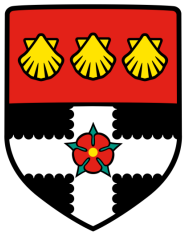

University of Reading

Department of Chemistry

Discovering Thermoelectric Materials with Modern Machine Learning Approaches

Luis M. Antunes

Supervisors: Dr Ricardo Grau-Crespo, Dr Keith Butler

A thesis submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Doctor of Philosophy in *Chemistry*

September 2024

Declaration

I, Luis M. Antunes, of the Department of Chemistry, University of Reading, confirm that this is my own work, and that all figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that failure to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

Luis M. Antunes
September 2024

Abstract

Machine learning is increasingly utilized to accelerate the discovery and design of new materials. Thermoelectrics are an important class of energy materials with the potential to help address pressing environmental challenges. This thesis presents novel machine learning-based methodologies for predicting material properties and generating crystal structures, with a focus on the discovery of new, and more effective, thermoelectric materials. First, a method is introduced for deriving distributed representations of materials solely from their chemical formulas, which demonstrates competitive performance in predicting various properties, such as formation energy and band gap. Next, an attention-based deep learning model is developed to predict thermoelectric transport properties, which incorporates the distributed representations, and proves capable of making useful predictions with a significantly reduced computational cost compared to traditional *ab initio* methods. Finally, a generative model is proposed that is capable of suggesting crystal structures for chemical compositions, which is vital for progressing from estimates of thermoelectric performance from composition, to deeper investigation based on structure. The results from these studies demonstrate the potential for modern machine learning techniques in the field of materials discovery, and particularly for accelerating the discovery of novel thermoelectrics.

Keywords: thermoelectrics, machine learning, large language models

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr Ricardo Grau-Crespo, who has been my unwavering partner on this voyage. His continual guidance, support, and encouragement have been invaluable, especially during the challenging times when things didn't go as planned. His patience and belief in me and my work helped me navigate some rough seas, and kept me on course. I would also like to sincerely thank my second supervisor, Dr Keith Butler, for his insightful feedback and steadfast belief in me and my ideas. His expertise and input have been instrumental in refining my work, and in helping me approach challenges from new perspectives. I owe them immensely for their mentorship, patience, and the profound impact they've had on this thesis and on my personal and professional growth.

I am extremely grateful for having been granted access to the supercomputing facilities, ARCHER/ARCHER2, the UK's national high-performance computing service, via the UK's HPC materials chemistry consortium and Young supercomputer, via the UK's materials and molecular modelling hub.

Finally, I would like to thank my family for their unwavering love, understanding, and endless support throughout this journey. Their encouragement and belief in me gave me the strength to persevere. They too have endured challenges along the way, and for that, I will be forever grateful.

Thank you to everyone who has supported me along this journey.

List of Publications

The work presented in this thesis has been published (or is accepted for publication) in the following articles:

1. **Antunes, L.M.**, Plata, J.J., Powell, A.V., Butler, K.T., Grau-Crespo, R. (2022). "Machine Learning Approaches for Accelerating the Discovery of Thermoelectric Materials." In: *Machine Learning in Materials Informatics: Methods and Applications*, pp. 1-32.

2. **Antunes, L.M.**, Grau-Crespo, R., Butler, K.T. (2022). "Distributed representations of atoms and materials for machine learning." *npj Computational Materials*, **8**(1), 44.

3. **Antunes, L.M.**, Butler, K.T., Grau-Crespo, R. (2023). "Predicting thermoelectric transport properties from composition with attention-based deep learning." *Machine Learning: Science and Technology*, **4**(1), 015037.

4. **Antunes, L.M.**, Butler, K.T., Grau-Crespo, R. (2024). "Crystal Structure Generation with Autoregressive Large Language Modeling." *Nature Communications* (in press). Also available as a pre-print: https://arxiv.org/abs/2307.04340.

Contents

List of Figures

List of Tables

List of Abbreviations

AIRSS	Ab initio Random Structure Searching
ALIGNN	Atomistic Line Graph Neural Network
AMCD	Average Minimum Composition Distance
AMSD	Average Minimum Structure Distance
BERT	Bidirectional Encoder Representations from Transformers
BTE	Boltzmann Transport Equation
CDVAE	Crystal Diffusion Variational Autoencoder
CEP	Crude Estimation of Property
CGCNN	Crystal Graph Convolutional Neural Network
CIF	Crystallographic Information File
CRTA	Constant Relaxation Time Approximation
CSP	Crystal Structure Prediction
DAS	Dual Adaptive Sampling
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DFT	Density Functional Theory
DOS	Density of States
ETI	Electronic-Transport-Informatics (Framework)
GAN	Generative Adversarial Network
GBTR	Gradient Boosting Tree Regression
GGA	Generalized Gradient Approximation
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
HTS	High-Throughput Screening
ICSD	Inorganic Crystal Structure Database
IFC	Inter-atomic Force Constant
LLM	Large Language Model
LSTM	Long Short Term Memory (Network)
MAE	Mean Absolute Error

MCTS	Monte Carlo Tree Search
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MP	Materials Project
MSE	Mean Squared Error
NLP	Natural Language Processing
NMT	Neural Machine Translation
NOMAD	NOvel MAterials Discovery (Database)
OQMD	Open Quantum Materials Database
PAW	Projector Augmented Wave (Method)
PBE	Perdew-Burke-Ernzerhof (Exchange-correlation functional)
PCA	Principal Component Analysis
PF	Power Factor
PUCT	Predictor Upper Confidence bound applied to Trees
RF	Random Forest
RLHF	Reinforcement Learning from Human Feedback
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC-AUC	Receiver Operating Characteristic - Area Under the Curve
SGD	Stochastic Gradient Descent
SMACT	Semiconducting Materials from Analogy and Chemical Theory
t-SNE	t-distributed Stochastic Neighbor Embedding
SOC	Spin-Orbit Coupling
SVD	Singular Value Decomposition
TE	Thermoelectric
UMAP	Uniform Manifold Approximation and Projection
USPEX	Universal Structure Predictor: Evolutionary Xtallography
VASP	Vienna Ab initio Simulation Package
VCA	Virtual Crystal Approximation
XRD	X-ray Diffraction

Chapter 1

Introduction

1.1 Machine Learning Approaches for the Discovery of New Thermoelectrics

The work in this thesis has been motivated by the desire to accelerate the discovery of new thermoelectric materials. While I ended up developing tools that will hopefully be applied beyond this field, the search for novel thermoelectrics exemplifies well the challenges this thesis addresses and the potential of the solutions offered. In this chapter, I give a brief introduction to thermoelectric materials and an overview of how machine learning (ML) approaches are being used in the field.

Thermoelectric materials are solids with a combination of thermal and electrical properties that allow them to be used for devices that convert heat into electricity, or that cool surfaces with an input of electrical power. Thermoelectric devices are attractive for electricity generation and for refrigeration applications due to their stability (absence of mobile parts) and reliability (little need for maintenance). But in order for them to be more widely adopted, their conversion efficiency must be increased from current levels [1].

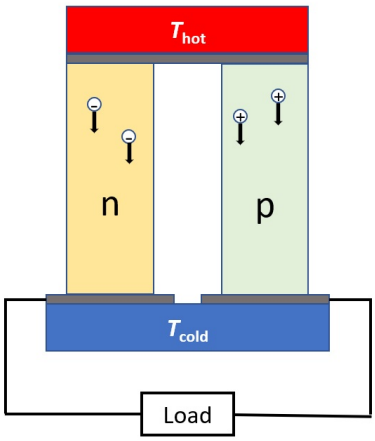

Figure 1.1: Scheme of a thermoelectric couple comprising an n-type and a p-type semiconductor.

A thermoelectric couple (Figure 1.1) comprises two semiconducting materials, one with n-type and one with p-type conductivity, assembled between a heat source at temperature T_{hot} and a heat sink at temperature T_{cold}. A thermoelectric module consists of an array

of these couples connected electrically in series and thermally in parallel. The efficiency of a thermoelectric generator, which contains the module plus ancillary components and associated electronics, depends strongly on the temperature difference $T_{\rm hot} - T_{\rm cold}$ and on the physical properties of the semiconducting materials in the thermoelectric couples. The latter are usually summarised in the figure of merit for the thermoelectric material:

$$zT = \frac{S^2 \sigma T}{\kappa}, \tag{1.1}$$

where S is the Seebeck coefficient or thermopower, which measures the voltage created per unit of temperature difference across the material; σ is the electrical conductivity; T is the absolute temperature; and κ is the thermal conductivity, which contains two main contributions: the lattice thermal conductivity $\kappa_{\rm latt}$ due to crystal vibrations, and the electronic thermal conductivity $\kappa_{\rm elec}$ due to heat-carrying diffusion of electrons in the solid. The higher the dimensionless figure of merit zT, the more efficient the material is in a thermoelectric device. That means that good thermoelectric materials must exhibit a large Seebeck coefficient, good electrical conductivity, but low thermal conductivity.

The search for high-zT materials is complicated by the fact that the transport properties in zT are interdependent. Metals have very good electrical conductivity, but generally (not always) [2] very poor Seebeck coefficients. Some electrically insulating materials, with wide electronic band gaps, exhibit large Seebeck coefficients, but the poor electrical conductivity prevents their use as thermoelectrics. On balance, the best thermoelectric materials are usually semiconductors, with charge carrier concentrations in the order of $\sim 10^{20}$ cm^{-3}. But even within semiconductors, the interdependence of the coefficients in zT means that it is difficult to find optimal materials. One of the most studied and used thermoelectric materials is Bi_2Te_3, which has a zT of around 1 at room temperature, and has been used in Peltier coolers for decades [3]. At high temperatures ($\sim$1000 K or above), Si-Ge alloys exhibit some of the highest values of zT [4, 5], and radioisotope thermoelectric generators based on these alloys have been used since the 1970's in NASA missions for space exploration [6]. But despite significant advances in our understanding of thermoelectric behavior in recent years, this has not yet translated into the development of new materials with widespread commercial applications.

Computer simulations have been widely used to improve the understanding of thermoelectric behavior, particularly because of the development of modern software for modeling electron (*e.g.* BoltzTraP [7]) and phonon transport (*e.g.* ShengBTE [8]) from first principles, based on solutions for Boltzmann's transport equation (BTE) for electrons and phonons, respectively. Traditionally, thermoelectric research has been driven by experiments, and simulations have been performed to rationalize observations. But given the cost of synthesizing materials and measuring the relevant transport properties, and the advances in modeling algorithms and computing hardware, computer simulations are increasingly playing a bigger role in the exploration of the vast chemical space of semiconductors for the discovery of new thermoelectric materials [9]. Central to this theoretical effort is the ability to perform predictions of transport properties in a high-throughput fashion. However, standard physics-based predictions of electronic and transport properties, while highly successful in rationalising thermoelectric behavior, tend to have too high a computational cost to allow for efficient exploration of chemical space.

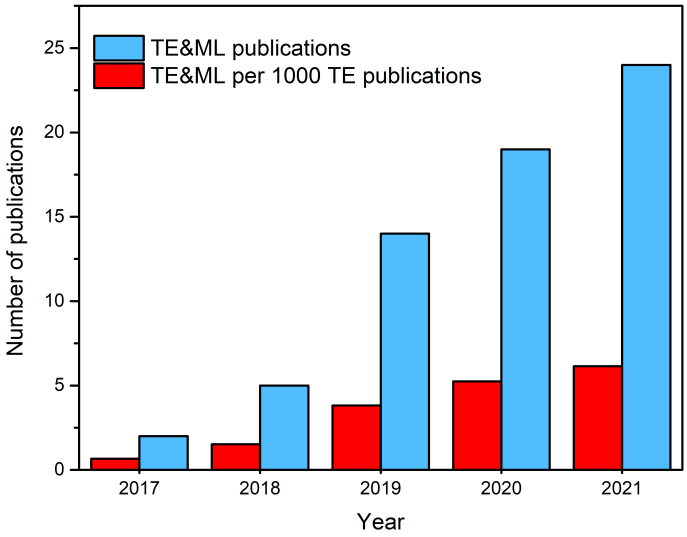

Figure 1.2: Blue bars: Papers including keywords "thermoelectric" (TE) *and* "machine learning" (ML) in the title or abstract, published in the last five years. Red bars: Number of those papers per thousand of papers including keyword "thermoelectric" in the title or abstract. Source: Web of Science.

This Introduction chapter provides an overview of the ML techniques that are being increasingly applied to accelerate the investigation and discovery of thermoelectric materials. ML techniques can use existing data to fit or train a statistical model, which can be then used to predict physical properties of compounds beyond the training set. Because ML allows bypassing the need for computationally expensive physics-based calculations, the search for molecules or materials with desired target properties can be vastly accelerated [10]. The application of ML to thermoelectric research is a nascent but rapidly expanding field, as illustrated in Fig. 1.2, and I will not provide here an exhaustive review of all contributions so far. I will instead highlight key developments and focus on how ML techniques can help accelerate the prediction of these properties, either by targeting the direct prediction of transport coefficients, or by accelerating the calculation of lower-level quantities that determine those coefficients. I will then summarize the current challenges and my perspective on the field.

1.1.1 Datasets for Machine Learning

The development of ML models for the prediction of thermoelectric transport properties requires the pre-existence of appropriate databases. A number of datasets focusing on thermoelectric properties have been shared publicly over the last decade. The datasets comprise a collection of stoichiometric and non-stoichiometric compounds and various corresponding physical properties relevant to thermoelectricity, such as electrical conductivity, the Seebeck coefficient, and lattice thermal conductivity at various temperatures. Some of the datasets are derived from computations based on theoretical methods, such as DFT-BTE, while others are derived from experimental measurements reported in the literature. Generally, these datasets range in size from 10^2 to 10^4 compounds. Table 1.1 lists datasets that can be used for ML-based prediction of thermoelectric properties.

Table 1.1: A list of publicly available datasets of thermoelectric properties that can be used for machine learning. Properties: E: Electrical conductivity; R: Electrical resistivity; S: Seebeck coefficient; P: Power factor; F: Thermoelectric figure of merit (zT); C: Semi-empirical lattice thermal conductivity; M: Semi-empirical intrinsic charge carrier mobility; O: Electron and hole effective masses; D: Density of states (DOS) effective mass; H: Electronic thermal conductivity; I: Ionic conductivity; L: Lattice thermal conductivity; T: Total thermal conductivity.

Dataset	Year	References	Source	Compounds	Properties
Wang *et al.*	2011	11, 12	theory	2,585	P, O
UCSB	2013	13, 14	experiment	282	E,R,S,P,F
Carrete *et al.*	2014	15, 16	theory	450	L
TE Design Lab	2016	17, 18	theory	2,701	C,M,D
Ricci *et al.*	2017	19, 20	theory	47,737	E,S,H
Xi *et al.*	2018	21, 22	theory	161	P
Chen *et al.*	2019	23, 24	experiment	100	L
Starrydata2	2019	25, 26	experiment	434	E,S,T
JARVIS-DFT	2020	27, 28	theory	21,900	E,S,P
Priya *et al.*	2021	29, 30	experiment	585	I
Jaafreh *et al.*	2021	31, 32	theory	119	L
Miyazaki *et al.*	2021	33, 34	theory	143	L
MIP-3d	2021	35, 36	theory	4,400	E, S
Tranås *et al.*	2022	37, 38	theory	122	L

An early dataset consisting of power factors computed from theoretically-derived electronic transport coefficients was described by Wang and co-workers in 2011, who computed the power factors for over 2,500 nanograined, sintered-powder materials from the AFLOWLIB database [11, 39]. The authors avoid invoking the CRTA by assuming the compounds exist as a sintered powder, which enables the derivation of the Seebeck coefficient and electrical conductivity from a simple and physically sound model that is based on the constant-mean-free-path approximation. A regression analysis reveals that the power factor is positively correlated with the band gap and the carrier effective mass, and that larger power factors are associated with larger numbers of atoms per primitive cell.

Perhaps the first reported large dataset containing electronic transport coefficients was curated by Gaultois and co-workers in 2013, and is often referred to as the UCSB (or UCSB-MRL) dataset, after the authors' affiliation (Materials Research Laboratory at University of California Santa Barbara) [13]. The database contains Seebeck coefficients, electrical conductivities and other data for 282 distinct stoichiometric and non-stoichiometric compounds at various temperatures (300 K, 400 K, 700 K, and 1000 K). The data were obtained from over 100 publications reporting experimental measurements, resulting in a database comprised of 1,093 distinct composition-temperature entries [40]. The original work did not use ML for analysis of the data, but was a useful demonstration of the power of data visualisation to gain insights into the property space of plausible thermoelectric materials. A web tool was also published to allow the visualisation of up to four parameters from the database (Figure 1.3). The UCSB database has been subsequently used in several studies to create statistical models to predict electronic transport properties. For example, Furmanchuk *et al.* reported the development of a regression model that predicted the Seebeck coefficient at various temperatures, using 927 entries from the database [41]. Mukherjee *et al.* used data from the UCSB database to develop a ML-based approach for the prediction of electrical conductivity [42]. Gaultois *et al.* used the database to build a web-based machine-learning model and recommendation engine for the real-time screening of thermoelectric materials properties [43].

Several larger databases derived from theoretical methods also exist. In 2017, Ricci *et*

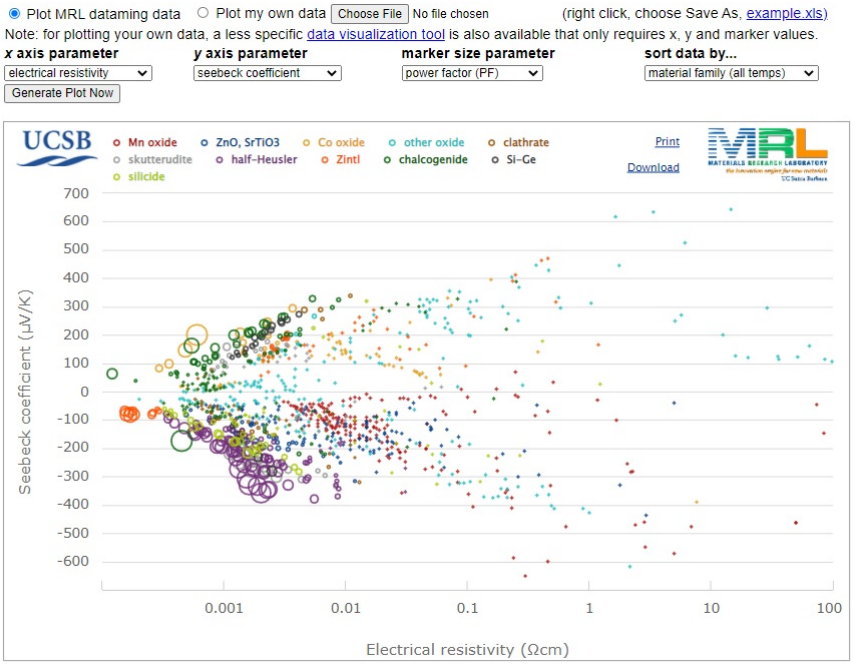

Figure 1.3: Screenshot of the web-based visualisation tool [44] accompanying the UCSB database. It allows the simultaneous visualisation of four parameters: abscissa, ordinate, marker size, and colour. Plots of Seebeck coefficients vs electrical resistivity give useful insights about the relative performance of different material families.

al. released a database of 47,737 compounds and their corresponding electrical conductivities and Seebeck coefficients at various temperatures and carrier concentrations [19, 45]. The properties were derived from DFT calculations, and the BoltzTraP software [7], using the BTE-RBA described above. Also, in 2020, Choudhary and co-workers [27] reported adding n- and p-type Seebeck coefficients and electrical conductivities (theoretically obtained using BoltzTrap) for 21,900 compounds to the JARVIS-DFT database [46] (Figure 1.4). Importantly, the electron transport coefficients in both of these databases were obtained using the CRTA. Since both databases share a number of compounds in common, Choudhary *et al.* compared the calculated n-type Seebeck coefficient for 9,434 compounds existing in both databases, at 600K and carrier concentration of $10^{20}/cm^3$, and found a mean absolute deviation of 18.8 $\mu V K^{-1}$ and coefficient of determination (R^2) of 0.87. The investigators attribute the differences between the datasets largely to the functionals used: the Ricci *et al.* database uses the GGA functional by Perdew *et al.* [47], while the JARVIS-DFT database uses the optB88vdW functional, which incorporates non-local correlation [48].

More recently, in 2021, Yao and co-workers reported the development of MIP-3d, a freely available database accessible online, which contains theoretically derived electron transport coefficients [35]. The database houses the results of DFT-based electronic property calculations for over 80,000 structures. Using a constant electron-phonon coupling approximation, [49] rather than the CRTA, the authors compute the Seebeck coefficient and the electrical conductivity (at 700K and doping level of 10^{20} cm^{-3}) for compounds in the database with band gaps $>$ 0.03 eV, which gave a total of more than 4,400 compounds. The TransOpt code [50], implementing the constant electron-phonon coupling approximation to deal with the scattering rates, was used to compute the transport properties. The authors validated their calculations by comparing various computed properties to those in the Materials Project,

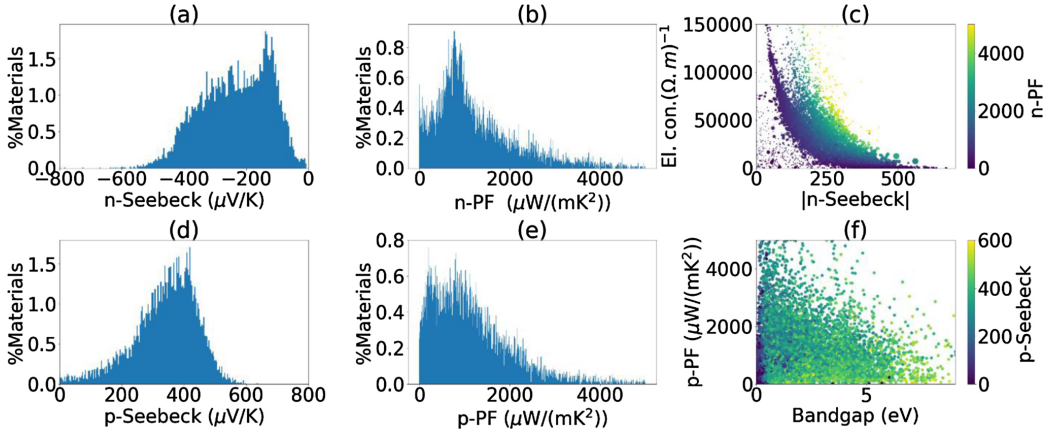

Figure 1.4: An overview of the computed thermoelectric properties for compounds of the JARVIS-DFT database, reproduced from reference 27. (a) n-type Seebeck coefficient distribution, (b) n-type power factors, (c) n-type electrical conductivity plotted against the absolute values of Seebeck-coefficient with colour-coded power-factor and size of the dots proportional to bandgaps, (d) p-type Seebeck coefficient distribution, (e) p-type power factors, (f) p-type power factor plotted against the bandgaps.

and found very good agreement. Amongst the top 5% of compounds, with high power factors and low sound velocities, there were many chalcogenides and compounds with heavy elements, such as Bi and Pb.

In contrast with databases of electronic transport coefficients, databases of lattice thermal conductivities are generally smaller in size, consisting typically of less than 10^3 compounds. This is likely to be a consequence of the data coming from experimental measurements, which are somewhat limited in number, or from theory, which is computationally quite involved for the calculation of lattice thermal conductivity. A dataset of lattice thermal conductivities was produced by Carrete *et al.*, as part of their search for half-Heusler semiconductors with low thermal conductivity [15]. The lattice thermal conductivities for 450 half-Heusler compounds were obtained by a combination of *ab initio* and ML methods: first, *ab-initio* methods were applied to a subset of 32 compounds, then that subset was used as a training set for a random forest regression model that was subsequently used to predict the lattice thermal conductivities of the remaining compounds. In a separate effort, Miyazaki and co-workers created a dataset containing the theoretical lattice thermal conductivities for 143 half-Heusler compounds [33]. The lattice thermal conductivities for all of the compounds in this dataset were computed using the Phono3py software library [51]. Finally, Jaafreh and co-workers assembled a dataset of the theoretical lattice thermal conductivities of 119 compounds reported in previous computational studies reported in the literature [31], and Tranås *et al.* produced a dataset of computed lattice thermal conductivities for 122 half-Heusler compounds using DFT and the temperature-dependent effective potential method [37, 52].

Datasets of experimentally-measured lattice thermal conductivities have also been assembled. Chen *et al.* collected the lattice thermal conductivities of 100 single crystal inorganic materials reported in the literature [23]. The dataset consists of 81 binary and 19 ternary compounds, and is diverse in terms of composition, space group, and the range of lattice thermal conductivities, which span 3 orders of magnitude.

In 2016, Gorai and co-workers released TE Design Lab, a database consisting of 2,701 compounds and their associated semi-empirical lattice thermal conductivities [17]. This is the first (and so far only) example of a semi-empirical dataset of thermoelectric properties. In

previous work [53, 54], Gorai *et al.* demonstrated a semi-empirical approach for computing the lattice thermal conductivity using simple descriptors for the acoustic and optical phonon modes. They derived a model for the lattice thermal conductivity that incorporates average atomic mass, average volume per atom, the number of atoms in the primitive cell, and the speed of sound in the material calculated from the DFT-derived bulk modulus, in addition to parameters fitted to experimental data. They found that the model was predictive only within one order of magnitude, across four orders of magnitude of experimental data.

1.1.2 Machine Learning Techniques to Accelerate the Calculation of Lattice Thermal Conductivities

Although in this thesis I will not report models for the prediction of thermal conductivities, this is a closely related problem that is important in the context of ML-based search for thermoelectric materials. Therefore I provide here an overview of previous work in this area and current challenges.

There are two main types of ML-based approaches to the calculation of lattice thermal conductivity. ML models can be trained, using suitable databases, to directly predict κ_{latt}. But it is also possible to use ML to accelerate the calculation of the forces or force constants needed within physics-based computational models for κ_{latt}. I summarise the two types of approach below.

Approaches Based on Thermal Conductivity Databases

The development of ML models for the direct prediction of lattice thermal conductivities faces two main obstacles: i) the lack of experimental or theoretical data and ii) the strong dependence of this property on a number of variables (temperature, particle size, nature of defects and their concentrations). Despite the data-rich environment that can be attributed to the recent rise of materials databases, data collection remains the main bottleneck for κ_{latt}-based models.

To the best of my knowledge, there have not been studies in which more that 150 materials have been used during training. For instance, Chen *et al.* collected 95 experimental values of κ_{latt} to build a model combining Gaussian process regression and recursive feature elimination [23]. Their results slightly improved previous works based on adding power coefficients as fitting parameters to the Debye-Callaway model to reproduce experimentally measured κ_{latt} [55]. Using κ_{latt} values obtained form DFT calculations does not guarantee larger datasets because of the high computational cost. The first dataset with calculated κ_{latt} values included only 101 binary compounds, belonging to rocksalt, zincblende, and wurtzite-type structural prototypes [56]. Combining Bayesian optimization with this limited dataset, while using only volume, density and element features, Seko *et al.* were able to find 221 materials with very low κ_{latt}, thus screening more than 54,000 structures. Similarly, Juneja *et al.* computed κ_{latt} for 120 dynamically stable, non-metallic materials containing binary, ternary and quaternary compounds [57]. They developed a Gaussian-process, regression-based ML model with four unique descriptors (maximum phonon frequency, Grüneisen parameter, average atomic mass and unit cell volume), predicting log-scaled κ_{latt} with a root mean square error (RMSE) of 0.21 against target values.

Due to the experimental and theoretical limitations for creating large datasets for κ_{latt}, most recent works have focused on developing strategies to create transferable, predictive models using small datasets. Incorporating the crude estimation of property, CEP, in the feature space stands as a common strategy that has worked reasonably well for different materials properties [58]. CEP is defined as a prediction of the targeted property with inexpensive

methods, even if their results are not very accurate. For instance, experimental band gaps are accurately predicted, even using small datasets, if the DFT-GGA band gap is included as a feature [58]. Following this strategy, Zhang *et al.* used the lattice thermal conductivity calculated through the Slack equation (which systematically underestimates κ_{latt}) as CEP, improving the accuracy of a ML model based on a dataset of 93 experimental measurements, as well as the use of a kernel ridge regression [58].

Small datasets can also work well by restricting the configurational space of materials. Different machine-learning-based models for predicting κ_{latt} have been built based on datasets with specific structural prototypes such as half-Heusler alloys [15, 33], germanides [59], and graphene alloys [60]. Although these models are usually more accurate, they show poor transferability across different families of materials. However, increasing the variability in the training data leads to a reduction of the accuracy of the model. In order to improve transferability and accuracy simultaneously, Juneja *et al.* has proposed the use of a localised regression-based Patchwork Kriging approach for a class-independent dataset [61]. Using this approach, the dataset, which covers a wide range of structural prototypes and compositions, is partitioned into smaller local subsets with respect to κ_{latt}. These subsets share some data-points in order to give the same response at the boundaries. This strategy drastically reduces the RMSE for log-scaled κ_{latt}, from 0.24 to 0.13 [61].

Feature selection constitutes a key step that not only modifies the accuracy of the model, but also promotes a clearer insight into the chemical and physical features underlying κ_{latt}. Chen *et al.* have demonstrated that bulk modulus and density can be combined to build a good descriptor of the anharmonicity of the crystal and their group velocities [23]. Similarly, Juneja *et al.* have found unexpected connections between electronic transport properties such as Seebeck coefficient, S, and electrical conductivity, σ, with κ_{latt} [62]. However, feature selection is also critical in determining the size of the training set and the applicability of the model. For instance, the use of materials properties, such as maximum phonon frequency or Grüneisen parameter, requires expensive lattice-dynamic calculations, which limits the number of materials included in the dataset, as well as the usefulness of the machine-learning model. For this reason, cheaper sets of features should be used to maximise its applicability. Jaafreh *et al.* have built a dataset based on calculated κ_{latt} using exclusively crystal and compositional features, in addition to temperature [31]. Crystal features are based on the Voronoi tessellation structure [63] of each material, whereas composition features are generated using the element properties. Accurate models are obtained independently of the machine-learning algorithm used during training and, most importantly, they can predict κ_{latt} for materials over a wide range of temperatures. The use of simple features that can be obtained from many databases facilitates the use of the model for the screening of κ_{latt} for more than 32,000 compounds.

Approaches Based on Accelerating the Calculation of Forces

The purely data-based prediction of κ_{latt}, as discussed above, is useful for examining trends across large chemical spaces, and for preliminary screening of promising low-κ_{latt} materials. But at the level of individual compounds, average errors in the order of 50% are too high to be relied upon for zT estimation. Additionally, databases of experimentally measured κ_{latt} values have important limitations, because such measurements are extremely sensitive to the procedure of synthesis or preparation of the material. Variables such as point defect concentration, average grain size, disorder or even isotope ratios not only modify κ_{latt} but also its behavior with respect to temperature.

In order to obtain more accurate values, and consider some of the aforementioned variables, some authors have opted for a bottom-up approach in which machine-learning algorithms are

not used to compute $\kappa_{\rm latt}$ directly, but instead to predict the forces needed for the calculation of $\kappa_{\rm latt}$ using state-of-the-art methods. Solving the phonon BTE [64], or combining molecular dynamics with Green-Kubo relations [65, 66], represent two of the most accurate approaches for obtaining $\kappa_{\rm latt}$. Accelerating these calculations opens the door to investigation of more sophisticated and realistic models of materials, allowing the consideration of the previously-mentioned synthetic variables. Although these two approaches are very different, they both rely on the calculation of forces, either via IFCs or interatomic potentials.

Several ML techniques have been proposed to build accurate interatomic potentials for molecular dynamics, which can be combined with Green-Kubo relations to obtain $\kappa_{\rm latt}$ values as accurate as the ones obtained from *ab-initio* molecular dynamics. This approach is very attractive because the Green-Kubo method allows the consideration of anharmonicity effects to all orders in the calculation of $\kappa_{\rm latt}$. For instance, Korotaev and Shapeev developed moment tensor potentials that allow for active learning as the way to generate a potential on the fly [67]. Using this method, they predicted $\kappa_{\rm latt}$ for partially-filled skutterudites, considering the role of disorder. The success of the machine-learned potential approach relies on the accuracy of those potentials to describe not only the harmonic part of the potential energy surface but also the anharmonic contributions. This has been recently explored by Verdi *et al.* [68], who used a kernel-based machine-learning model implemented in the VASP code to investigate a paradigmatic anharmonic material, ZrO_2, and its lattice thermal conductivity is predicted with high accuracy. There are significant on-going efforts to improve the accuracy of machine-learned potentials for the description of harmonic and anharmonic vibrational properties. For example, deep neural networks have been used for the development of interatomic potentials for β-Ga_2O_3, reproducing accurately phonon dispersion and the anisotropic behavior of $\kappa_{\rm latt}$ [69]. George *et al.* have recently shown how to fit Gaussian approximation potential models that accurately predict vibrational properties and investigated the performance in the prediction of thermal conductivity [70].

An alternative, related approach is to use machine-learning techniques to accelerate the calculation of the force constants needed for the solution of the phonon BTE. There are some codes, based on regularised linear regression or compressed sensing techniques, that reduce between one and two orders of magnitude the computational cost of calculating IFCs [71, 72]. As discussed above, traditional approaches based on finite-displacement of some atoms in supercells require hundreds or even thousands of supercell single-point calculations for the prediction of 2nd and 3rd-order IFCs. However, these new approaches are based on the distortion of *all* atoms of the supercell, and have the computational advantage of requiring much fewer single-point DFT calculations (typically only a few tens) to obtain the IFCs (Fig. 1.5).

The principle behind these techniques is based on the linear relationship between the forces, F, and displacements, u, via the IFCs, Φ:

$$F_i^\alpha = -\Phi_{ij}^{\alpha\beta} u_j^\beta - \frac{1}{2}\Phi_{ijk}^{\alpha\beta\gamma} u_j^\beta u_k^\gamma ..., \tag{1.2}$$

where i, j, k and α, β and γ represent atoms and Cartesian coordinates, respectively. Force constants are extracted from a regression of these linear equations or using compressed sensing, which is a technique for recovering sparse solutions from incomplete data. In this type of approach, the calculated force constants are very sensitive to the amplitude and distribution of the distortions. For instance, small displacements (0.01-0.05 Å) following a Gaussian distribution work reasonably well to extract second-order IFCs. However, small displacements can lead to high numerical errors if higher-order IFCs are included in the model. The average displacement amplitude used with the direct approach (finite-displacements), implemented in packages such as thirdOrder.py-ShengBTE [8] or AAPL [73], is typically 0.01-0.02 Å, because

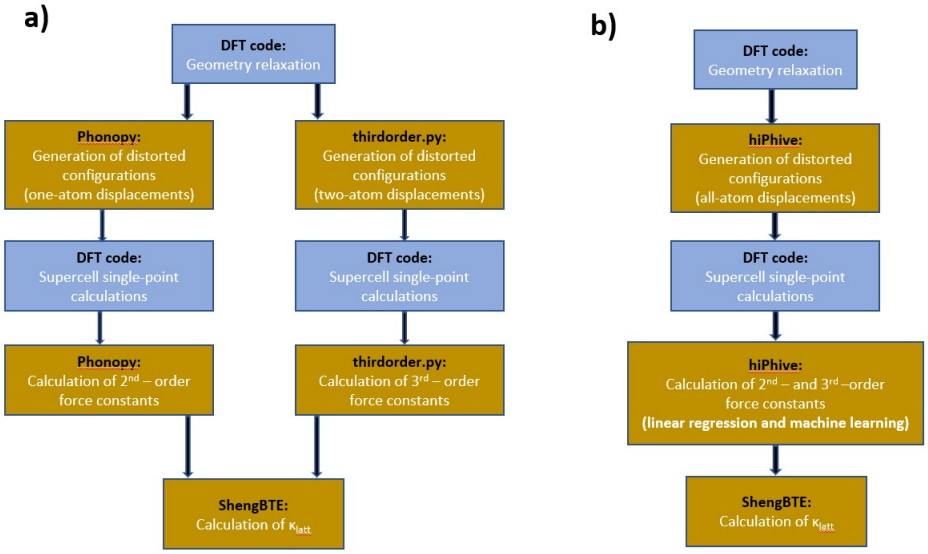

Figure 1.5: Comparison between the workflows of a) the traditional method to obtain κ_{latt} using the DFT-BTE approach with systematic atom displacements, and b) the machine-learning-accelerated approach implemented in the hiPhive code [72]

.

each distorted supercell is exclusively used for the calculation of a specific third-order IFC. Using the regression approach implemented in the hiPhive code, in which all distorted supercells are used to calculated all IFCs, larger displacement amplitudes are needed to disentangle the contribution of high-order IFCs [74]. For large displacements (>0.1Å), Gaussian distributions are not encouraged because they can produce supercells with too-short interatomic distances. There are different approaches to overcome this problem. One simple solution is using distorted supercells generated via a Monte Carlo algorithm, which penalises displacements producing too-short interatomic distances. Although this approach works well with simple structures, building the right training dataset for complex materials with different types of bond strength can be challenging. More complex approaches based on dual adaptive sampling, DAS, have proven that it is possible to predict accurate IFCs for materials with chemical bond hierarchy [75]. Yang *et al.* developed a DAS method that generates an effective training set covering a wide spectrum of thermodynamic conditions and a wide temperature range, obtaining accurate values of κ_{latt} for $CoSb_3$ [75]. Once the training set is properly built, these techniques are extremely powerful for the high-throughput prediction of κ_{latt} at a reduced cost. This strategy has been effectively applied to rocksalt and zincblende compounds [76], ternary and quaternary chalcogenides [77, 78], skutterudites [75] or clathrates [79].

1.1.3 Machine Learning Techniques to Accelerate the Calculation of Electron Transport Coefficients

In contrast with the prediction of lattice thermal conductivity, there have been fewer studies aimed at learning models of electron transport coefficients, although the datasets available for this task are generally much larger. Table 1.2 presents a list of studies involving the use of ML for the prediction of electron-transport properties.

