atom: B(IP)	<=	8.93925 eV
Elemental charge of B atom: B(Z)	<=	46.5
Radius of B atom: Brs	>=	1.02150 Å
Predicted lattice parameter from hiSISSO rung 1	<=	4.00535 Å
Predicted lattice parameter from rung 1	<=	4.01571 Å
Coverage 1.0		
Electron affinity of A atom	<= AND >=	1.31958eV -0.350936 eV
Radius(val) of A atom: A(rval)	<=	2.26450 Å
Radius of B atom: B(rs)	>=	1.02150 Å

Table 4.2: Second set of subgroups discovered with varying coverage parameter

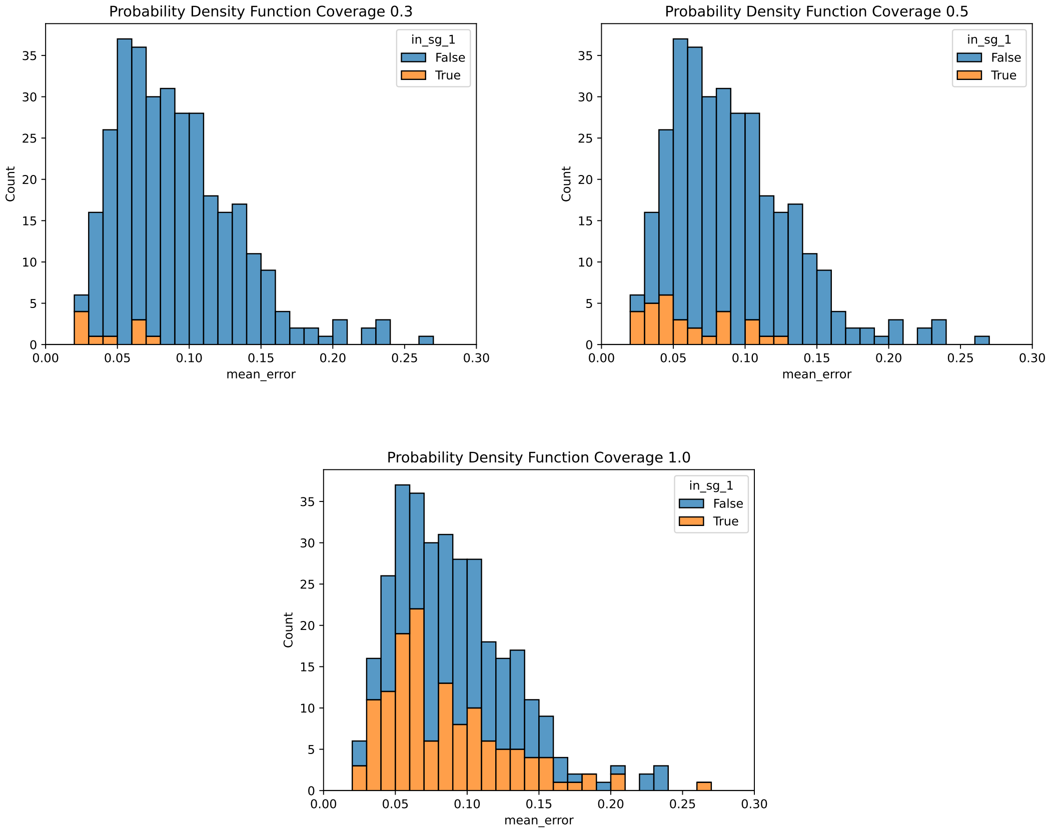


Figure 4.3: The second subgroup plotted with the probability density function to show the distribution and spread of errors between the oxide perovskites inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0.

The coverage parameter was again increased as seen in Figure 4.3 in order to see the affect it has on the distribution, and in this case, the first clear observation is a much smaller subgroup obtained. The subgroup (in orange) for coverage 0.3 for example is <10% of the size of the dataset, which already is a failure according to the benchmarks that we set. We also observe a much clearer difference size of the subgroup as we increase the coverage from 0.3 to 1.0. This indicates instability in the subgroups discovered.

The cumulative density function curve is then plotted to investigate the separation further.

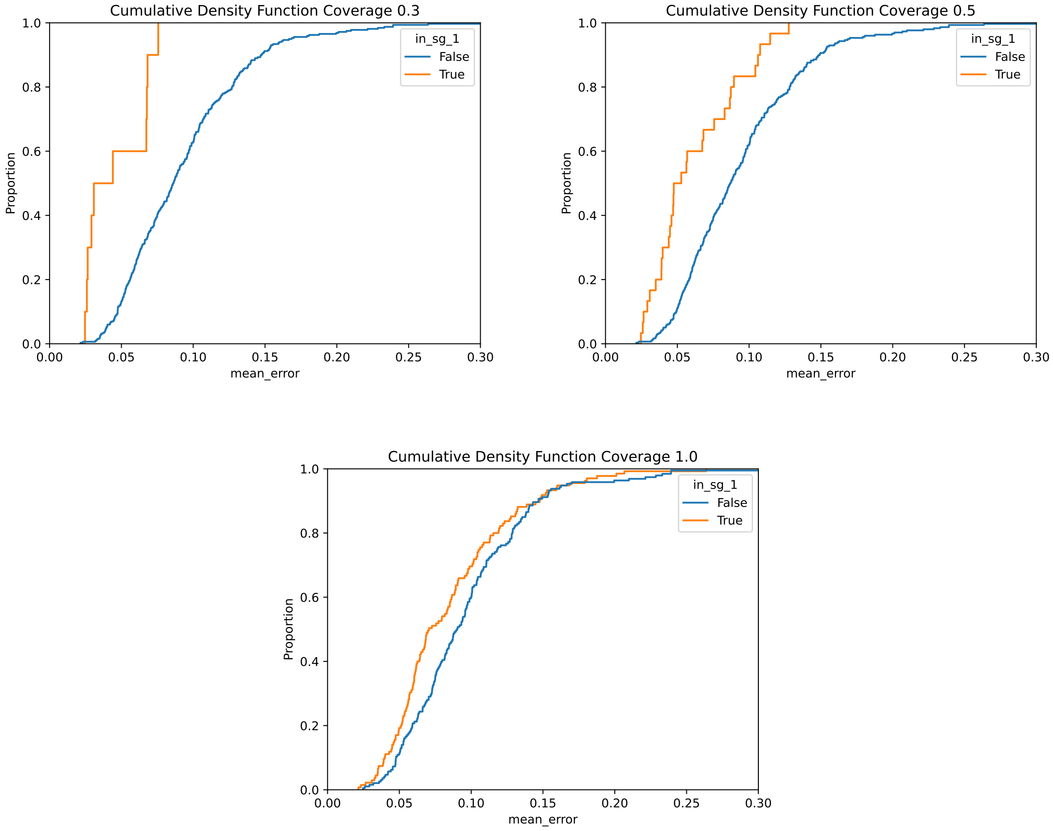


Figure 4.4: Cumulative Density Function for the second subgroup of materials discovered inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0.