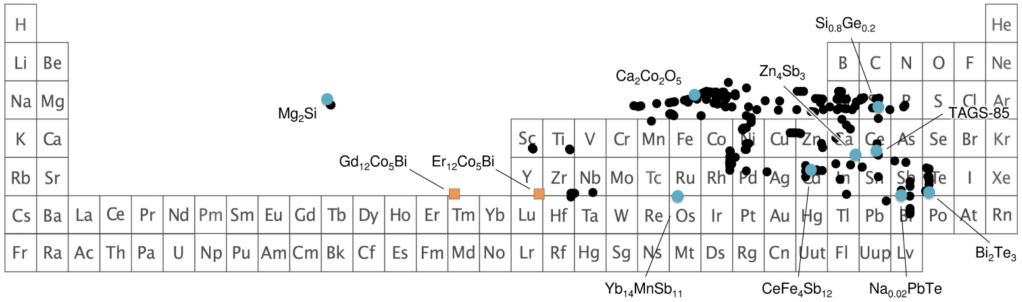

Figure 1.6: Scheme, reproduced from reference 43, illustrating the differences in composition space between known thermoelectrics from the UCSB database and those discovered in the work of Gaultois *et al.* [43]. Most thermoelectrics lie close together in composition space (blue and black dots), but the machine learning-based recommendation engine predicted new compounds (orange squares) that were different, chemically and structurally, from existing known thermoelectrics.

Table 1.2: A list of studies involving the use of machine learning to predict various electron transport properties. S is the Seebeck coefficient, σ is the electrical conductivity, and PF is the power factor (i.e. σS^2).

Study	Year	Ref.	Dataset	Algorithms	Targets
Gaultois *et al.*	2016	43	UCSB, custom	random forest classification	S, σ
Chen *et al.*	2016	80	Ricci *et al.*	clustering with DBSCAN [81]	S, σ
Furmanchuk *et al.*	2018	41	UCSB	random forest regression	S
Mukherjee *et al.*	2020	42	UCSB, custom	gradient boosting regression	σ
Choudhary *et al.*	2020	27	JARVIS-DFT	gradient boosting regression	S, PF
Sheng *et al.*	2020	82	Xi *et al.*	gradient boosting regression	p-type PF
Yoshihama *et al.*	2021	83	XRD[a], starrydata2	Gaussian process regression	S, σ
Pimachev *et al.*	2021	84	custom	neural net, random forest	S
Na *et al.*	2021	85	UCSB	neural net regression	S, σ, PF

[a] X-ray diffraction data from the AtomWork-Adv database [86, 87]

One of the first applications of ML to the estimation of electron transport properties was the work of Gaultois and co-workers, who reported the development of a ML-based thermoelectric material recommendation engine [43, 88]. Using a dataset constructed from various experimental and theoretical sources, including the UCSB dataset, the authors train a random forest classifier to predict whether a compound will have a Seebeck coefficient and electrical resistivity (in addition to thermal conductivity and band gap) that fall within acceptable ranges for thermoelectric application. The materials are represented using a "tuned blend" of descriptors that are developed in-house, and incorporate information from a variety of sources, including the periodic table. Using leave-one-out cross-validation, and error histograms for visualisation, the authors conclude that the model is somewhat skewed towards making false negative predictions in the case of resistivity, and false positive predictions in the case of the Seebeck coefficient, but nevertheless makes reliable predictions overall. Based on the outputs of the model, the authors make a number of recommendations of new compounds with the potential for thermoelectric application, and further investigate $Er_{12}Co_5Bi$ and $Gd_{12}Co_5Bi$ in particular. The recommendations seemed counter-intuitive, since their compositions are not typical of known thermoelectrics (Figure 1.6). Nevertheless, the predicted high electrical conductivity and modest Seebeck coefficient (and low thermal conductivity) were all confirmed experimentally.

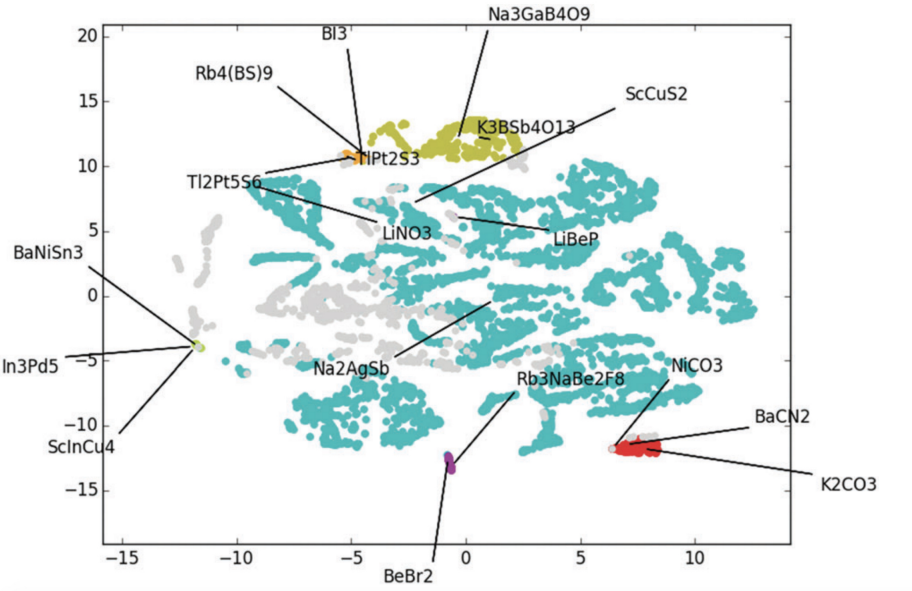

Figure 1.7: A visualisation of the clustering results obtained using DBSCAN, reproduced from reference 80. The clustered high-dimensional data has undergone dimensionality reduction with t-SNE, and the axes have no physical meaning. Clustering analysis reveals 6 large clusters, as depicted by the various colours in the plot, with each of the clusters exhibiting distinct ranges of electron transport coefficients.

Another early application of ML to the calculation of electron transport properties was the work of Chen *et al.* [80], who reported an analysis of the results of computing the theoretical transport coefficients of the compounds comprising the Ricci *et al.* database [19]. Instead of developing a regression model to predict the Seebeck coefficient or the electrical conductivity, the authors carried out a clustering analysis with the DBSCAN algorithm [81]. Each material in a set of 5,431 candidate thermoelectric materials from the database was given a descriptor comprised of 58 properties, excluding the calculated Seebeck coefficient and electrical conductivity, since these were the properties that they were attempting to model. Their clustering analysis revealed 6 clusters (Figure 1.7), and they found that each of the clusters contained distinct ranges of electron transport coefficients. Such an approach can be used in quantitative models, such as a cluster-rank-model, as suggested by the authors.

More recent studies have generally involved the use of regression models. An early example is a study from Furmanchuk and co-workers [41], who reported using the UCSB database [13] to learn a regression model that predicted the Seebeck coefficient for stoichiometric and non-stoichiometric crystalline solids between 300K and 1000K. After an initial curation step of the dataset to remove duplicates, 927 entries from the UCSB dataset were used to train a random forest [89] regression model. The authors crafted a descriptor consisting of anywhere from 50 to 452 features, and achieved an R^2 of between 0.70 and 0.80. The features employed consisted of material-specific properties such as the number of valence electrons, the largest atomic number, and measured thermal conductivity data. To verify that model performance extended beyond the original dataset, the authors collected 20 compounds from the literature, and report that the model achieves an $R^2 \geq 0.88$ on this external dataset. The authors also provided a publicly accessible web application that allows anyone to use their model [90].

Another study that makes use of the UCSB dataset and empirical values of electron transport coefficients is that of Mukherjee *et al.*, who attempt to predict experimentally

measured electrical conductivities as well as relaxation times [42]. The authors begin by assembling a dataset of 124 semiconducting compounds from the literature and from the UCSB dataset. The compounds have different structure types, such as rocksalt, wurtzite, and zinc-blende, and have experimentally measured electrical conductivities in the range of 10^{-3} to 10^{5} S cm^{-1} at 300K. A feature selection process led to the selection of 8 different features, including boiling point, melting point, molar heat capacity, electron affinity, and ionisation energy. A gradient boosting tree regression method was employed [91], which has been shown to be useful for smaller datasets. The model achieved an RMSE of 0.22 S cm^{-1} and an R^2 of 0.98 for the prediction of log-scaled electrical conductivity. Furthermore, using the predicted electrical conductivities, the authors predict the electron relaxation times, which outperforms the relaxation times obtained from a deformation potential model.

Datasets containing theoretically-derived values of electron transport coefficients have also been used in ML studies. In 2020, Choudhary and co-workers report creating a dataset by computing the Seebeck coefficients for 21,900 compounds in the JARVIS-DFT database [27]. They further train ML models on these data to create predictors of the Seebeck coefficient and power factor. Instead of building regression models, they train models that classify a material as a high-performance thermoelectric, if its Seebeck coefficient is predicted to be greater than 100 μVK^{-1} for p-type, or less than 100 μVK^{-1} for n-type materials, and if its power factor is greater than 1000 μWm^{-1}K^{-2}. They found that gradient boosting decision trees, combined with force-field inspired descriptors described previously by the authors [92], result in the best performance.

A recurring theme in the prediction of material properties using ML approaches is that there is often a scarcity of data for the problem at hand. Training data can be created from first-principles calculations, but first-principles calculations themselves can be computationally expensive, and require manual intervention. To mitigate this obstacle, Sheng *et al.* report using an Active Learning approach [93] to predict the power factors of diamond-like chalcogenides and pnictides [82]. Starting from a previously created database [21], 158 compounds are selected as the initial set of ground-truth, DFT-derived examples. An additional 342 compounds are selected by enumerating possible combinations of cations and anions, and together with the previous set of 158 compounds comprise a search space. Using a query-by-committee approach, with different kinds of models, such as support vector regression, gradient boosting regression, and random forest regression, the 15 candidates which exhibit the most prediction variance are selected at the beginning of each iteration, and subjected to DFT calculation to determine their power factors. Each iteration of Active Learning begins with a training step on the available DFT data, using composition-only descriptors such as valence electron number, atomic weight, electronegativity, etc., and the learning loop ends when either the extrapolated Pearson R is greater than 0.90, or 10 iterations have elapsed. All the models eventually converged to low RMSE ($\sim$ 4 μWcm^{-1}K^{-2}) and high Pearson R ($>$ 0.90). The gradient boosting regression model performed the best in the last iteration of the learning loop, and was used to predict the power factor for all compounds in the entire search space. The results of these predictions led the authors to make generalisations about what kinds of compounds are likely to have higher power factors, such as binary pnictides.

Descriptors used for the prediction of electron transport coefficients typically consist of compositional information, such as the ratios of atoms in a compound, known atomic properties such as electronegativity, mass, etc., and structural information, such as volume per atom. A different approach is described by Yoshihama and co-workers, using DFT-derived X-ray diffraction patterns [83]. The authors assembled a dataset of 1,116 examples using electron transport properties from the starrydata2 database [25] and X-ray diffraction data from the AtomWork-Adv database [87]. The descriptor for a compound consisted of tem-

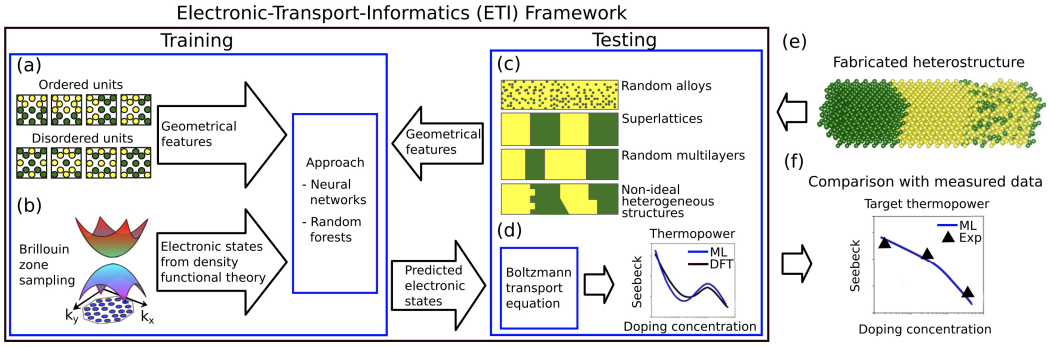

Figure 1.8: An overview of the ETI framework, reproduced from reference 84. (a) 16-atom units of ordered and disordered fragments are used to train machine-learning algorithms, (b) electronic properties are computed from theory, (c) machine-learning models predict the energy values of collections of valence and conduction bands for configuration of various sizes, (d) the Boltzmann Transport Equation is used to compute the Seebeck coefficient from predicted electronic properties, (e) a fabricated heterostructure is subjected to analysis, and (f) the predictions of the Seebeck coefficient are compared to experimental values.

perature, atomic composition, and the X-ray diffraction data, and the target properties were the electrical conductivity and Seebeck coefficient (in addition to direct prediction of ZT and thermal conductivity). Different types of model were created, but Gaussian process regression produced the best results. Within a defined applicability domain, the model produced an MAE of 1.28×10^4 $(\Omega\ \mathrm{m})^{-1}$ for electrical conductivity, and an MAE of 10.7 $\mu\mathrm{VK}^{-1}$ for the Seebeck coefficient. The authors subsequently used the models to examine 610 compounds with unknown transport properties, and identified compounds with high zT, including several containing Pb, Te, and Se, as well as those with more atoms per unit cell in the crystal.

Although the majority of studies reported thus far involve bulk 3D solids, some work is beginning to emerge on other classes of materials, such as heterostructures. Pimachev *et al.* reported the development of an electronic-transport-informatics (ETI) framework for the prediction of thermopower of fabricated silicon/germanium semiconducting heterostructures (Figure 1.8) [84]. Their approach rests on the hypothesis that the relationship between electronic band structures and smaller, localised collections of atoms, can be extrapolated to larger collections. They trained neural network and random forest models to predict the energy values of collections of valence and conduction bands for small (16-atom) ordered and disordered configurations, with the expectation that the models would transfer their knowledge to larger systems. A number of experiments were performed by the authors to validate their approach, and one such attempt involved the prediction of the band structures of a Si_4Ge_4 superlattice. The predictions matched the DFT results closely, producing an MAE of 34.2 meV for the neural network model, and 38.2 meV for the random forest model. The authors emphasised that fabricated heterostructures contain structural complexities that complicate the application of *ab-initio* approaches. To demonstrate the effectiveness of their approach on such systems, the models were applied to systems such as n-type Si(5Å)/Ge(7Å) superlattices grown along the [001] direction at 300K, and n-type $Si_{0.7}Ge_{0.3}$ alloys at 300K, to obtain the bands to be used in conjunction with the BTE, through which the Seebeck coefficients were derived. The resulting cross-plane and in-plane Seebeck coefficients at different carrier concentrations were in good agreement with experimental observations.

In another departure from pure bulk solids, Na and co-workers reported the development of a deep neural network model to predict the electron transport coefficients of doped materials

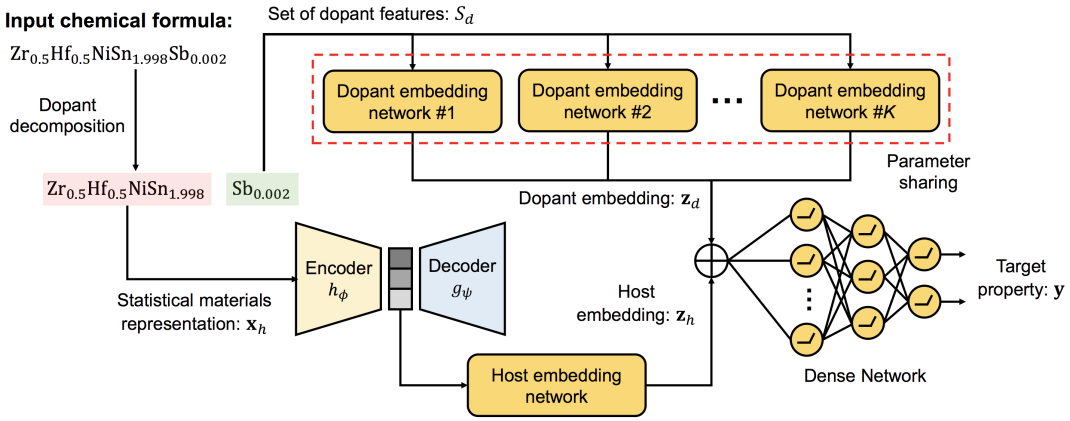

Figure 1.9: Architecture of the DopNet model, reproduced from reference 85. The yellow components represent either single neurons, or feed-forward networks, with ReLU activation. The input to the architecture is the formula $Zr_{0.5}Hf_{0.5}Sn_{1.998}Sb_{0.002}$, and the output is the target property, $\mathbf{y}$.

considering the nature and concentration of the doping elements [85]. Doping is a common strategy to enhance the thermoelectric performance of materials, and it is well known that the addition of even a small amount of dopants can have a drastic effect on thermoelectric behavior [94, 95]. The authors devised a deep neural network architecture that they named DopNet, which accepts as input a non-stoichiometric composition, and predicts a numerical target property (Figure 1.9). The input is first split into host and dopant elements, and each then converted into feature vectors consisting of the statistics of the elemental attributes of the atoms present. The inputs are subsequently embedded using a learned autoencoder, and concatenated before being fed into a deep feed-forward neural network. A dataset of 573 distinct composition-temperature entries was derived from the UCSB dataset, and the performance of DopNet in regression tasks was compared to models created using more traditional machine-learning algorithms, such as Suppport Vector Regression, Gaussian Process Regression, and Gradient Boosting Tree Regression (GBTR). Evaluation using 10-repeated 3-fold cross-validation revealed that the DopNet model outperformed all other machine-learning algorithms, achieving an R^2 of 0.86 on the Seebeck coefficient prediction task, and an R^2 of 0.64 on the electrical conductivity prediction task. To further assess the utility of the DopNet model, the authors sourced 18 compounds from the literature that have been studied experimentally at 700K, and that were absent from the dataset used to train and evaluate the model, and attempted to predict the reported zT values. DopNet achieves an MAE of 0.13 versus an MAE of 0.41 achieved by the GBTR model, a large improvement which the authors attribute to the enhanced ability of the deep neural network architecture to learn the highly non-linear relationships involved in doping effects.

1.1.4 Current Challenges and Perspectives

A number of themes emerge from the studies described above, involving the application of ML to the prediction of transport properties from databases. Generally, datasets tend to be small, with only two known datasets containing in the order of 10^4 examples, and the remaining all containing in the order of 10^3 examples, or less. Moreover, most datasets are derived from theory. While there are clear and valid reasons for the lack of experimentally-derived datasets, practitioners must remain mindful of the limitations of theoretically-derived values.

For example, values typically result from computations that are carried out under the CRTA, which is often a poor approximation, and commonly used functionals are known to produce quite inaccurate band gaps. Furthermore, the majority of studies are constrained to bulk 3D systems, and do not explore more intricate systems, such as heterostructures and superlattices. Related to this is the observation that many of the studies are limited to certain kinds of bulk systems, such as chalcogenides, pnictides, and half-Heusler compounds. Chemical space is vast, and the great majority of it remains unexplored for its thermoelectric potential.

Regarding the ML solutions that have been implemented, most involve the use of classical ML algorithms, such as random forests, ridge regression, and gradient boosting trees (Table 1.3). While other fields of computational materials science have gradually seen the introduction of the latest advancements in Deep Learning techniques [96], such as transformer networks [97] and convolutional graph neural networks [98], the field of thermoelectrics has yet to see any substantial adoption of these techniques. One reason for the lack of use of Deep Learning may be the small size of thermoelectric datasets, and the consequent fear of overfitting models with many parameters, but recent research has shown that this concern may be exaggerated [99]. Additionally, the descriptors used tend to consist of enumerations of known atomic and bulk properties, such as atomic radius, electronegativity, and melting point, for example. But recent research, particularly in Natural Language Processing, has demonstrated the superior nature of distributed, or non-local, representations [100]. These representations replace traditional descriptors, and there has been some work on developing such representations for materials [101–103]. However, such representations have not yet been used to predict thermoelectric transport properties.

Table 1.3: A list of machine-learning (ML) techniques used by various studies described in this chapter, along with references with more information.

ML Technique	Ref.	Used by Ref.
Active learning	93	67
Compressed sensing	104	72
Gaussian process regression	105	23, 57, 83
Gradient boosting regression	91	42, 27, 82
Kernel ridge regression	106	58, 68
Neural networks	96	69, 84, 85
Random forests	107	43, 41, 84

Going forward, there are a number of future directions that would advance the state of the art. With respect to datasets, a need exists for larger databases with many more compounds, perhaps in the order of 10^5, or greater. Such a dataset would need to be computed from first-principles, but the structures of hundreds of thousands of compounds exist in openly accessible materials databases [108, 109]. The opportunity to apply more appropriate theoretical treatments would also increase the quality of the data, and the size of the chemical space covered would allow models to better generalize to less familiar systems. Another direction that can be explored, and that addresses the problem of data scarcity, is to train models that learn atomic scale dynamics, instead of training models to learn to predict transport properties directly. This approach has the advantage that data can be generated as required, from molecular dynamics simulations, and does not depend on first-principles computations, or experimental measurement, for transport coefficients. In this spirit, Xie *et al.* developed Graph Dynamical Networks, that learn the dynamics of atoms in materials [110]. Similar approaches are already being used for the determination of lattice thermal conductivity, but they may also be used for electrical conductivity determination (see Noritake *et al.* [111], for example).

With respect to ML algorithms, it is likely that Deep Learning will begin to see adoption

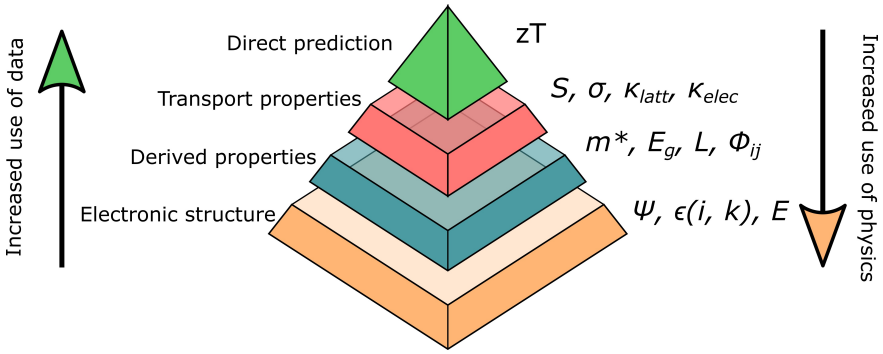

Figure 1.10: Scheme illustrating the different levels at which machine-learning techniques can be applied in the prediction of the thermoelectric figure of merit.

in the field, as will the use of distributed representations in place of traditional, local representations. Graph neural network models of inorganic structure have been shown to work very well when predicting electronic and bulk material properties [112], and it would be very interesting to see what such models could glean from a thermoelectric dataset that includes material structure information. On the other hand, structure information is often not available in thermoelectric property datasets, and models must make use of composition alone. Recently, a transformer-based model has demonstrated superior performance when predicting electronic and bulk material properties from composition alone [113]. It would be exciting to apply such a model to thermoelectric datasets. Indeed, scanning composition space offers tremendous potential [114], and such a model would provide the means for making accurate and fast predictions of the thermoelectric properties of vast numbers of compositions.

In this chapter, I have discussed the application of ML techniques to the investigation of thermoelectric materials, focusing on the accelerated prediction of the transport coefficients contributing to the thermoelectric figure of merit zT. I have contrasted these approaches with purely physics-based approaches to compute these properties. However, it is important to note that the classification of existing approaches according to their use of physics principles or data is not binary: there is a spectrum of models incorporating variable proportions of physics and data. This is illustrated in Figure 1.10. ML models can be used to attempt to predict zT directly from databases, using little physics in the model itself (although some physics would enter in the selection of descriptors). A lower-level approach, of which some examples have been given in this chapter, involves the use ML to individually predict the transport coefficients S, σ, κ_{elec}, κ_{latt} that appear in zT. ML can also be used one level below, predicting properties that affect the transport coefficients, *e.g.* effective mass m^*, band gap E_{g}, Lorenz number L, carrier relaxation time τ, or deformation potential, which are related to electronic transport, or interatomic force constants $\Phi_{ij}^{\alpha\beta}$, $\Phi_{ijk}^{\alpha\beta\gamma}$, which are related to phonon transport. In this approach level, the ML predictions can be incorporated in physical models to obtain the transport coefficients. The lowest-level approach consists of using ML for the prediction of the electronic structure $\epsilon(i,\vec{k})$ and total energy of the system as functions of the atomic coordinates, thus replacing the DFT calculations. All electron and phonon transport properties can then be derived using physical principles, still at a relatively low computational cost because the DFT simulations are often the most time-consuming part in the calculation of transport coefficients. The lower the level at which ML is used, the more physics is involved in the model. In future studies of thermoelectric behavior of materials, it is likely that ML techniques will be used following multi-level approaches, where lower-level approaches are used to create large datasets to train higher-level approaches.

Finally, I have only discussed the prediction of properties that are relevant to the figure of merit zT of the thermoelectric materials. But ML methods can also give useful insights about other properties of interest for thermoelectric behavior. For example, the dopability of the material is a crucial consideration in the design of a thermoelectric material, as has been argued in the review by Gorai *et al.*[9] Predicting the optimal charge carrier concentration of a compound is not very useful if the solubility of dopants and the intrinsic defect chemistry of the material does not allow that carrier concentration to be reached. The stability of materials, with respect to phase separation and the formation of compositional inhomogeneities, is also a very important aspect in the design of thermoelectric materials.

Overall, the problem of discovering new thermoelectric materials is a complex playground for ML. But it is precisely the challenging nature of the problem that makes it so compelling. The potential for thermoelectric materials to transform the energy consumption profile on Earth is astounding, and they are crucial if nations are intent on meeting carbon emission targets. The search for thermoelectric materials that are cheap to produce, made of abundant elements, and non-toxic is an important, worthwhile and fascinating endeavour that is bound to increasingly attract the attention of computational scientists, and of ML practitioners in particular.

1.2 Aims and Objectives of the Thesis

The central objective of this thesis is to design and develop novel and efficient ML-based tools as part of a workflow for discovering promising thermoelectric materials (and potentially other functional materials) in unexplored chemical spaces. This requires the following specific objectives:

- Design, train, and test efficient representations of materials compositions that are suitable for deep learning algorithms.
- Develop a deep learning model that is capable of predicting properties of interest for thermoelectric applications (e.g. the Seebeck coefficient) from knowledge of composition alone.
- Create a crystal structure generation tool that is capable of fast prediction of crystal structures from composition, in such a way that high-throughput *ab initio* confirmation of the ML-predicted properties is possible.

While for the purposes of this thesis, the focus is on the discussion on thermoelectric materials, the ML-based tools developed have wider appeal. Therefore, this thesis aims to make a substantial contribution to the field of materials informatics, paving the way for faster discovery of useful materials.

Chapter 2

Methodology

2.1 Deep Learning

Deep learning is an ML technique that has led to significant advances in many areas of research, such as natural language processing and computer vision [96]. Although it originates from the idea of an artificial neural network, developed in the 1940s and 1950s, the first practical demonstrations of the technique arrived in 2012, when Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton reported state-of-the-art computer vision results. Their model, AlexNet, won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a large margin [115]. For the first time, this work demonstrated how a GPU could facilitate the training of a neural network with 60 million tunable parameters. It was also a synthesis of various novel concepts at the time, including convolutional layers, the ReLU activation function, and the "dropout" technique for addressing overfitting. This work ushered in a new ML era. Since then, the approach has been applied in countless settings, at increasingly larger scales, with a growing toolbox of techniques, and a stunning track record of success. Deep learning has become an established science, and a distinct field of research. In this section, I will provide a precise description of the basic techniques that constitute the foundation for deep learning.

2.1.1 Overview of Artificial Neural Networks

At the core of deep learning is the artificial neural network, a computational model of distributed and collective information processing inspired by the networks of neurons in biological brains. The neural network appears to have been first introduced in 1943 by Warren S. McCulloch and Walter Pitts, and by Frank Rosenblatt in 1958 in the form of the Perceptron model [116, 117].

The model is comprised of a collection of "neurons", which are information processing units that accept a weighted sum of inputs from other units, and produce as output the result of an activation function. The inspiration for this design comes from the observed "all or none" nature of biological neurons, where a neuron either fires completely or not at all, depending on whether it reaches a threshold of stimulation. This binary character of biological activity leads naturally to a model of computation based on binary logic, where a neuron can be "on" (firing) or "off" (not firing).

While the originators of the neural network model may have envisioned this threshold-based behavior closely resembling the operations of logical gates (e.g., AND, OR, NOT), which are the fundamental building blocks of digital computation, modern neural networks are typically not binary in their activity, but rather output a continuous range of values. Rather than a neuron being in an "on" or "off" state, it can be interpreted as having a certain level of

activity (e.g. a "firing rate").

In practice, a precise biological interpretation is unnecessary. In 1969, Minsky and Papert showed that Perceptrons could not learn certain functions that were not linearly separable, such as the XOR function [118]. This result led to widespread pessimism towards neural networks. However, it eventually became clear that these limitations could be overcome through the introduction of an additional layer of neurons. David Rumelhart and colleagues demonstrated how a multi-layer neural network could be trained, and hence the modern version of the neural network was conceived [119].

Generally, an artificial (or deep) neural network is composed of multiple layers of neurons, typically organized into an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to all neurons in the subsequent layer via weighted connections. The network is trained by adjusting these weights based on errors in its predictions using a method known as backpropagation. The training process typically employs an optimization algorithm, such as stochastic gradient descent, to minimize the difference between the network's predictions and the expected outcomes.

2.1.2 Training: Backpropagation and Optimization

Formally, an artificial neural network can be described as a function $f(\mathbf{x};\theta)$, where $\mathbf{x} \in \mathbb{R}^n$ is an n-dimensional input vector and θ represents the network's parameters. The network's parameters generally consist of the connection weights and biases. The network is organized in L layers, with each layer l containing a set of neurons. For a single layer l, the output $\mathbf{h}^{(l)}$ is computed as:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}), \tag{2.1}$$

where $\mathbf{W}^{(l)}$ is the weight matrix for layer l, $\mathbf{b}^{(l)}$ is the bias vector, and σ is the activation function (e.g., ReLU, sigmoid). The input layer is simply $\mathbf{h}^{(0)} = \mathbf{x}$, and the output of the final layer, $\mathbf{h}^{(L)}$, represents the network's prediction.

The network is trained by minimizing a loss function $\mathcal{L}(\mathbf{y}, f(\mathbf{x};\theta))$, where $\mathbf{y}$ is the target output. Training is an iterative process which adjusts θ, with the aim of reducing the difference between the predicted and true outputs, as quantified by the loss function. This optimization is typically performed using stochastic gradient descent (SGD) and backpropagation.

Backpropagation is the method used to compute the gradient of the loss function with respect to θ. It is an algorithm that efficiently applies the chain rule of calculus by computing the gradient layer-by-layer, beginning from the output layer and moving backward, reusing intermediate results and avoiding redundant calculations. For each layer l, the gradient of the loss function with respect to the weights $\mathbf{W}^{(l)}$ and biases $\mathbf{b}^{(l)}$ is computed as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \delta^{(l)}\mathbf{h}^{(l-1)^\top}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}} = \delta^{(l)}, \tag{2.2}$$

where $\delta^{(l)}$ represents the gradient of the loss function with respect to layer l, which is recursively calculated starting from the output layer. The gradient of the loss function is thus computed with respect to the parameters of each layer.

SGD is an iterative algorithm which updates θ based on these gradients. The training process requires a dataset, $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, consisting of N examples, where each example consists of an input $\mathbf{x}_i$, and its corresponding target output $\mathbf{y}_i$. At each iteration t, a sampling of one or more examples, $\mathbf{x}_i$ and corresponding target $\mathbf{y}_i$, is selected. The parameters are updated as:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\partial \mathcal{L}(\mathbf{y}_i, f(\mathbf{x}_i;\theta))}{\partial \theta^{(t)}}, \tag{2.3}$$

where $\eta \in \mathbb{R}$ is the learning rate, and $\frac{\partial \mathcal{L}}{\partial \theta}$ represents the gradient computed through backpropagation.

Typically, the network is exposed multiple times to the entire dataset during training. Each exposure to the full dataset is termed an *epoch*. Algorithm 1 describes the neural network training procedure in detail:

Algorithm 1 Neural Network Training with Backpropagation and SGD

1: **Initialize:** Randomly initialize weights $\mathbf{W}^{(l)}$ and biases $\mathbf{b}^{(l)}$ for each layer l.
2: **while** not converged, or for a fixed number of epochs **do**
3: **for** each batch $\mathbf{X}_{\text{batch}}, \mathbf{Y}_{\text{batch}}$ **do**
4: **i. Sample batch:** Select a batch of training examples $\mathbf{X}_{\text{batch}} = \{\mathbf{x}_i\}$ and corresponding targets $\mathbf{Y}_{\text{batch}} = \{\mathbf{y}_i\}$.
5: **ii. Forward pass:** For each layer l, compute the output of the network:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}),$$

where $\mathbf{h}^{(0)} = \mathbf{X}_{\text{batch}}$.
6: **iii. Compute loss:** Calculate the loss function $\mathcal{L}(\mathbf{Y}_{\text{batch}}, \hat{\mathbf{Y}}_{\text{batch}})$, where $\hat{\mathbf{Y}}_{\text{batch}}$ is the output from the final layer.
7: **iv. Backpropagation:** Compute the gradient of the loss with respect to the parameters of each layer:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}} \quad \text{for each layer } l.$$

8: **v. Update parameters:** Update the parameters using the gradients and the learning rate η:

$$\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}, \quad \mathbf{b}^{(l)} = \mathbf{b}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}}.$$

9: **end for**
10: **end while**

Training is considered complete (or *converged*) when the model's performance on a held-out validation dataset is deemed satisfactory. One common technique to prevent overtraining is early stopping, where training is halted once the validation performance no longer improves. Given the large number of degrees of freedom (i.e., the network's parameters), care must be taken to avoid overfitting, which occurs when the model fits the training data too closely, resulting in poor generalization to unseen data. Techniques such as regularization, dropout, and cross-validation can also be used to mitigate this risk.

2.1.3 Supervised and Unsupervised Learning

ML training algorithms are broadly categorized as either *supervised* or *unsupervised*.

In supervised learning, a model is trained on a labeled dataset, where each input $\mathbf{x}_i$ is associated with a corresponding target $\mathbf{y}_i$. The goal of supervised learning is to learn a function $f(\mathbf{x})$ that maps inputs to their corresponding outputs, minimizing the error between the predicted target $\hat{\mathbf{y}}$ and the true target $\mathbf{y}$. The most common supervised learning tasks are *classification*, where the target is a discrete label, and *regression*, where the target is a continuous value.

In unsupervised learning, a model is trained on a dataset that does not consist of labeled exemplars. In this setting, there are only $\mathbf{x}_i$, and there are no $\mathbf{y}_i$. The aim is to automatically discover the underlying structure inherent in the dataset. Common unsupervised learning tasks are *clustering*, where similar examples are grouped together, and *dimensionality reduction*, where the number of features used to describe an example is reduced, with the aim of simplifying a supervised learning problem, or to aid in visualization of the dataset.

Artificial neural networks are used in both of these paradigms. In the context of supervised learning, the inputs are mapped to outputs consisting of a categorical vector (for classification) or a single continuous output (for regression). In the context of unsupervised learning, variants of the feed-forward architecture described in the previous sections are typically used, and include models such as autoencoders and Generative Adversarial Networks (GANs) [120, 121]. Whether trained with labeled or unlabeled data, the underlying principles of neural network architecture and backpropagation remain the same.

2.2 Local and Distributed Representations

In an ML problem, data can be represented in various ways. Two common forms of representation are *local* and *distributed* representations.

2.2.1 Local Representations

A local representation is the more traditional approach often used in classical ML. In a local representation, each component x_i of the feature vector $\mathbf{x} = [x_1, x_2, ..., x_n]$ directly corresponds to a specific, interpretable attribute of the data. For example, in a dataset consisting of measurements of various species of flower, a feature vector might include components for petal length, petal width, and color. Each of these dimensions has a concrete and intelligible meaning, and each feature is (ideally) independent of the others.

Local representations have the advantage that they are typically intuitive and interpretable. They often form the basis of classical ML models such as decision trees, linear regression, and support vector machines. However, they may struggle with capturing complex relationships between features, or generalizing well when the data is highly non-linear or abstract. Some forms of data, such as a word from a large natural language corpus, are simply not easily describable using local representations.

2.2.2 Distributed Representations

A distributed representation, on the other hand, is more abstract. Rather than each dimension of the feature vector representing a specific, concrete attribute, a distributed representation encodes information across many dimensions. Each component x_i of the feature vector $\mathbf{x} \in \mathbb{R}^n$ may not have a clear, interpretable meaning by itself. Instead, the data is represented as a point in a high-dimensional space, and the meaning arises from the combination of values across all dimensions. The downside is that a distributed representation is not directly interpretable. However, they are ideal for representing abstract and subtle aspects of an example, such as the meaning of a word in a sentence. Moreover, metrics can be defined over a collection of these vectors to capture semantic relationships between data points. Two common metrics are the *Euclidean distance*,

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{2.4}$$

where $\mathbf{x}_1 = [x_{11}, x_{12}, \ldots, x_{1n}]$ and $\mathbf{x}_2 = [x_{21}, x_{22}, \ldots, x_{2n}]$, and the *cosine similarity*,

$$\text{cosine similarity}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|} \tag{2.5}$$

where $\mathbf{x}_1 \cdot \mathbf{x}_2$ is the dot product of the vectors, and $\|\mathbf{x}_1\|$ and $\|\mathbf{x}_2\|$ are the magnitudes (or norms) of the vectors.

These metrics can quantify how "close" two examples are in the high-dimensional space, with the assumption that smaller distances correspond to higher semantic similarity. While these metrics are often also applied to local representations, proximity in feature space does not always correspond to meaningful semantic similarity, as they do in the case of distributed representations.

Finally, distributed representations allow for transformations that can yield new vectors while preserving semantic coherence. For example, vector addition and subtraction can be used to create new examples based on the relationships between existing ones. An example is the creation of "sentence" vectors by the element-wise summation of the word vectors for the words comprising the sentence [122].

2.3 The Transformer Architecture

The Transformer architecture, introduced by Vaswani *et al.* in the paper "Attention is All You Need" [97], began a paradigm shift in the field of Natural Language Processing (NLP) by moving away from recurrent architectures like RNNs and LSTMs, which had been predominantly used. It introduced a purely attention-based mechanism that allows for more efficient parallelization during training, and is currently the foundation for state-of-the-art models such as BERT and GPT [123, 124].