The cumulative density function curves shown in Figure 4.4 show that the separation of the dataset between those in the subgroup and outside of it quickly diminishes as the coverage parameter is increased to 1.0, this is evidenced by the convergence of the blue and orange line for coverage of 1.0. This difference shows that with a lower coverage, SGD here does a better job of separating the low and high error materials, which is interesting to note considering this is the second round of SGD on the second model developed with SISSO. However, this is done with a very small number of materials (a smaller subgroup), hence the subgroup found is still not significant as decided by our previous metric discussed.

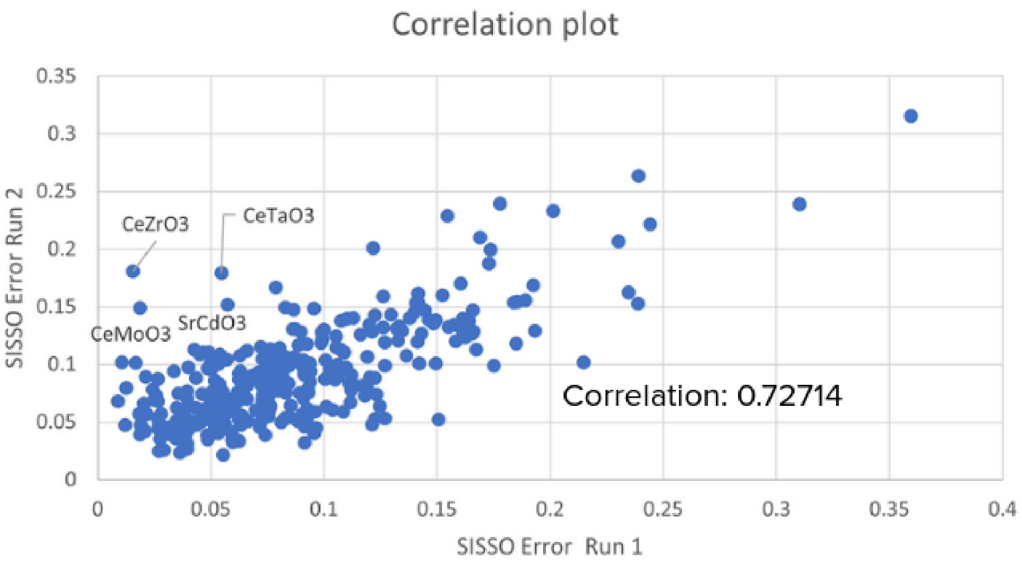


Figure 4.5: Correlation plot showing Pearson correlation between two SISSO models as well as the outlier materials.

Following the metrics defined, the correlation between the models should now be studied. Pearson correlation is used to study the relation between the two SISSO models developed, as seen in the correlation plot in Figure 4.5 The correlation reflects if the two models are in relatively good agreement. If it is low then they are separate models. This check is used to see if the iterative approach leads to a separate local model or not. Given the relatively high correlation of 0.72, it is a good indicator that the two models are in some agreement. This is a good indicator that the two SISSO models that were found, have close correlation in describing the material property in question, bulk modulus and are consistent between the first dataset and the reduced dataset.

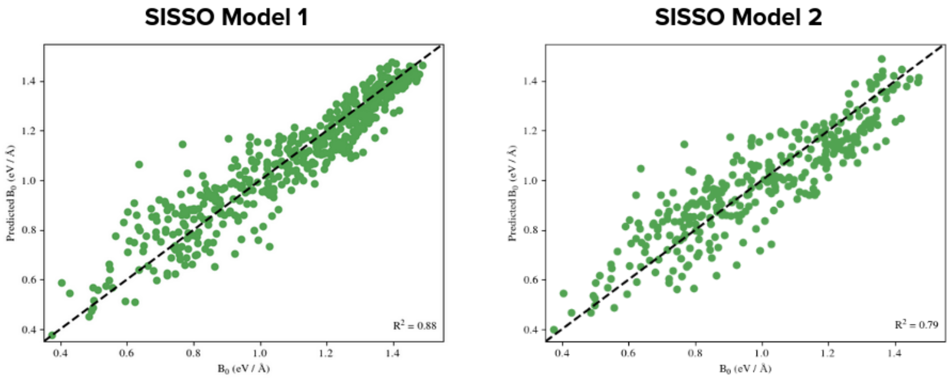


Figure 4.6: Parity plots for the two SISSO models

Parity plots also allow us to study the SISSO machine learning plot for the models, and show us the performance of the machine learning model through comparing the differences between predicted and actual bulk modulus, as well as their deviation from the prediction. From the parity plots in Figure 4.6, it becomes evident and is logical that the second model, which is trained on the reduced dataset with low error materials removed, has more deviation between the predicted and actual values of bulk modulus. This ties into the hypothesis of only one model to be found, as the training to predict bulk modulus performed best on the first model with the original dataset, and the second model SISSO found shows poorer performance as seen by the scatter of the datapoints.

The initial results indicate to us that there is only one global model discoverable, the reason for the choice of training on oxides to develop a model, were mainly the availability of the oxide dataset as well as to investigate if a local subgroup discovered could be applicable to another chemically different group of Perovskites (Halides in this case). However, the initial results of which the subgroup discovered was unable to identify any possible candidates for the next round of the workflow.

Hence, following this development in the work, the datasets are expanded using both sulfide and halide atoms, to see the effect the C atom (for a perovskite ABC_3) will have on the overall models developed, and to improve the generalizability of the model.

4.2 Second Active Workflow cycle

4.2.1 Sulfide Oxide Dataset

In order to determine how extendable the model is to sulfides, a sulfide-oxide mix of dataset was chosen in order to have a bridge between the sulfides and the oxides. The atoms in the f-block of the periodic table are avoided in order to prevent any possible inaccuries performing FHI-aims calculations. The procedure was carried out using the subgroup discovered with coverage 0.3 (in Figure 4.1), with samples randomly selected, 30 samples inside of the first subgroup and 70 are outside of it, leading to a total of 100 sulfide perovskites. This is done to prevent a bias in selection of the sulfides simulated. This dataset includes the 504 oxide perovskites from the original dataset, with the additional 100 sulfide perovskites that are added in.