The Transformer architecture is based on an *encoder-decoder* structure, where both the encoder and decoder consist of multiple layers of identical sub-units, which in turn consist of feed-forward neural network layers. One of its key innovations is its use of multi-head attention, which allows the model to focus on different parts of the input sequence when making predictions.

In this section, I will provide an overview of the Transformer architecture, and of the concept of multi-head attention.

2.3.1 Overview of the Transformer

The Transformer was originally developed for the task of sequence-to-sequence translation. In such a scenario, one would like to transform a sequence of words in one language (e.g. English) to a sequence of words in another language (e.g. French). Thus, the name "Transformer" is apt, as the model transforms one sequence into another. In NLP, traditional models for such tasks included RNNs, which generally iterate over each word in the input sequence, building up to a single distributed representation, which is then used as the basis for a decoding step, which produces each output word sequentially, until a terminating output token is produced. Various mechanisms for emphasizing (or attending to) certain input and output tokens were, over time, introduced atop this architecture. While these models greatly advanced the state-of-the-art in machine translation, they were difficult to scale, to meet the increasing demands of real-time translation. The Transformer architecture was developed to address the shortcomings of the recurrent models.

The Transformer model consists of two main components: the *encoder* and the *decoder*. Each of these components is composed of multiple layers of sub-modules that contain multi-

head attention blocks and feed-forward neural networks. The encoder processes the input data and generates a set of hidden representations, while the decoder uses these hidden representations, along with the previously generated tokens, to produce the output sequence. See Figure 2.1.

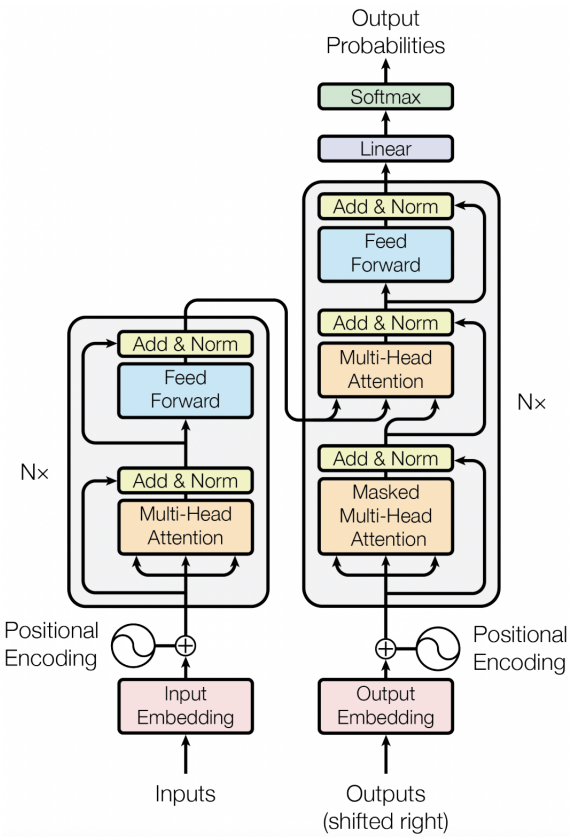

Figure 2.1: The Transformer model architecture, from Vaswani *et al.*. [97]

2.3.2 The Attention Mechanism

At the core of the Transformer is the concept of *attention*, which is a mechanism for computing a weighted representation of the input by focusing on each position in the sequence. The most common form of attention is known as *self-attention*. In self-attention, a weighted representation is computed for each element in the sequence in relation to each other element. This allows the model to capture the importance of a relationship between two elements regardless of the distance between them in the sequence.

Self-attention is computed using three vectors: a *query* vector, a *key* vector, and a *value* vector. For each element of the sequence, the output is a weighted sum of the value vectors, with the weights determined by the similarity between the query and key vectors.

The attention mechanism is what allows the Transformer to process all input tokens in parallel, unlike sequential models such as RNNs.

2.3.3 Multi-Head Attention and Positional Encoding

More formally, the input to the model consists of a sequence. The sequence, $X_{\text{in}} \in \mathbb{R}^{n \times d_{\text{in}}}$, is comprised of d_{in}-dimensional representations for each of the n constituent elements (e.g.

words) of the sequence. These d_{in}-dimensional representations may be categorical vectors in the case of words, for example, or local or pre-trained distributed representations. The first step involves the *positional encoding* of the sequence's elements into X_{in}, resulting in $X_{\text{enc}} \in \mathbb{R}^{n \times d_{\text{model}}}$, where d_{model} is given as a hyperparameter. Positional encoding is used in the Transformer to inject sequence order information, since the model lacks an inherent sequential structure, such as in RNNs. The positional encoding for each position pos and dimension i is defined as:

$$\begin{aligned} PE(pos, 2i) &= \sin(pos/10000^{2i/d_{\text{model}}}) \\ PE(pos, 2i+1) &= \cos(pos/10000^{2i/d_{\text{model}}}) \end{aligned} \tag{2.6}$$

where pos is the position in the sequence, and i is the dimension index. These alternating sine and cosine functions allow the model to encode positional information across different frequencies, capturing both short- and long-range dependencies. The final, encoded input to the model, X_{enc}, is the sum of the input embeddings X_{in} and the positional encodings PE:

$$X_{\text{enc}} = X_{\text{in}} + PE \tag{2.7}$$

This is followed by the sequential application of a number of Transformer blocks. Each Transformer block begins by performing a multi-head self-attention operation. (Figure 2.2) The self-attention operation allows the model to learn to attend to the relationships between the elements of the sequence, in the context of the task. The "attention weights" are encoded into a $n \times n$ matrix, associated with each of h attention heads, by applying the $softmax$ operation to a scaled dot-product of a query, $Q_i \in \mathbb{R}^{n \times d_K}$, and a transposed key, $K_i^T \in \mathbb{R}^{d_K \times n}$, where $d_K = d_{\text{model}}/h$ specifies the key (and query) dimension for an attention head.

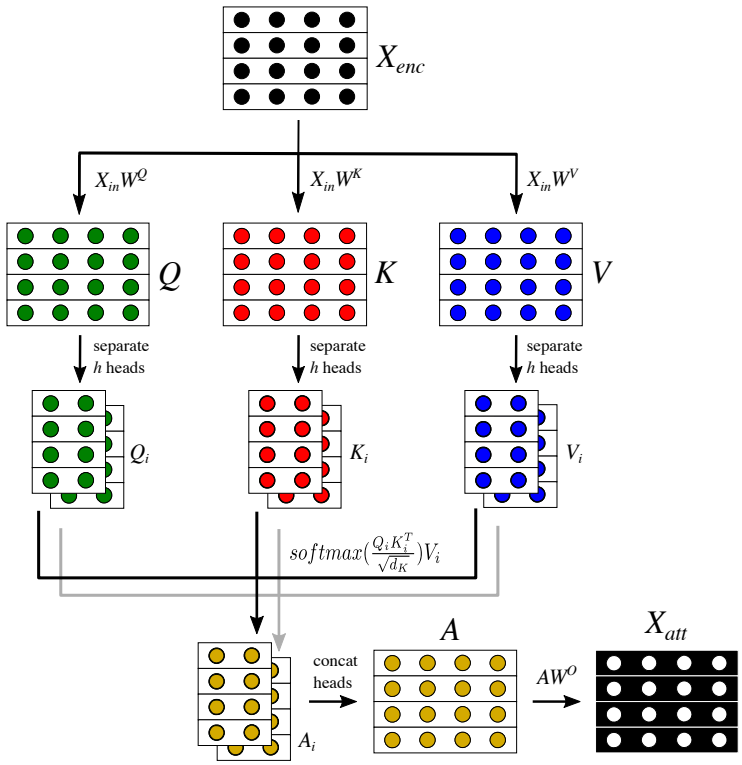

Figure 2.2: Depiction of the multi-head self-attention operation. The input, $X_{\text{enc}} \in \mathbb{R}^{n \times d_{\text{model}}}$, consists of the positionally encoded representations for n constituent elements of the sequence, each with d_{model} components. Linear transformations are applied to the input to produce the query, $Q \in \mathbb{R}^{n \times d_{\text{model}}}$, the key, $K \in \mathbb{R}^{n \times d_{\text{model}}}$, and the value, $V \in \mathbb{R}^{n \times d_{\text{model}}}$, using the learned parameters W^Q, W^K, and W^V, respectively. The query, key and value are each subsequently separated into h heads (indexed here by i). The corresponding queries, Q_i, keys, K_i, and values, V_i, are combined to produce the attention products, A_i, by multiplying the $softmax$ of a scaled dot-product of the queries and keys with the values. After the attention products are concatenated to produce A, a linear transformation of A using the learned parameters W^O produces the output of multi-head self-attention, $X_{\text{att}} \in \mathbb{R}^{n \times d_{\text{model}}}$.

The Transformer block follows the multi-head self-attention operation with layer normalization [125], dropout [126], and feed-forward $ReLU$ operations (Figure 2.3). The output of a Transformer block, $X_{\text{out}} \in \mathbb{R}^{n \times d_{\text{model}}}$, thus consists of the same dimensions as the input, which allows multiple Transformer blocks to be connected serially.

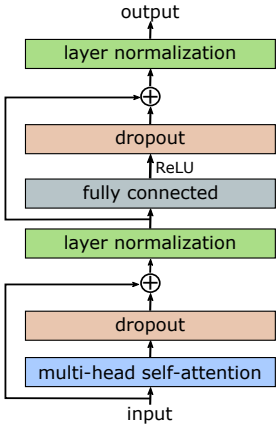

Figure 2.3: Components of the Transformer block.

2.4 Autoregressive Large Language Modeling

Autoregressive models, in the context of deep learning, are a class of generative models used for the production of text, via the prediction of the next token of text given the preceding tokens. Such models undergo an unsupervised *pre-training* procedure, where they are tasked with predicting which token is most likely to follow a sequence of preceding tokens.

While language modeling pre-dates deep learning, Alec Radford and colleagues demonstrated in 2018 that subjecting deep neural networks to autoregressive pre-training resulted in state-of-the-art natural language models [124]. These models, which came to be known as the Generative Pre-trained Transformer (GPT), are based on the decoder portion of the Transformer architecture, and have since become foundational to the modern NLP workflow. It was proposed that the Transformer architecture could be scaled up in size, suggesting that improved performance could be achieved by increasing the amount of training data, computational power, and model parameters. Developments such as ChatGPT, a revolutionary chatbot capable of achieving human-level performance on many intellectual tasks, have since validated these ideas, as the size of these GPT models has increased from millions to hundreds of billions of parameters [127].

The following sections introduce the fundamentals of language modeling and describe how LLMs use autoregressive pre-training to achieve state-of-the-art performance on a variety of NLP tasks.

2.4.1 Language Modeling

Language modeling refers to the estimation of the probability distribution over sequences of tokens in a given language. Formally, given a sequence of tokens $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the task of a language model is to estimate the joint probability $P(\mathbf{x})$. This is often factorized as the product of conditional probabilities using the chain rule:

$$P(\mathbf{x}) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)\ldots P(x_n|x_1, \ldots, x_{n-1}) \tag{2.8}$$

The objective of the model is to maximize the likelihood of the observed sequences in a training dataset. In practice, language models are typically trained on large corpora of text, and are produced through Maximum Likelihood Estimation (MLE). Traditionally, language models have been limited to estimating distributions over fixed numbers of tokens, and are

referred to as n-gram language models. However, deep neural network-based language models, such as the GPT model, are much more powerful and flexible, with more recent versions having practically no limit on the number of preceding tokens which are considered when predicting the next token.

2.4.2 Large Language Models and Autoregressive Pre-training

Large language models, such as GPT, are characterized by the incorporation of a deep neural network with typically millions or more parameters. These models must undergo an unsupervised pre-training step before they can be fine-tuned for more specialized, downstream tasks, such as sentiment classification, for example. Fine-tuning usually involves replacing the model's original output layer with a layer specialized for the downstream task. The original output layer produces next-token probabilities. Thus, the pre-trained (or foundation) model generates text one token at a time, with each predicted token being conditioned on the previously generated tokens. The model follows the autoregressive process:

$$P(x_t|x_1, x_2, \ldots, x_{t-1}) \tag{2.9}$$

Autoregressive pre-training requires a vocabulary, $\mathcal{V}$, and an ordered list of tokens $\mathcal{U} = (u_1, ..., u_n)$, with $u_i \in \mathcal{V}$, such as may be obtained by concatenating the tokenized documents of a corpus end-to-end. The objective is to maximize the following likelihood:

$$\mathcal{L}(\theta;\mathcal{U}) = \sum_i \log P(u_i|u_{i-c}, ..., u_{i-1}; \theta) \tag{2.10}$$

where c is the size of a context window, P is the conditional probability distribution to be modelled, and θ the parameters of a neural network. Therefore, $\mathcal{J}(\theta;\mathcal{U}) = -\mathcal{L}$ is the objective to be minimized, using stochastic gradient descent to adjust the parameters. In practice, the cross-entropy loss between the model's predicted probabilities and the true one-hot encoded labels is used to establish the difference between the actual and desired outputs.

Chapter 3

Distributed Representations of Atoms and Materials

3.1 Introduction

A central problem in materials science is the rational design of materials with specific properties. Typically, useful materials have been discovered serendipitously [128]. With the advent of ubiquitous and capable computing infrastructure, materials discovery has been increasingly aided by computational chemistry, especially density functional theory (DFT) simulations [129]. Such theoretical calculations are indispensable when investigating the properties of novel materials. However, they are computationally intensive, and performing such analysis on large numbers of compounds (there are more than 10^{10} chemically sensible stoichiometric quaternary compounds possible [114]) becomes impractical with today's computing technology. Moreover, certain chemical systems, such as those with very strongly correlated electrons, or with high levels of disorder, remain a theoretical challenge to DFT [130, 131].

The application of ML to materials science aims to ameliorate some of these problems, by providing alternate computational routes to properties of interest. There have been numerous examples of the successful application of ML to chemical systems. Techniques from ML have been used to predict very local and detailed properties, such as atomic and molecular orbital energies and geometries [132] or partial charges [133], and also global properties, such as the formation energy and band gap of a given compound [134–137].

For a ML algorithm to work effectively, the objects of the system of interest must be converted into faithful representations that can be consumed in a computational context. Deriving such representations has been a main focus for researchers in ML, and in the case of Deep Learning, such representations are typically learned automatically, as part of the training process [96]. Related to this are the concepts of Unsupervised Learning, where patterns in the data are derived without the use of labels or other forms of supervision [138], and Semi-supervised Learning, where a small amount of labelled data is combined with large amounts of unlabelled data [139–142]. Indeed, given that most data is unlabelled, such techniques are very valuable. Some of the most successful and widely used algorithms, such as Word2Vec from the field of Natural Language Processing (NLP), use unsupervised learning to derive effective representations of the objects in the system of interest (words, in this case) [100, 143].

The most basic object of interest in chemical systems is very often the atom. Thus, there have already been several investigations examining the derivation of effective machine representations of atoms in an unsupervised setting [102, 103, 144], and other investigations have aimed to learn good atomic representations in the context of a supervised learning task [145, 146]. A learned representation of an atom generally takes the form of an embedding,

which can be described as a relatively low-dimensional space in which higher-dimensional vectors can be expressed. Using embeddings in a ML task is advantageous, as the number of input dimensions is typically lower than if higher-dimensional sparse vectors were used. Moreover, embeddings which are semantically similar reside closer together in space, which provides a more principled structure to the input data. Such representations should allow ML models to learn a task more quickly and effectively.

A widely held hypothesis in ML is that unlabelled data can be used to learn effective representations. In this work, we introduce an approach for learning atomic representations using an unsupervised approach. This approach, which we name SkipAtom, is inspired by the Skip-gram model in NLP, and takes advantage of the large number of inorganic structures in materials databases. We also investigate forming representations of chemical compounds by pooling atomic representations. Combining vectors by various pooling operations to create representations of systems composed from parts (e.g. sentences from words) is a common technique in NLP, but apparently remains largely unexplored in materials informatics [147]. The analogy we explore here is that atoms are to compounds as words are to sentences, and our results demonstrate that effective representations of compounds can be composed from the vector representations of the constituent atoms. Finally, a common problem when searching chemical space for new materials is that the structure of a compound may not be known. Since the properties of a material are typically tightly coupled to its structure, this creates a significant barrier [148]. Here, we compare our models, which operate on representations derived from chemical formulas only, to benchmarks that are based on models that use structural information. We find that, for certain tasks, the performance of the composition-only models is comparable.

3.2 Methods

3.2.1 Representations of Atoms and Compounds

There are various strategies for providing an atom with a machine representation. These range from very simple and unstructured approaches, such as assigning a random vector to each atom, to more sophisticated approaches, such as learning distributed representations. A distributed representation is a characterization of an object attained by embedding in a continuous vector space, such that similar objects will be closer together.

Similarly, a compound may be assigned a machine representation. Again, these representations may be learned on a case-by-case basis, or they may be formed by composing existing representations of the corresponding atoms.

Atomic Representations

We are interested in deriving representations of atoms that can be used in a computational context, such as a ML task. Intuitively, we would like the representations of similar atoms to be similar as well. Given that atoms are multifaceted objects, a natural choice for a computational descriptor for an atom might be a vector: an n-tuple of real numbers. Vector spaces are well understood, and can provide the degrees of freedom necessary to express the various facets that constitute an atom. Moreover, with an appropriately selected vector space, such atomic representations can be subjected to the various vector operations to quantify relationships and to compose descriptions of systems of atoms, or compounds.

Random Vectors

The simplest approach to assigning a vector description to an atom is to simply draw a random vector from $\mathbb{R}^n$, and assign it to the atom. Such vectors can come from any distribution desired, but in this report, such vectors will come from the standard normal distribution, $\mathcal{N}(0, 1)$.

One-hot Vectors

One-hot vectors, common in ML, are binary vectors that are used for distinguishing between various categories. One assigns a vector component to each category of interest, and sets the value of the corresponding component to 1 when the vector is describing a given category, and the value of all other components to 0. More formally, a one-hot n-dimensional vector $\mathbf{v}$ is in the set $\{0, 1\}^n$ such that $\sum_{i=1}^{n} v_i = 1$, where v_i is a component of $\mathbf{v}$. A unique one-hot vector is assigned to each category. In the context of this report, a category is an atom (Figure 3.1a).

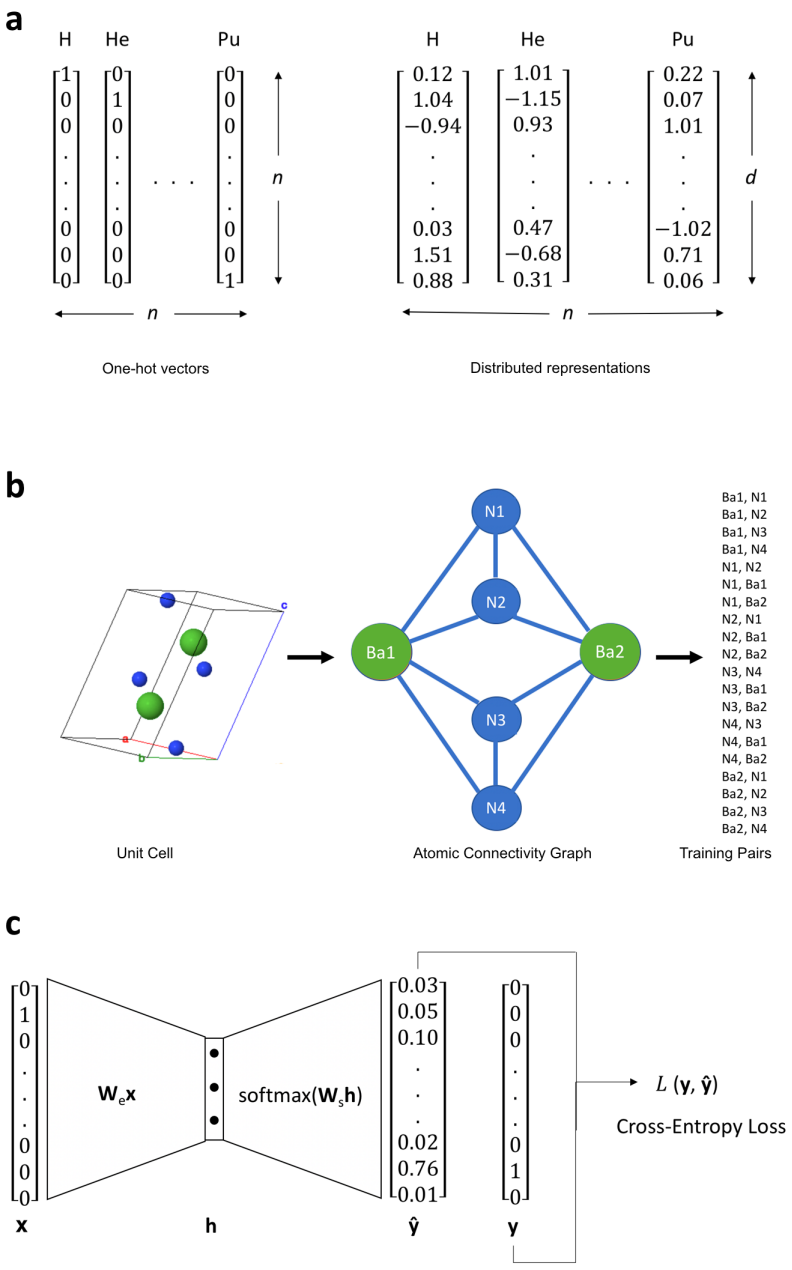

Figure 3.1: **a** Scheme illustrating one-hot and distributed representations of atoms. In the diagram, there are n atoms represented, and d is the adjustable number of dimensions of the distributed representation. Note that the atoms in this example are H, He and Pu, but they could be any atom. **b** Scheme describing how training data is derived for the creation of SkipAtom vectors. Here, a graph representing the atomic connectivity in the structure of Ba_2N_4 is depicted, and the resulting target-context atom pairs derived for training. The graph is derived from the unit cell of Ba_2N_4. **c** Scheme describing how the SkipAtom vectors are derived through training. Here, a one-hot vector, $\mathbf{x}$, representing a particular atom is transformed into an intermediate vector $\mathbf{h}$ via multiplication with matrix $\mathbf{W}_\mathrm{e}$. The matrix $\mathbf{W}_\mathrm{e}$ is the embedding matrix, whose columns will be the final atom vectors after training. Training consists of minimizing the cross-entropy loss between the output vector $\hat{\mathbf{y}}$ and the one-hot vector representing the context atom, $\mathbf{y}$. The output $\hat{\mathbf{y}}$ is obtained by applying the softmax function to the product $\mathbf{W}_\mathrm{s}\mathbf{h}$.

Atom2Vec

If one may know a word by the company it keeps, then the same might be said of an atom. In 2018, Zhou and coworkers described an approach for deriving distributed atom vectors that involves generating a co-occurrence count matrix of atoms and their chemical environments, using an existing database of materials, and applying Singular Value Decomposition (SVD) to the matrix [102]. The number of dimensions of the resulting atomic vectors is limited to the number of atoms used in the matrix.

Mat2Vec

A popular means of generating word vectors in NLP is through the application of the Word2Vec algorithm, wherein an unsupervised learning task is employed [143]. Given a corpus (a collection of text), the goal is to predict the likelihood of a word occurring in the context of another. A neural network architecture is employed, and the learned parameters of the projection layer constitute the word vectors that result after training. In 2019, Tshitoyan and coworkers described an approach for deriving distributed atom vectors by making direct use of the materials science literature [103]. Instead of using a database of materials, they assembled a textual corpus from millions of scientific abstracts related to materials science research, and then applied the Word2Vec algorithm to derive the atom representations.

SkipAtom

In the NLP Skip-gram model, an occurrence of a word in a corpus is associated with the words that co-occur within a context window of a certain size. The task is to predict the context words given the target word. Although the aim is not to build a classifier, the act of tuning the parameters of the model so that it is able to predict the context of a word results in a parameter matrix that acts effectively as the embedding table for the words in the corpus. Words that share the same contexts should share similar semantic content, and this is reflected in the resulting learned low-dimensional space. Analogously, atoms that share the same chemo-structural environments should share similar chemistry.

In the SkipAtom approach, the crystal structures of materials from a database are used in the form of a graph, representing the local atomic connectivity in the material, to derive a dataset of connected atom pairs (Figure 3.1b). Then, similarly to the Skip-gram approach of the Word2Vec algorithm, Maximum Likelihood Estimation is applied to the dataset to learn a model that aims to predict a context atom given a target atom.

More formally, a materials database consists of a set of materials, M. A material, $m \in M$, can be represented as an undirected graph, consisting of a set of atoms, A_m, comprising the material, and bonds $B_m \subseteq \{(x, y) \in A_m \times A_m | x \neq y\}$, which are unordered pairs of atoms. The task is to maximize the average log probability:

$$\frac{1}{|M|} \sum_{m \in M} \sum_{a \in A_m} \sum_{n \in N(a)} \log p(n|a) \tag{3.1}$$

where $N(a)$ are the neighbors of a (not including a itself); more specifically: $N(a) = \{x \in A_m | (a, x) \in B_m\}$.

In practice, this means that the cross-entropy loss between the one-hot vector representing the context atom and the normalized probabilities produced by the model, given the one-hot vector representing the target atom, is minimized (Figure 3.1c).

The graph representing a material can be derived using any approach desired, but in this work, an approach is used which is based on Voronoi decomposition [149], which identifies nearest neighbors using solid angle weights to determine the probability of various coordination environments [150, 151]. (See Supplementary Note 3 in Appendix A for more information about how the graphs are derived.)

The result of SkipAtom training is a set of vectors, one for each atom of interest (Figure 3.1a), that reflects the unique chemical nature of the represented atom, as well as its relationship to other atoms.

A complicating factor in the procedure just described is that some atoms may be underrepresented in the database, relative to others. This will result in the parameters of those infrequently occurring atoms receiving fewer updates during training, resulting in lower quality representations for those atoms. This is an issue when learning word representations as well, and there have been several solutions proposed in the context of NLP [152, 153]. Borrowing from these solutions, we apply an additional, optional processing step to the learned vectors, termed *induction*. The aim is to adjust the learned vectors so that they reside in a more sensible area of the representation space. To achieve this, each atom is first represented as a triple, given by its periodic table group number and row number, and its electronegativity. Then, for each atom, the closest atoms are obtained, in terms of the cosine similarity between the vectors formed from these triples. Using the learned embeddings for these closest atoms, a *mean nearest-neighbor representation* is derived, and the induced atom vector, $\hat{\mathbf{u}}$, is formed by adding the original atom vector, $\mathbf{u}$, to the mean nearest neighbor:

$$\hat{\mathbf{u}} = \mathbf{u} + \frac{1}{N}\sum_{k=0}^{N} e^{-k}\mathbf{v}_k \tag{3.2}$$

where N is the number of closest atoms to consider, and $\mathbf{v}_k$ is the learned embedding of the k^{th} nearest atom from the sorted list of nearest atoms. In this work, the nearest 5 atoms are considered.

Compound Representations

Atom vectors by themselves may not be directly useful, as most problems in materials informatics involve chemical compounds. However, atom vectors can be combined to form representations of compounds.

Atom Vector Pooling

The most basic and general way of combining atom vectors to form a representation for a compound is to perform a pooling operation on the atom vectors corresponding to the atoms in the chemical formula for the compound. There are three common pooling operations: sum-pooling, mean-pooling, and max-pooling.

Sum-pooling involves performing component-wise addition of the atom vectors for the atoms in the chemical formula. That is, for a chemical compound whose formula is comprised of m constituent elements, and a set of atom vectors, $\mathbf{v} \in V$, the compound vector, $\mathbf{w}$, is given in this case by:

$$\mathbf{w} = \sum_{k=1}^{m} c_k \mathbf{v}_k \tag{3.3}$$

where $\mathbf{v}_k$ is the corresponding atom vector for the k^{th} constituent element in the formula, and c_k is the relative number of atoms of the k^{th} constituent element (which need not be a whole number, as in the case of non-stoichiometric compounds).

Mean-pooling involves performing component-wise addition of the atom vectors for the atoms in the chemical formula, followed by dividing by the total number of atoms in the formula. In this case:

$$\mathbf{w} = \frac{\sum_{k=1}^{m} c_k \mathbf{v}_k}{\sum_{k=1}^{m} c_k} \tag{3.4}$$

Finally, *max-pooling* involves taking the maximum value for each component of the vectors being pooled. In this case:

$$\mathbf{w} = \max_{k=1}^{m} c_k \mathbf{v}_k \tag{3.5}$$

where $\max$ returns a vector where each component has the maximum value of that component across n input vectors.

ElemNet (Mean-pooled One-hot Vectors)

If we assign a unique one-hot vector to each atom, and perform mean-pooling of these vectors when forming a representation for a chemical compound, then the result is the same as the input representation for the ElemNet model [145]. Such a compound vector is sparse (as most compounds do not typically contain more than 5 or 6 atom types). Each component of the vector contains the unit normalized amount of the atom in the formula. For example, for H_2O, the component corresponding to H would have a value of 0.66 whereas the component corresponding to O would have a value of 0.33, and all other components would have a value of zero.

Bag-of-Atoms (Sum-pooled One-hot Vectors)

In NLP, the Bag-of-Words is a common representation used for sentences and documents. It is formed by simply performing sum-pooling of the one-hot vectors for each word in the text. Similarly, we can conceive of a Bag-of-Atoms representation for chemical informatics, where sum-pooling is performed with the one-hot vectors for the atoms in a chemical formula. The result is a list of counts of each atom type in the formula. This is an unscaled version of the ElemNet representation. Crucially, this sum-pooling of one-hot vectors is more appropriate for describing compounds than it is for describing natural language sentences, as there is no significance to the order of atoms in a chemical formula as there is for the order of words in a sentence.

3.2.2 Evaluation Tasks

A number of diverse materials ML tasks are utilized to evaluate the effectiveness of the pooled atom vector representations, and the quality of the SkipAtom representation. In total, ten previously described tasks are utilized, and are broadly divided into two categories: those used for evaluating the pooling approach, and those used for evaluating the SkipAtom approach. To evaluate the pooling approach, nine tasks are chosen, and are described in Table 3.1.

Table 3.1: The predictive tasks utilized in this study to evaluate the atom vector pooling approach. All datasets and benchmarks for the tasks above are described in [154], with the exception of the Formation Energy task, which is described in [145].

Task	Type	Examples	Structure?	Method
Band Gap (eV)	Regression	4,604	No	Experiment [135]
Band Gap (eV)	Regression	106,113	Yes	DFT-GGA [155, 156]
Bulk Modulus (log(GPa))	Regression	10,987	Yes	DFT-GGA [157]
Shear Modulus (log(GPa))	Regression	10,987	Yes	DFT-GGA [157]
Refractive Index (n)	Regression	4,764	Yes	DFPT-GGA [158]
Formation Energy (eV/atom)	Regression	275,424	Yes	DFT [145, 159]
Bulk Metallic Glass Formation	Classification	5,680	No	Experiment [160, 161]
Metallicity	Classification	4,921	No	Experiment [135]
Metallicity	Classification	106,113	Yes	DFT-GGA [155, 156]

The tasks were chosen to represent the various scenarios encountered in materials data science, such as the availability of both smaller and larger datasets, the need for either regression or classification, the availability of material structure information, and the means (experiment or theory) by which the data is obtained. The OQMD (Open Quantum Materials Database) Formation Energy task [145, 159] requires a different training protocol, as it was derived from a different study than the other eight tasks that are used for the pooling approach, which were sourced from the Matbench test suite [154].

To evaluate the SkipAtom representation, the Elpasolite Formation Energy task was utilized. The task and the model were initially described in the paper that introduced Atom2Vec (an alternative approach for learning atom vectors) [102]. The task consists of predicting the formation energy of elpasolites, which are comprised of a quaternary crystal structure, and have the general formula ABC_2D_6. The target formation energies for 5,645 examples were obtained by DFT [162]. The input consists of a concatenated sequence of atom vectors, each representing the A, B, C, and D atoms. We reproduce the approach here, for comparison against the Atom2Vec results.

All tasks require a representation of a material as input, and produce a prediction of a physical property as output, in either a regression or classification setting. Moreover, with the exception of the Elpasolite Formation Energy task, all tasks make use of the same model architecture (described in detail below).

3.2.3 Pooling Approach Evaluation

For the purposes of evaluation, the atom and compound vectors were utilized as inputs to feed-forward neural networks. All results for evaluating the pooling approach were obtained using a 17-layer feed-forward neural network architecture based on ElemNet [145]. The network was comprised of 4 layers with 1,024 neurons, followed by 3 layers with 512 neurons, 3 layers with 256 neurons, 3 layers with 128 neurons, 2 layers with 64 neurons, and 1 layer with 32 neurons, all with ReLU activation. For regression tasks, the output layer consisted of a single neuron and linear activation. For classification tasks, the output layer consisted of a single neuron and sigmoid activation (as only binary classification was performed). Instead of using dropout layers for regularization, as in the ElemNet approach, L2 regularization was used, with a regularization constant of 10^{-5}. The goal during training was to minimize the Mean Absolute Error loss (for regression tasks), or the Binary Cross-entropy loss (for classification tasks). All pooling approach experiments utilized a mini-batch size of 32, and a learning rate of 10^{-4} along with the Adam optimizer (with an epsilon parameter of 10^{-8}) [163]. As described in the paper that introduces the Matbench test set [154], k-fold cross-validation

was performed to evaluate the compound vectors in regression tasks, with the same random seed to ensure the same splits were used each time. For classification tasks, stratified k-fold cross-validation was performed. As required by the benchmarking protocol, 5 splits were used (with the exception of the OQMD Formation Energy prediction task, which used 10 splits). Because the variance was high for some tasks after k-fold cross-validation, repeated k-fold cross-validation was performed, to reduce the variance [164]. All training was carried out for 100 epochs, and the best performing epoch was chosen as the result for that split. By following this protocol, a direct and fair comparison can be made to results reported previously using the same Matbench test set [154].

3.2.4 Elpasolite Formation Energy Prediction

The results for evaluating the SkipAtom approach were obtained using the Elpasolite neural network architecture and protocol, originally described in the paper that introduces Atom2Vec [102]. The input to the neural network is a vector constructed by concatenating 4 atom vectors, representing each of the 4 atoms in an Elpasolite composition. The single hidden layer consists of 10 neurons, with ReLU activation. The output layer consists of a single neuron, with linear activation. L2 regularization was used, with a regularization constant of 10^{-5}. The goal during training was to minimize the Mean Absolute Error loss. The training protocol differs slightly in this report, and 10-fold cross-validation was performed, utilizing the result after 200 epochs of training. The same random seed was used for all experiments, to ensure the same splits were utilized. A mini-batch size of 32 was utilized, and a learning rate of 10^{-3} along with the Adam optimizer (with an epsilon parameter of 10^{-8}) was chosen [163].

3.2.5 SkipAtom Training

Learning of the SkipAtom vectors involved the use of the Materials Project database [165]. To assemble the training set, 126,335 inorganic compound structures were downloaded from the database. Each of these structures was converted into a graph representation using an approach based on Voronoi decomposition [149–151], and a dataset of co-occurring atom pairs was derived. (See Supplementary Note 3 in Appendix A for more information on graph derivation.) A total of 15,360,652 atom pairs were generated, utilizing 86 distinct atom types. The architecture consisted of a single hidden layer with linear activation, whose size depended on the desired dimensionality of the learned embeddings, and an output layer with 86 neurons (one for each of the utilized atom types) with $softmax$ activation. The training objective consisted of minimizing the cross-entropy loss between the predicted context atom probabilities and the one-hot vector representing the context atom, given the one-vector representing the target atom as input. Training utilized stochastic gradient descent with the Adam optimizer, with a learning rate of 10^{-2} and a mini-batch size of 1,024, for 10 epochs.

3.2.6 Data Availability

The data that support the findings described this chapter are available as follows: The materials data that was used to learn the SkipAtom embeddings are publicly available online at https://materialsproject.org/. The elpasolite formation energy training data are publicly available online at https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.117.135502, in the Supplemental Material section. The datasets comprising the Matbench tasks are publicly available at https://hackingmaterials.lbl.gov/automatminer/datasets.html. The Mat2Vec pretrained embeddings are publicly available online and can be downloaded by following the instructions at https://github.com/materialsintelligence/mat2vec. The Atom2Vec embeddings

are publicly available online and can be obtained from https://github.com/idocx/Atom2Vec. The processed data that is used in this study, as well as scripts for reproducing the experiments, can be found on the GitHub repository at the address https://github.com/lantunes/skipatom. Any other relevant data from this work is available from the authors upon reasonable request.

3.2.7 Code Availability

The code for creating and using the SkipAtom vectors is open source, released under the GNU General Public License v3.0. The code repository is accessible online, at:
https://github.com/lantunes/skipatom
The repository also contains pre-trained 200-dimensional SkipAtom vectors for 86 atom types that can be immediately used in materials informatics projects.

3.3 Results and Discussion

3.3.1 Evaluation of Atom Vectors

A common technique for making high-dimensional data easier to visualize is t-SNE (t-Stochastic Neighbor Embedding) [166]. Such a technique reduces the dimensionality of the data, typically to 2 dimensions, so that it can be plotted. Visualizing learned distributed representations in this way can provide some intuition regarding the quality of the embeddings and the structure of the learned space. In Figure 3.2, the 200-dimensional learned SkipAtom vectors are plotted after utilizing t-SNE to reduce their dimensionality to 2. It is evident that there is a logical structure to the data. We see that the alkali metals are clustered together, as are the light non-metals, for example. The relative locations of the atoms in the plot reflect chemo-structural nuances gleaned from the dataset, and are not arbitrary.

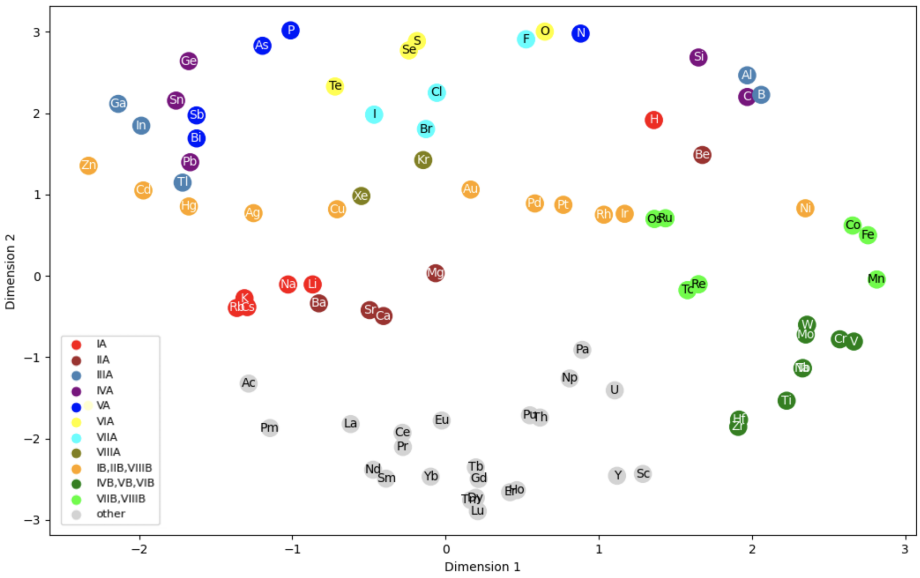

Figure 3.2: Dimensionally-reduced SkipAtom atom vectors with an original size of 200 dimensions. The vectors were reduced to 2 dimensions using t-SNE. (See also Supplementary Figures 1 and 2 in Appendix A for results of dimensionality reduction with PCA.)

To properly evaluate the quality of a learned distributed representation, they are utilized in the context of a task, and their performance compared to other representations. Here, we use the Elpasolite Formation Energy prediction task, and compare the performance of the SkipAtom vectors to the performance of other representations, namely, to Random vectors, One-hot vectors, Mat2Vec and Atom2Vec vectors. In the original study that introduced the task, atom vectors were 30- and 86-dimensional. We trained SkipAtom vectors with the same dimensions, and also with 200 dimensions, and evaluated them. The results are summarized in Table 3.2.

Table 3.2: Elpasolite Formation Energy prediction results after 10-fold cross-validation; mean best formation energy MAE on the test set after 200 epochs of training in each fold. Batch size was 32, learning rate was 0.001. Note that Dim refers to the dimensionality of the atom vector; the size of the input vector is 4 $\times$ Dim. All results were generated using the same procedure on identical train/test folds.

Representation	**Dim**	**MAE (eV/atom)**
Atom2Vec	30	0.1477 $\pm$ 0.0078
SkipAtom	30	0.1183 $\pm$ 0.0050
Random	30	0.1701 $\pm$ 0.0081
Atom2Vec	86	0.1242 $\pm$ 0.0066
One-hot	86	0.1218 $\pm$ 0.0085
SkipAtom	86	0.1126 $\pm$ 0.0078
Random	86	0.1190 $\pm$ 0.0085
Mat2Vec	200	0.1126 $\pm$ 0.0058
SkipAtom	**200**	**0.1089 $\pm$ 0.0061**
Random	200	0.1158 $\pm$ 0.0050

For all embedding dimension sizes, SkipAtom outperforms the other representations on the Elpasolite Formation Energy task (Mat2Vec vectors were only available for this study in 200 dimensions, and Atom2Vec vectors, by virtue of how they are created, cannot have more dimensions than atom types represented). In Figure 3.3, a plot of how the mean absolute error changes during training demonstrates that the SkipAtom representation achieves better results from the beginning of training, and maintains the performance throughout.

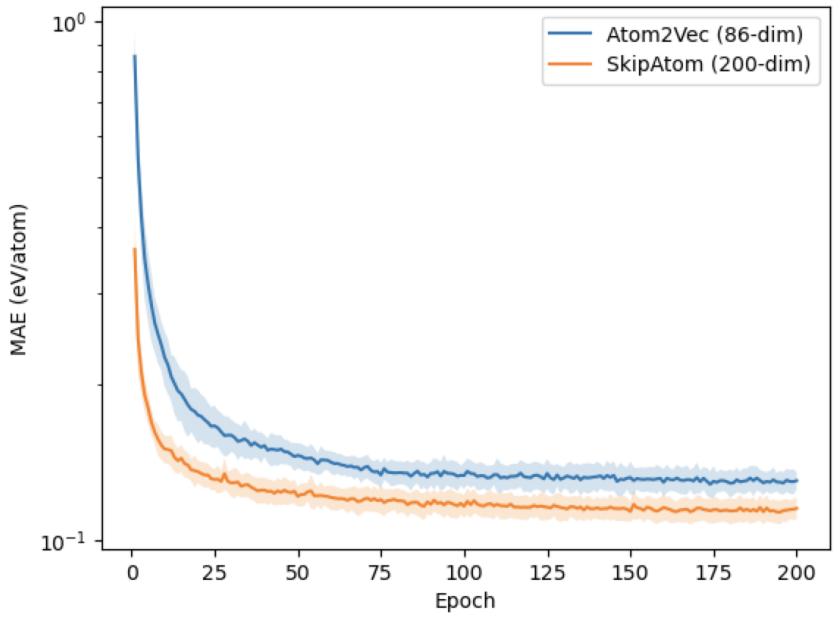

Figure 3.3: Mean absolute error during training for the Elpasolite Formation Energy prediction task, for the Atom2Vec and SkipAtom representations. The average MAE over 10 folds is plotted.

3.3.2 Evaluation of Compound Vectors

Similar to atom vectors, compound vectors formed by the pooling of atom vectors can be dimensionally reduced, and visualized with t-SNE, or with PCA (Figure 3.4a). In Figure 3.4b, a sampling of several thousand compound vectors, formed by the sum-pooling of one-hot vectors, were reduced to 2 dimensions using t-SNE, and plotted. Additionally, since each compound vector represents a compound in the OQMD dataset, which contains associated formation energies, a color is assigned to each point in the plot denoting its formation energy. A clear distinction can be made across the spectrum of compounds and their formation energies. The vector representations derived from the composition of atom vectors appear to have preserved the relationship between atomic composition and formation energy.

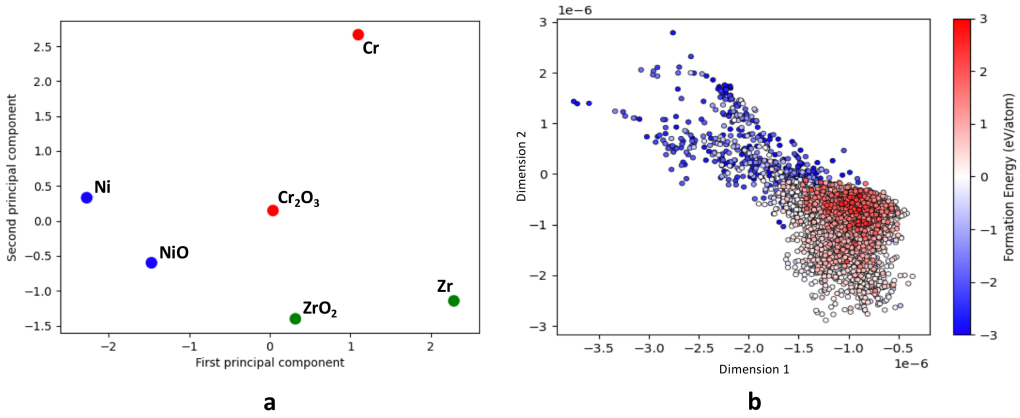

Figure 3.4: **a** 200-dimensional SkipAtom vectors for Cr, Ni, and Zr, and their mean-pooled oxides, dimensionally reduced using PCA. **b** Plot of a sampling of the dimensionally-reduced compound vectors for the OQMD Dataset Formation Energy task, mapped to their associated physical values. The points are sum-pooled one-hot vectors reduced using t-SNE with a Hamming distance metric. The sum-pooled one-hot representation was the best performing for the task.

Again, as with atom vectors, the quality of a compound vector is best established by comparing its performance in a task. To evaluate the quality of pooled atom vectors, 9 predictive tasks were utilized, as described in Table 3.1. The performance on the benchmark regression tasks is summarized in Table 3.3, and the performance on the benchmark classification tasks is summarized in Table 3.4. Finally, the performance on the OQMD Formation Energy prediction task is summarized in Table 3.5.

Table 3.3: Benchmark regression task results after 2-repeated 5- or 10-fold cross-validation; mean best MAE on the test set after 100 epochs of training in each fold. All results were generated using the same procedure on identical train/test folds. TBG refers to the Theoretical Band Gap task (MAE in eV), BM to the Bulk Modulus task (MAE in log(GPa)), SM to the Shear Modulus task (MAE in log(GPa)), and RI to the Refractive Index task (MAE in n). These tasks make use of structure information. EBG refers to the Experimental Band Gap task (MAE in eV), and it makes use of composition only. Only the best results for each representation are reported. The pooling procedure varies between results; blue results represent sum-pooling, red results represent mean-pooling, and teal results represent max-pooling. Numbers in parentheses represent the standard deviation to one part in 10^4. See the Supplementary Information tables in Appendix A for more detailed results.

Representation	Dim	EBG	TBG	BM	SM	RI
SkipAtom	86	0.3495*(20)*	0.2791*(8)*	0.0789*(2)*	**0.1014***(1)*	0.3275*(4)*
Atom2Vec	86	0.3922*(87)*	0.2692*(8)*	0.0795*(5)*	0.1029*(0)*	0.3308*(16)*
Bag-of-Atoms / One-hot	86	0.3797*(22)*	0.2611*(8)*	0.0861*(2)*	0.1137*(5)*	0.3576*(2)*
ElemNet / One-hot	86	0.4060*(72)*	**0.2582***(3)*	0.0853*(1)*	0.1155*(1)*	0.3409*(16)*
One-hot	86	0.3823*(46)*	0.2603*(4)*	0.0861*(3)*	0.1140*(2)*	0.3547*(13)*
Random	86	0.4109*(58)*	0.3180*(16)*	0.0908*(4)*	0.1195*(2)*	0.3593*(6)*
Mat2Vec	200	0.3529*(7)*	0.2741*(2)*	**0.0776***(0)*	**0.1014***(2)*	**0.3236***(17)*
SkipAtom	200	**0.3487***(85)*	0.2736*(8)*	0.0785*(0)*	**0.1014***(0)*	0.3247*(15)*
Random	200	0.4058*(4)*	0.3083*(21)*	0.0871*(1)*	0.1163*(2)*	0.3543*(6)*

Table 3.4: Benchmark classification task results after 2-repeated 5-fold stratified cross-validation; mean best ROC-AUC on the test set after 100 epochs of training in each fold. All results were generated using the same procedure on identical train/test folds. TM refers to the Theoretical Metallicity task, and makes use of structure information. BMGF refers to the Bulk Metallic Glass Formation task, and EM to the Experimental Metallicity task. These last two do not make use of structure information. Only the best results for each representation are reported. The pooling procedure varies between results; blue results represent sum-pooling, red results represent mean-pooling, and teal results represent max-pooling. See the Supplementary Information tables in Appendix A for more detailed results.

Representation	Dim	TM	BMGF	EM
SkipAtom	86	0.9520 ± 0.0002	0.9436 ± 0.0010	0.9645 ± 0.0012
Atom2Vec	86	0.9526 ± 0.0001	0.9316 ± 0.0012	0.9582 ± 0.0008
Bag-of-Atoms / One-hot	86	0.9490 ± 0.0002	0.9277 ± 0.0004	0.9600 ± 0.0012
ElemNet / One-hot	86	0.9477 ± 0.0001	0.9322 ± 0.0014	0.9485 ± 0.0007
One-hot	86	0.9487 ± 0.0003	0.9289 ± 0.0016	0.9599 ± 0.0014
Random	86	0.9444 ± 0.0000	0.9274 ± 0.0006	0.9559 ± 0.0021
Mat2Vec	200	**0.9528 ± 0.0002**	0.9348 ± 0.0024	**0.9655 ± 0.0014**
SkipAtom	200	0.9524 ± 0.0001	**0.9349 ± 0.0019**	0.9645 ± 0.0008
Random	200	0.9453 ± 0.0001	0.9302 ± 0.0016	0.9541 ± 0.0002

Table 3.5: OQMD Dataset Formation Energy prediction results after 10-fold cross-validation; mean best formation energy MAE on the test set after 100 epochs of training in each fold. All results were generated using the same procedure on identical train/test folds.

Representation	Dim	Pooling	MAE (eV/atom)
SkipAtom	86	sum	0.0420 ± 0.0005
Atom2Vec	86	sum	0.0396 ± 0.0004
Bag-of-Atoms / One-hot	**86**	**sum**	**0.0388 ± 0.0002**
ElemNet / One-hot	86	mean	0.0427 ± 0.0007
Random	86	sum	0.0440 ± 0.0004
Mat2Vec	200	sum	0.0401 ± 0.0004
SkipAtom	200	sum	0.0408 ± 0.0003
Random	200	sum	0.0417 ± 0.0004

In the benchmark regression and classification task results, there isn't a clear atom vector or pooling method that dominates. The 200-dimensional representations generally appear to perform better than the smaller 86-dimensional representations. Though not evident from Tables 3 and 4, sum- and mean-pooling outperform max-pooling (see Supplementary Note 1 and Supplementary Tables 1 to 10 in Appendix A). The pooled Mat2Vec representations are notable, in that they achieve the best results in 4 of the 8 benchmark tasks, while pooled SkipAtom representations are best in 2 of the 8 benchmark tasks. Pooled Random vectors tend to under-perform, though not always by a very large margin. This may not be so surprising, since random vectors exhibit quasi-orthogonality as their dimensionality increases, and thus may have the same functional characteristics as one-hot vectors [167]. On the OQMD Formation Energy prediction task, the Bag-of-Atoms representation yields the best results, significantly outperforming both the distributed representations, and the mean-pooled one-hot representation originally used in the ElemNet paper, that introduced the task.

A noteworthy aspect of these results is how the pooled atom vector representations compare to the published state-of-the-art values on the 8 benchmark tasks from the Matbench test suite. Figure 3.5 depicts this comparison. Indeed, the models described in this report

outperform the existing benchmarks on tasks where only composition is available (namely, the Experimental Band Gap, Bulk Metallic Glass Formation, and Experimental Metallicity tasks). Also, on the Theoretical Metallicity task and the Refractive Index task, the pooled SkipAtom, Mat2Vec and one-hot vector representations perform comparably, despite making use of composition information only.

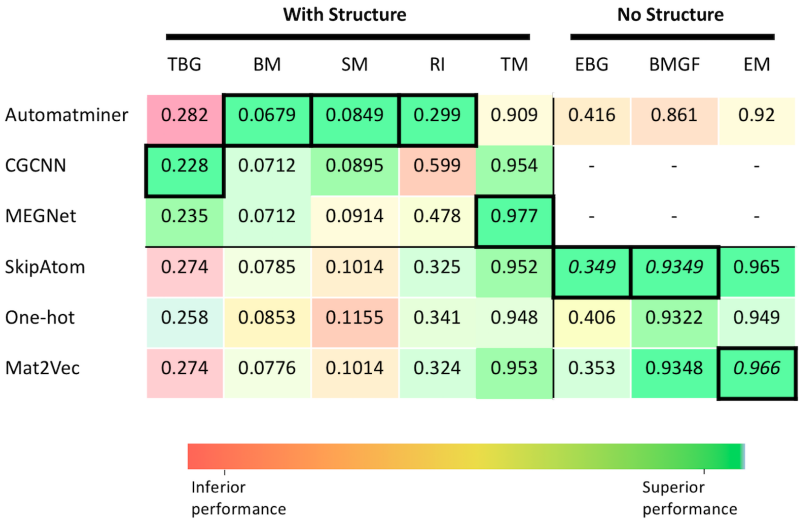

Figure 3.5: A comparison between the results of the methods described in the current work and existing state-of-the-art results on benchmark tasks. TBG refers to the Theoretical Band Gap task (MAE in eV), BM to the Bulk Modulus task (MAE in log(GPa)), SM to the Shear Modulus task (MAE in log(GPa)), RI to the Refractive Index task (MAE in n), and TM to the Theoretical Metallicity task (ROC-AUC). These tasks make use of structure information. EBG refers to the Experimental Band Gap task (MAE in eV), BMGF to the Bulk Metallic Glass Formation task (ROC-AUC), EM to the Experimental Metallicity task (ROC-AUC). These tasks make use of composition only. The results that are outlined in bold represent the best score for that task. Italicized results represent an improvement over existing best scores. As described in the Methods section of this chapter, the same approach was used to obtain the results for all of the algorithms in the table.

3.4 Conclusions

NLP researchers have learned many lessons regarding the computational representations of words and sentences. It could be fruitful for computational materials scientists to borrow techniques from the study of Computational Linguistics. Above, we have described how making an analogy between words and sentences, and atoms and compounds, allowed us to borrow both a means of learning atom representations, and a means of forming compound representations by pooling operations on atom vectors. Consequently, we draw the following conclusions: i) effective computational descriptors of atoms can be derived from freely available and growing materials databases; ii) effective computational descriptors of compounds can be easily constructed by straightforward pooling operations of the atom vectors of the constituent atoms; iii) representations of material composition (without structure) can be useful for predicting certain properties, and can play a useful role in hierarchical screening studies where subsequent more expensive steps account for structure.

SkipAtom performs as well as state-of-the-art embeddings, while offering significant ad-

vantages in terms of flexibility and ease of implementation. The SkipAtom representation can be derived from a dataset of readily accessible compound structures. Moreover, the training process is lightweight enough that it can be performed on a good quality laptop on a scale of minutes to several hours (given the atom pairs). This highlights some important differences between SkipAtom and Atom2Vec and Mat2Vec. Training of the Mat2Vec representation requires the curation of millions of journal abstracts, and a subsequent classification step for retaining only the most relevant abstracts. Additionally, pre-processing of the tokens in the text must be carried out to identify valid chemical formulae through the use of custom rules and regular expressions. On the other hand, since SkipAtom makes direct use of the information in materials databases, no special pre-processing of the chemical information is required. Although the procedures for creating Mat2Vec and SkipAtom vectors have been incorporated into publicly available software libraries, the conceptually simpler SkipAtom approach leaves little room for ambiguity that might result from manually written chemical information extraction rules. When compared to Atom2Vec, a principal difference is that SkipAtom vectors are not limited in size by the number of atom types available. This allows larger SkipAtom vectors to be trained, and, as is evident from the results described above, larger vectors generally perform better on tasks. (See Supplementary Note 5 in Appendix A for an analysis of embedding size.) Overall, we believe SkipAtom is a more accessible tool for computational materials scientists, allowing them to readily train expressive atom vectors on chemical databases of their choosing, and to take advantage of the growing information in these databases over time. (See Supplementary Note 6 in Appendix A for an analysis of training dataset size.)

The ElemNet architecture demonstrated that the incorporation of composition information alone could result in good performance when predicting chemical properties. In this work, we have extended the result, and shown how such an approach performs in a variety of different tasks. Perhaps surprisingly, the combination of a deep feed-forward neural network with compound representations consisting of composition information alone results in competitive performance when comparing to approaches that make use of structural information. We believe this is a valuable insight, since high-throughput screening endeavours, in the search for new materials with desired properties, often target areas of chemical space where only composition is known. We envision performing large sweeps of chemical space, in relatively shorter periods of time, since structural characteristics of the compounds would not need to be computed, and only composition would be used. The results presented here could motivate more extensive and computationally cheaper screening.

Going forward, there are a number of different avenues that can be explored. First, the atom vectors generated using the SkipAtom approach can be explored in different contexts, such as in combination with structural information. For example, graph neural networks, such as the MEGNet architecture [112], can accept as input any atom representation one chooses. It would be interesting to see if starting with pre-trained SkipAtom vectors could improve the performance of these models, where structure information is also incorporated. (See Supplementary Note 2 in Appendix A for preliminary results with MEGNet.) Alternatively, chemical compound vectors formed by pooling SkipAtom vectors can be directly concatenated with vectors that contain structure information, thus complementing the pooled atom vectors with more information. A candidate for encoding structure information is the Coulomb Matrix (in vectorized form), a descriptor which encodes the electrostatic interactions between atomic nuclei [168]. Finally, one limitation of the SkipAtom approach is that it does not provide representations of atoms in different oxidation states. Since it is (often) possible to unambiguously infer the oxidation states of atoms in compounds, it is, in principle, possible to construct a SkipAtom training set of pairs of atoms in different oxidation states. The number of atom types would increase by several fold, but would still be within limits that allow for ef-

ficient training. Note that by using a motif-centric learning framework, the oxidation states of transition metal elements have been effectively learned based on local bonding environments, using a graph neural network framework [169]. It would be interesting to explore the results of forming compound representations using such vectors for atoms in various oxidation states. (See Supplementary Note 4 in Appendix A for a preliminary experiment demonstrating the learning of representations for Fe(II) and Fe(III).)

Chapter 4

Thermoelectric Transport Property Prediction with Deep Neural Networks

4.1 Introduction

As discussed in Chapter 1 of this thesis, finding good thermoelectric materials with the right combination of properties is a difficult task, because of the interdependence of the properties that appear in the figure of merit. Other factors, like abundance and toxicity, further complicate the search for good candidate materials. While thermoelectricity has been a known phenomenon since the early 1800s [170, 171], relatively few materials have been discovered that are effective enough for practical applications. Well-studied thermoelectric materials, such as Bi_2Te_3 and PbTe, are suitable for various applications, but are often too expensive or too toxic for widespread adoption [172]. If thermoelectric generators are to be deployed on a scale large enough to have a positive environmental impact, new materials are needed [173].

The search for novel thermoelectrics is an active field of research [174–176]. A range of promising thermoelectric materials have been discovered experimentally, either serendipitously, or as a result of chemical intuition. In the low-temperature range (near room temperature), where thermoelectric materials are typically used for cooling applications or low-grade heat recovery, top performances are achieved with Bi_2Te_3-based alloys (*e.g.* $zT = 1.2$ and power factor of 45 μWcm^{-1}K^{-2} for $(Bi_{1-x}Sb_x)_2Te_3$ at room temperature [177]). Materials based on PbTe exhibit some of the best performances in the temperature range between 500 K and 900 K (*e.g.* zT of 2.5 at around 800 K in p-doped $Pb_{1-x}Sr_xTe$, with a maximal power factor above 30 μWcm^{-1}K^{-2}) [178]. At very high temperatures, such as those used in radioisotope thermoelectric generators ($\sim$1000 K or above), Si-Ge alloys exhibit some of the highest figures of merit (*e.g.* peak zT of about 1.3 at 1173 K in an n-type nanostructured SiGe bulk alloy, corresponding a maximal power factor of $\sim$30 μWcm^{-1}K^{-2}) [179]. Other families of compounds that are attracting considerable attention as promising thermoelectric materials include the metal chalcogenides (*e.g.* SnSe, Cu_2Se) [180–182], skutterudites (*e.g.* $CoAs_3$, $CoSb_3$) [183], Zintl compounds (*e.g.* $YbZn_2Sb_2$) [184], clathrates (*e.g.* $Sr_8Ga_{16}Ge_{30}$) [185], Heusler and Half-Heusler compounds (*e.g.* TiNiSn, ZrNiSn) [186–188], and metal oxides (*e.g.* $NaCo_2O_4$, $Ca_3Co_4O_9$) [189, 190]. Hole-doped polycrystalline SnSe is the record-holder in terms of thermoelectric figure of merit, and is reported to exhibit a zT of 3.1 at 783K [191]. In principle, there are no theoretical or thermodynamic limits for the possible values of zT [192], so there is hope that materials with even higher values of zT can be found.

In addition to trial-and-error exploration, and the rational design of materials, computational techniques based on the combination of density functional theory (DFT) and high-throughput screening (HTS) are becoming increasingly prevalent in the search for new thermoelectrics [193–195]. The first report of such an approach was made in 2006 by Madsen, who screened a dataset of 1,630 Sb-containing compounds derived from existing crystal structure databases, and based on the results of *ab initio* calculations, identified LiZnSb as an interesting thermoelectric material [196]. Since then, a number of studies involving the use of HTS in the search for new thermoelectric candidates have followed [11, 21, 197–203]. The increasing availability of distributed computing infrastructure, along with the development of workflow management software [159, 204–210], has enabled the growing adoption of this approach.

While DFT-based HTS is becoming more prevalent, there remains a large gap between the size of chemical space that is accessible with this approach, and the size of the space of all possible inorganic materials. As discussed in Chapter 1, in order to bridge that gap and to further accelerate computational predictions of thermoelectric behavior, techniques involving the use of ML are being increasingly used in the search for new thermoelectric materials [211–215]. Data for these ML approaches can come from either theoretical calculations, or from physical experiments. HTS studies have been producing *ab initio* results for thousands of materials, and these results can be assembled into datasets that are usable with ML algorithms. Since experimental data is scarcer, the outputs of *ab initio* calculations are often the source of data for ML approaches. Using ML to learn models that predict the output of *ab initio* calculations is sensible, since invoking an ML model is much faster (and less computationally expensive) than carrying out an *ab initio* calculation. ML models of various thermoelectric properties, such as the Seebeck coefficient [41, 43, 84, 216], electrical conductivity [42, 83], power factor [27, 82, 217, 218], lattice thermal conductivity [15, 23, 31, 33, 37, 219–226], and even zT [85, 227–231], have been developed.

In this chapter, I report the use of attention-based deep learning, together with existing datasets derived from high-throughput DFT calculations [232], to predict the thermoelectric transport properties of a material. The input to the model is a representation of a material's composition, and optionally the material's band gap. The output is a collection of predictions for a range of temperatures, for various doping levels, and for *n* and *p* doping types. This structure-free approach allows us to scan regions of materials space of hypothetical but plausible compounds, whose structures are not known. Our multi-output approach creates a thermoelectric behavior profile for a material at a number of different conditions, which offers advantages over narrower models that only make predictions for specific conditions.

4.2 Methods

4.2.1 Datasets

Our models are trained on the dataset published in 2017 by Ricci *et al.* [19] (henceforth the *Ricci database*). This is a freely available electronic transport database containing the computationally derived electronic transport properties for 47,737 inorganic compounds with stoichiometric compositions. The properties listed include the Seebeck coefficient, the electrical conductivity, and the electronic thermal conductivity, obtained using DFT in the generalized gradient approximation (GGA), and the BTE through the BoltzTraP computer software [7], under the constant relaxation time approximation (CRTA). They also associate the computed band gap with each entry, amongst several other properties. For each compound, the aforementioned properties were determined at various temperatures (100 K to 1300 K in 100 K increments), for p- and n-doping types, and at 5 doping levels (ranging from 10^{16} to 10^{20}

cm^{-3}). Moreover, each property is a tensor quantity reported as a 3×3 matrix. The database is altogether quite large, with 18,617,430 data points if one considers only the values of the diagonal elements S_{xx}, S_{yy}, and S_{zz} (*i.e.* 47,737 compounds $\times$ 13 temperatures $\times$ 2 doping types $\times$ 5 doping levels $\times$ 3 diagonal elements). Another important consideration is that there are duplicate compounds in the database in terms of composition (corresponding to possible polymorphs). While there are 47,737 unique compounds in the database when structure is considered, there are only 34,628 unique compositions. In this study, we form a dataset of compositions from the Ricci database and their associated thermoelectric transport properties. For cases where there are multiple entries with the same composition, we obtain the DFT-derived energy per atom of each polymorph, and use the transport properties and band gap of the entry corresponding to the polymorph with the lowest energy per atom.

Additionally, we form a dataset consisting solely of compositions and their associated electronic band gaps derived from DFT, by combining data from the Materials Project [155] and the Ricci database. We obtained 126,335 structures and their associated electronic band gaps from the Materials Project, which corresponded to 89,444 unique compositions, which are used to train the band gap predictor. Where there were multiple structures for a composition, again we used the band gap of the polymorph with the lowest computed energy per atom.

The Ricci database has some important limitations. As discussed in Ref. [19] and elsewhere (see Ref. [233] for a recent perspective), the use of the GGA and CRTA in the prediction of electronic transport can lead to large discrepancies with respect to experiment. In particular, GGA band structures generally exhibit too narrow gaps and too large bandwidths, which tends to exaggerate the electronic conductivity. The CRTA, especially when unaccompanied by physically-sound prediction of relaxation times, misses important differences in scattering mechanisms across compounds. Furthermore, the calculations in Ref. [19] did not consider spin-orbit coupling (SOC), which often has an important effect on the electron transport properties of materials [234]. Inevitably, any ML model based on this dataset will carry over these limitations of the underlying data, hindering the quality of the predictions with respect to experimental values. However, our approach establishes a protocol capable of efficiently mapping composition to thermoelectric behavior, which can be easily refined once more accurate databases become available. This is important because, in addition to the improvement of existing *ab initio* databases, there are ongoing efforts to create large databases of thermoelectric properties from experiment [235], so we anticipate our model will keep evolving following the expansion of such datasets.

4.2.2 ML Models

We build ML models that predict the Seebeck coefficient, the electrical conductivity, and the power factor using data from the Ricci database. Our multi-output regression models [236, 237] produce predictions of transport properties at 13 temperatures, 5 doping levels, for 2 doping types, given a material's composition and (optionally) band gap. The task is to predict the mean of the diagonal elements of the Seebeck tensor, $(S_{\mathrm{xx}} + S_{\mathrm{yy}} + S_{\mathrm{zz}})/3$, henceforth referred to as the Seebeck coefficient, S, and the mean of the diagonal elements of the electrical conductivity tensor, $(\sigma_{\mathrm{xx}} + \sigma_{\mathrm{yy}} + \sigma_{\mathrm{zz}})/3$, henceforth referred to as the electrical conductivity, σ. The values for electrical conductivity in the Ricci database are reported per unit of relaxation time. Hence, in this report, electrical conductivity, σ, will more precisely refer to electrical conductivity per unit relaxation time, σ/τ. The target power factor, $S^2\sigma$, is also predicted, and is defined here as the mean of the directional power factors, $(S_{\mathrm{xx}}^2\sigma_{\mathrm{xx}} + S_{\mathrm{yy}}^2\sigma_{\mathrm{yy}} + S_{\mathrm{zz}}^2\sigma_{\mathrm{zz}})/3$. It will be denoted by PF, and also given per unit of relaxation time.

More formally, the task is to learn a function $f : \mathcal{X} \to \mathcal{Y}$, given a training set $\mathcal{D} =$

$\{(\mathbf{x}_i, \mathbf{y}_i) \mid 1 \leq i \leq k\}$, with $\mathbf{x}_i \in \mathcal{X}$, $\mathbf{y}_i \in \mathcal{Y}$, and k labeled examples. Here, the $\mathbf{x}_i$ represent a multi-dimensional input describing the features of an exemplar, and $\mathbf{y}_i$ represent a multi-dimensional target associated with $\mathbf{x}_i$. A training procedure is used to find f, and involves the minimization of a loss, $L : \mathcal{Y} \times \hat{\mathcal{Y}} \to \mathbb{R}$, that specifies the degree of disagreement between the true values $\mathcal{Y}$, and $\hat{\mathcal{Y}}$, the output of f given members of $\mathcal{X}$.

Here, we use two different forms of f: a Random Forest [89], and an attention-based deep neural network based on the CrabNet architecture, which is the state-of-the-art tool for property prediction from materials composition, as demonstrated in the work by Sparks *et al.* [113]. The CrabNet architecture incorporates a multi-head self-attention mechanism, originally introduced in the Transformer deep learning model [238], which provides the added advantage of enhanced interpretability. Traditionally, a Transformer transforms an input sequence to an output sequence using an encoder followed by a decoder. However, CrabNet consists strictly of an encoder, followed by a number of Residual blocks [239]. Moreover, instead of a sequence of words, CrabNet operates on a bag of atoms, and consequently, instead of using a positional encoding of the input, it encodes the relative amounts of atoms present.

The input to the model thus consists of a material's composition. Formally, the input, $X_{\text{in}} \in \mathbb{R}^{n \times d_{\text{in}}}$, consists of d_{in}-dimensional representations for the n constituent elements of the composition. The first step involves the encoding of the relative amounts of atoms into X_{in}, referred to as *fractional encoding* (see [113] for more details), resulting in $X_{\text{enc}} \in \mathbb{R}^{n \times d_{\text{model}}}$, where d_{model} is given as a hyperparameter. This is followed by the sequential application of a number of Transformer blocks. Each Transformer block begins by performing a multi-head self-attention operation. (Figure 2.2) The self-attention operation allows the model to learn to attend to the relationships between the atoms of the composition, in the context of the task. The "attention weights" are encoded into a $n \times n$ matrix, associated with each of h attention heads, by applying the $softmax$ operation to a scaled dot-product of a query, $Q_i \in \mathbb{R}^{n \times d_K}$, and a transposed key, $K_i^T \in \mathbb{R}^{d_K \times n}$, where $d_K = d_{\text{model}}/h$ specifies the key (and query) dimension for an attention head. The output of a Transformer block, $X_{\text{out}} \in \mathbb{R}^{n \times d_{\text{model}}}$, thus consists of the same dimensions as the input.

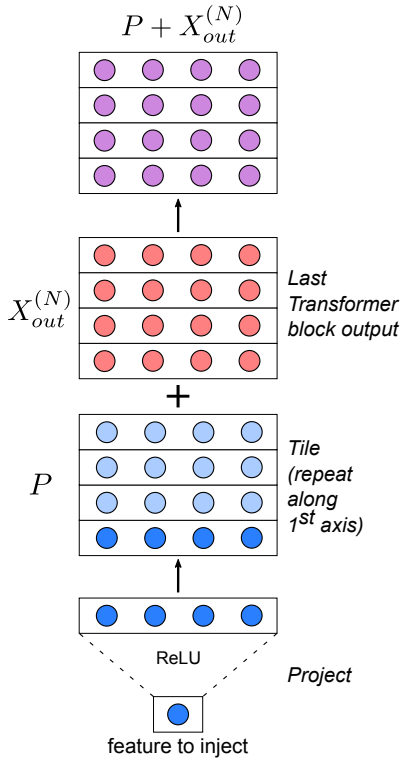

Figure 4.1: A depiction of how additional features, such as band gap, are incorporated into the CrabNet architecture. The addition operation refers to element-wise addition. The projection of the feature involves trainable parameters.

Since it may also be desirable to provide additional information beyond composition to the model, we augment the CrabNet architecture so that additional features may be provided. There are a number of ways this could be accomplished, but we choose to borrow an approach from computer vision [240], and perform a projection on v input features, $\mathbf{u} \in \mathbb{R}^v$, followed by a tiling operation, so that the resulting projected features, $P \in \mathbb{R}^{n \times d_{\text{model}}}$, have the same dimensions as the output of a Transformer block. Finally, we perform element-wise addition, $P + X_{\text{out}}^{(N)}$, where N is the number of Transformer blocks, and $X_{\text{out}}^{(N)}$ denotes the output of the last Transformer block (Figure 4.1). While any number of extra features may be supplied to the model this way, in this work, we (optionally) supply a single feature, the band gap E_{g}, associated with the material.

Finally, the output $P + X_{\text{out}}^{(N)}$ is given to three separate output heads. Each output head consists of a series of Residual blocks, followed by a fully connected linear layer that produces the final predictions for each of S, σ, and PF. This multi-head architecture has advantages in terms of convenience, efficiency, and also usually provides better overall performance on the task when compared to using a separate (single-head) model for each property predicted. (See Supplementary Table 1 in Appendix B for a comparison of the performance of architectures with different output head numbers.) For clarity, and to differentiate it from the original CrabNet architecture, we refer to this model as CraTENet (Compositionally-restricted attention-based ThermoElectrically-oriented Network); its architecture is illustrated in Figure 4.2.

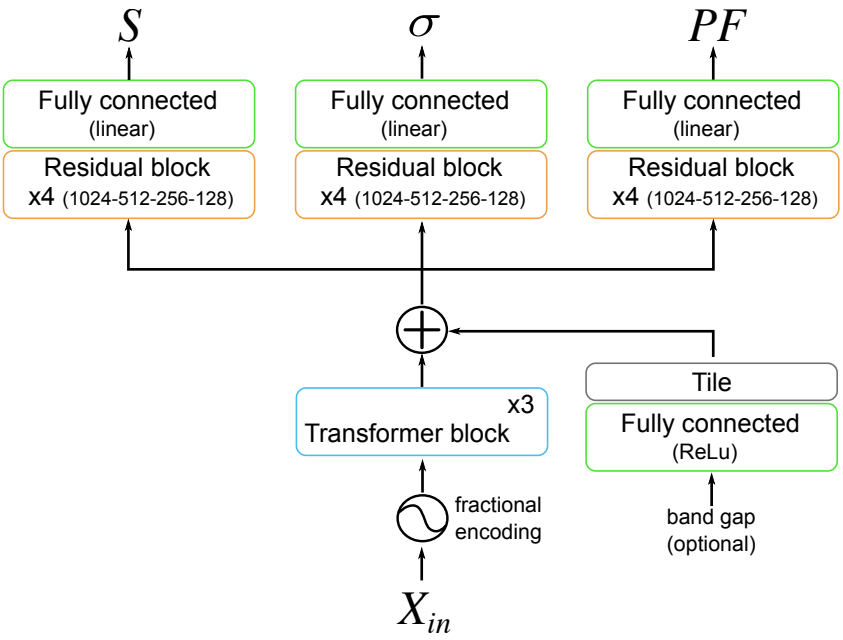

Figure 4.2: The multi-head attention-based architecture, CraTENet, used in this study. Each of the three output heads are multi-valued, containing the prediction of the Seebeck (S), electrical conductivity (σ), and power factor (PF), at different temperatures, doping levels, and doping types.

The CraTENet model thus expects a dataset consisting of compositions, $X_i \in \mathbb{R}^{n \times d_{\text{in}}}$, and associated thermoelectric transport properties, $\mathbf{y}_i^S, \mathbf{y}_i^\sigma, \mathbf{y}_i^{PF} \in \mathbb{R}^m$, where $\mathbf{y}_i^S$, $\mathbf{y}_i^\sigma$, and $\mathbf{y}_i^{PF}$, represent the S, σ, and PF transport values, respectively, at all temperatures, doping levels and doping types, each an m-dimensional vector. Optionally, a band gap, $E_{\text{g}_i} \in \mathbb{R}$, may be associated with X_i. The dataset is thus $\{((X_i, E_{\text{g}_i}), (\mathbf{y}_i^S, \mathbf{y}_i^\sigma, \mathbf{y}_i^{PF})) \mid 1 \leq i \leq k\}$, where k is the number of examples.