To evaluate the SISSO model for this sulfide-oxide dataset, absolute error plots as well as the validation RMSE are plotted as seen in Figure 4.7. In this case can be seen that the model dimension 5 has the lowest validation RMSE, with a much clearer indication of convergence toward the fifth dimension. The mean and median of the absolute error for the fifth dimension can also be seen as lower, giving us a good indication of where the optimum model can be found.

The first set of subgroups discovered are detailed in Table 4.3, and it can be seen that there are fewer parameters involved in the conditions for the subgroup. Interesting to note is the influence of the elemental charge of the C atom (<=12) in the subgroup, which

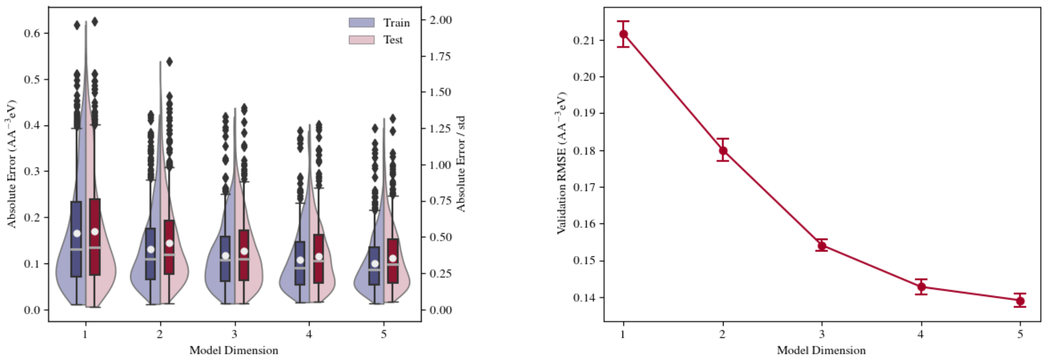


Figure 4.7: Distribution of absolute errors and RMSE for the Oxide-Sulfide dataset. The model dimension with the lowest RMSE is model 5, the Validation RMSE decreases with increasing model dimension.

Coverage 0.3		
Property	Condition	Value
Higher Occupied Molecular Orbital of A atom	$<$	-2.90100 eV
Lower Unoccupied Molecular Orbital of B atom	$>=$	-4.68356 eV
Elemental charge of C atom	$<$	12
Coverage 0.5		
Higher Occupied Molecular Orbital of A atom	$<$	-2.90100 eV
Elemental charge of C atom	$<=$	12
Coverage 1.0		
Ac	$<$	2.5
Elemental charge of C atom	$<=$	12

Table 4.3: First set of subgroups discovered on the new dataset

shows that there is some significance of the C atom in the SGD process, and limits us here to only oxides. Here the elemental charge condition tells us that the model still works best in predicting bulk modulus for oxide perovskites. This is logical given that the model is trained mostly on oxide perovskites, however it is still noteworthy as this shows that the inclusion of the sulfide perovskites in the dataset does not greatly impact the model.

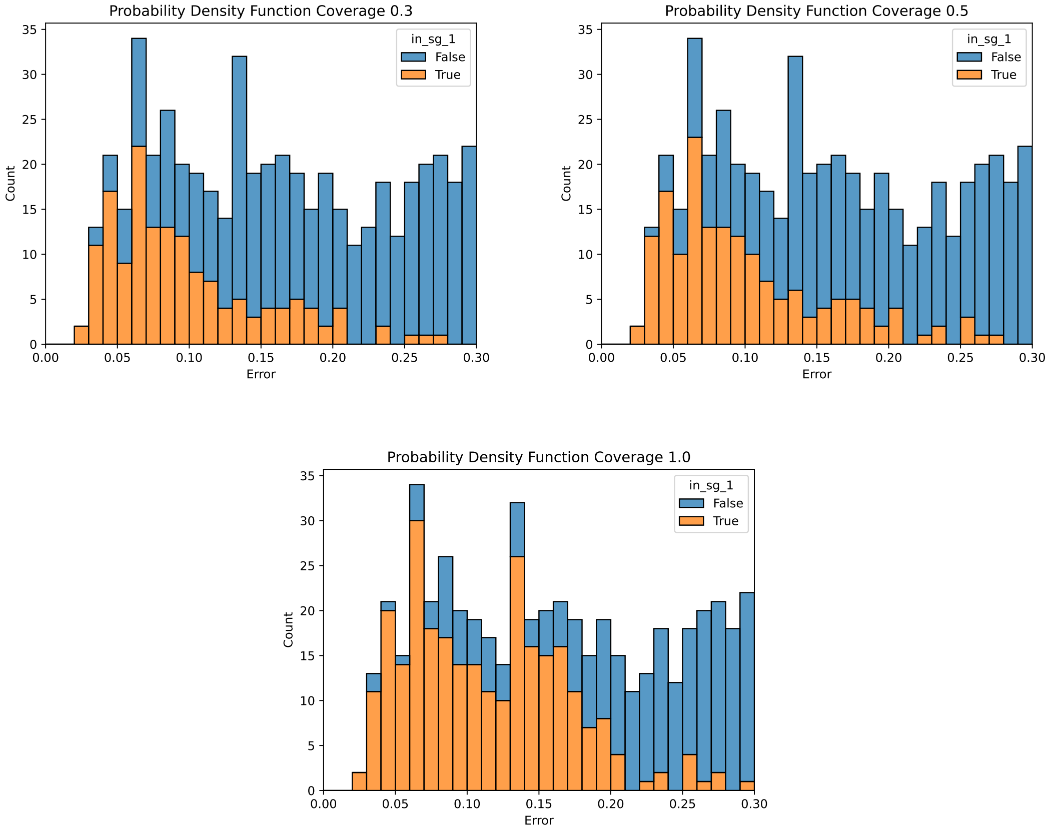


Figure 4.8: Probability Density Function for the subgroup with Oxide-Sulfide Perovskites discovered inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0.

The new first subgroup is now plotted in Figure 4.8 with the probability density function to show the distribution and spread of errors between the oxide perovskites inside and outside the subgroup. The separation of low and high error materials is better, as evidenced by mostly orange in the histograms for lower errors, with few or no high error materials in the distribution. It is also worth noting that the size of the subgroup fits our metric. The coverage parameter was again increased in order to see the affect it has on the distribution, and in this case, we observe minor differences in the separation of distribution of low error materials from high error ones (outside of the subgroup), as we increase the coverage from 0.3 to 1.0. This indicates stability in the subgroup despite changing the coverage parameter.