As in the CrabNet and Roost models, the CraTENet model learns the heteroscedastic aleatoric uncertainty (*i.e.* how the variance of the predicted variable depends on the independent variables), explicitly through the loss function [241, 242]. Here, the calculated variance is a measure of the uncertainty associated with the incompleteness of the descriptor used (which is why the calculated variance decreases considerably when the band gap information is added to the descriptor). This variance is different from the epistemic variance related to the quality of the model parameterization. Whereas the CrabNet and Roost models use a Robust L1 loss to estimate the uncertainty, we find that a Robust L2 loss, which places an L2 distance on the residuals, results in superior performance for this task (see Supplementary Note 2 and Supplementary Table 4 in Appendix B). The loss, L_p, for a particular thermoelectric transport property p, is given by:

$$L_p = \frac{1}{2k} \sum_{i=1}^{k} \sum_{j=1}^{m} (\hat{y}_{ij}^p - y_{ij}^p)^2 \exp(-\ln \hat{s}_{ij}^p) + \ln \hat{s}_{ij}^p \tag{4.1}$$

where k is the number of examples in the dataset, and m is the number of components of the output vector $\mathbf{y}_i^p$. The prediction of the i^{th} example is $\hat{\mathbf{y}}_i^p$, and $\hat{y}_{ij}^p$ the j^{th} component of the i^{th} prediction (also considered the predictive mean in this context). The corresponding target value is y_{ij}^p. Finally, the predictive aleatoric variance for the j^{th} component of the i^{th}

prediction is given by $\hat{s}_{ij}^{p}$. The form of this loss arises from the assumption that the uncertainty in the observations follows a Gaussian distribution. Also, the term $\exp(-\ln \hat{s}_{ij}^{p})$ is used in place of the term $1/\hat{s}_{ij}^{p}$ for numerical stability reasons, such as to avoid a potential division by zero. Since the model utilizes a separate output head for each of the three thermoelectric transport properties being learned, the overall loss, L, to be minimized is given by:

$$L = \alpha L_S + \beta L_\sigma + \gamma L_{PF} \tag{4.2}$$

where α, β, and γ are constants which weight the importance of each of the terms in the loss L. In this work, $\alpha = \beta = \gamma = 1$.

Finally, we also train a band gap predictor from composition, using the original CrabNet model and the expanded band gap dataset described previously. The fact that the band gap predictor can be trained with a much larger dataset than the one used for training the CraTENet model justifies our attempt to use the band gap as an additional input to CraTENet for the prediction of transport coefficients. As shown in the Results and Discussion section, if the band gap predictor is sufficiently accurate, the inclusion of the predicted band gap in the CraTENet input can lead to overall performance enhancement, even if composition remains the only global input of the model.

4.2.3 ML Model Training and Evaluation

For all CraTENet and CrabNet models, the input, X_{in}, consisted of $n = 8$ elements, and was zero-padded if the composition consisted of less than 8 elements. Each element in the input was described with a SkipAtom distributed representation [243] with dimensions $d_{\text{in}} = 200$. (We performed experiments, as described in Supplementary Note 1 and Supplementary Table 3 in Appendix B, to determine the performance of different descriptors). The default architectural hyperparameters of the original CrabNet model were used without further tuning. Specifically, both models consisted of $h = 4$ attention heads in each of 3 sequential Transformer blocks; the hyperparameter d_{model} was set to 512. The output (or output head) consisted of 4 sequential Residual blocks, with 1024, 512, 256, and 128 nodes respectively. For all neural network training procedures, a mini-batch size of 128 and a learning rate of 10^{-4} was used, which were derived from a hyperparameter grid search. The Adam optimizer, with an epsilon parameter of 10^{-8}, was used [163]. All neural network models were implemented using the TensorFlow [244] and Keras [245] software libraries.

The input for the Random Forest models was a descriptor described by Meredig *et al.* [148], as implemented in the Matminer software library [246]. It is a local descriptor of composition, containing properties such as atomic fractions, electronegativities, and radii. In some experiments, we concatenate an unscaled band gap feature to the descriptor. The Random Forest model hyperparameters were determined using a grid search. The number of estimators was set to 200, the maximum depth was set to 110, the maximum number of features was set to 36, and bootstrapping was used. We used the implementation provided in the Scikit-learn software library [247].

Because the electrical conductivity values in the Ricci database are given per unit of relaxation time τ, which is an exceedingly small number (of the order of 10^{-15} s), the target values for σ and PF are numerically quite large. The values also vary by orders of magnitude, reflecting the distribution across metallic, semiconducting and insulating conductivity ranges. For these reasons, the models learn $\log_{10}\sigma$ and $\log_{10} PF$ instead. All output targets are standardized by removing the mean and scaling to unit variance. The band gap, when it is provided to the CraTENet model, is given in eV units and unscaled.

Neural network model training was carried out in one of two contexts: a 90-10 holdout experiment, or a 10-fold cross-validation experiment. For 90-10 holdout experiments, we split

the dataset $\mathcal{D}$ into a set $\mathcal{A}$ consisting of 90% of the data, and a set $\mathcal{B}$ consisting of 10% of the data. For the neural network models, set $\mathcal{A}$ was further split into a training set $\mathcal{T}$ consisting of 90% of $\mathcal{A}$, and a validation set $\mathcal{V}$ consisting of 10% of $\mathcal{A}$. Early stopping was used (with a patience of 50) to determine the optimal number of epochs to train, using $\mathcal{V}$ as the validation set. Then, the model was re-trained on all of $\mathcal{A}$ for the number of epochs determined to be optimal, again starting from random parameters. Test set $\mathcal{B}$ was then used to evaluate performance of the re-trained model (see [248] for more information on this approach). The Random Forest models were trained on $\mathcal{A}$, and evaluated on $\mathcal{B}$. The same random seed was used throughout when creating the splits, to ensure identical splits for all experiments.

For the 10-fold cross-validation experiments, we followed the same procedure as for the 90-10 holdout experiments, except that we create 10 mutually exclusive splits, each consisting of 10% of $\mathcal{D}$ for testing and 90% of $\mathcal{D}$ for training, using the same random seed for all experiments, and repeating the hold-out procedure for each of the 10 splits. The performance on $\mathcal{B}$ was averaged across the 10 splits to yield the final performance of the model.

The objective of all neural network training experiments was to minimize either the Robust L1 or Robust L2 loss. The objective of Random Forest training was to minimize the mean squared error (MSE) criterion. The mean absolute error (MAE) and coefficient of determination (R^2) metrics were used to assess model performance. To produce the final neural network models to be used for inference on composition space outside the datasets used for training and evaluation, we train the models on all available data $\mathcal{D}$ for a number of epochs determined from the corresponding 10-fold cross-validation experiment, by averaging the number of epochs required for each fold. The final Random Forest models to be used for inference were simply trained on all available data $\mathcal{D}$.

4.2.4 DFT Calculations

We performed a small number of DFT calculations in systems not found in the Ricci database, for testing purposes. All calculations were carried out using the Vienna Ab initio Simulation Package (VASP) [249, 250], and the calculation settings were chosen to follow the work of Ricci *et al.* [19] as closely as possible. The Perdew-Burke-Ernzerhof (PBE) [47] exchange-correlation functional, which is based on the GGA, was used in conjunction with the projector augmented-wave method [251, 252] to describe the interaction between core and valence electrons. All structures were fully relaxed until the force on each atom is below 0.02 eV/Å. Spin polarization was on, and magnetic moments on the ions were initialized in a high-spin ferromagnetic configuration, and then allowed to relax to the spin groundstate. A self-consistent static calculation was performed using 90 k-points/Å^{-3} (in terms of reciprocal lattice volume) for systems with band gaps ≥ 0.5 eV, and 450 k-points/Å^{-3} for systems with band gaps < 0.5 eV. Subsequently, a non-self-consistent calculation was performed to evaluate the band structures on a uniform k-point grid, with 1,000 k-points/Å^{-3} for systems with band gaps ≥ 0.5 eV, and 1,500 k-points/Å^{-3} for systems with band gaps < 0.5 eV. Spin-orbit coupling was not considered.

The Seebeck coefficient, S, and the electrical conductivity, σ, were computed using the BoltzTraP2 software package [253]. Interpolation was first performed by sampling 5 irreducible k-points for each k-point from the VASP output. The band structure was then integrated to obtain sets of Onsager coefficients. The temperature range 100K to 1300K was explored, in increments of 100K, at 5 different doping levels (10^{16} to 10^{20} cm^{-3}), for both n and p doping types. We verified that our *ab initio* procedure emulates the one that was used to create the Ricci database by comparing our results to those of the Ricci database for a number of compounds (see Supplementary Figure 3 in Appendix B).

4.2.5 Data Availability

The data that support the findings described in this chapter are openly available. The Ricci database of thermoelectric transport coefficients is publicly available online at: https://datadryad.org/stash/dataset/doi:10.5061/dryad.gn001. The Materials Project data that was used to train the band gap predictor and form a composition search space are publicly available online at: https://materialsproject.org/. The pre-trained SkipAtom embeddings that were used as input to the neural network models are located at: https://github.com/lantunes/skipatom. The OQMD data that was used to provide structures for the SMACT-generated selenides are publicly available online at: https://oqmd.org/.

4.2.6 Code Availability

The code with the CraTENet implementation, and for pre-processing the data and reproducing the experiments, is open source, released under the MIT License. The code repository is accessible online, at: https://github.com/lantunes/CraTENet.

4.3 Results and Discussion

4.3.1 Thermoelectric Property Prediction

Both the CraTENet model and a Random Forest model were trained on the 34,628 entries of the Ricci database. To establish the generalization error of the models, 10-fold cross-validation was performed. Since multi-target regression of thermoelectric transport properties on composition is essentially a new task, unreported in the literature, there are no existing benchmarks to compare with. We created simple baseline models, such as linear regression with a Meredig feature vector, or simply taking the median of the target values, but these models performed considerably worse than the ML models presented here. To simplify presentation, we leave out the baseline results.

The results of 10-fold cross-validation are presented in Table 4.1. For the remainder of this chapter, "CraTENet" will refer to either the version of the model which does not accept a band gap input or to the CraTENet model in general, depending on the context, whereas "CraTENet+gap" will specifically refer to the version of the model which requires a band gap input. As is evident from the results in Table 4.1, the models which utilize the band gap clearly outperform those which do not. The band gap is thus an important predictor of thermoelectric transport properties. In both the case where band gap is or is not provided, the CraTENet model outperforms the Random Forest model in terms of MAE. The Random Forest performs better in terms of R^2, but generally only when band gap is absent. Moreover, the models appear to perform slightly better when predicting the $\log \sigma$ than the Seebeck. Prediction of the $\log PF$ appears to be the most problematic, with the R^2 for this property being noticeably lower than for the other two properties. The best thermoelectric materials have Seebeck coefficients in the order of several hundreds of μV/K, so the resulting MAE is still reasonably small by comparison.

Table 4.1: Ten-fold cross-validation results for each of the transport properties for the CraTENet and Random Forest (RF) models, both with and without a provided band gap, in terms of MAE and R^2. Each value represents the mean result across 10 folds, across all temperatures, doping levels and doping types. Bold values represent the best result for a class of models (*i.e.* with or without band gap) for a particular property.

	S		$\log \sigma$		$\log PF$	
	MAE (μV/K)	R^2	MAE	R^2	MAE	R^2
CraTENet	**114**	0.780	**0.576**	0.768	**0.452**	0.616
RF	141	**0.798**	0.696	**0.780**	0.476	**0.632**
CraTENet+gap	**49**	**0.962**	**0.260**	**0.968**	**0.380**	0.731
RF+gap	54	0.961	0.301	0.964	0.398	**0.737**

The results in Table 4.1 represent predictions made for all temperatures, doping levels and doping types. However, it is useful to understand how the models perform for different cross-sections of the data. For example, the 10-fold cross-validation results as a function of doping type are presented in Table 4.2. To obtain the values in Table 4.2, only the predictions for a given doping type were considered when computing the metrics, across all doping levels and temperatures. The CraTENet model appears to perform better on the p-type predictions, though it depends on which metric one considers. In Figure 4.3, 10-fold cross-validation results are presented as a function of temperature and doping level. It is interesting (and useful to know) that the PF predictions are worse, in terms of R^2 values, at lower temperatures and higher doping levels. The MAE, on the other hand, does not show significant variations with the conditions of temperature and doping, remaining constant at around 0.40 for $\log PF$. The ability of the model to find the most promising compounds for further study depends on the magnitude of the error relative to the width of the distribution of values. If the absolute error is roughly constant, the ability of the model to discriminate between compounds can be expected to be worse for a dataset that is more narrowly distributed. In this sense, the R^2 is a better metric because it is related to the ratio between the mean squared error and the variance. Supplementary Figure 14 in Appendix B shows that at high doping levels the distribution of values is narrower, and therefore the R^2 (as well as our ability to select the best compounds) decreases. The effect of temperature is a bit less pronounced, but because increasing temperature also tends to widen the distribution, the R^2 is slightly better at high temperatures. The variations in the distribution of power factors at different conditions are related to the balance between the Seebeck coefficient and the conductivity in metallic *vs.* gapped materials in the calculations of Ref. [19]; further details can be found in the Supplementary Material in Appendix B.

Table 4.2: Ten-fold cross-validation performance of the CraTENet model as a function of doping type. Each value represents the mean result for each doping type across all 10 folds, across all temperatures and doping levels. Bold values represent the best result between p- and n-doping types for a class of models (*i.e.* with or without band gap) for a particular property.

		S		$\log \sigma$		$\log PF$	
	Doping	MAE (μV/K)	R^2	MAE	R^2	MAE	R^2
CraTENet	p-type	119	**0.636**	0.589	**0.775**	0.465	**0.631**
CraTENet	n-type	**109**	0.627	**0.562**	0.758	**0.439**	0.594
CraTENet+gap	p-type	**49**	**0.945**	0.260	**0.972**	0.388	**0.747**
CraTENet+gap	n-type	50	0.925	0.260	0.962	**0.371**	0.709

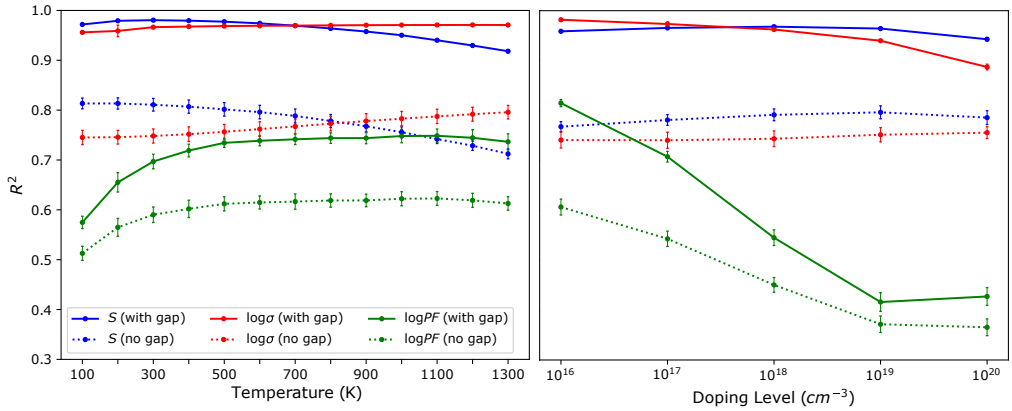

Figure 4.3: Ten-fold cross-validation performance (R^2) of the CraTENet model as a function of temperature and doping level. On the left, each point represents the mean performance for each temperature across all 10 folds, across all doping levels and doping types. On the right, each point represents the mean performance for each doping level across all 10 folds, across all temperatures and doping types. The dotted series represent the model's results without a provided band gap.

To understand how the predictions compare to the "true" values (*i.e.* the target DFT values), and how the prediction errors are distributed, it is useful to plot the true versus the predicted values, and also the distribution of absolute errors, as in Figure 4.4. The plots show that most predictions lie close to the true values. Moreover, the distribution of absolute errors indicates that the majority of errors are less than the overall MAE values.

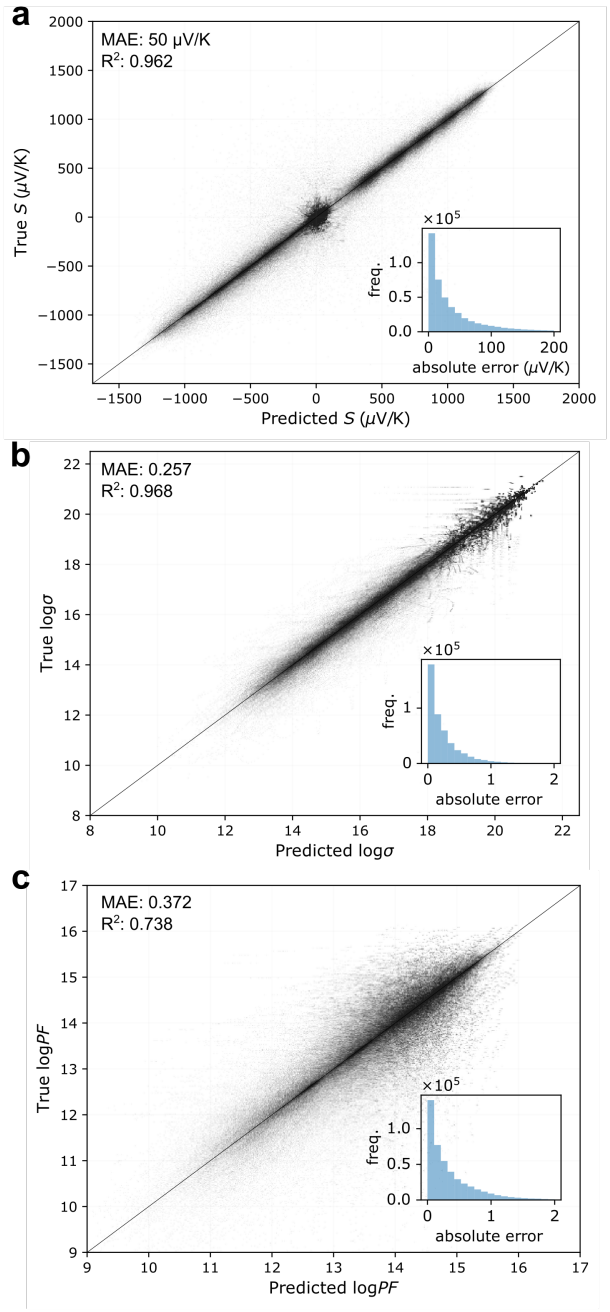

Figure 4.4: True values vs. predicted values of the test set of a 90-10 holdout experiment using the CraTENet+gap model, for each of the transport properties, across all temperatures, doping levels, and doping types. Each plot contains 450,190 points, as there are 3,463 compositions in the test set, each with 130 (13 temperatures $\times$ 5 doping levels $\times$ 2 doping types) associated values. The inset plots depict the distribution of absolute errors.

As the CraTENet model performs best when access to a band gap is available, it is important to understand how the performance of the model depends on the quality of the band gap provided, since, in many contexts, an experimental or *ab initio* band gap may not be available. In screening scenarios, the band gap could originate from a predictive model. Thus, to understand how the CraTENet model depends on the quality of the band gap, we performed sensitivity experiments, by incrementally degrading high quality band gaps (*i.e.* derived from

an initio methods) by adding Gaussian noise, and then supplying these "lower-quality" band gaps to the model. The results are presented in Figure 4.5. In the figure, the horizontal axis along the top of the plot represents the resulting MAE (in eV) after a certain percentage of noise has been added to the band gaps. For example, when 10% noise has been added to the *ab initio* band gaps, the MAE when comparing these corrupted gaps to the true gaps is 0.065 eV. Figure 4.5 shows, as might be expected, that when more noise is added to the band gaps, the performance of the model falls. However, some thermoelectric transport properties are more robust (or more sensitive) to changes in the band gap quality. For example, in the case of the prediction of the Seebeck, even with band gaps exhibiting an MAE of 0.30 eV, the model is still able to achieve an R^2 of 0.85, in comparison to an R^2 of below 0.80 when no band gap is provided. However, in the case of $\log \sigma$, the model is much more sensitive. Current state-of-the-art band gap predictors that operate on composition alone typically achieve an MAE of 0.30-0.45 eV [254]. However, band gap predictor performance is expected to improve over time, and this will further increase the utility of the CraTENet model in screening scenarios with predicted band gaps.

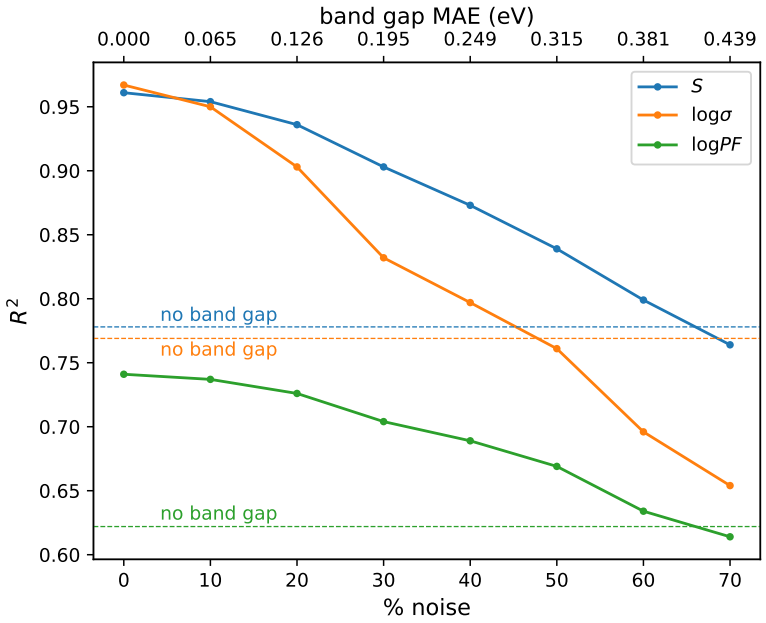

Figure 4.5: Performance of the CraTENet+gap model (in terms of R^2) as a function of band gap quality. A 90-10 holdout experiment was performed, and the actual gaps in the test set were corrupted by adding increasing amounts of Gaussian noise, before the performance of the model was assessed. The dotted lines represent the performance of the CraTENet model without a provided band gap.

4.3.2 Band Gap Prediction

A dataset consisting of compositions and their corresponding DFT-derived band gaps was formed by taking all of the unique compositions in the Materials Project, and consisted of 89,444 entries. A CrabNet model was trained on this dataset, using the minimization of the Robust L1 loss as the objective. To establish the generalization error of the model, 10-fold cross-validation was performed (as described in the Methods). Across the 10 folds, the model achieved a mean R^2 of 0.71, and a mean MAE of 0.38 eV. A final model was trained on all

89,444 entries for 101 epochs, which was determined to be the ideal number of epochs required (*i.e.* the mean number of epochs required across the 10 folds). This band gap predictor was subsequently used to provide band gaps when scanning composition space where structure and band gaps were unknown.

4.3.3 Searching Composition Space for New Thermoelectrics

Materials Project Compounds not in the Ricci Database

Of the 126,335 structures we obtained from the Materials Project, we derived 89,444 unique compositions. Since the compounds in the Ricci database originate from the Materials Project, we obtained 54,816 unique compositions when removing the compositions found in the Ricci database. This collection of 54,816 compositions forms a sizeable and convenient search space, since GGA band gaps have already been computed for these compounds, and their structures are known. Moreover, we verified that the distributions of compositions in this dataset and the Ricci database are similar (see Supplementary Figure 2 in Appendix B). Thus, we apply our CraTENet+gap model to this space, in an attempt to surface novel compounds which may represent promising thermoelectrics. We verify the quality of our predictions by performing *ab initio* calculations for a small subset of these compounds.

Making predictions for tens of thousands of compounds with the CraTENet model is computationally inexpensive in comparison with *ab initio* calculations, since inference is fast, aided by the use of GPUs and the inherent parallelism in neural networks. After performing inference on this space, we selected 23 materials from this collection that spanned a range of different thermoelectric properties and band gaps. For example, the predicted Seebeck values ranged from -1200 to 1200 $\mu V/K$. When comparing the values produced using the CraTENet+gap model and those obtained through *ab initio* methods, we found that the R^2 was between 0.87 and 0.88, and the MAE was between 72 and 79 $\mu V/K$ (Figure 4.6 and Supplementary Figure 15 in Appendix B). Although the agreement is generally good, there are some outliers (notably related to compositions SbTeIr and $LiNbN_2$). The performance of the model at specific compositions is difficult to rationalise, as it reflects both how well similar compositions are represented in the training set and the error related to the incompleteness of composition as a descriptor.

Moreover, we extracted the top 1000 compounds by predicted power factor, for each of p and n doping types (the lists are provided in the dataset accompanying this chapter). We selected 3 p-type selenides for performing *ab initio* calculations: GaCuTeSe, InCuTeSe, and CeSbSe. These compounds do not appear to have been studied as thermoelectrics before, but they seem promising as they include elements like Cu, In, Sb, and Te that are present in well-known thermoelectrics. After carrying out *ab initio* calculations, we found generally good agreement with the CraTENet predictions (Figure 4.7; see Supplementary Figures 4-9 in Appendix B for more comprehensive plots of the predictions).

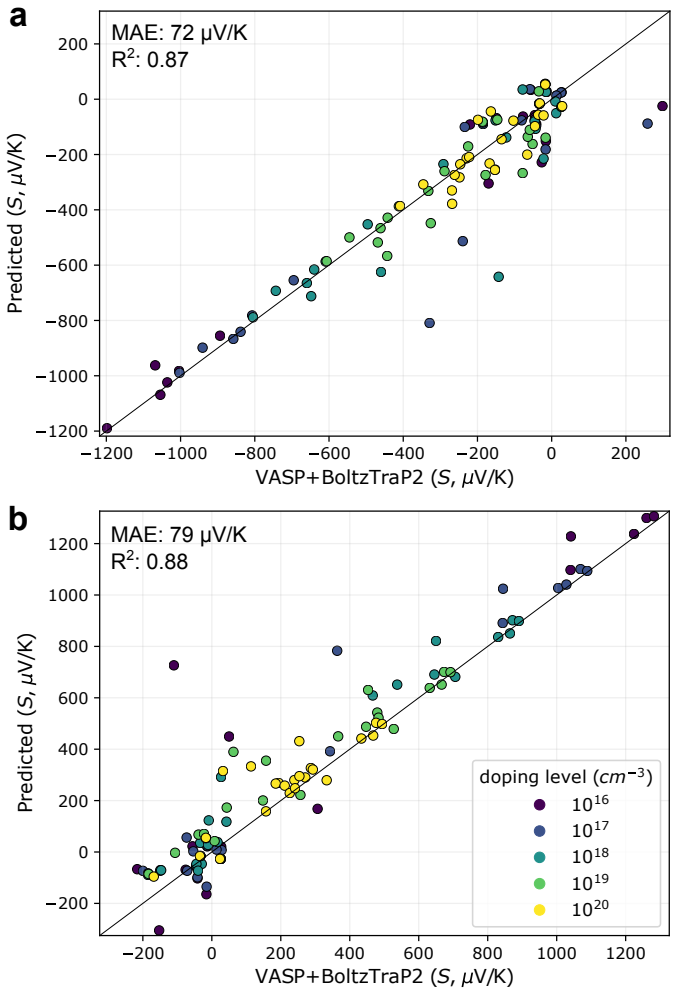

Figure 4.6: Seebeck coefficients at 700 K predicted with CraTENet+gap *vs.* those computed using the *ab initio* approach, for 23 Materials Projects compounds not found in the Ricci database, with **a)** n-type doping, and **b)** p-type doping. Each point represents a particular compound at a particular doping level (*e.g.* SbTeIr at $10^{20}\mathrm{cm}^{-3}$).

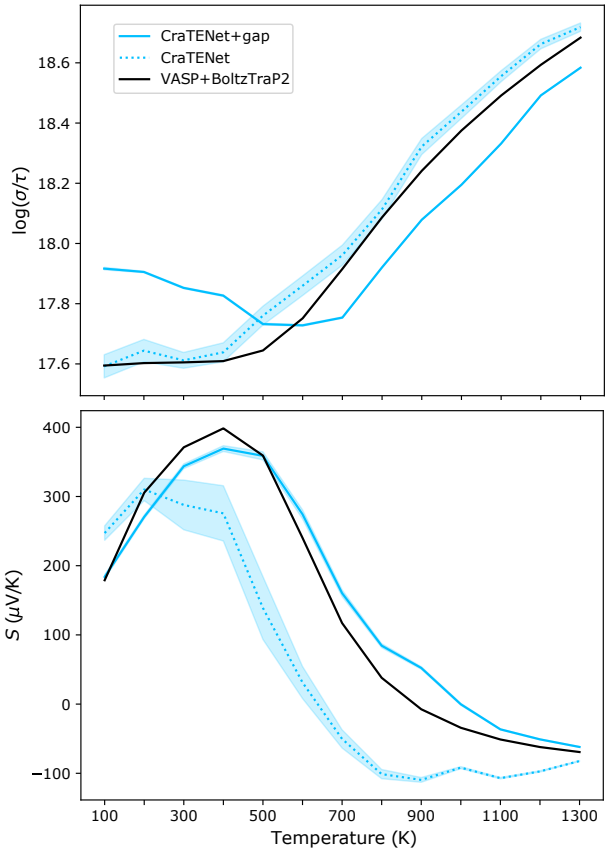

Figure 4.7: Predictions of the Seebeck and $\log \sigma$ for GaCuTeSe using the CraTENet models and the *ab initio* procedure, for p-type doping, at a level of 10^{19} cm^{-3}. The band gap value used, 0.387 eV, was obtained from the Materials Project. The shaded regions represent the $\pm$ standard deviation (*i.e.* the square root of the predicted variance).

Hypothetical Selenides

Since the CraTENet model requires only composition as input, it is conceivable that arbitrarily large hypothetical composition spaces could be generated and then processed by the model. SMACT is a software library that facilitates the generation of composition spaces, while adhering to chemical bonding rules, resulting in compositions which are chemically sensible [255]. Selenium-based materials are very promising thermoelectrics, because they exhibit similar properties as record-holding thermoelectric tellurides, but with the advantage that Se is much more Earth-abundant and cheaper than Te. We then chose to focus on creating a composition space of ternary selenides. Using SMACT, we generated 269,846 ternary selenide compositions, containing elements with an atomic number less than 84 (to avoid the heavy radioactive elements). The CraTENet and CraTENet+gap models were then used to make predictions of the thermoelectric transport properties of these compositions. As the CraTENet+gap model requires a band gap, we use our composition-only CrabNet band gap predictor as the source of the band gaps for this space. Since there is uncertainty in the band gap prediction, we make a separate prediction of thermoelectric transport properties using the predicted gap, the predicted gap plus the standard deviation, and the predicted gap minus the standard deviation. We find that this technique is useful for understanding the sensitivity of

the predictions to the band gap value for a particular composition.

Having made predictions on these SMACT-generated selenides, we then rank the compositions by power factor (as described in the previous section). We make the top 1000 compositions publicly accessible in the code and dataset repository accompanying this chapter. There are several interesting selenides in that list, involving elements like bismuth (*e.g.* $LiBiSe_2$) or thallium (*e.g.* $NaTlSe_2$) which are often present in known thermoelectric materials. To the best of our knowledge, these compounds have not been studied as thermoelectrics in the literature. To validate the model's predictions, we carried out *ab initio* calculations on these two compounds, given that their structures are reported in the OQMD database [159]. A comparison of the predictions and the *ab initio* values for each is provided in Figures 4.8a and 4.8b. (See also Supplementary Figures 10-13 in Appendix B for more comprehensive plots of the predictions).

In the absence of DFT-calculated band gaps as input, the performance of the CraTENet model for these compounds is not as impressive in predicting the DFT-calculated values of the transport coefficients. The model using the predicted band gaps as an input seems to perform generally better than the model with no gap, but the deviations are still considerable, especially at high temperatures. All models, for example, overestimate the electrical conductivity of $LiBiSe_2$ by at least half an order of magnitude. Still, the DFT calculations confirm, within their own limitations, that these compounds have attractive values of the electronic transport coefficients; they deserve further investigation, either using more accurate theoretical predictions with methods beyond the GGA and the CRTA, or experimentally. Clearly, the main use of the methods presented here cannot be the quantitative prediction of the transport properties of individual compounds, but rather the identification of interesting candidates in unexplored regions of the compositional space.

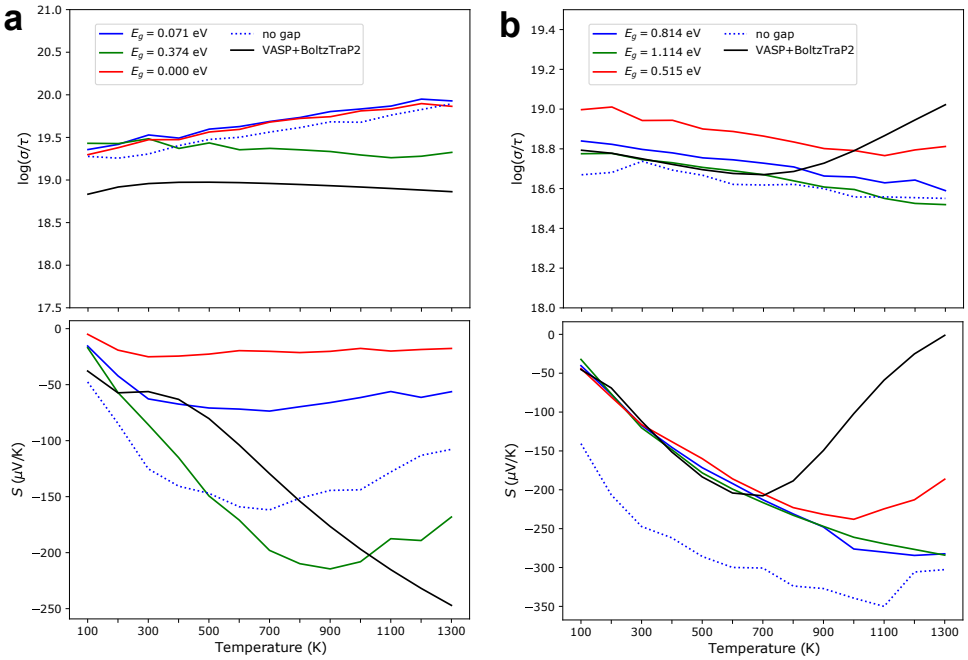

Figure 4.8: Predictions of the Seebeck and $\log \sigma$ for **a**) $LiBiSe_2$, and **b**) $NaTlSe_2$ using the CraTENet models and the *ab initio* procedure, for n-type doping, at a level of 10^{20} cm^{-3}. Predicted band gap values were used: blue represents the initial prediction, green represents the prediction plus the predicted standard deviation, and red represents the prediction minus the predicted standard deviation (*i.e.* square root of the predicted variance).

4.4 Conclusions

Approaches based on HTS combined with ML seem promising for suggesting novel candidate materials, since very large areas of chemical space can be examined quickly and efficiently. Here, we have shown that such an approach can be used to identify promising candidate thermoelectric materials based on the screening of potential compositions only, optionally supplemented with band gaps.

Several aspects of the approach described here contribute to its utility. First, the use of multi-output regression is helpful, and well-suited to the problem, since thermoelectric transport properties are dependent on factors such as temperature, doping level, and doping type. Conversely, an approach that requires parameters such as the temperature, doping level and doping type as input is problematic, since it increases the dimensionality of the input space, and also leads to inputs that resemble each other closely, as a result of the combinatorial nature of such a dataset [256].

Second, we believe that regression is a more useful choice for this learning task when compared to classification, in the context of searching for new materials. Several existing studies have involved the training of classification models of thermoelectric properties [43, 257]. These classification approaches involve predicting whether a thermoelectric property is in a desired range, or above (or below) a specified threshold. We argue that regression models, such as ours, provide a level of increased utility via their finer-grained predictions, which is critical when sifting through many thousands of potential candidates. A binary classifier simply provides no convenient means of differentiating the candidates labelled as promising. Although there is room for improvement in the quality of the predictions made by

our regression models, we find that at the current performance level, the approach is effective at surfacing promising candidates.

Third, the use of an attention-based model, in combination with the Robust L2 loss, both leads to superior performance and provides unique advantages. The learned attention weights provide an opportunity to interpret the predictions made for a composition [258], and this could be a useful aspect of using the CraTENet model when analyzing individual materials (rather than in bulk, as we have focused on here). Additionally, the Robust L2 loss is especially useful in that it allows the model to learn to quantify the uncertainty arising from mapping the composition (and optionally band gap) to thermoelectric properties. This provides users with a quantitative measure of the certainty of a prediction.

Future work will involve follow-up investigations of the candidate materials proposed here, using more rigorous *ab initio* methods. Should the candidates continue to appear promising, attempts may be made to synthesize the materials and measure their thermoelectric properties in the laboratory. In terms of the model itself, future work may involve augmenting the objective so that it takes into account the shape of the underlying manifold on which the multiple target values exist [259]. It is important to note that optimal thermoelectric transport properties are not the only criteria that establishes a material as a practical thermoelectric; other properties, such as dopability and stability, need to be considered. Thus, the computational discovery of novel thermoelectrics will be aided by the development of a suite of predictive models.

It is clear that the approach we describe depends heavily on the quality of the data it is trained on. The Ricci database was derived using theoretical constraints such as the CRTA for solving the Boltzmann transport equation, and the GGA for the exchange correlation functionals, which have important limitations. However, the approach we describe here can continue to be used with future databases of computed thermoelectric properties that will be obtained with more accurate theoretical methods, with improved data quality.

Finally, to demonstrate the predictions made by the CraTENet model, we have deployed an internet-accessible web browser-based application, located at
https://thermopower.materialis.ai, that allows a user to submit a material's composition and (optionally) its band gap, and returns thermoelectric transport property predictions for the material, as made by the CraTENet model.

Chapter 5

Crystal Structure Generation with Large Language Models

5.1 Introduction

In this thesis, we have focused on the prediction of properties of materials based solely on their composition. However, to properly define a material, both composition and structure are essential. The accurate evaluation of properties in a solid using *ab initio* techniques requires structural knowledge. Therefore, to confirm ML predictions of properties from composition, a first step is to generate a plausible structure for each composition.