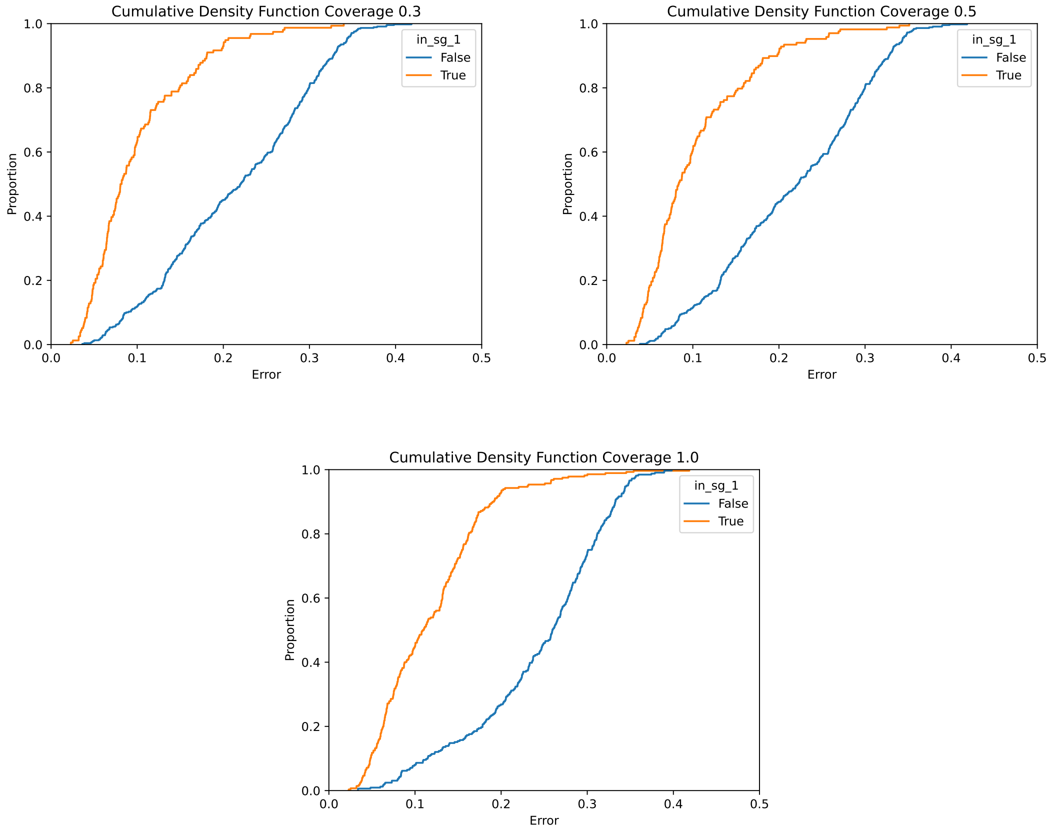


Figure 4.9: Cumulative Density Function for the second subgroup of materials discovered inside (orange) and outside (blue) the subgroup, with coverage parameter adjusted from 0.3 to 1.0.

The cumulative density function in Figure 4.9 shows us the separation of the dataset between those in the subgroup and outside of it, and in the case of the second subgroup, this difference is more starkly contrasted as the coverage parameter is increased. Here we notice that between the coverage of 0.3 to 1.0, there is no significant change in the separation. This indicates that the subgroups are stable with respect to the change in coverage parameter. The separation of the subgroups in this case is the best that we have seen.

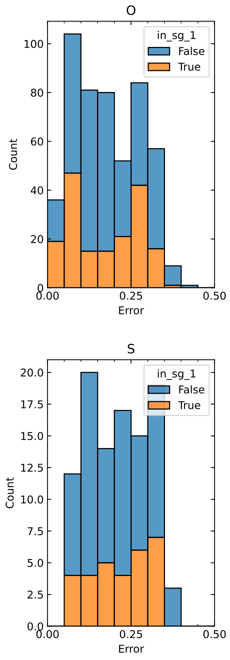


Figure 4.10: Proportion of C atom in the subgroup

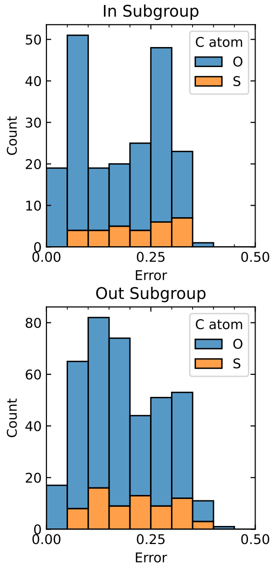


Figure 4.11: Distribution of C atom in the subgroup

From the histograms in Figures 4.10 and 4.11, the proportion of the C atom (Oxygen or Sulfur) can be seen relative to the entire dataset, in terms of in or out of the subgroup found. As the dataset is mainly trained on oxide perovskites, it is logical that the majority of perovskites that are in the subgroup with lower error would be oxide perovskites. It is interesting to note however that the proportion (rather than the sheer count) of sulfide perovskites relative to the number of sulfide perovskites in the dataset, is similar to that of the oxides.

4.2.2 Sulfide-Halide-Oxide Dataset

For the second expanded dataset, a sulfide-halide-oxide mix was chosen to see how this combination would then influence the finding of a global model, and the eventual first subgroup discovered. This dataset includes the 504 oxide perovskites from the original dataset, with an additional 100 sulfide perovskites that are added in, and finally the 50 halide perovskites that are included.

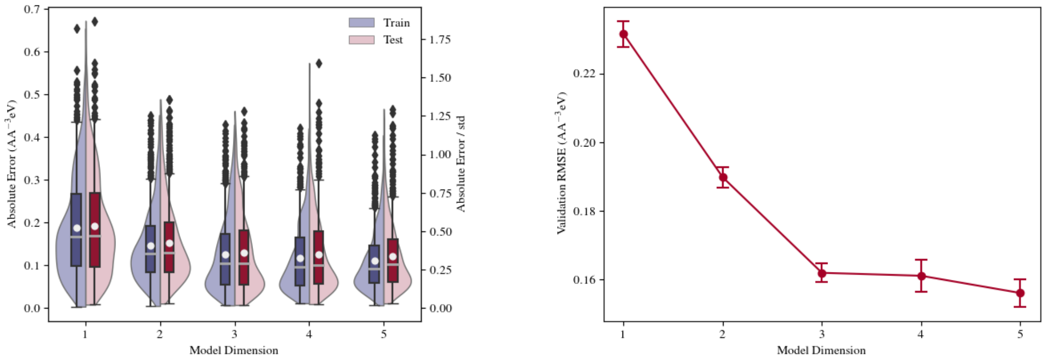


Figure 4.12: Distribution of absolute errors and RMSE for the Sulfide-Halide-Oxide dataset. The model dimension with the lowest RMSE is model 5, the Validation RMSE decreases with increasing model dimension.

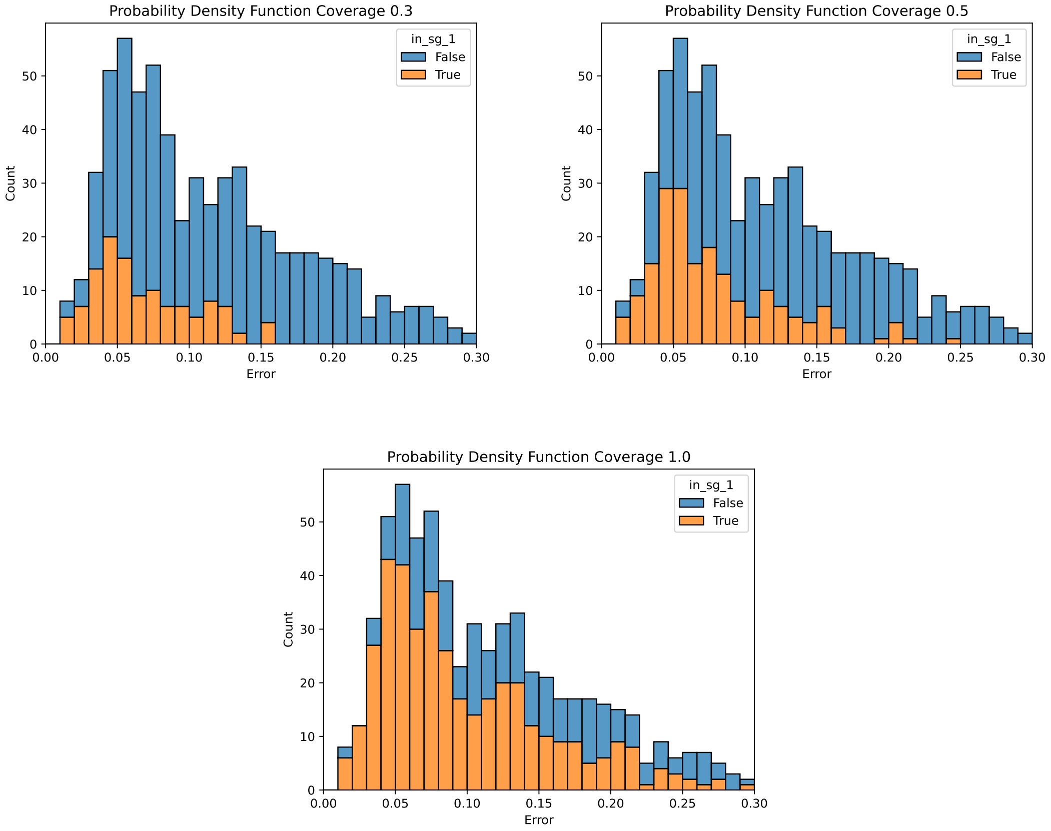
The absolute error plots as well as the validation RMSE are plotted again in Figure 4.12 to evaluate the new model. In this case for the Sulfide-Halide-Oxide perovskites dataset, it can be seen that the model dimension 5 has the lowest validation RMSE, with a clear indication of convergence toward the fifth dimension. The mean and median of the absolute error for the fifth dimension can also be seen as lower, thus the SISSO model of the fifth dimension was chosen to perform subgroup discovery.

The subgroups found for the new sulfide-halide perovskite dataset are listed in Table 4.4. It is interesting to note that the parameters involving the C atom are not predominant in

Coverage 0.3		
Property	Condition	Value
Ionization Potential of A atom	>=	5.47762 eV
Radius (r_{val}) of A atom	<=	1.87700 Å
Electron affinity of B atom	<	0 eV
Ionic Potential of B atom	<=	8.59751 eV
Radius(r_{val}) of B atom	<=	0.975250 Å
Radius(r_{val}) of C atom	<=	0.756950 Å
Coverage 0.5		
Ionization Potential of A atom	<=	7.34643 eV
Elemental charge of A atom: A(Z)	>=	15.5
Electron affinity of B atom	<	0 eV
Ionic Potential of B atom	<=	8.59751 eV
Radius(r_{val}) of B atom	<=	0.975250 Å
Coverage 1.0		
Electron affinity of A atom	>	0 eV
Radius(r_{val}) of B atom	<=	0.975250 Å

Table 4.4: First subgroups discovered on new dataset with varying coverage parameter

the subgroup selection space, perhaps indicating less effect of this choice on the dataset. Another recurrent condition, is the influence of the radius of the B atom (rval), which limits the dataset again, to B atoms in the area of transition metals. Also, the subgroups are based mainly on oxides.


Probability Density Function Coverage 0.3

Probability Density Function Coverage 0.5

Probability Density Function Coverage 1.0

Figure 4.13: Probability Density Function for the subgroup with Sulfide-Halide-Oxide Perovskites discovered with varying coverage parameters

The new first subgroup is now plotted in Figure 4.13, with the probability density function to show the distribution and spread of errors between the oxide perovskites inside and outside the subgroup. Here it can be observed that there is a good size of the subgroup, as well as good separation of high and low error materials (as indicated by the orange bar in the histogram). This is true even for higher coverage at 1.0. The coverage parameter was again increased in order to see the affect it has on the distribution, and in this case, we observe minor differences in the separation of distribution of low error materials from high error ones (outside of the subgroup), as we increase the coverage from 0.3 to 1.0. This indicates stability in the subgroup.

The cumulative density function plotted in Figure 4.14 shows low overlap between the two cumulative error distributions. It is also noteworthy that between the coverage of 0.3 to 1.0, there is no significant change in this overlap. This indicates stability in the subgroup found.

From the histograms in Figure 4.15 and 4.16, the proportion of the C atom (Sulfur, Oxygen or Halide atom) can be seen relative to the entire dataset, in terms of in or out of the subgroup found. Interestingly, the halide atom subset of the data is entirely out of the subgroup found, indicating higher errors for the predicted values of Bulk modulus for the halide perovskites. This is reduced in the sulfide perovskites, with a proportion of them being in the subgroup found. This observation is unsurprising, as sulfide perovskites can be seen to serve as an inbetween to oxide and halide perovskites in terms of chemistry. Given that the model is trained largely on oxide perovskites, it is logical then that the halide perovskite data is not included in the subgroup discovered.