To elucidate the structures of unknown materials, a Crystal Structure Prediction (CSP) approach is often employed, which attempts to derive the ground state crystal structure for a given chemical composition under specific physical conditions [260]. CSP approaches are relatively computationally expensive, typically involving *ab initio* techniques [261]. They often begin with the generation of candidate structures. Examples are the AIRSS [262, 263] and USPEX [264] approaches. Initializing the search space with sensible structures increases the likelihood of success, and decreases the amount of computation required. It is therefore expected that effective crystal structure generation tools would help accelerate the prediction of structures using CSP methods.

Increasingly, generative modeling approaches based on autoencoder architectures and generative adversarial networks (GANs) [121] have been used to generate crystal structures [265–269]. Indeed, generative modeling has become commonplace, an outcome catalyzed by astounding advancements in the computational generation of images, audio and natural language over the last several years [270]. The Large Language Model (LLM), backed by the Transformer architecture [238], is the approach behind state-of-the-art performance on natural language processing tasks. This approach begins with a generative pre-training step, which is autoregressive in nature, involving the unsupervised task of predicting the next token given a sequence of preceding tokens [124]. When such models are scaled to billions of parameters, their effectiveness becomes quite remarkable, as tools such as ChatGPT [127] demonstrate.

LLMs have recently been used in the context of materials science [271–277]. These attempts have been focused on using existing and publicly accessible LLMs, training and tuning LLMs for natural language generation tasks involving chemical subject matter, or training LLMs on a corpus of expanded chemical compositions for the purposes of generating unseen compositions. However, the potential of training LLMs on textual representations of crystal structures has not been considered. A sole exception is a recent pre-print by Flam-Shepherd and Aspuru-Guzik, where the idea of generating the structures of molecules, materials, and protein binding sites with LLMs has been preliminarily explored [278].

Here, we report an LLM specifically designed for crystal generation. This model is distinctively trained on textual representations of inorganic crystal structures, specifically in the Crystallographic Information File (CIF) format [279], instead of relying solely on natural language corpora, or chemical compositions alone. The motivation for this approach originates from two conjectures: The first states that a sequence of symbols (i.e. tokens) is an appropriate representation modality for many predictive tasks, including those involving chemical structure. The idea of representing any domain with a sequence of tokens may at first seem counter-intuitive. However, consider that even images can be represented this way, and be subject to the autoregressive language modeling of pixels [280]. This challenges the notion that domain-specific representations, such as graphs for chemical structure [112], are necessary for superior performance. The second conjecture states that LLMs learn more than simply *surface statistics* and the conditional probability distribution of tokens. Indeed, autoregressive pre-training involving next-token prediction may result in learning an effective *world model*: an internalized causal model of the processes generating the target phenomena. A model which simply learns spurious correlations in the data is less desirable, as it may have greater difficulty in generalizing beyond the training distribution. Recent studies have demonstrated that LLMs trained on sequences of board game play (e.g. chess and Othello) do indeed track the state of the board, and probes of the internal activations of the model reveal the existence of representations of various abstract concepts specific to the domain [281, 282]. We therefore asked whether a model trained to predict the 3-dimensional coordinates of atoms, digit-by-digit, could learn the chemistry implicit in crystal structures, and generate unseen structures, borrowing from its model of the world of atoms.

As such, we herein describe the CrystaLLM model, a tool for crystal structure generation trained on an extensive corpus of CIF files representing the structures of millions of inorganic solid-state materials. Unlike small molecule organic compounds, the generative modeling of inorganic crystals presents unique challenges: the structures are complex and periodic, are not readily described by simple graphs, are imbued with different forms of symmetry, and can be constructed from more than 100 different elements. Even so, the model is capable of reliably generating correct CIF syntax and physically plausible crystal structures for many classes of inorganic compounds. Moreover, we demonstrate how sampling from the model can be improved using the Monte Carlo Tree Search (MCTS) algorithm [283, 284] together with a pre-trained graph-based neural network predictor of formation energy.

5.2 Methods

CrystaLLM is a Transformer-based, decoder-only language model of the CIF file format, trained autoregressively on a corpus of millions of CIF files (Figure 5.1a). Rather than training on structural representations derived from the CIF files, the model is directly trained on the standardized and tokenized text contents of the CIF files. During training, the model is given a sequence of tokens from the corpus of CIF files, and is tasked with predicting the tokens which follow each of the given tokens. Once the model is trained, it can be used to generate new CIF files, conditioned on some starting sequence of tokens. Generating a CIF file involves repeatedly sampling tokens from the model, conditioning on the accumulated generated content, until a terminating condition is reached (Figure 5.1b).

To assess the ability of the model to generate structures, a test set of approximately 10,000 randomly chosen CIF files is withheld from a training set of approximately 2.2 million CIF files, and the model is tasked with generating CIF files beginning from prompts constructed from the test set. Moreover, we assemble what we call a *challenge set*, which consists of 70 structures, 58 of which were obtained from the recent literature, and were not in the training

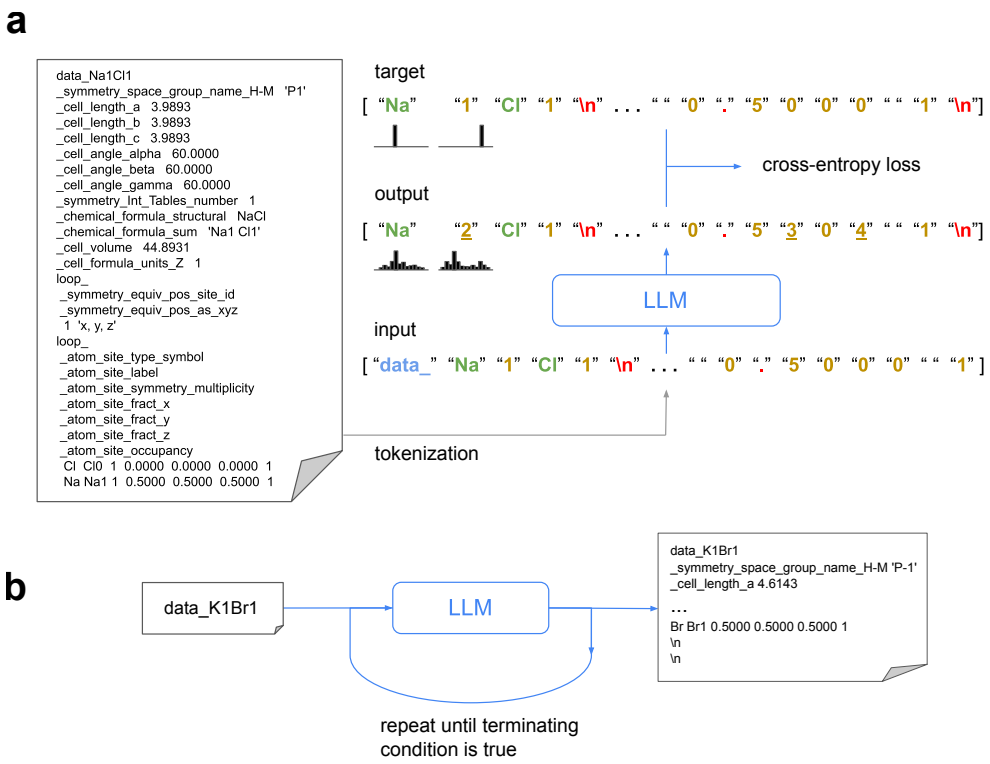

Figure 5.1: **a** Core concepts in training a Large Language Model of CIF files: A CIF file (left) is converted into a sequence of symbols, through tokenization. The sequence is processed by the model, which produces a list of probability distributions over the vocabulary, for each corresponding symbol in the input. The resulting predicted probability distributions are evaluated against the target distributions (which contain the entire probability mass on the correct subsequent token), using the cross-entropy loss metric. The target tokens are the input tokens shifted one spot to the left, as the objective is to predict the next token given a sequence of preceding tokens. The tokens are categorized as CIF tags (blue), atoms (green), numeric digits (gold), and punctuation (red). Output tokens (not actually sampled during training) represent the tokens assigned the highest probability by the model. Underlined tokens represent predicted distributions assigning a relatively low probability to the correct next token. **b** Generation of a CIF file: First, a prompt is constructed by concatenating the symbol `data_` with the desired cell composition, which is then tokenized and processed by the model. Next, a token is sampled from the predicted distribution for the upcoming token in the sequence. Finally, the sampled token is added to the accumulating contents of the CIF file. This procedure continues iteratively until a predefined terminating condition is met (e.g. two consecutive newline tokens are sampled).

set. The remaining 12 structures are from the training set, and are included as representatives of different structural classes. They serve to assess the model's ability to recover what it has seen in training, and as a means of comparing the model's generations of seen and unseen structures. (Supplementary Table 1 in Appendix C contains the full list of the challenge set compounds, and their sources.) The permutative nature of the dataset, with many structures having been derived by substituting atoms into pre-defined templates, results in a test set with the potential for some structures to closely resemble those of the training set. The challenge set provides a source of structures that are guaranteed to have been produced through a different process. Moreover, the challenge set constitutes a manageable set of compounds that reflects a variety of solid-state structural classes, allowing for a fine-grained picture of the model's capabilities. The test set, on the other hand, is better suited for a bulk assessment, and originates from the same distribution as the training set.

The following terminology is used in the remainder of this article: A *formula*, *reduced formula*, or *reduced composition*, refers to the empirical formula, or formula unit, which is the simplest, whole-number ratio of atoms in the compound. An example of a formula is Ba_2MnCr. A *cell composition* is a chemical formula referring to the total number of atoms of each type in the unit cell of a crystal. It represents the chemical formula of the compound as it would appear in the crystallographic unit cell, which might contain Z formula units. An example of a cell composition is $Ba_6Mn_3Cr_3$, with a Z of 3.

5.2.1 Training and Learned Representations

Training consists of iteratively sampling sequences of tokens, of fixed length, and adjusting the model's parameters so that it becomes progressively better at predicting which token should follow a preceding sequence. (See the Methods, and Supplementary Note 2 in Appendix C, for more information on the model architecture and training.) Since it has been observed that LLM performance improves as the number of model parameters is increased [285], we train a *small* model, consisting of 25 million parameters, and a *large* model, consisting of 200 million parameters.

To monitor the progress of training, we withhold a validation set that constitutes 10% of the set held-out for training. Over the course of training, the model continues to improve in terms of its total cross-entropy loss on the validation set, even after 90,000 iterations (see Supplementary Figure 2 in Appendix C). We note, however, that improvements appear to become smaller with more training time.

As a consequence of the model's architecture, each token in a processed sequence is mapped to a distinct learned vector representation using an embedding table, whose parameters are adjusted during training. The result is that, through autoregressive training, distributed representations are learned for each symbol in the vocabulary. The vocabulary consists of symbols for atoms, space groups, and numeric digits. (See Supplementary Note 1 in Appendix C for a detailed description of the vocabulary and the tokenization procedure.) The training process appears to result in sensible representations of these various symbols. Plots of dimensionally-reduced atom and space group vectors demonstrate a logical structure, where similar entities cluster together, indicating that intrinsic properties and relationships are captured. (See Supplementary Figure 3 in Appendix C for plots of the learned atom vectors, and Supplementary Figure 4 in Appendix C for a plot of the learned space group vectors.) Moreover, examination of the learned numeric digit vectors reveals that numerical relationships are captured in the representations, as measurements of cosine and Euclidean distances between the learned digit vectors demonstrate a logical spatial relationship. (See Supplementary Figure 5 in Appendix C.) While not explored further in this work, we note that

distributed representations of chemical entities, such as atoms, are useful for the prediction of materials properties [243, 286].

5.2.2 Dataset Curation

The dataset was assembled by obtaining structures from the Materials Project [155], the OQMD [159], and NOMAD [287], which were originally optimized using density functional theory (DFT) simulations. Specifically, the structures from the Materials Project were downloaded in April 2022, and from NOMAD in April 2023. We use version 1.5 of the OQMD, which was released in October 2021. In total, approximately 3.6 million structures were obtained. This dataset consists of compounds containing anywhere from 1 to 10 elements, with most consisting of 3 or 4 elements. The elements up to and including atomic number 94 are present, with the exception of polonium, astatine, radon, francium, and radium. The dataset contains roughly 800,000 unique formulas, and 1.2 million unique cell compositions. When paired with space groups, there are 2.3 million unique cell composition-space group pairs. (See Supplementary Figure 1 in Appendix C.) To choose between duplicate structures containing the same cell composition and space group, the structure with the lowest volume per formula unit was selected. The 2.3 million structures in this dataset were converted to CIF files using the pymatgen library [288], and were used for training. The CIF files were created with the pymatgen option for symmetry finding tolerance set to 0.1 Å. All floating point numbers in the files were rounded to 4 decimal places. The dataset was split randomly into train, validation, and test sets, such that the training set consisted of 2,047,889 CIF files, the validation set 227,544 CIF files, and the test set 10,286 CIF files.

5.2.3 CIF Syntax Standardization and Tokenization

The dataset of CIF files was standardized and tokenized prior to training. The vocabulary consisted of CIF tags, space group symbols, element symbols, numeric digits, and various punctuation symbols, for a total of 371 symbols. After tokenization, the training set consisted of 768 million tokens. See Supplementary Note 1 in Appendix C for further details.

5.2.4 Generative Pre-training

The generative pre-training step requires a vocabulary, $\mathcal{V}$, and an ordered list of tokens $\mathcal{U} = (u_1, ..., u_n)$, with $u_i \in \mathcal{V}$. We want to maximize the following likelihood:

$$\mathcal{L}(\theta; \mathcal{U}) = \sum_i \log P(u_i | u_{i-c}, ..., u_{i-1}; \theta) \tag{5.1}$$

where c is the size of a context window, P is the conditional probability distribution to be modelled, and θ the parameters of a neural network. We therefore minimize $\mathcal{J}(\theta; \mathcal{U}) = -\mathcal{L}$, using stochastic gradient descent to adjust the parameters. We use a multi-layer Transformer decoder [289] for the neural network, as described in [124]. Our model consists of 25 million parameters, with 8 layers, 8 attention heads, and an embedding size of 512. We decay the learning rate from 10^{-3} to 10^{-4} over the course of training, and use a batch size of 32. For further details, see Supplementary Note 2 in Appendix C.

5.2.5 Evaluation of Generated Structures

A CIF file is said to be *valid* if: 1) the declared space group is consistent with the generated structure, 2) the generated bond lengths are reasonable, and 3) the declared atom site multiplicity is consistent with the cell composition. To check if the generated structure is consistent

with the printed space group, we use the `SpacegroupAnalyzer` class of the pymatgen library, which uses the spglib library [290]. To check if bond lengths are reasonable, we first use a Voronoi-based nearest-neighbour algorithm in pymatgen to identify bonded atoms; then, we establish expected bond lengths based on the electronegativity difference between the bonded atoms, and their ionic or covalent radii. We classify a structure as having reasonable bond lengths if all the detected bond lengths are within 30% of the corresponding expected bond lengths. See Supplementary Note 3 in Appendix C for more details on how the validity of a generated CIF file is established.

In some scenarios, we wish to determine whether a generated structure matches a target structure, which typically represents a ground-truth structure. To determine whether two structures are a match, we use the pymatgen `StructureMatcher` class, which performs a structural similarity assessment of two crystals. We use a fractional length tolerance of 0.2, a site tolerance of 0.3 Å, and an angle tolerance of 5 degrees, which are the default values in pymatgen. Both structures are reduced to primitive cells before matching, and are scaled to equivalent volume.

5.2.6 Benchmark Evaluations

CSP Tasks

To evaluate CrystaLLM on the Perov-5, Carbon-24, MP-20 and MPTS-52 benchmarks, we consider two different scenarios: 1) the model is trained only on the benchmark training sets, and 2) the model is trained on the full 2.3 million-structure dataset minus the validaton and test set structures of the MPTS-52 dataset. For the first scenario, both the small and large model architectures are used. We use the same 60-20-20 train/validation/test splits used in the CDVAE study [266] for the Perov-5, Carbon-24, and MP-20 datasets, and we use the same 27,380/5,000/8,096 train/validation/test split used in the DiffCSP study for the MPTS-52 dataset. These models are trained for a fixed number of iterations: the Perov-5 model is trained for 1,750 iterations, the Carbon-24 model is trained for 8,000 iterations, the MP-20 model is trained for 5,000 iterations, and the MPTS-52 model is trained for 3,500 iterations. For the second scenario, we train a model with the small model architecture on the full 2.3 million-structure dataset minus the structures of the MPTS-52 validation and test sets. The model is trained for 100,000 iterations. We decay the learning rate from 10^{-3} to 10^{-4} over the course of training, and use a batch size of 32, for all models. For both scenarios, we take the structures of the test set(s), and prompt the models with only the cell compositions of these structures. Models are given 20 attempts to generate a structure. We use top-k sampling with $k = 10$ and a temperature of 1.0 for all models and in both scenarios.

To establish the match rate and RMSE, we use the same procedure defined in the DiffCSP study. Specifically, we use the pymatgen `StructureMatcher` class, with a fractional length tolerance of 0.3, a site tolerance of 0.5 Å, and an angle tolerance of 10 degrees, to determine if a generation attempt matches the ground truth structure. The RMSE, normalized by $\sqrt[3]{V/N}$ (where V is the volume of the lattice and N is the number of sites), is computed between the corresponding ground truth structure and each matching generated structure. The test set's average RMSE is computed by taking the lowest RMSE for each entry's matching generated structure.

Unconditional Generation Tasks

To evaluate CrystaLLM on the unconditional generation tasks, we train a model on the training sets of each of the Perov-5, Carbon-24 and MP-20 datasets, using both the small and large

model architectures. We use the same 60-20-20 train/validation/test splits used in the CDVAE study [266]. These models are trained for a fixed number of iterations: the Perov-5 model is trained for 5,000 iterations, the Carbon-24 model is trained for 8,000 iterations, and the MP-20 model is trained for 5,000 iterations. We decay the learning rate from 10^{-3} to 10^{-4} over the course of training, and use a batch size of 32, for all models. Models are then given 10,000 generation attempts, starting from the prompt '`data_`'. Each generation attempt results in both a generated cell composition and structure. We use top-k sampling with $k = 30$ and temperatures of 0.5 and 0.7 for all models.

To establish the unconditional generation metrics, we follow the same procedure defined in the CDVAE study. Specifically, structural fingerprints are created using the `CrystalNNFingerprint` class with the "ops" preset, and compositional fingerprints are created using the `ElementProperty` class with the "magpie" preset, both provided by the matminer library [246]. For the coverage metrics, we use the standard cutoff values: for MP-20, we use a structure cutoff of 0.4 and a composition cutoff of 10; for Carbon-24 and Perov-5, we use a structure cutoff of 0.2 and a composition cutoff of 4.

5.2.7 Monte Carlo Tree Search Decoding

The MCTS search tree is constructed iteratively, as the search proceeds. We maintain a tree width of 5, and maximum tree depth of 1,000. The PUCT constant c_{puct} is set at 1.0. The expansion involves adding child nodes based on predicted probabilities. When a node has a probability of 0.99 or greater, it becomes the only child node, and bypasses the rollout step. During the rollout step, the CrystaLLM model is prompted with token sequences until a terminating condition is met, up to a maximum of 1,000 tokens. Evaluation is conducted using the ALIGNN model of formation energy per atom. The ALIGNN model is given the generated CIF file, and the predicted formation energy per atom (in eV) is used to compute the reward. The backpropagation step accumulates outcomes in the tree nodes, scoring each based on the quality of the generated structure, with a reward constant λ of 2.0. For all compounds, we perform 1,000 search iterations. See Supplementary Note 4 in Appendix C for a more detailed description of the algorithm.

5.2.8 Uniqueness and Novelty of Generated Materials

To assess the model's ability to generate materials unseen in training, the model is prompted with '`data_`' 1,000 times, each resulting in a CIF file. We use top-k sampling with $k = 10$ and a temperature of 1.0. (In principle, the chosen temperature should affect the trade off between novelty rate and how reasonable the generated structures are, so temperature should be considered a parameter to be optimized in future studies.) To establish uniqueness and novelty of the generated structures, we use the pymatgen `StructureMatcher` class, with a fractional length tolerance of 0.2, a site tolerance of 0.3 Å, and an angle tolerance of 5 degrees. A generated compound is considered *unique* if it represents a structural type that appears only once amongst all compounds generated during the experiment, under the specified tolerances for lattice dimensions and atomic positions configured for the `StructureMatcher` class. A generated compound is considered *novel* if it is structurally distinct from all of the compounds in the dataset used to train the model.

5.2.9 DFT Calculations

For the pyrochlore case study, a small number of DFT calculations were performed using VASP, following as closely as possible the settings used in the OQMD project (where most of

the pyrochlore structures seen in training were taken from). For example, the recommended PAW potential was used for each element: Zr_sv for zirconium, Hf_pv for hafnium, Lu_3 for lutetium, Pr_3 for praseodymium, Ce_3 for cerium (for the remaining elements, the name of the PAW potential simply matched the element's symbol). The Perdew-Burke- Ernzerhof (PBE) exchange-correlation functional [47], in the generalized-gradient approximation, was used in all calculations. Hubbard (PBE+U) corrections were applied for transition metal elements with unfilled d levels (U_{eff}=3.8 eV for Mn and 3.1 eV for V). Although the cell parameters reported here correspond to the conventional cubic cell with 8 formula units, the DFT calculations were performed using the primitive cell with two formula units, and sampling of the reciprocal space corresponding to that primitive cell was performed using a 7x7x7 grid, as done for all pyrochlore calculations in the OQMD project.

For the DFT calculation of the energy against hull of the unconditionally generated compounds, we also used the VASP code, following the Materials Project settings [291], i.e. same functional (PBE), U_{eff} parameters, PAW potentials, etc. to ensure compatibility with reference compounds in the hull. Structures generated with CrystaLLM were relaxed to the nearest local minima within the generated unit cell, without symmetry constraints on the atomic coordinates (we applied small random displacements of less than 0.1 Å to the initial coordinates). All the DFT calculations converged, electronically and ionically, within the standard convergence thresholds in the Materials Project setup.

5.2.10 Web Application

The web application is made available at https://crystallm.com. The user of the application is presented with a text field requiring a formula to be entered. Optionally, they may provide the number of formula units (Z), the desired space group, and the size of the model. Once they press the `Generate` button, a request is sent to a GPU server which has the model in memory. The request is converted into a prompt, and the generated contents are returned to the user. If no Z is provided, we scan through Z values of 1, 2, 3, 4, 6, and 8, and return the first valid structure generated by the model. We validate the generated structure using the same procedure described previously, checking that the generated structure is consistent in terms of the printed space group, and other elements of the CIF file. If no valid structure can be found, the user is presented with an informative error message, including the option to view the generated content. Requests typically take several seconds to process, but can take longer if no Z is provided and the model has trouble generating a valid structure for the attempted Z values. Generated structures are displayed in a web browser-based 3D structure viewer provided by the Crystal Toolkit framework, upon which the front-end of the web application is built [292].

5.2.11 Data Availability

The structures used in the experiments described in this chapter were obtained from the Materials Project (https://materialsproject.org/), the OQMD (https://oqmd.org/), and NOMAD (https://nomad-lab.eu/). All structures were made available by those sources under the Creative Commons Attribution 4.0 License [293].

All trained models, training sets, and artifacts generated by the models have been deposited to Zenodo. The files are publicly accessible at: https://zenodo.org/records/10642388. All files are released under the CC-BY 4.0 license.

5.2.12 Code Availability

The code for training and using the CrystaLLM model is open source, released under the MIT License. The code repository is accessible online, at:
https://github.com/lantunes/CrystaLLM.

5.3 Results and Discussion

5.3.1 Generalizing to Unseen Structures

To evaluate the ability of the model to generate an unseen structure, the model is prompted with the structure's cell composition, and allowed to generate up to 3,000 tokens. The prompt includes the first line of the CIF file, which consists of the data block header, containing the cell composition of the structure. Subsequently, the model is prompted with both the structure's cell composition and space group, and again allowed to generate up to 3,000 tokens. The prompt includes the first several lines of the pre-processed CIF file, up to the line containing the specification of the space group. Prompting the model with both the cell composition and space group allows us to assess how reliant the model is on the space group. This process is repeated for all CIF files of the held-out test set (10,286 in total).

The generated CIF files are then assessed for correctness and quality. Any syntactically incorrect CIF files are declared invalid. Syntactically correct CIF files are subjected to further analysis, and are considered to be valid only if specific criteria are met, such as being consistent in terms of generated structure and declared space group, and having reasonable bond lengths (see Supplementary Note 3 in Appendix C for further details on the validation of generated CIF files). The results of evaluating the generation of the CIF files of the test set using the small model are presented in Table 5.1.

Table 5.1: Performance of the small model on the held-out test set. The percentages represent the fraction of test set compounds which meet the corresponding criteria. For example, the first row represents the percentage of test set compounds where the declared space group in the generated CIF file is consistent with the generated structure. Valid generated length refers to the length of a valid generated CIF file in terms of the number of tokens.

	No Space Group	**With Space Group**
Space Group Consistent	98.8%	99.1%
Atom Site Multiplicity Consistent	99.4%	99.4%
Bond Length Reasonableness Score	0.9878 $\pm$ 0.0686	0.9878 $\pm$ 0.0671
Bond Lengths Reasonable	94.6%	94.6%
Valid	93.8%	94.0%
Longest Valid Generated Length	1145	970
Average Valid Generated Length	331.885 $\pm$ 42.567	339.002 $\pm$ 41.361

The CIF files generated by prompting the model with the cell composition and space group were compared to the corresponding CIF files of the test set using a structure matching algorithm. The fraction of matching structures is presented in Table 5.2.

Table 5.2: Structure matching results for the test set when the space group is included in the prompt. The *Reduced Unseen* column represents the results for formulas that were not seen in training with any Z. [1]

	All	**Reduced Unseen**
At least 1 match within 3 attempts	88.1%	86.3%
All 3 attempts matching	67.4%	70.0%
Matched on 1st attempt	78.4%	78.7%

We further examined how closely the generated cell parameters resembled the actual cell parameters, for the cases where there was a structural match. We took the first matching structure for samples that had at least one generated structure matching the test set structure, and measured the R^2 and mean absolute error (MAE) for the true versus generated cell lengths, the true versus generated (i.e. printed) volume, and the implied (from generated cell parameters) versus generated volume. The results are presented in Figure 5.2.

[1] For example, if Na1Cl1 were in training, Na2Cl2 may be in *All* but not in *Reduced Unseen*.

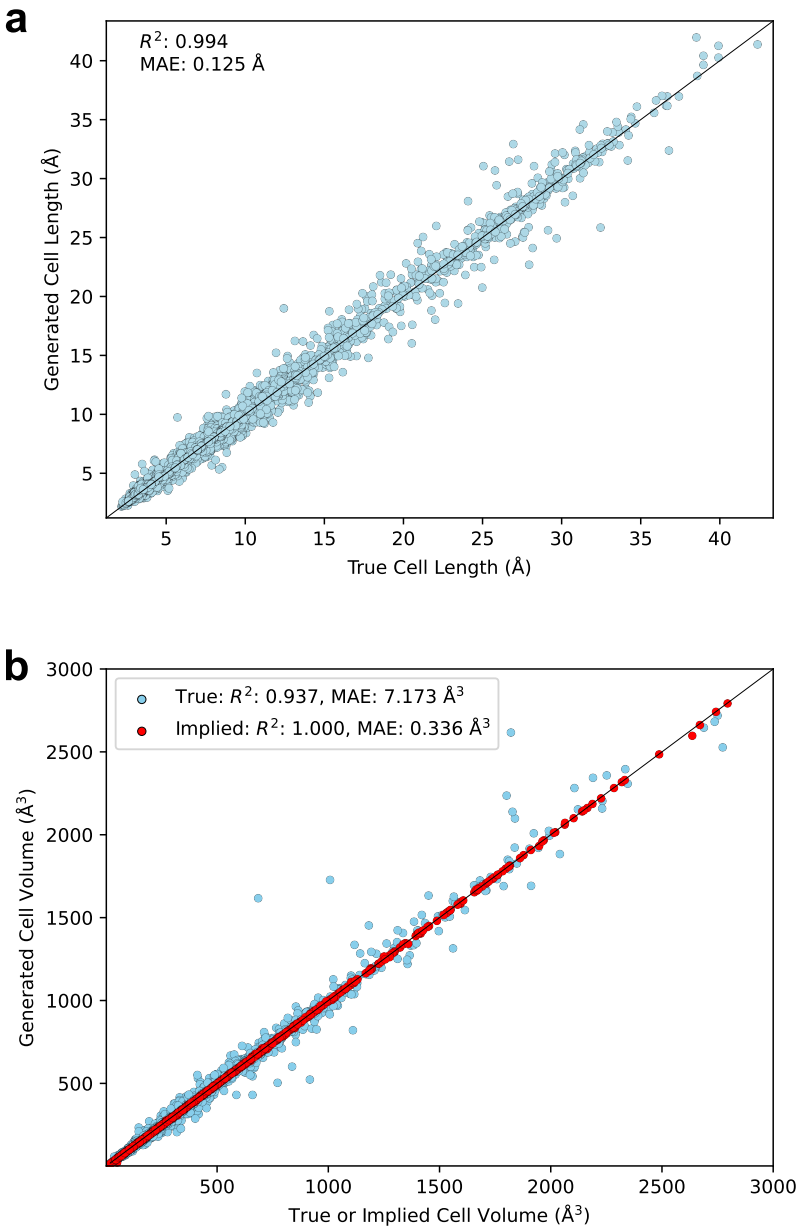

Figure 5.2: **a** The generated cell lengths for matching structures of the test set vs. the true cell lengths, when space group is included. **b** The generated cell volumes for matching structures of the test set vs. either the true cell volumes, or the cell volumes implied from the generated cell parameters, when space group is included.

To further assess the model's ability to generalize to unseen structures, we prompted the model with the cell compositions of the challenge set. The challenge set contains 58 structures not seen in training. These structures were all manually sourced from the recent literature, and represent experimentally characterized materials. Crucially, these compounds originate through a process different from the process which generated the training set (namely, a high-throughput DFT analysis of hypothetical materials). They also represent a variety of different structural classes, such as intermetallics, silicates, sulfides and selenides, borates, phosphates,

carbonates, and complex mixed-anion compounds.

Both the small and large models were prompted with the cell compositions of the challenge set, both with and without the space group. A total of 100 attempts were made to generate a structure from the given cell composition (and optionally space group). We record the successful generation rate, representing the fraction of compounds where at least one valid CIF file was generated in the 100 attempts, and the true match rate, representing the fraction of compounds where there was a structural match between a valid generated structure and the true structure reported in the literature. The results are presented in Table 5.3 and Supplementary Tables 2 to 5 in Appendix C.

Table 5.3: Results of the small and large models on the challenge set, both with a space group ('s.g.') and without. The first row represents the percentage of cases where the model was able to generate a valid structure within 100 attempts. The second row represents the percentage of cases where a generated structure matched the true structure, for the compounds seen in training. The last row represents the percentage of cases where a generated structure matched the true structure, for unseen compounds only.

	Small model		**Large model**	
	no s.g.	**with s.g.**	**no s.g.**	**with s.g.**
Successful Generation Rate	85.7%	88.6%	87.1%	91.4%
Match Rate (Seen)	50.0%	50.0%	83.3%	83.3%
Match Rate (Unseen)	25.9%	34.5%	37.9%	41.4%

The results in Table 5.3 indicate that inclusion of the space group in the prompt increases the likelihood of generating a valid structure, and of generating a match with the true structure. The large model appears to be superior to the small model in all categories. While the models can recover the reported structure more often when the structure was seen in training, it is noteworthy that they are able to generate unseen structures which match the reported structure in up to 40% of the cases.

5.3.2 Comparison with Other ML-based Approaches

Generative models of materials based on advanced ML techniques have been developed recently, some concurrently with this work. Due to the unavailability of source code and complete benchmarking results for all these emerging models, conducting an in-depth comparison between the approaches remains challenging. Nevertheless, here we present a comparison with other ML-based approaches. CDVAE [266], DiffCSP [294], DiffCSP++ [295] and UniMat [296] are examples of diffusion-based approaches, whereas Gruver *et al.* [297] introduced a fine-tuned version of the LLaMA-2 model [298] for crystal structure generation. DiffCSP focuses on crystal structure prediction through an equivariant diffusion process, while CDVAE uses a diffusion-based approach within a variational autoencoder framework for generating periodic materials. DiffCSP++ augments the equivariant diffusion process by introducing support for space-group constrained generation, through the incorporation of prior knowledge of Wyckoff positions which constrain the diffusion process. UniMat re-purposes the 3D U-Net architecture [299, 300] for unconditional generation, and generation conditioned on composition, and is trained on a large dataset of millions of structures.

We compare CrystaLLM to these models in both conditional and unconditional generation settings. In the Crystal Structure Prediction (CSP, or conditional) generation setting, we compare CrystaLLM to these models on four benchmarks: Perov-5 [301, 302], Carbon-24 [303], MP-20 [155], and MPTS-52 [304]. The Perov-5 dataset consists of 18,928 perovskites,

Carbon-24 consists of 10,153 carbon allotropes, MP-20 consists of 45,231 stable inorganic materials of various classes, while MPTS-52 consists of 40,476 various inorganic materials. MPTS-52 is by far the most complex dataset, with up to 52 atoms in the unit cells of the constituent structures. In the unconditional generation setting, CrystaLLM is compared to these models on the Perov-5, Carbon-24, and MP-20 benchmarks.

The benchmark datasets have each been split into training, validation and test sets. All models are trained solely on the training set. For the CSP task, the models are used to generate 20 structures for each of the cell compositions of the test set. The models are evaluated in terms of the *match rate*, which is the fraction of compositions for which the true structure was generated within n attempts (we tried n=1 and 20), and the average root mean squared error (RMSE) of the closest candidate for each test set structure. For the unconditional generation task, the models are given 10,000 generation attempts, and are evaluated in terms of metrics such as validity rate and coverage. In the validity tests, following the metrics presented in other studies, a structure is defined as valid if no interatomic distances are below 0.5 Å, and a composition is defined as valid if a charge neutral combination of the constituent atoms in the generated stoichiometry is possible. Coverage measures how closely the generated materials match the distribution of ground truth materials. Coverage precision is a measure of how many generated materials are a close match to materials from the ground truth set and is an indication of the quality of the generated materials. Being a close match is defined by distance between the materials with a pre-defined metric (see Supplementary Note 5 in Appendix C for more details). Coverage recall measures how many of the ground truth materials are matched by at least one generated material, and is a measure of how diverse the generated materials are. For example, a generating process could have high COV-P by simply generating the same valid material each time, but COV-R would be low in this instance. The AM(S/C)D measures are similar to coverage statistics, but measure the minimum distance between a generated material and the materials in the ground truth set; these measures are also separated across structural matching (AMSD) and composition matching (AMCD). While the COV-R and COV-P metrics have become established for the purposes of evaluating generative models of materials, we note that they must be interpreted cautiously, as they have several drawbacks. Primarily, the metrics do not fully account for the novelty of the generated materials, focusing instead on similarity, which depends on arbitrarily set thresholds. This can favor models which are overfit to the dataset, and not necessarily generalizable. Moreover, the metrics can be sensitive to the relative sizes of the test and generated sets, which can lead to potentially misleading scores, since a larger generated set together with a smaller test set might result in artificially high COV-R values, while a smaller generated set could inflate COV-P values. The results for the CSP task are presented in Table 5.4, and the results for the unconditional generation task are presented in Tables 5.5 and 5.6.

For the CSP task, we present results for three different versions of CrystaLLM. Versions a and b are trained on the benchmark data only and differ in the size of the model used. Version c is trained on the full 2.3M training points minus the test set of MPTS-52 and is included to demonstrate how the results improve with the size of training data, but is not directly comparable to other models due to the different training data sets. For the unconditional generation task, we present results for both the small and large CrystaLLM models, with different sampling temperatures. Supplementary Table 8 in Appendix C contains comprehensive results for the unconditional generation task.

Table 5.4: Benchmark CSP results. Numbers in bold indicate the best n=20 result, while italicized numbers represents the best n=1 result, amongst the models trained only on the benchmark training sets, where n represents the number of samples generated for each structure of the benchmark test set. *a* Results for the small model architecture trained only on the benchmark training sets. *b* Results for the large model architecture trained only on the benchmark training sets. *c* Results for the small model architecture trained on the original 2.3M-structure dataset without the structures of the MPTS-52 validation or test sets. The CDVAE and DiffCSP results are taken from Jiao *et al.* [294].

		Perov-5		Carbon-24		MP-20		MPTS-52	
Model	n	Match Rate	RMSE	Match Rate	RMSE	Match Rate	RMSE	Match Rate	RMSE
CDVAE	1	45.31	0.1138	17.09	0.2969	33.90	0.1045	5.34	0.2106
CDVAE	20	88.51	0.0464	88.37	0.2286	66.95	0.1026	20.79	0.2085
DiffCSP	1	**52.02**	**0.0760**	17.54	0.2759	51.49	0.0631	12.19	0.1786
DiffCSP	20	**98.60**	**0.0128**	**88.47**	0.2192	**77.93**	0.0492	**34.02**	0.1749
CrystaLLM [a]	1	47.95	0.0966	**21.13**	**0.1687**	55.85	0.0437	17.47	0.1113
CrystaLLM [a]	20	98.26	0.0236	83.60	0.1523	75.14	0.0395	32.98	0.1197
CrystaLLM [b]	1	46.10	0.0953	20.25	0.1761	**58.70**	**0.0408**	**19.21**	**0.1110**
CrystaLLM [b]	20	97.60	0.0249	85.17	**0.1514**	73.97	**0.0349**	33.75	**0.1059**
CrystaLLM [c]	1	-	-	-	-	-	-	28.30	0.0850
CrystaLLM [c]	20	-	-	-	-	-	-	47.45	0.0780

Table 5.5: Validity and Coverage metrics for the unconditional generation tasks. Numbers in bold indicate the best results for the given task. The LM-CH (character-level tokenization) and LM-AC (atom+coordinate-level tokenization) results are from Flam-Shepherd *et al.* [278]. The LLaMA 70B results are from Gruver *et al.* [297]. τ represents the sampling temperature.