Through our tests of expanding the generalizability of the dataset, the results indicated that the model is influenced by the majority perovskite trained on (in this case oxides), and introduction of new perovskite types has little influence on the overall model as well as the subgroups discovered. Hence, the workflow was halted here, and a second round of SGD was not carried out, given the conclusions that the introduction of the new perovskites in the dataset proved not to aid in the model prediction or provide more interesting local models from SGD.

Cumulative Density Function Coverage 0.3

Cumulative Density Function Coverage 0.5

Cumulative Density Function Coverage 1.0

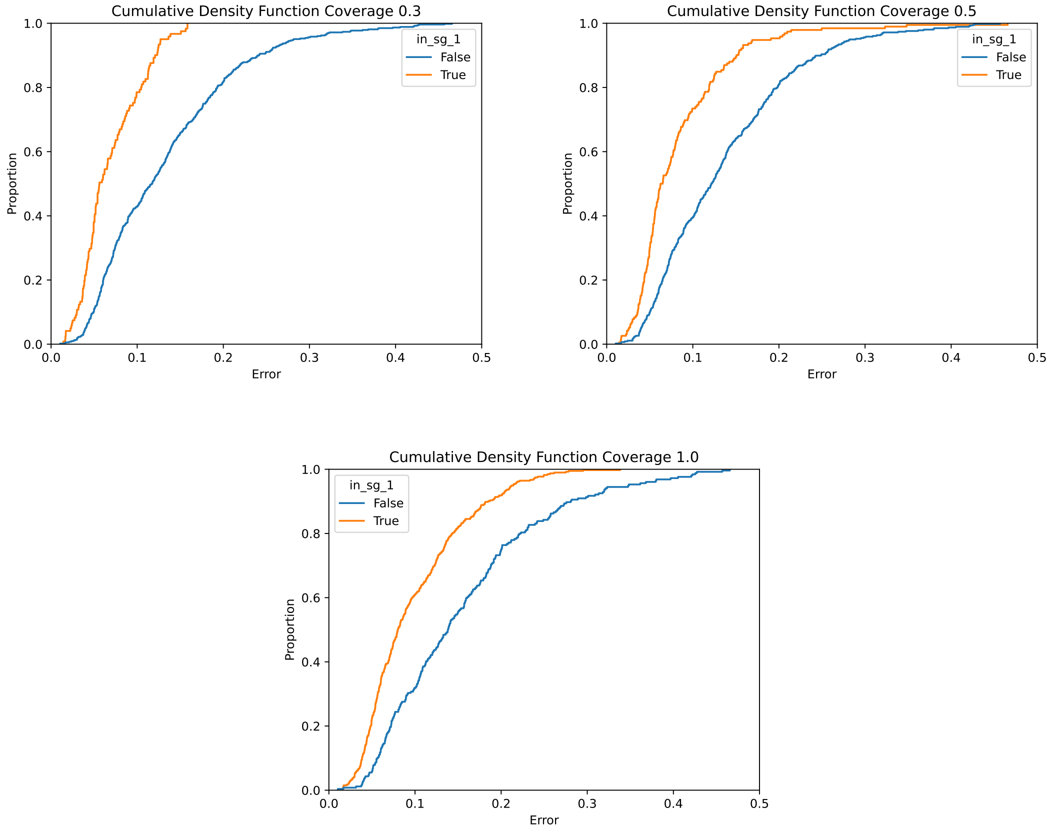


Figure 4.14: Cumulative Density Function for the second subgroup of materials discovered with varying coverage parameters

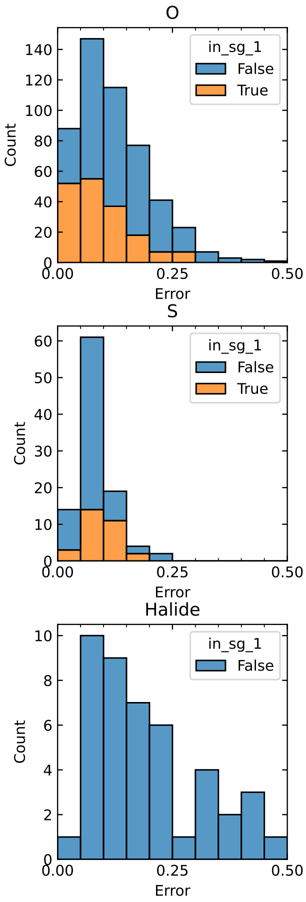


Figure 4.15: Proportion of C atom in the subgroup (coverage 0.3)

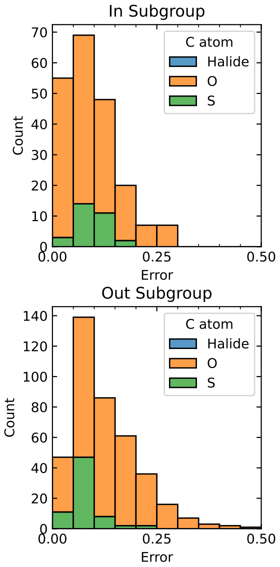


Figure 4.16: Distribution of C atom in the subgroup (coverage 0.3)

5 Conclusion and outlook

The proposal for this work was to develop an active workflow that develops a global machine learning model for predicting bulk modulus using the Sure-independence screening and sparsifying operator method (SISSO), while remaining locally relevant with a small number of local models found using subgroup discovery, as opposed to the typical approach of Bayesian Optimization with GPR, which can result in a model which may not be generalized beyond the specific solution.

To evaluate our workflow, metrics were developed which involved the size of the subgroup discovered, the correlation between the SISSO models, and the distribution of materials with low and high error. These were evaluated with probability density function and cumulative density function graphs. The initial results, with the parity plots (Figure 4.5) showing poorer performance of the second SISSO model, as well as the high correlation number in the correlation plot (Figure 4.6) indicate to us that there is only one global model discoverable, and the subgroup discovered was unable to identify any possible candidates for the next round of the workflow. The datasets were then expanded using both sulfide and halide atoms, to see the effect the C atom (for a perovskite ABC_3) would have on the overall models developed, and to improve the generalizability of the model. This inclusion of sulfide and halide atoms, to make the model more generalisable, resulted in some indication there is some influence of the C atom, however from our results it did not significantly influence the SGD process.

Noteworthy conclusions to be drawn are firstly the parameters involving the C atom are not predominant in the subgroup selection space, perhaps indicating less effect of this choice on the dataset. Another recurrent condition, is the influence of the radius of the B atom (rval), which limits the dataset to having perovskites with B atoms in the area of transition metals. Lastly, the models were found to work best for oxide perovskites, which is a logical conclusion given the training of the model on a dataset containing mainly oxide perovskites.

The SGD in our work was still unable to separate the materials between those with low and high error completely, as shown in the probability distribution function curves plotted. One possibility is an inherent problem with focusing on low errors, meaning that every model will have some distribution of errors, and which samples have a low error may just be random and not systematic. Another possibility is symmetric wells in the input feature space as within the A and B elements in the perovskites across the periodic table there are symmetries in elemental features, resulting in 'symmetry wells' where the material feature in question is the same. These could both impair the SGD process and the results obtained.

The active workflow developed, while having shortcomings, is a good first step as providing an alternative to the traditional Bayesian method, and allows for more room in the process for studying the workflow as opposed to a 'black-box' approach. For the future development of the active workflow, based off the initial conclusions set, firstly, different metrics (other than negative median shift) can be experimented with and tabulated, in order to investigate the effect of this metric and SGD, as well as to see if some metrics are able to provide complete separation of high and low error materials. Further, given the good performance of the global model on B atoms with transition metals can be further studied to understand the reasons behind this phenomenon, and also to improve the accuracy of the workflow. Lastly, a final test of the C atom should be carried out with sulfide perovskite datasets of a higher proportion than oxide perovskites, to give a conclusion as to whether the C atom significantly affects the SGD results.

Bibliography

[1] Kirstin Alberi et al. “The 2019 materials by design roadmap”. In: *Journal of Physics D: Applied Physics* 52.1 (2018), p. 013001.

[2] Helge S Stein and John M Gregoire. “Progress and prospects for accelerating materials science with automated and autonomous workflows”. In: *Chemical science* 10.42 (2019), pp. 9640–9649.

[3] Mitsutaro Umehara et al. “Analyzing machine learning models to accelerate generation of fundamental materials insights”. In: *npj Computational Materials* 5.1 (2019), pp. 1–9.

[4] Sarat Kumar Sarvepalli. “Deep learning in neural networks: the science behind an artificial brain”. In: *Liverpool Hope University, Liverpool* (2015).

[5] Turab Lookman et al. “Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design”. In: *npj Computational Materials* 5.1 (2019), pp. 1–17.

[6] Ankit Agrawal and Alok Choudhary. “Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science”. In: *Apl Materials* 4.5 (2016), p. 053208.

[7] Claudia Draxl and Matthias Scheffler. “The NOMAD laboratory: from data sharing to artificial intelligence”. In: *Journal of Physics: Materials* 2.3 (2019), p. 036001.

[8] Valentin Stanev et al. “Machine learning modeling of superconducting critical temperature”. In: *npj Computational Materials* 4.1 (2018), pp. 1–14.

[9] Burr Settles. “Active learning literature survey”. In: (2009).

[10] Yuan Tian et al. “Role of uncertainty estimation in accelerating materials development via active learning”. In: *Journal of Applied Physics* 128.1 (2020), p. 014103.

[11] Eric Brochu, Vlad M Cora, and Nando De Freitas. “A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning”. In: *arXiv preprint arXiv:1012.2599* (2010).

[12] Peter I Frazier. “Bayesian optimization”. In: *Recent Advances in Optimization and Modeling of Contemporary Problems*. INFORMS, 2018, pp. 255–278.

[13] Carl Edward Rasmussen and C Williams. “Gaussian processes for machine learning the mit press”. In: *Cambridge, MA* 32 (2006), p. 68.

[14] P Hohenberg and WJPR Kohn. “Density functional theory (DFT)”. In: *Phys. Rev* 136 (1964), B864.

[15] Walter Kohn and Lu Jeu Sham. “Self-consistent equations including exchange and correlation effects”. In: *Physical review* 140.4A (1965), A1133.

[16] Volker Blum et al. “Ab initio molecular simulations with numeric atom-centered orbitals”. In: *Computer Physics Communications* 180.11 (2009), pp. 2175–2196.

[17] Ernest R Davidson and David Feller. “Basis set selection for molecular calculations”. In: *Chemical Reviews* 86.4 (1986), pp. 681–696.

[18] Jianqing Fan and Jinchi Lv. “Sure independence screening for ultrahigh dimensional feature space”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.5 (2008), pp. 849–911.

[19] Runhai Ouyang et al. “SISSO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates”. In: *Physical Review Materials* 2.8 (2018), p. 083802.

[20] Christopher Sutton et al. "Identifying domains of applicability of machine learning models for materials science". In: *Nature communications* 11.1 (2020), pp. 1–9.

[21] Mario Boley et al. "Creedo—scalable and repeatable extrinsic evaluation for pattern discovery systems by online user studies". In: *KDD 2015 Workshop on Interactive Data Exploration and Analytics (IDEA'15), Sydney, Australia*. 2015.

[22] Stefano Curtarolo et al. "AFLOW: An automatic framework for high-throughput materials discovery". In: *Computational Materials Science* 58 (2012), pp. 218–226.

[23] David Hicks et al. "AFLOW-XtalFinder: a reliable choice to identify crystalline prototypes". In: *npj Computational Materials* 7.1 (2021), pp. 1–20.

[24] Maja-Olivia Lenz et al. "Parametrically constrained geometry relaxations for high-throughput materials science". In: *npj Computational Materials* 5.1 (2019), pp. 1–10.

[25] AS Verma and A Kumar. "Bulk modulus of cubic perovskites". In: *Journal of alloys and compounds* 541 (2012), pp. 210–214.