	Validity (%)		Coverage (%)	
Method	Struct	Comp	COV-R	COV-P
MP-20				
CDVAE	**100.0**	86.70	99.15	99.49
DiffCSP	**100.0**	83.25	99.71	**99.76**
DiffCSP++	99.94	85.12	99.73	99.59
UnitMat	97.20	89.40	**99.80**	99.70
LM-CH	84.81	83.55	99.25	97.89
LM-AC	95.81	88.87	99.60	98.55
LLaMA 70B (τ=1.0)	96.50	86.30	96.80	98.30
LLaMA 70B (τ=0.7)	99.60	**95.40**	85.80	98.90
CrystaLLM small (τ=0.5)	94.97	93.80	97.58	95.75
CrystaLLM large (τ=0.5)	96.21	**95.40**	96.78	96.60
Perov-5				
CDVAE	**100.0**	98.59	99.45	98.46
DiffCSP	**100.0**	98.85	**99.74**	98.27
DiffCSP++	**100.0**	98.77	99.60	98.80
UnitMat	**100.0**	98.80	99.20	98.20
LM-CH	**100.0**	98.51	99.60	**99.42**
LM-AC	**100.0**	98.79	98.78	99.36
CrystaLLM small (τ=0.5)	99.83	**99.24**	97.91	98.95
CrystaLLM large (τ=0.7)	99.82	98.92	98.28	98.92
Carbon-24				
CDVAE	**100.0**	-	99.80	83.08
DiffCSP	**100.0**	-	99.90	97.27
DiffCSP++	99.99	-	**100.0**	88.28
UnitMat	**100.0**	-	**100.0**	96.50
CrystaLLM small (τ=0.5)	99.86	-	99.80	98.96
CrystaLLM large (τ=0.5)	99.90	-	99.75	**99.52**

Table 5.6: Average Minimum Distance metrics for the unconditional generation tasks. Numbers in bold indicate the best results for the given task. τ represents the sampling temperature.

Method	AMSD-R	AMSD-P	AMCD-R	AMCD-P
MP-20				
CDVAE	0.154	0.188	3.620	4.014
UnitMat	0.097	0.119	**2.410**	2.410
CrystaLLM small (τ=0.5)	0.106	0.095	3.729	**1.762**
CrystaLLM large (τ=0.7)	**0.090**	**0.077**	3.299	2.114
Perov-5				
CDVAE	0.048	0.059	**0.696**	**1.270**
UnitMat	0.046	0.074	0.711	1.399
CrystaLLM small (τ=0.7)	**0.025**	0.027	1.055	1.287
CrystaLLM large (τ=0.5)	0.027	**0.020**	1.144	1.319
Carbon-24				
CDVAE	0.048	0.134	0.000	0.000
UnitMat	0.018	0.052	0.000	0.000
CrystaLLM small (τ=0.7)	**0.015**	0.021	0.000	0.000
CrystaLLM large (τ=0.5)	0.018	**0.010**	0.000	0.000

In the CSP task, CrystaLLM outperforms DiffCSP on three out of four benchmarks in terms of RMSE for both n=20 and n=1, and in terms of match rate when constrained to only a single generation attempt. This is achieved even in the most challenging of the benchmarks, MPTS-52, which contains structures with larger unit cells and more atoms. In the unconditional generation task, CrystaLLM is competitive with the other models, and also achieves strong results in terms of compositional validity on MP-20 and Perov-5, and obtains the highest COV-P value on Carbon-24. Furthermore, the best AMSD metrics are achieved by CrystaLLM on all three benchmarks.

CrystaLLM has important advantages when compared to the other models. In comparison to the diffusion-based methods, CrystaLLM supports both conditional and unconditional generation seamlessly, without requiring any architectural adjustments. More (or less) information is simply provided in the prompt, accordingly. Conversely, DiffCSP requires architectural augmentation to support unconditional generation, and CDVAE also requires an architectural adjustment to support conditional generation. Another important advantage is that CrystaLLM natively supports space-group constrained generation, with no changes or external processing required. Conversely, DiffCSP++ was devised as a separate approach dedicated to handling space-group constrained generation. It relies on a template retrieval and substitution method when the space group is unknown. In contrast, CrystaLLM generates a suitable space group automatically, with no extra work required. The DiffCSP++ template-based approach consequently makes it difficult to propose structures when no suitable template exists, which is a limitation that CrystaLLM does not have. CDVAE and UniMat do not support space group-constrained generation.

In comparison to the fine-tuned LLaMA-2 model, the largest CrystaLLM model has 200 million parameters, whereas the smallest fine-tuned LLaMA-2 model has 7 billion parameters, a difference of more than an order of magnitude in the number of parameters. The smaller size of CrystaLLM makes it easier to deploy for inference tasks, and much more accessible for training and fine-tuning. Additionally, while the fine-tuned LLaMA-2 model supports the constructs of natural language in its prompts, the flexibility of its inputs suggests that CrystaLLM may be conditioned on other properties of the structure as well, including those not traditionally included in the CIF format.

Finally, as a neural language model, CrystaLLM can leverage the established practice

of fine-tuning, allowing the pre-trained model to be adapted for the prediction of materials properties. There is far less precedent in fine-tuning models based on diffusion and variational autoencoder architectures for tasks involving regression or classification.

The differences above between CrystaLLM and previous methods indicate that CrystaLLM has the unique advantage of being a more flexible, general-purpose model, capable of supporting a number of different generation use cases, without requiring a switch between architectural variants, or different models entirely, and which can be deployed in a cost-effective manner. CrystaLLM can alternate seamlessly between unconditional generation (when neither composition nor space group is known), generation conditioned on composition only, and generation conditioned on both composition and space group. Notably, it supports the conditioning of structure generation on specific symmetry space groups without being restricted, in principle, to the availability of known templates, a capability unique to CrystaLLM.

5.3.3 Examples of Generated Structures

To further examine the model's ability to generalize to unseen scenarios, we prompted the model with various formulas, and examined its output. The results are presented in Figure 5.3.

An example of the model generalizing to a formula that had been seen in training, but with different space groups, is presented in Figure 5.3a. The formula, Ba_2MnCr, was in the held-out test set, with the $R\bar{3}m$ space group. That combination of formula and space group had not been seen in training. The model generated a structure matching the one in the test set on the first attempt, when the space group was provided.

The model also demonstrated the ability to generate plausible structures for formulas not seen in training with any Z. An example is the quaternary compound CsCuTePt. This compound was not in the training set, but was in the held-out test set (with Z=4). The model generated a structure matching the one in the test set, in the $F\bar{4}3m$ space group, on the third attempt when the space group was provided. The generated structure is presented in Figure 5.3b.

Finally, in Figure 5.3c is the generated structure of $YbMn_6Sn_6$ [305], an example of the model generalizing to structural motifs with atoms not seen in training. This formula was not seen in training for any Z, and was not in the held-out test set. However, $ZrMn_6Sn_6$ was seen in training, in the *P6/mmm* space group. The model generated a structure in the same space group on the first attempt, without the space group being provided. The generated structure matched the $ZrMn_6Sn_6$ structure, with Yb substituted for Zr, and with cell parameters and atomic coordinates adjusted accordingly. This demonstrates the model performing a structure prediction by analogy procedure, as commonly used by materials scientists for discovery [114, 306], despite never having been provided with the procedure to do this.

Rutiles

Rutiles are a class of binary compounds that adopt a tetragonal unit cell, in the $P4_2/mnm$ space group (Z=2), as is seen in TiO_2, from which this class of materials adopts its name. The general formula for rutile oxides is MO_2, where M is a metallic species in the $+4$ oxidation state. Rutile fluorides are also known, where the metal is in the $+2$ oxidation state.

The model's training dataset consisted of essentially all of the rutiles one might expect to be able to find in nature. Therefore, to test the model's ability to generate unseen rutiles, we requested the generation of theoretically possible, but unlikely compounds, such as AuO_2. With gold in a highly unlikely $+4$ oxidation state, AuO_2 is not expected to be formed under most conditions. However, the model was able to imagine what the structure of such a

compound might be (when the space group is provided). While TiO_2 has cell parameters a=4.594Å, c=2.959Å, the generated rutile gold variant has a=4.838Å c=3.429Å, reflecting the increased volume occupied by the larger gold atoms (Figure 5.3d).

Spinels

Spinels are a group of ternary compounds with general formula AB_2X_4. The most common combination of elements in spinels is one where A is a cation in the $+2$ oxidation state, B is a cation in the $+3$ oxidation state, and X, normally a chalcogen, is a -2 anion. Spinels form cubic close-packed structures, with eight tetrahedral, and four octahedral sites, normally in the $Fd\bar{3}m$ space group.

To explore the model's ability to generate unseen spinels, we selected the thiospinel Sm_2BS_4, which was absent from both the training and test sets. The model was able to generate the expected spinel structure when the cell composition and space group were provided (Figure 5.3e). During training, the model encountered a number of different oxy-, thio-, and selenospinels, and this likely contributed to its ability to generate this compound.

Elpasolites

Elpasolites are quaternary compounds with the general formula ABC_2X_6. The A and C species are typically alkali metal cations in the $+1$ oxidation state, B is usually a transition metal cation in the $+3$ oxidation state, and X is a halogen anion. The elpasolites are often referred to as "double perovskites", since their structures are related to perovskites by the doubling of their unit cell dimensions, and the replacement of the M^{2+} cation with alternating M^{+} and M^{3+} cations. Elpasolites crystallize in the $Fm\bar{3}m$ space group, and are the most common quaternary crystal system reported in the Inorganic Crystal Structure Database (ICSD) [307]. We wondered if the CrystaLLM model could generate elpasolites not seen during training.

We selected an elpasolite from the held-out test, that was not seen in training: the fluoride KRb_2TiF_6. The model was able to generate the correct elpasolite structure when the cell composition and space group was provided (Figure 5.3f).

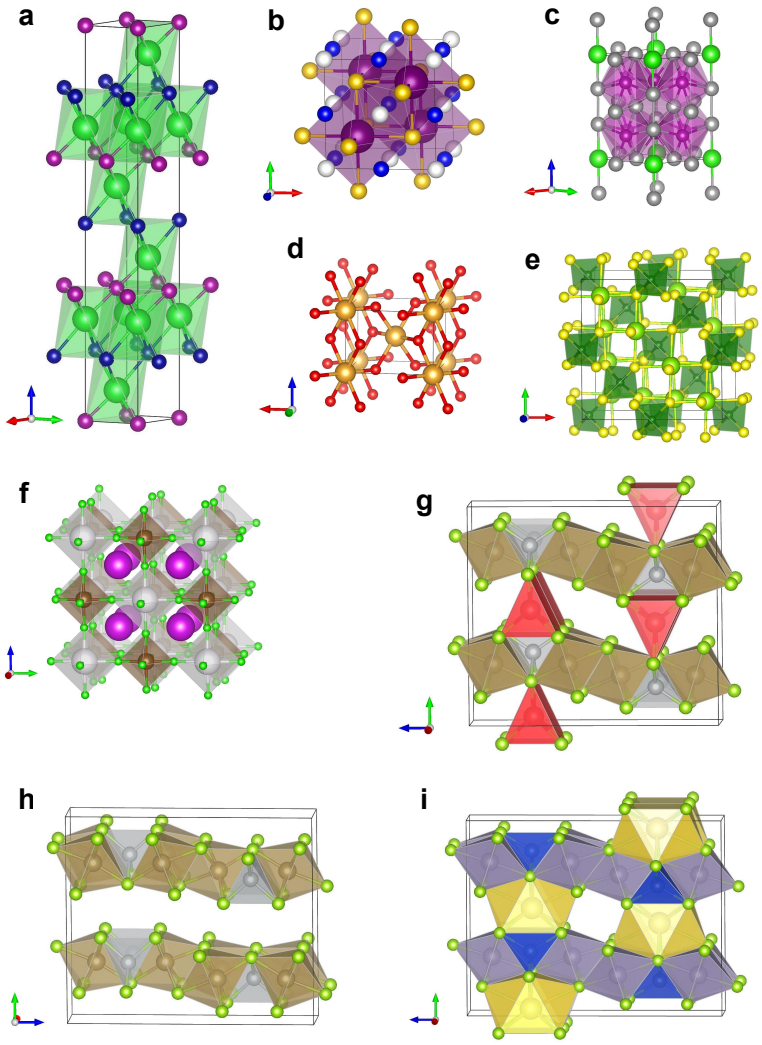

Figure 5.3: The generated structures of various inorganic compounds. **a** Ba_2MnCr. Cell parameters: a, b: 3.778 Å, c: 27.503 Å, α, β: 90.0°, γ: 120.0°. Color scheme: Ba: green, Mn: purple, Cr: blue. **b** CsCuTePt. Cell parameters: a, b, c: 7.153 Å, α, β, γ: 90.0°. Color scheme: Cs: purple, Cu: blue, Te: gold, Pt: white. **c** $YbMn_6Sn_6$. Cell parameters: a, b: 5.488 Å, c: 8.832 Å, α, β: 90.0°, γ: 120.0°. $ZrMn_6Sn_6$, in the training set, possessed the same structure, but with the following cell parameters: a, b: 5.364 Å, c: 8.933 Å, α, β: 90.0°, γ: 120.0°. Color scheme: Yb: green, Mn: magenta, Sn: grey. **d** AuO_2. Cell parameters: a, b: 4.838 Å, c: 3.429 Å, α, β, γ: 90.0°. Color scheme: Au: yellow, O: red. **e** Sm_2BS_4. Cell parameters: a, b, c: 10.884 Å, α, β, γ: 90.0°. Color scheme: Sm: light green, B: green, S: yellow. **f** KRb_2TiF_6. Cell parameters: a, b, c: 8.688 Å, α, β, γ: 90.0°. Color scheme: K: white, Rb: purple, Ti: brown, F: green. **g** $LiTa_2NiSe_5$ (a: 3.517 Å, b: 13.362 Å, c: 15.156 Å, Z=4), which resembles the recently reported structure in [308]. **h** Ta_2NiSe_5, seen in training. **i** $NaSn_2CuSe_5$, seen in training.

Pyrochlores

The general formula for the pyrochlores is $A_2B_2O_7$, where A, a trivalent cation, and B, a tetravalent cation, are either rare-earths or transition metals (other oxidation states, e.g.

combining monovalent and pentavalent cations, are also possible, but we focus here on the trivalent/tetravalent pyrochlores). Pyrochlores crystallize in the $Fd\bar{3}m$ space group (Z=8). There are many combinations of A and B that are possible for this structure, by using lanthanide ions, actinide ions, and Y(III) for the A species, and various transition metal ions, as well as Ti(IV), Zr(IV), and Hf(IV) for the B species. We investigated whether CrystaLLM could generate valid pyrochlore structures for any unseen combinations, and whether it could estimate reasonable cell parameters in line with the trends observed for the pyrochlore series, as the cell parameters are expected to be correlated with the ionic radii of the A and B cations.

We created a space of pyrochlores consisting of 144 compounds by producing different combinations of A and B species. Of these, 54 were seen in training. We selected 10 compounds from among the 90 not seen in training, and attempted 3 generations with the model, for each. The cell composition and space group were included in the prompt. All generations resulted in valid pyrochlore structures (Table 5.7).

Table 5.7: Values of mean generated cell length for the selected pyrochlores not seen in training, over 3 generation attempts.

Formula	**Cell Length** (Å)
$Ce_2Hf_2O_7$	10.75 ± 0.07
$Ce_2Mn_2O_7$	10.50 ± 0.22
$Ce_2V_2O_7$	10.53 ± 0.09
$La_2Mn_2O_7$	10.21 ± 0.07
$La_2V_2O_7$	10.48 ± 0.06
$Lu_2Hf_2O_7$	10.30 ± 0.08
$Lu_2Zr_2O_7$	10.45 ± 0.12
$Pr_2Mn_2O_7$	10.40 ± 0.08
$Pr_2V_2O_7$	10.51 ± 0.06
$Pr_2Hf_2O_7$	10.80 ± 0.06

We subsequently performed DFT relaxation calculations on the first generated structure for each of the 10 compounds. One case, $Ce_2V_2O_7$, posed challenges in calculation under the generalized gradient approximation and was thus excluded from further analysis. The DFT-derived value of the cell parameter for each of the remaining compounds is plotted against the mean value generated by CrystaLLM in Figure 5.4. A good agreement exists between the DFT-derived and generated cell lengths, with an R^2 of 0.62 and MAE of 0.08 Å being exhibited. This example illustrates CrystaLLM's capability to accurately estimate cell parameters of compounds not seen in training with any structure.

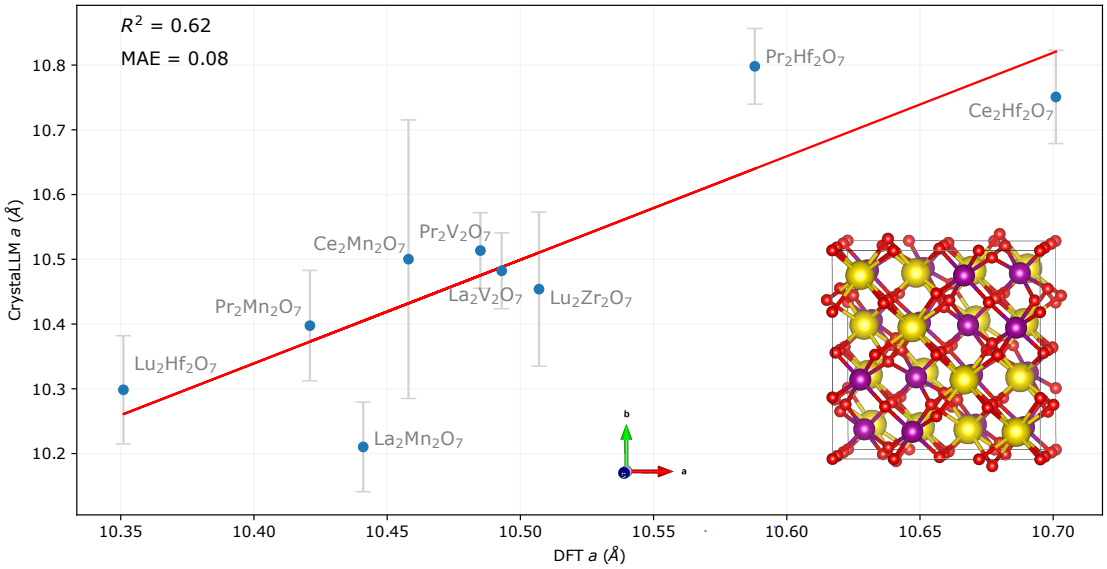

Figure 5.4: The generated vs. DFT-derived value of the cell parameter a for selected pyrochlores not in the training dataset. The error bars represent the $\pm$ standard deviation of the value of the a cell parameter for the three generation attempts (all of which resulted in the pyrochlore structure), while the y-coordinate of the points represents the mean value of the cell parameter across the three attempts. The inset represents the structure of the generated pyrochlore $Pr_2Mn_2O_7$, with cell parameters a, b, c: 10.34 Å, α, β, γ: 90.0°. Color scheme: Pr = yellow, Mn = purple, O = red.

Problematic Cases

While the model seems capable of generating structures for many different classes of inorganic crystals, it does nonetheless have difficulty in certain cases. All of the cases appear to involve systems that are rare, and under-represented in the training dataset, or missing from the training set altogether. More precisely, we define a *template* as a unique combination of the reduced composition ratio, the space group, and Z. For example, the combination of the reduced composition ratio 1:1:3:4, space group *Cmcm*, and $Z = 4$, represents a unique template. There are 25,921 unique templates in the dataset.

The problematic cases in the challenge set are largely represented by unseen templates, and templates for which there are few examples. For example, validation rates were low for $Mg_7Pt_4Ge_4$, the structure of which was reported recently to exist in the $P6_3mc$ space group (Z=2) [309]. In this case, there were only 38 examples of 7:4:4 systems in the training dataset, none contained Mg or Pt, and none were in the $P6_3mc$ space group.

The small version of the model also seems to struggle with generating phosphates, sulfates, carbonates, and organic-inorganic hybrid structures. Examples include carbonate hydroxide minerals, such as $Co_2CO_3(OH)_2$ [310] and $Cu_2CO_3(OH)_2$ (malachite). While present in the dataset, they belong to a group of analogous structures for which there are only a handful of examples. While both the small and large versions of the model can generate $Mn_4(PO_4)_3$, they generally fail to generate a valid structure for $Ca_5(PO_4)_3(OH)$ (hydroxyapatite). A common theme is the appearance of multiple oxyanions, which can give rise to more complex arrangements of atoms, for which the model may not have seen enough examples. In contrast, the model can generate compounds of the perovskite class reliably. However, over 5,000

examples of the ABX_3 (X=O,F) system in the $Pm\bar{3}m$ space group were seen in training. Finally, structures represented by CIF files with a relatively large number of tokens also pose challenges for the models.

Future versions of the model will consider strategies for addressing these occurrences of class imbalance.

5.3.4 Heuristic Search for Low-Energy Structures

The examples generated in the previous section were produced through top-k random sampling of the model. Essentially, as the CIF file is generated, each subsequent token is sampled randomly from amongst the top k tokens, according to their probabilities. (See Supplementary Note 2.4 in Appendix C for a detailed description of top-k sampling.) However, random sampling may not necessarily result in the most desirable sequence, and consequently, there are more strategic approaches for constructing sequences that incorporate the probability distributions produced by the model, along with additional heuristics. An example of a heuristic search is Beam Search [311], which is commonly used in natural language contexts to improve the quality of generated sequences. Another popular heuristic search algorithm is MCTS, which has traditionally been used in the context of planning and games, but has recently also been used to increase the quality of generated natural language, through incorporation with LLMs [312].

Here, we employ the MCTS algorithm, informed by CrystaLLM, to generate a collection of sequences, which is expected to progressively yield sequences of increasingly higher quality as the search advances. In this implementation, each node in the tree represents a cumulative context of tokens. The algorithm operates through a series of steps, including selection, expansion, rollout, evaluation, and backpropagation. The search tree is constructed iteratively, as the search proceeds (Figure 5.5). In the selection phase, nodes are chosen using the PUCT algorithm (Predictor-Upper Confidence bound applied to Trees) [313, 314], which is a principled means of obtaining a balance between exploring untried nodes, and exploiting promising nodes. The expansion involves adding child nodes based on predicted probabilities. During the rollout step, the CrystaLLM model is prompted with token sequences until a terminating condition is met, leading to the evaluation of the completed sequence. Evaluation is conducted using the ALIGNN (Atomistic Line Graph Neural Network) model of formation energy per atom [315], while the backpropagation step accumulates outcomes in the tree nodes, scoring each based on the quality of the generated structure. (See Supplementary Note 4 in Appendix C for a more detailed description of the algorithm.) The objective is to produce structures with lower formation energy per atom, E_f, and the incorporation of the ALIGNN model allows for a fast and sufficiently accurate estimate of the target property.

When compared to random sampling, MCTS improves the overall validity rate for a compound, and also generally produces lower energy structures. To evaluate the MCTS decoding procedure, we took the 20 most problematic cases of the challenge set where the validity rate was greater than 0, and performed 1,000 generation attempts using random top-k sampling, and 1,000 iterations of MCTS. The results are presented in Table 5.8.

Table 5.8: Results of MCTS decoding for the 20 most problematic cases of the challenge set. The percentages represent the fraction of cases with the corresponding improvement after using MCTS decoding, when compared to random sampling. The first row represents the percentage of cases where the validity rate improved. The second row represents the percentage of cases where the minimum E_{f} obtained was improved. The third row represents the percentage of cases where the mean E_{f} was improved.

	No Space Group	**With Space Group**
Validity Rate Improvement	95.0%	60.0%
Minimum E_{f} Improvement	85.0%	65.0%
Mean E_{f} Improvement	70.0%	65.0%

When no space group is provided in the prompt, the validity rate improves in 95% of the cases, and the minimum E_{f} attained improves in 85% of cases. (See Supplementary Tables 6 and 7 in Appendix C for more detailed results.) In some cases, the validity rate increases as the search proceeds when using MCTS (see Supplementary Figure 6 in Appendix C).

To further test the performance of MCTS, we applied the procedure to 102 novel compounds generated unconditionally by CrystaLLM (see the following section "Generating Novel Materials" for details of the unconditional generation). On these materials, we performed MCTS decoding with 1,000 iterations each, using ALIGNN to provide feedback. After MCTS, the ALIGNN energy decreased (or remained constant) for all the compositions, with an average energy change of -153 $\pm$ 15 meV/atom (compared to the structures generated without MCTS). The mean E_{hull} for the 102 structures, as calculated by DFT, improved by -56 $\pm$ 15 meV/atom on average, to 0.34 eV/atom; 22 of those structures were within 0.1 eV/atom of the hull. Further demonstration of the statistical significance of ALIGNN-based MCTS, and details of the results, are provided in Supplementary Note 6 in Appendix C. Future improvements of the energy estimators will increase the effectiveness of the MCTS approach.

5.3.5 Generating Novel Materials

The discovery of novel and stable compounds can expand the capabilities of materials science. To understand the potential of using CrystaLLM for generating novel and feasible crystalline solids, we used the large model trained on the 2.3M-structure dataset to generate 1,000 structures unconditionally, and assessed the stability of the novel compounds among them, using DFT. Of the 1,000 generated CIF files, 900 were valid, and 891 represented structurally distinct (i.e. unique) materials. There were 102 structures which were novel when compared to the training dataset (established using structure matching). We performed DFT relaxation of the 102 novel structures, and compared the energy of each structure with the convex hull as given by the Materials Project. The mean E_{hull} of the 102 novel structures was 0.40 eV/atom. Notably, we found that 20 structures were within 0.1 eV/atom of the hull, including 3 with $E_{\mathrm{hull}} = 0.00$ eV/atom. Figure 5.6 depicts the 4 most stable of the novel compounds. (See Supplementary Table 9 in Appendix C for comprehensive results for the 20 most stable compounds.)

Inspection of the novel materials revealed that the model generated a mix of ionic, semi-ionic and metallic compounds. The compounds with lower energy above the hull tended to be ionic and sem-ionic in nature. This could be due to the model being better at learning the coordination rules of ionic and semi-ionic compounds, as they are typically more defined and stricter than those for metallic compounds. For example, in most oxides, Fe will be coordinated tetrahedrally or octahedrally to oxygen. For metallic compounds, it is less defined, *a priori*,

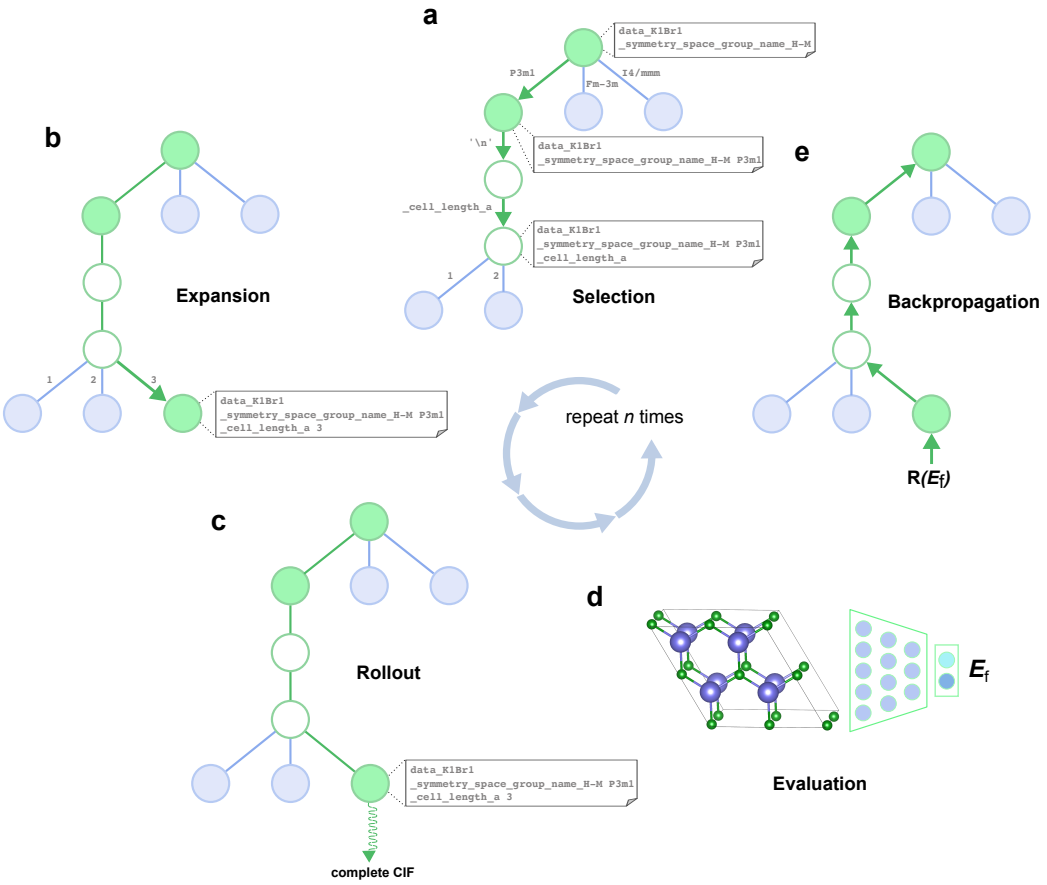

Figure 5.5: Schematic depiction of the Monte Carlo Tree Search decoding procedure. CIF files are generated as a tree is iteratively constructed, with each iteration guiding the generation of subsequent structures towards more desirable parameters (e.g. lower formation energy per atom). The nodes in the tree represent the cumulative contents of a CIF file at various points. **a** The Selection step involves descending the tree by choosing the most promising node at each level, using a variant of the PUCT algorithm. **b** During Expansion, an unexplored child node is randomly selected and added to the tree. If a node has only one highly probable child (represented as empty nodes), the child node bypasses the Rollout step. **c** The Rollout step involves prompting the model with the contents of the selected node, and sampling from the model until a terminal condition is met, so as to obtain a complete CIF file and an estimate of the value of a node. **d** The generated structure is validated and scored, incorporating the prediction of the structure's formation energy per atom, as given by a pre-trained neural network. **e** Finally, the score is backpropagated through the selected nodes, which store the accumulated results of each iteration. The resulting generated CIF file, if valid, is returned.

what the coordination patterns should be. In fact, many metallic compounds only stabilize due to disorder thanks to the configurational entropy (effects which are not considered here). The model has therefore a better chance of generating a stable ionic material than a stable ordered metallic compound.

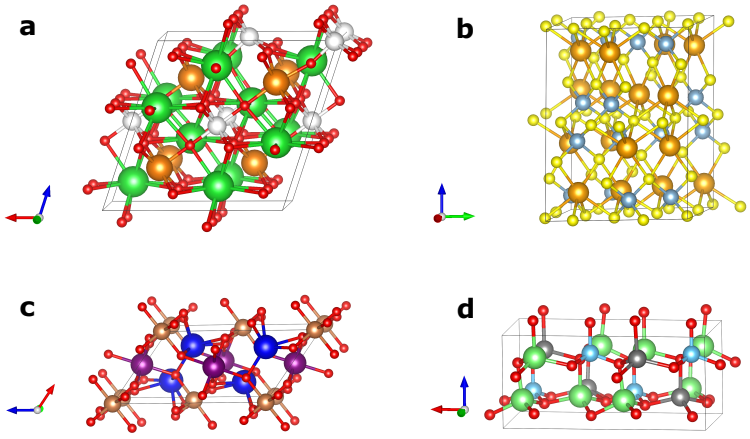

Figure 5.6: The four lowest-energy novel structures generated unconditionally by the large model. **a** $Ba_4Na_2Ir_2O_{11}$ Z=2, Cm. Cell parameters: a: 10.308 Å, b: 5.995 Å, c: 10.269 Å, α, γ: 90.0°, β: 108.5°. Color scheme: Ba: green, Na: orange, Ir: white, O: red. E_{hull}=0.00 eV/atom. **b** $NaAlS_2$ Z=16, $P2_1$. Cell parameters: a: 10.233, b: 10.277 Å, c: 13.703 Å, α, γ: 90.0°, β: 100.9°. Color scheme: Na: orange, Al: grey, S: yellow. E_{hull}=0.00 eV/atom. **c** Ca_2YSbO_6 Z=2, $P2_1/c$. Cell parameters: a: 5.651 Å, b: 5.853 Å, c: 9.850 Å, α, γ: 90.0°, β: 125.0°. Color scheme: Ca: blue, Y: purple, Sb: bronze, O: red. E_{hull}=0.00 eV/atom. **d** Li_2FeSiO_4 Z=4, $Pna2_1$. Cell parameters: a: 10.988 Å, b: 6.278 Å, c: 5.026 Å, α, β, γ: 90.0°. Color Scheme: Si: light blue, Fe: dark grey, Li: light green, O: red. E_{hull}=0.02 eV/atom.

5.3.6 Beyond Element Substitution

Although CrystaLLM appears to be very effective at finding appropriate template systems for a given cell composition, and making the necessary adjustments of cell parameters to substitute different atoms, it appears capable of going further, synthesizing information from different template systems. An example is the selenide $LiTa_2NiSe_5$, which is obtained by lithium intercalation into Ta_2NiSe_5 [308].

The compound $LiTa_2NiSe_5$ was not present in the training set; however, the layered material Ta_2NiSe_5 was (Figure 5.3g,h). As $LiTa_2NiSe_5$ was included in the challenge set, we performed 100 generation attempts with the model. While the model was not able to recover the lowest energy structure reported, it did produce structures with close resemblance to low-energy polymorphs. Upon closer examination of the dataset, we found that $NaSn_2CuSe_5$ was present (Figure 5.3i), which likely provided some precedent for the intercalation of atoms between layered structures. It thus appears that the model is capable of integrating information from different template systems to form new structural predictions.

5.3.7 The CrystaLLM.com Web Application

To allow for easy and open access to the CrystaLLM model, we make it available through a web application, published at https://crystallm.com. The application allows users to enter in a reduced formula, and optionally a value for Z and the desired space group. The option to select the model size is also provided. The request is sent to the model, and the resulting structure (or the CIF contents, if the structure is invalid) is presented to the user. By making the model easily accessible, we hope to contribute a potentially useful tool to the materials structure research community. We also hope to receive feedback from users that may help improve future versions of the model.

5.4 Conclusions

Here, we have shown that LLMs of the CIF format are able to generate inorganic crystal structures for a variety of known classes. Indeed, the model is able to produce valid and sensible arrangements of atoms in 3-dimensional space by generating *xyz* coordinates digit-by-digit. The model also seems to have captured the relationship between space group symbols and the symmetries inherent in the structures it generates.

We chose to build a language model of the CIF format (instead of a simplified format, for example, which might include a minimal vocabulary) for several reasons. First, the CIF format is not particularly verbose. The model learns the grammatical structure of the format fairly quickly. We can thus avoid having to devise an intermediate format that requires inter-conversion between more common formats, which could also be error prone. Second, we believe that having the model learn to generate the more redundant parts of the CIF format, such as the cell volume, and Z, which are inferable from prior inputs, helps the model to perform better overall.

A number of approaches for crystal structure generation have been reported [316–319]. These approaches generally require the existence of pre-defined structural templates, and are followed by the procedural or machine learning-assisted substitution of atoms and adjustment of cell parameters, under the constraint of a specified space group. These types of approaches can also be enhanced to increase the structural diversity of generated materials, by allowing partial substitutions and adjusting substitution probabilities [320]. Conversely, CrystaLLM automatically selects the templates which can be applied to a given composition, utilizing the implicit templates it has absorbed through autoregressive training. Moreover, the model can automatically adjust cell parameters to accommodate the atoms in the unit cell. It can also produce structures based on templates it has not explicitly encountered in training, borrowing from its internalized concepts of chemical structure. In comparison with recently reported diffusion-based ML methods for crystal generation (CDVAE [266] and DiffCSP [294]), not only does CrystaLLM outperform them on established benchmarks in several aspects, but it also offers additional advantages in terms of flexibility (e.g. in using symmetry as input) and the potential for fine-tuning.

While the CrystaLLM model can generate sensible structures, this does not by itself make it suitable, as is, for CSP. Just as natural language LLMs, such as GPT-3 and -4, are not suitable chatbots without further fine-tuning and alignment, the CrystaLLM model will also need to be fine-tuned for more advanced tasks. Fine-tuning involves an additional and separate training step, where the model's parameters are adjusted in the context of a different task. This may also involve altering the model's output layer, such as to make it suitable for a regression task. Models can be fine-tuned using a variety of techniques, but supervised learning and reinforcement learning [321] are most common. One might use reinforcement learning, for

example, when a task is not clearly defined as a supervised learning problem. When fine-tuning natural language LLMs for chatbot applications, it is common to use Reinforcement Learning from Human Feedback (RLHF) [322, 323]. With RLHF, the idea is to gather data from human annotators to be used to train a reward model, which scores generated text according to its desirability. The reward model is then used as part of a reinforcement learning-based tuning of the LLM. In CSP, one would like to produce ground-state structures (for some given physical conditions). One could thus imagine an analogous procedure where CrystaLLM is fine-tuned for the goal of generating low-energy structures, via feedback from an external evaluator of the generated structure's energy, resulting in *Reinforcement Learning from Thermodynamic Feedback*. This procedure would also require a reward model, and such a model should ideally provide a timely estimate of a structure's energy. This excludes time-consuming approaches such as DFT. A viable approach could make use of a separate machine learning-based model of formation energy, such as one based on ALIGNN. Indeed, neural network potentials have been used to accelerate the prediction of crystal structures, and the identification of potentially stable materials [324, 325].

There are several limitations with the current approach. First, none of the structures of the dataset have site-occupancy disorder (fractional site occupancies). Therefore, CrystaLLM cannot generate disordered structures, and may not successfully generate structures for combinations of cell composition and space group that imply a disordered structure. An example is $K_2NaTiOF_5$, which is reported to be an elpasolite, in the $Fm\bar{3}m$ space group (Z=4), with F and O species sharing the same crystal site [326]. Another limitation is that the CIF files of the dataset were not all created using the same level of theory. The training set is derived from a combination of DFT sources using different settings, functionals, etc., which may make it difficult for the model, in some instances, to learn a consistent relationship between cell composition and detailed structure [327].