6 Annex

6.1 Code used for fitting

The code used for fitting of the Birch-Murnaghan equation for the bulk modulus calculations is given below in Listing 6.1.

```
#!/usr/bin/python3.7

"""
The parameters in the Birch-Murnaghan equation of state

  E(V) = E0 + B0*V/B’ * [(V0/V)^B’ / (B’-1) + 1] - B0 V0 / (B’-1)

are fit to energies for different lattice constants.

  E0: Energy in equilibrium
  V0: Volume in equilibrium
  B0: Bulk modulus in equilibrium
      B := - V (dP/dV)_T
      P := - (dE/dV)_S
  B’: Pressure derivative of B  (assumed constant in this model)
      B’ := (dB/dP)_T

See also: http://en.wikipedia.org/wiki/Birch-Murnaghan_equation_of_state
"""
```

```python

import sys

import numpy as np
from scipy.optimize import fmin_powell

USAGE = """%prog [options] <input_file>

The <input_file> should contain two columns, the first containing the volume
and the second total energies.

No unit conversions are done. If you have used Angstroms^3 and eV in the input
file, the bulk modulus is returned in eV/Anstrom^3.  You can convert it using
GNU units:

  units "<B0> eV/angstrom^3" "GPa"\
"""

def Birch_Murnaghan(paras, V):
    """Evaluate Birch Murnaghan equation-of-state for [E0, V0, B0, B’]"""
    E0, V0, B0, Bprime = paras
    T1a = B0*V/Bprime
    T1b = ((V0/V)**Bprime / (Bprime-1.) + 1.)
    T1 = T1a * T1b
    T2 = - B0 * V0 / (Bprime - 1)
    E = E0 + T1 + T2
    return E

def sum_of_squares(paras, Vs, Es):
    """Calculate squared norm of residue vector for paras."""
    E_BMs = Birch_Murnaghan(paras, Vs)
    return np.sum((Es - E_BMs)**2)


def initial_guess(Vs, Es):
    """Construct initial guess for V0, E0, ..."""
    n = len(Es)
    assert n == len(Vs)
    if n < 3:
        parser.error("Need at least three energies for fit")  # 4?

    E0 = np.min(Es)
    i0 = np.argmin(Es)   # index of minimum
    V0 = Vs[i0]
    if i0 > 0:
        dE_left = (Es[i0] - Es[i0-1]) / (Vs[i0] - Vs[i0-1])
        V_left = (Vs[i0] + Vs[i0-1]) / 2.
    else:
        dE_left = 0.
        V_left = Vs[i0]
    if i0 < n-1:
        dE_right = (Es[i0+1] - Es[i0]) / (Vs[i0+1] - Vs[i0])
        V_right = (Vs[i0+1] + Vs[i0]) / 2.
    else:
        dE_right = 0.
        V_right = Vs[i0]
    ddE = (dE_left - dE_right) / (V_left - V_right)
    B0 = ddE * V0
    Bprime = 3.5  # Typical value according to [1]
    return np.array([E0, V0, B0, Bprime], dtype=’float’)


def get_default_VV(V0, Vs):
    """Get default set of lattice constants aa for plotting"""
    Vmin = min(np.min(Vs), V0)
    Vmax = max(np.max(Vs), V0)
    Vrange = Vmax - Vmin
    VV = np.linspace(Vmin - 0.15*Vrange, Vmax + 0.15*Vrange, 200)
    return VV

def plot(VV, EE, Vs, Es, V_text="$V$ in [a^3]"):
```

```python
    import matplotlib as mpl
    import matplotlib.pyplot as plt
    fig = plt.figure()
    ax = fig.add_subplot(1,1,1)

    # Plot input data points
    ax.plot(Vs, Es, ’o’, label="Calculated energies")

    # Plot Birch-Murnaghan equation of states
    ax.plot(VV, EE, ’-’, label="Birch-Murnaghan fit")

    # Decorate and plot
    ax.legend()
    ax.set_xlabel(V_text)
    ax.set_ylabel("$E$ in [E]")
    def on_q_exit(event):
        if event.key == "q": sys.exit(0)
    plt.connect(’key_press_event’, on_q_exit)
    plt.show()


def main():
    import optparse

    # Parse command line
    parser = optparse.OptionParser(usage=USAGE)
    parser.add_option("-p", "--plot", action="store_true",
                      help="Plot E(a) curve using matplotlib")
    parser.add_option("-i", "--info", action="store_true",
                      help="Output info about Birch-Murnaghan e.o.s. & quit")
    parser.add_option("-r", "--range", action="store", nargs=3, type="float",
                      help="Output range <start> <stop> <nstep>")
    parser.add_option("-o", "--output", "--datafile", action="store",
                      help="Output fitted curve to OUTPUT")
    parser.add_option("-e", "--reference", action="store", type="float",
                      help="Reference energy", default=0.)
    parser.add_option("-l", "--lattice-constant", metavar="FACTOR",
                      help="""\
Assume <input_file> contains lattice constants a instead of volumes V
and calculate V=FACTOR*a^3. Use, e.g., FACTOR=0.25 for fcc.""")

    options, args = parser.parse_args()
    if options.info:
        print(__doc__)
        sys.exit(0)
    if len(args) != 1:
        parser.error("Need exactly one argument")
    input_file_name = args[0]

    use_lattconst = options.lattice_constant is not None
    if use_lattconst:
        unit_cell_frac = float(options.lattice_constant)

    # Read input data
    data = np.loadtxt(input_file_name)   # Input data as array
    if use_lattconst:
        lattconsts, Es = data.T              # Separate 1st and 2nd col
        Vs = unit_cell_frac * lattconsts**3
    else:
        Vs, Es = data.T              # Separate 1st and 2nd col
    Es -= options.reference

    # Sort input data by V
    VEs = sorted([list(zip(Vs, Es))])          # List of (V, E) pairs sorted by V
    Vs, Es = np.array(VEs,dtype=’float’).T             # Convert back to numpy arrays

    # Construct initial guesses
    paras = initial_guess(Vs, Es)

    # Optimize parameters
    paras, fopt, direc, n_iter, n_funcalls, warnflag \
        = fmin_powell(sum_of_squares, paras, (Vs, Es), full_output=True)
    E0, V0, B0, Bprime = paras

    # Final output
    print("Final residue: %g [E]^2." % fopt)
    print("=== Final parameters (output units match input units)")
    outlist = []
    outlist.append(" E0  = %18.12g [E]" % E0)
    outlist.append(" V0  = %18.12g [a^3]" % V0)
    if use_lattconst:
        a0 = (V0 / unit_cell_frac)**(1./3.)
        outlist.append(" a0  = %18.12g [a]" % a0)
    outlist.append(" B0  = %18.12g [E/a^3]" % B0)
    outlist.append(" B0’ = %18.12g" % Bprime)
    for line in outlist:
        print(line)

    if options.range is None:
        VV = get_default_VV(V0, Vs)
    else:
        start, stop, nstep = options.range
        VV = np.linspace(start, stop, nstep)

    EE = Birch_Murnaghan(paras, VV)


    # Optionally plot
    if options.output is not None:
        f = open(options.output, "w")
        for line in outlist:
            f.write("# %s\n" % line)
        if use_lattconst:
            np.savetxt(f, np.array([aa, VV, EE]).T)
        else:
            np.savetxt(f, np.array([VV, EE]).T)
```

```python
        f.close()
    if options.plot:
        try:
            if use_lattconst:
                aa = (VV / unit_cell_frac)**(1./3.)
                plot(aa, EE, lattconsts, Es, "$a$ in [a]")
            else:
                plot(VV, EE, Vs, Es)
        except ImportError:
            parser.error("Cannot use --plot without matplotlib")

if __name__ == "__main__":
    main()
```

Listing 6.1: Birch-Munaghan fit