Nevertheless, we believe that CrystaLLM will be a useful tool for crystal structure generation, which is quickly becoming a critical step in large scale materials discovery [320, 328], and materials informatics. We plan to explore fine-tuning the model for physical property prediction tasks, such as the prediction of lattice thermal conductivity, where experimental data is relatively scarce [215]. The architecture of the model allows it to be fine-tuned for either composition-based or structure-based prediction tasks. This implies that CrystaLLM may be the basis for a general-purpose materials informatics model, which can be used for generative tasks, and fine-tuned for property prediction tasks that require either composition or structure. If the model is able to transfer what it has learned about the world of atoms to these various predictive problems, it may prove to be a quite flexible tool relevant to many aspects of materials chemistry.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis presents the development of novel ML algorithms designed to address various aspects of the materials discovery process. In particular, the focus has been on the discovery of new and promising thermoelectric materials. The algorithms presented are built upon current and emerging deep learning methodologies. They draw extensively from advancements in NLP, particularly the Transformer architecture—a result that is notable given the thesis's focus on a problem in materials chemistry. This synthesis of two seemingly disparate fields, namely, NLP and materials chemistry, demonstrates the fruitfulness of interdisciplinary research.

First, I designed and evaluated an approach for learning distributed representations of atoms and materials, named SkipAtom, which can be used to produce effective representations of materials from their compositions alone, for use with deep neural network models. I showed that SkipAtom outperforms existing atom vector representations like Atom2Vec and Mat2Vec on materials prediction tasks, including formation energy prediction. Through benchmarking on a variety of regression and classification tasks, I demonstrated that SkipAtom-derived compound representations, which rely solely on composition, achieve competitive results compared to models that incorporate structure. This suggests that SkipAtom can be an effective tool for rapid, high-throughput materials screening where structural data may be unavailable.

Second, I developed a Transformer-based model, named CraTENet, for predicting electronic transport properties of thermoelectric materials, specifically the Seebeck coefficient, the electrical conductivity, and the power factor. I demonstrated that CraTENet can generate useful predictions across various temperatures, doping levels, and doping types, providing an efficient and practical alternative to traditional machine learning approaches. Its attention-based architecture allows for greater interpretability by surfacing important relationships between atomic elements within a material's composition, and the incorporation of the band gap as an optional input, when available, improves prediction quality significantly.

Finally, I created CrystaLLM, a crystal structure generation tool based on autoregressive large language modeling. A fast model for CSP is needed in any workflow that attempts to identify promising materials within unexplored chemical spaces. Here, I showed that CrystaLLM can generate valid crystal structures for a wide variety of materials classes, including those not seen during training. The model demonstrated the ability to generalize to combinations of unseen compositions and space groups, generating structures that are physically plausible and chemically sensible. It also proved useful in the context of unconditioned generation, proposing stable materials unseen in the training dataset. When integrated with the MCTS decoding procedure, the quality of generated materials improves noticeably.

To illustrate how these tools can be integrated into a workflow for discovering novel

thermoelectric materials, consider the flowchart in Figure 6.1.

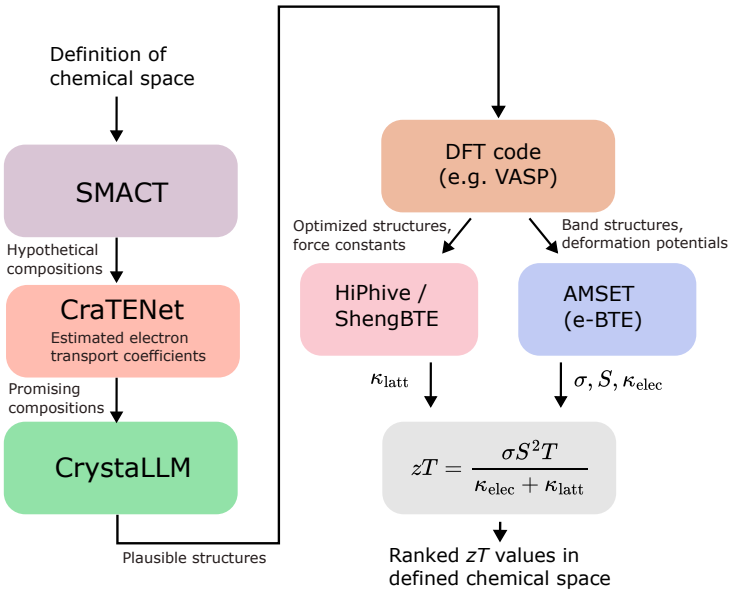

Figure 6.1: A flowchart depicting a potential thermoelectric materials discovery pipeline and workflow, which incorporates the tools described in this thesis.

The CraTENet model, incorporating pre-trained SkipAtom representations, can be used to filter a large space of chemical compositions generated by SMACT, by producing predictions for thermoelectric transport properties from composition alone. The compositions with the most promising combination of predicted properties can then be given to the CrystaLLM model, which will propose structures for those compositions. Finally, the proposed structures can be subjected to a DFT pipeline, involving a more rigorous investigation of the materials and their properties. While this final step can be the most time-consuming and labor-intensive, the tools presented here reduce the search space to only the most promising candidates.

The ML-based tools developed in this thesis are also of wider interest, beyond thermoelectric applications. The pre-trained SkipAtom representations, for instance, can be used in any context where atoms and materials are involved. The SkipAtom software library additionally allows for customized training on any set materials, leading to atomic representations which reflect the nature of the dataset. Similarly, the CrystaLLM model is not constrained to any specific class of materials, and can be used for the crystal structure generation of materials of any kind. Like SkipAtom, the CrystaLLM software library allows for training a CrystaLLM model on a custom dataset, further expanding the scenarios in which the model can be used.

6.2 Future work

While the algorithms introduced in this thesis can be used to augment and enhance the materials discovery process, there are a number of ways in which they can be further investigated, expanded and improved.

Given that pre-trained SkipAtom embeddings are often competitive on tasks with approaches that include structure information, it could be worthwhile to attempt screening a large, targeted portion of materials space for a specific property using composition alone. This would be computationally cheaper than explicitly requiring structure information. Additionally, it would be interesting to investigate the performance of existing graph neural network

models, such as MEGNet [112], when they begin with pre-trained SkipAtom vectors as the atomic representations. The pre-trained SkipAtom embeddings could accelerate training, and possibly improve performance. Finally, it should be possible to learn atom vectors that take into account oxidation states, using the SkipAtom approach. Indeed, such an investigation has recently been reported [329]. A collection of oxidation state-based atom representations would be a unique tool for materials informaticists.

The accuracy of the predictions of thermoelectric behavior from composition, as produced by CraTENet, could be improved through a number of different approaches. The single most important development would be the creation of a more accurate *ab initio* database of electron transport properties (given the problem that the Ricci database used in this study is based on the GGA flavor of DFT, which is problematic for the prediction of electron transport). Two further extensions would make possible the prediction, at the ML level, of the full thermoelectric figure of merit, zT: 1) the ability to estimate sensible relaxation times for electron transport, and 2) the ability to predict the phonon-related (lattice) thermal conductivity. For the former, training data could be obtained from either *ab initio* calculations involving electron-lattice interactions (for example, in deformation potential approximations, as done in the AMSET code), or by contrasting electrical conductivities per unit of relaxation times predicted at DFT level with experimental conductivities. In terms of the prediction of lattice thermal conductivities, I anticipate that structure should be part of the input, but that could be achieved by integrating the CrystaLLM tool in the process.

The crystal structure generation and prediction model, CrystaLLM, could be improved in a number of ways. Perhaps the most obvious improvement would be to incorporate all of the recent LLM architectural developments since GPT-2, upon which CrystaLLM is currently based. These include techniques for improving positional representations, such as RoPE (Rotary Positional Embeddings) [330]. Other techniques, such as KV (key-value) caching, and multi-query decoding [331], could be used to accelerate inference, making the model even faster to use in practice. It would also be interesting to experiment with simplified versions of the CIF syntax that the model is trained to produced, such as using a lower decimal precision. Additionally, sampling to achieve a more balanced representation of the structural templates the model sees in pre-training is another avenue of investigation that is expected to be fruitful: LLMs are reliant on the quality of data seen in training, thus it is likely that large improvements could be made to the model's performance by simply optimizing the quality of the dataset. Another desired extension would be the ability to predict fractional occupancy of lattice sites, essentially extending the model to site-disordered materials. This is important, because most thermoelectric and other functional materials are solid solutions with some degree of site disorder. Certainly, this introduces complexity in the further *ab initio* evaluation of properties, because DFT calculations are normally performed in fully ordered crystal cells. (Methods like the virtual crystal approximation, VCA, which define fractional occupancies of lattice sites in DFT simulations, tend to perform poorly in the prediction of most properties.) However, this is a well-studied problem, and ensemble approaches can be used to calculate effective properties of solid solutions (e.g. [332]). Finally, it would be exciting to incorporate the most recent state-of-the-art estimators of formation energy into the MCTS decoding procedure. On the Matbench Discovery benchmark [333], which measures the ability of an ML model to predict solid-state thermodynamic stability, the best model currently attains an F1 score of 0.763, which is a large improvement over the ALIGNN model's score of 0.567.

In summary, while there are many directions in which the models I present here could evolve, I believe that this work has made a substantial contribution to the rapidly developing field of materials informatics. I look forward to new developments in the field, and hope to further contribute in the future.

References

[1] H. J. Goldsmid, *Introduction to Thermoelectricity*, 1st ed., ser. Springer Series in Materials Science №121. Springer-Verlag Berlin Heidelberg, 2010.

[2] M. Markov, X. Hu, H.-C. Liu, N. Liu, S. J. Poon, K. Esfarjani, and M. Zebarjadi, "Semi-metals as potential thermoelectric materials," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[3] I. T. Witting, T. C. Chasapis, F. Ricci, M. Peters, N. A. Heinz, G. Hautier, and G. J. Snyder, "The Thermoelectric Properties of Bismuth Telluride," *Advanced Electronic Materials*, vol. 5, no. 6, p. 1800904, 2019.

[4] G. Joshi, H. Lee, Y. Lan, X. Wang, G. Zhu, D. Wang, R. W. Gould, D. C. Cuff, M. Y. Tang, M. S. Dresselhaus, G. Chen, and Z. Ren, "Enhanced Thermoelectric Figure-of-Merit in Nanostructured p-type Silicon Germanium Bulk Alloys," *Nano Letters*, vol. 8, no. 12, pp. 4670–4674, 2008, pMID: 18973391.

[5] E. K. Lee, L. Yin, Y. Lee, J. W. Lee, S. J. Lee, J. Lee, S. N. Cha, D. Whang, G. S. Hwang, K. Hippalgaonkar, A. Majumdar, C. Yu, B. L. Choi, J. M. Kim, and K. Kim, "Large Thermoelectric Figure-of-Merits from SiGe Nanowires by Simultaneously Measuring Electrical and Thermal Transport Properties," *Nano Letters*, vol. 12, no. 6, pp. 2918–2923, 2012, pMID: 22548377.

[6] R. R. Furlong and E. J. Wahlquist, "US space missions using radioisotope power systems," *Nuclear News*, vol. 42, pp. 26–35, 1999.

[7] G. K. Madsen and D. J. Singh, "BoltzTraP. A code for calculating band-structure dependent quantities," *Computer Physics Communications*, vol. 175, no. 1, pp. 67–71, 2006.

[8] W. Li, J. Carrete, N. A. Katcho, and N. Mingo, "ShengBTE: a solver of the Boltzmann transport equation for phonons," *Computer Physics Communications*, vol. 185, no. 6, p. 1747–1758, 2014.

[9] P. Gorai, V. Stevanović, and E. S. Toberer, "Computationally guided discovery of thermoelectric materials," *Nature Reviews Materials*, vol. 2, no. 9, pp. 1–16, 2017.

[10] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, no. 7715, pp. 547–555, 2018.

[11] S. Wang, Z. Wang, W. Setyawan, N. Mingo, and S. Curtarolo, "Assessing the Thermoelectric Properties of Sintered Compounds via High-Throughput Ab-Initio Calculations," *Physical Review X*, vol. 1, no. 2, p. 021012, 2011.

[12] https://journals.aps.org/prx/supplemental/10.1103/PhysRevX.1.021012/supplement.pdf.

[13] M. W. Gaultois, T. D. Sparks, C. K. Borg, R. Seshadri, W. D. Bonificio, and D. R. Clarke, "Data-driven review of thermoelectric materials: Performance and resource considerations," *Chemistry of Materials*, vol. 25, no. 15, pp. 2911–2920, 2013.

[14] https://citrination.com/datasets/150557/show_files.

[15] J. Carrete, W. Li, N. Mingo, S. Wang, and S. Curtarolo, "Finding Unprecedentedly Low-Thermal-Conductivity Half-Heusler Semiconductors via High-Throughput Materials Modeling," *Physical Review X*, vol. 4, no. 1, p. 011019, 2014.

[16] https://journals.aps.org/prx/supplemental/10.1103/PhysRevX.4.011019 (Supplementary Material).

[17] P. Gorai, D. Gao, B. Ortiz, S. Miller, S. A. Barnett, T. Mason, Q. Lv, V. Stevanović, and E. S. Toberer, "TE Design Lab: A virtual laboratory for thermoelectric material design," *Computational Materials Science*, vol. 112, pp. 368–376, 2016.

[18] https://tedesignlab.org.

[19] F. Ricci, W. Chen, U. Aydemir, G. J. Snyder, G.-M. Rignanese, A. Jain, and G. Hautier, "An ab initio electronic transport database for inorganic materials," *Scientific Data*, vol. 4, no. 1, pp. 1–13, 2017.

[20] https://datadryad.org/stash/dataset/doi:10.5061/dryad.gn001.

[21] L. Xi, S. Pan, X. Li, Y. Xu, J. Ni, X. Sun, J. Yang, J. Luo, J. Xi, W. Zhu *et al.*, "Discovery of high-performance thermoelectric chalcogenides through reliable high-throughput material screening," *Journal of the American Chemical Society*, vol. 140, no. 34, pp. 10 785–10 793, 2018.

[22] https://pubs.acs.org/doi/10.1021/jacs.8b04704 (Supporting Information).

[23] L. Chen, H. Tran, R. Batra, C. Kim, and R. Ramprasad, "Machine learning models for the lattice thermal conductivity prediction of inorganic materials," *Computational Materials Science*, vol. 170, p. 109155, 2019.

[24] https://www.sciencedirect.com/science/article/pii/S0927025619304549 (Supplementary Data).

[25] Y. Katsura, M. Kumagai, T. Kodani, M. Kaneshige, Y. Ando, S. Gunji, Y. Imai, H. Ouchi, K. Tobita, K. Kimura *et al.*, "Data-driven analysis of electron relaxation times in PbTe-type thermoelectric materials," *Science and Technology of Advanced Materials*, vol. 20, no. 1, pp. 511–520, 2019.

[26] https://github.com/starrydata/starrydata_datasets.

[27] K. Choudhary, K. F. Garrity, and F. Tavazza, "Data-driven Discovery of 3D and 2D Thermoelectric Materials," *Journal of Physics: Condensed Matter*, vol. 32, no. 47, p. 475501, 2020.

[28] https://www.ctcms.nist.gov/~knc6/JVASP.html.

[29] P. Priya and N. Aluru, "Accelerated design and discovery of perovskites with high conductivity for energy applications through machine learning," *npj Computational Materials*, vol. 7, no. 1, pp. 1–12, 2021.

[30] https://figshare.com/s/10b18051e26fa4d4f18c.

[31] R. Jaafreh, Y. S. Kang, and K. Hamad, "Lattice Thermal Conductivity: An Accelerated Discovery Guided by Machine Learning," *ACS Applied Materials & Interfaces*, vol. 13, no. 48, pp. 57 204–57 213, 2021, pMID: 34806862.

[32] https://pubs.acs.org/doi/10.1021/acsami.1c17378 (Supporting Information).

[33] H. Miyazaki, T. Tamura, M. Mikami, K. Watanabe, N. Ide, O. M. Ozkendir, and Y. Nishino, "Machine learning based prediction of lattice thermal conductivity for half-Heusler compounds using atomic information," *Scientific Reports*, vol. 11, no. 1, pp. 1–8, 2021.

[34] https://www.nature.com/articles/s41598-021-92030-4#Sec7 (Supplementary Information).

[35] M. Yao, Y. Wang, X. Li, Y. Sheng, H. Huo, L. Xi, J. Yang, and W. Zhang, "Materials informatics platform with three dimensional structures, workflow and thermoelectric applications," *Scientific Data*, vol. 8, no. 1, pp. 1–10, 2021.

[36] http://www.mip3d.org/.

[37] R. Tranås, O. M. Løvvik, O. Tomic, and K. Berland, "Lattice thermal conductivity of half-Heuslers with density functional theory and machine learning: Enhancing predictivity by active sampling with principal component analysis," *Computational Materials Science*, vol. 202, p. 110938, 2022.

[38] https://www.sciencedirect.com/science/article/pii/S0927025621006376#tblA.2.

[39] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. H. Taylor, L. J. Nelson, G. L. Hart, S. Sanvito, M. Buongiorno-Nardelli *et al.*, "AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations," *Computational Materials Science*, vol. 58, pp. 227–235, 2012.

[40] The original publication describing the UCSB database states that it consists of over 18,000 data points. We independently verified that the database consists of 1,093 entries with 17 associated components (such as temperature, Seebeck coefficient, etc.), for a total of 18,581 data points. Moreover, 282 unique compositions are represented in the database at various temperatures. Each database entry represents a unique composition-temperature pair.

[41] A. Furmanchuk, J. E. Saal, J. W. Doak, G. B. Olson, A. Choudhary, and A. Agrawal, "Prediction of seebeck coefficient for compounds without restriction to fixed stoichiometry: A machine learning approach," *Journal of Computational Chemistry*, vol. 39, no. 4, pp. 191–202, 2018.

[42] M. Mukherjee, S. Satsangi, and A. K. Singh, "A Statistical Approach for the Rapid Prediction of Electron Relaxation Time Using Elemental Representatives," *Chemistry of Materials*, vol. 32, no. 15, pp. 6507–6514, 2020.

[43] M. W. Gaultois, A. O. Oliynyk, A. Mar, T. D. Sparks, G. J. Mulholland, and B. Meredig, "Perspective: Web-based machine learning models for real-time screening of thermoelectric materials properties," *APL Materials*, vol. 4, no. 5, p. 053213, 2016.

[44] http://www.mrl.ucsb.edu:8080/datamine/thermoelectrics.jsp.

[45] The original publication describing the Ricci *et al.* database states that the database consists of 48,000 compounds. We have independently confirmed that the database contains exactly 47,737 compounds, of which 36,628 represent unique compositions.

[46] K. Choudhary, K. F. Garrity, A. C. Reid, B. DeCost, A. J. Biacchi, A. R. H. Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone *et al.*, "The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design," *npj Computational Materials*, vol. 6, no. 1, pp. 1–13, 2020.

[47] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Physical Review Letters*, vol. 77, no. 18, p. 3865, 1996.

[48] J. Klimeš, D. R. Bowler, and A. Michaelides, "Chemical accuracy for the van der Waals density functional," *Journal of Physics: Condensed Matter*, vol. 22, no. 2, p. 022201, 2009.

[49] J. Xi, D. Wang, Y. Yi, and Z. Shuai, "Electron-phonon couplings and carrier mobility in graphynes sheet calculated using the Wannier-interpolation approach," *The Journal of Chemical Physics*, vol. 141, no. 3, p. 034704, 2014.

[50] X. Li, Z. Zhang, J. Xi, D. J. Singh, Y. Sheng, J. Yang, and W. Zhang, "TransOpt. A code to solve electrical transport properties of semiconductors in constant electron–phonon coupling approximation," *Computational Materials Science*, vol. 186, p. 110074, 2021.

[51] A. Togo, L. Chaput, and I. Tanaka, "Distributions of phonon lifetimes in Brillouin zones," *Physical Review B*, vol. 91, p. 094306, Mar 2015.

[52] O. Hellman and I. A. Abrikosov, "Temperature-dependent effective third-order interatomic force constants from first principles," *Physical Review B*, vol. 88, no. 14, p. 144301, 2013.

[53] J. Yan, P. Gorai, B. Ortiz, S. Miller, S. A. Barnett, T. Mason, V. Stevanović, and E. S. Toberer, "Material descriptors for predicting thermoelectric performance," *Energy & Environmental Science*, vol. 8, no. 3, pp. 983–994, 2015.

[54] E. S. Toberer, A. Zevalkink, and G. J. Snyder, "Phonon engineering through crystal chemistry," *Journal of Materials Chemistry*, vol. 21, no. 40, pp. 15 843–15 852, 2011.

[55] S. Miller, P. Gorai, B. Ortiz, A. Goyal, D. Gao, S. Barnett, T. Mason, G. Snyder, Q. Lv, V. Stevanović, and E. Toberer, "Capturing Anharmonicity in a Lattice Thermal Conductivity Model for High-Throughput Predictions," *Chemistry of Materials*, vol. 29, no. 6, pp. 2494–2501, 2017.

[56] A. Seko, A. Togo, H. Hayashi, K. Tsuda, L. Chaput, and I. Tanaka, "Prediction of Low-Thermal-Conductivity Compounds with First-Principles Anharmonic Lattice-Dynamics Calculations and Bayesian Optimization," *Physical Review Letters*, vol. 115, no. 20, 2015.

[57] R. Juneja, G. Yumnam, S. Satsangi, and A. Singh, "Coupling the High-Throughput Property Map to Machine Learning for Predicting Lattice Thermal Conductivity," *Chemistry of Materials*, vol. 31, no. 14, pp. 5145–5151, 2019.

[58] Y. Zhang and C. Ling, "A strategy to apply machine learning to small datasets in materials science," *npj Computational Materials*, vol. 4, no. 1, 2018.

[59] D. Zhang, A. Oliynyk, G. Duarte, A. Iyer, L. Ghadbeigi, S. Kauwe, T. Sparks, and A. Mar, "Not Just Par for the Course: 73 Quaternary Germanides RE4 M2 XGe4 (RE = La-Nd, Sm, Gd-Tm, Lu; M = Mn-Ni; X = Ag, Cd) and the Search for Intermetallics with Low Thermal Conductivity," *Inorganic Chemistry*, vol. 57, no. 22, pp. 14 249–14 259, 2018.

[60] D. Visaria and A. Jain, "Machine-learning-assisted space-transformation accelerates discovery of high thermal conductivity alloys," *Applied Physics Letters*, vol. 117, no. 20, 2020.

[61] R. Juneja and A. Singh, "Guided patchwork kriging to develop highly transferable thermal conductivity prediction models," *Journal of Physics: Materials*, vol. 3, p. 024006, 2020.

[62] ——, "Unraveling the role of bonding chemistry in connecting electronic and thermal transport by machine learning," *Journal of Materials Chemistry A*, vol. 8, no. 17, pp. 8716–8721, 2020.

[63] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009, vol. 501.

[64] M. Omini and A. Sparavigna, "An Iterative Approach to the Phonon Boltzmann Equation in the Theory of Thermal Conductivity," *Physica B*, vol. 212, no. 2, pp. 101–112, 1995.

[65] R. Kubo, "Statistical-Mechanical Theory of Irreversible Processes. I. General Theory and Simple Applications to Magnetic and Conduction Problems," *Journal of the Physical Society of Japan*, vol. 12, no. 6, pp. 570–586, 1957.

[66] M. S. Green, "Markoff Random Processes and the Statistical Mechanics of Time-Dependent Phenomena. II. Irreversible Processes in Fluids," *The Journal of Chemical Physics*, vol. 22, no. 3, pp. 398–413, 1954.

[67] P. Korotaev and A. Shapeev, "Lattice dynamics of $\mathrm{Yb}_x\mathrm{Co}_4\mathrm{Sb}_{12}$ skutterudite by machine-learning interatomic potentials: Effect of filler concentration and disorder," *Physical Review B*, vol. 102, p. 184305, Nov 2020.

[68] C. Verdi, F. Karsai, P. Liu, R. Jinnouchi, and G. Kresse, "Thermal transport and phase transitions of zirconia by on-the-fly machine-learned interatomic potentials," *npj Computational Materials*, vol. 7, no. 1, pp. 1–9, 2021.

[69] R. Li, Z. Liu, A. Rohskopf, K. Gordiz, A. Henry, E. Lee, and T. Luo, "A deep neural network interatomic potential for studying thermal conductivity of β-Ga_2O_3," *Applied Physics Letters*, vol. 117, no. 15, p. 152102, 2020.

[70] J. George, G. Hautier, A. P. Bartók, G. Csányi, and V. L. Deringer, "Combining phonon accuracy with high transferability in Gaussian approximation potential models," *The Journal of Chemical Physics*, vol. 153, no. 4, p. 044104, 2020.

[71] F. Zhou, W. Nielson, Y. Xia, and V. Ozoliņš, "Lattice Anharmonicity and Thermal Conductivity From Compressive Sensing of First-Principles Calculations," *Physical Review Letters*, vol. 113, p. 185501, 2014.

[72] F. Eriksson, E. Fransson, and P. Erhart, "The Hiphive Package for the Extraction of High-Order Force Constants by Machine Learning," *Advanced Theory And Simulations*, vol. 2, no. 5, p. 1800184, 2019.

[73] J. J. Plata, P. Nath, D. Usanmaz, J. Carrete, C. Toher, M. de Jong, M. D. Asta, M. Fornari, M. Buongiorno Nardelli, and S. Curtarolo, "An Efficient and Accurate Framework for Calculating Lattice Thermal Conductivity of Solids: AFLOW-AAPL Automatic Anharmonic Phonon Library," *npj Computational Materials*, vol. 3, no. 45, p. 45, 2017.

[74] F. Eriksson, E. Fransson, and P. Erhart, "Efficient construction of linear models in materials modeling and applications to force constant expansions," *npj Computational Materials*, vol. 6, p. 135, 2020.

[75] H. Yang, Y. Zhu, E. Dong, Y. Wu, J. Yang, and W. Zhang, "Dual adaptive sampling and machine learning interatomic potentials for modeling materials with chemical bond hierarchy," *Physical Review B*, vol. 104, p. 094310, 2021.

[76] Y. Xia, V. I. Hegde, K. Pal, X. Hua, D. Gaines, S. Patel, J. He, M. Aykol, and C. Wolverton, "High-Throughput Study of Lattice Thermal Conductivity in Binary Rocksalt and Zinc Blende Compounds Including Higher-Order Anharmonicity," *Physical Review X*, vol. 10, p. 041029, Nov 2020.

[77] J. J. Plata, V. Posligua, A. Marquez, J. Fernández Sanz, and R. Grau-Crespo, "Fast, accurate and non-empirical determination of the lattice thermal conductivities of I-III-VI2 chalcopyrite semiconductors," *ChemRxiv*, 2021.

[78] K. Pal, C. W. Park, Y. Xia, J. Shen, and C. Wolverton, "Scale-invariant Machine-learning Model Accelerates the Discovery of Quaternary Chalcogenides with Ultralow Lattice Thermal Conductivity," *arXiv preprint arXiv:2109.03751*, 2021.

[79] J. Brorsson, A. Hashemi, Z. Fan, E. Fransson, F. Eriksson, T. Ala-Nissila, A. V. Krasheninnikov, H.-P. Komsa, and P. Erhart, "Efficient Calculation of the Lattice Thermal Conductivity by Atomistic Simulations with Ab Initio Accuracy," *Advanced Theory and Simulations*, p. 2100217, 2021.

[80] W. Chen, J.-H. Pöhls, G. Hautier, D. Broberg, S. Bajaj, U. Aydemir, Z. M. Gibbs, H. Zhu, M. Asta, G. J. Snyder *et al.*, "Understanding thermoelectric properties from high-throughput calculations: trends, insights, and comparisons with experiment," *Journal of Materials Chemistry C*, vol. 4, no. 20, pp. 4414–4426, 2016.

[81] M. Ester, H.-P. Kriegel, J. Sander, and X. 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*. AAAI Press, 1996, p. 226–231.

[82] Y. Sheng, Y. Wu, J. Yang, W. Lu, P. Villars, and W. Zhang, "Active learning for the power factor prediction in diamond-like thermoelectric materials," *npj Computational Materials*, vol. 6, no. 1, pp. 1–7, 2020.

[83] H. Yoshihama and H. Kaneko, "Design of thermoelectric materials with high electrical conductivity, high Seebeck coefficient, and low thermal conductivity," *Analytical Science Advances*, vol. 2, no. 5-6, pp. 289–294, 2021.

[84] A. K. Pimachev and S. Neogi, "First-principles prediction of electronic transport in fabricated semiconductor heterostructures via physics-aware machine learning," *npj Computational Materials*, vol. 7, no. 1, pp. 1–12, 2021.

[85] G. S. Na, S. Jang, and H. Chang, "Predicting thermoelectric properties from chemical formula with explicitly identifying dopant effects," *npj Computational Materials*, vol. 7, no. 1, pp. 1–11, 2021.

[86] https://crystdb.nims.go.jp/.

[87] Y. Xu, M. Yamazaki, and P. Villars, "Inorganic materials database for exploring the nature of material," *Japanese Journal of Applied Physics*, vol. 50, no. 11S, p. 11RH02, 2011.

[88] http://thermoelectrics.citrination.com.

[89] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[90] http://info.eecs.northwestern.edu/SeebeckCoefficientPredictor.

[91] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[92] K. Choudhary, B. DeCost, and F. Tavazza, "Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape," *Physical Review Materials*, vol. 2, no. 8, p. 083801, 2018.

[93] B. Settles, "Active Learning Literature Survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[94] B. C. Sales, D. Mandrus, and R. K. Williams, "Filled skutterudite antimonides: a new class of thermoelectric materials," *Science*, vol. 272, no. 5266, pp. 1325–1328, 1996.

[95] D. P. Young, P. Khalifah, R. J. Cava, and A. P. Ramirez, "Thermoelectric properties of pure and doped FeMSb (M= V, Nb)," *Journal of Applied Physics*, vol. 87, no. 1, pp. 317–321, 2000.

[96] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[97] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[98] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, ser. ICLR '17, 2017.

[99] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 242–252.

[100] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1188–1196.

[101] L. M. Antunes, R. Grau-Crespo, and K. T. Butler, "Distributed Representations of Atoms and Materials for Machine Learning," *arXiv preprint arXiv:2107.14664*, 2021.

[102] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, and S.-C. Zhang, "Learning atoms for materials discovery," *Proceedings of the National Academy of Sciences*, vol. 115, no. 28, pp. E6411–E6417, 2018.

[103] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, "Unsupervised word embeddings capture latent knowledge from materials science literature," *Nature*, vol. 571, no. 7763, pp. 95–98, 2019.

[104] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[105] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[106] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT press, 2012, ch. 14.4.3, pp. 492–493.

[107] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[108] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. a. Persson, "The Materials Project: A materials genome approach to accelerating materials innovation," *APL Materials*, vol. 1, no. 1, p. 011002, 2013.

[109] G. Bergerhoff, I. Brown, F. Allen *et al.*, "Crystallographic databases," *International Union of Crystallography, Chester*, vol. 360, pp. 77–95, 1987.

[110] T. Xie, A. France-Lanord, Y. Wang, Y. Shao-Horn, and J. C. Grossman, "Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials," *Nature Communications*, vol. 10, no. 1, pp. 1–9, 2019.

[111] F. Noritake, K. Kawamura, T. Yoshino, and E. Takahashi, "Molecular dynamics simulation and electrical conductivity measurement of $Na_2O\bullet 3\,SiO_2$ melt under high pressure; relationship between its structure and properties," *Journal of Non-Crystalline Solids*, vol. 358, no. 23, pp. 3109–3118, 2012.

[112] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, "Graph networks as a universal machine learning framework for molecules and crystals," *Chemistry of Materials*, vol. 31, no. 9, pp. 3564–3572, 2019.

[113] A. Y.-T. Wang, S. K. Kauwe, R. J. Murdock, and T. D. Sparks, "Compositionally restricted attention-based network for materials property predictions," *npj Computational Materials*, vol. 7, no. 1, pp. 1–10, 2021.

[114] D. W. Davies, K. T. Butler, A. J. Jackson, A. Morris, J. M. Frost, J. M. Skelton, and A. Walsh, "Computational screening of all stoichiometric inorganic materials," *Chem*, vol. 1, no. 4, pp. 617–627, 2016.

[115] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[116] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[117] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, p. 386, 1958.

[118] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. MIT Press Limited, 1969.

[119] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[120] M. A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.

[121] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[122] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[123] J. Devlin, M. Chang, K. Lee, K. Toutanova, C. Doran, and T. Solorio, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *ACL, Minneapolis, Minnesota (Jun 2019)*, pp. 4171–4186, 2019.

[124] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving Language Understanding by Generative Pre-Training," 2018.

[125] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[126] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[127] "Introducing ChatGPT," https://openai.com/blog/chatgpt, accessed: 2023-06-21.

[128] F. J. DiSalvo, "Challenges and opportunities in solid-state chemistry," *Pure Appl. Chem.*, vol. 72, no. 10, pp. 1799–1807, 2000.

[129] A. Zunger, "Inverse design in search of materials with target functionalities," *Nature Rev. Chem.*, vol. 2, no. 4, pp. 1–16, 2018.

[130] C. Duan, F. Liu, A. Nandy, and H. J. Kulik, "Putting Density Functional Theory to the Test in Machine-Learning-Accelerated Materials Discovery," *J. Phys. Chem. Lett.*, vol. 12, pp. 4628–4637, 2021.

[131] S. D. Midgley, S. Hamad, K. T. Butler, and R. Grau-Crespo, "Bandgap Engineering in the Configurational Space of Solid Solutions via Machine Learning: (Mg,Zn)O Case Study," *J. Phys. Chem. Lett.*, vol. 12, pp. 5163–5168, 2021.

[132] Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller III, "OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features," *J. Chem. Phys.*, vol. 153, no. 12, p. 124111, 2020.

[133] A. Raza, A. Sturluson, C. M. Simon, and X. Fern, "Message passing neural networks for partial charge assignment to metal–organic frameworks," *J. Phys. Chem. C*, vol. 124, no. 35, pp. 19 070–19 082, 2020.

[134] F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, "Crystal structure representations for machine learning models of formation energies," *Int. J. Quantum Chem.*, vol. 115, no. 16, pp. 1094–1101, 2015.

[135] Y. Zhuo, A. Mansouri Tehrani, and J. Brgoch, "Predicting the Band Gaps of Inorganic Solids by Machine Learning," *J. Phys. Chem. Lett.*, vol. 9, no. 7, pp. 1668–1673, 2018.

[136] D. W. Davies, K. T. Butler, and A. Walsh, "Data-driven discovery of photoactive quaternary oxides using first-principles machine learning," *Chem. Mater.*, vol. 31, no. 18, pp. 7221–7230, 2019.

[137] N. Artrith, "Machine learning for the modeling of interfaces in energy storage and conversion materials," *J. Phys. Energy*, vol. 1, no. 3, p. 032002, 2019.

[138] G. E. Hinton, T. J. Sejnowski *et al.*, *Unsupervised Learning: Foundations of Neural Computation*. MIT press, 1999.

[139] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2010.

[140] C. Duan, F. Liu, A. Nandy, and H. J. Kulik, "Semi-supervised Machine Learning Enables the Robust Detection of Multireference Character at Low Cost," *J Phys. Chem. Lett.*, vol. 11, no. 16, pp. 6640–6648, 2020.

[141] Y. Zhang and A. A. Lee, "Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning," *Chem. Sci.*, vol. 10, no. 35, pp. 8154–8163, 2019.

[142] H. Huo, Z. Rong, O. Kononova, W. Sun, T. Botari, T. He, V. Tshitoyan, and G. Ceder, "Semi-supervised machine-learning classification of materials synthesis procedures," *npj Comput. Mater.*, vol. 5, no. 1, pp. 1–7, 2019.

[143] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Preprint at https://arxiv.org/abs/1301.3781*, 2013.

[144] S. K. Chakravarti, "Distributed Representation of Chemical Fragments," *ACS Omega*, vol. 3, no. 3, pp. 2825–2836, 2018.

[145] D. Jha, L. Ward, A. Paul, W.-k. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, "ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition," *Sci. Rep.*, vol. 8, no. 1, pp. 1–13, 2018.

[146] R. E. Goodall and A. A. Lee, "Predicting materials properties without crystal structure: Deep representation learning from stoichiometry," *Nature Commun.*, vol. 11, no. 1, pp. 1–9, 2020.

[147] J. Mitchell and M. Lapata, "Vector-based Models of Semantic Composition," in *Proceedings of ACL-08: HLT*, 2008, pp. 236–244.

[148] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton, "Combinatorial screening for new materials in unconstrained composition space with machine learning," *Phys. Rev. B*, vol. 89, no. 9, p. 094104, 2014.

[149] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites." *J. Reine Angew. Math.*, vol. 1908, pp. 97 – 102, 1908.

[150] N. E. Zimmermann and A. Jain, "Local structure order parameters and site fingerprints for quantification of coordination environment and crystal structure similarity," *RSC Adv.*, vol. 10, no. 10, pp. 6063–6081, 2020.

[151] H. Pan, A. M. Ganose, M. Horton, M. Aykol, K. A. Persson, N. E. Zimmermann, and A. Jain, "Benchmarking Coordination Number Prediction Algorithms on Inorganic Crystal Structures," *Inorg. Chem.*, vol. 60, no. 3, pp. 1590–1603, 2021.

[152] M. T. Pilehvar and N. Collier, "De-Conflated Semantic Representations," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1680–1690.

[153] ——, "Inducing Embeddings for Rare and Unseen Words by Leveraging Lexical Resources," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Vol. 2, Short Papers*. Association for Computational Linguistics, 2017, pp. 388–393.

[154] A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain, "Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm," *npj Comput. Mater.*, vol. 6, no. 1, pp. 1–10, 2020.

[155] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder *et al.*, "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation," *APL Mater.*, vol. 1, no. 1, p. 011002, 2013.

[156] S. P. Ong, S. Cholia, A. Jain, M. Brafman, D. Gunter, G. Ceder, and K. A. Persson, "The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles," *Comput. Mater. Sci.*, vol. 97, pp. 209–215, 2015.

[157] M. De Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C. K. Ande, S. Van Der Zwaag, J. J. Plata *et al.*, "Charting the complete elastic properties of inorganic crystalline compounds," *Sci. Data*, vol. 2, no. 1, pp. 1–13, 2015.

[158] I. Petousis, D. Mrdjenovich, E. Ballouz, M. Liu, D. Winston, W. Chen, T. Graf, T. D. Schladt, K. A. Persson, and F. B. Prinz, "High-throughput screening of inorganic compounds for the discovery of novel dielectric and optical materials," *Sci. Data*, vol. 4, no. 1, pp. 1–12, 2017.

[159] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, "Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD)," *JOM*, vol. 65, no. 11, pp. 1501–1509, 2013.

[160] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Comput. Mater.*, vol. 2, no. 1, pp. 1–7, 2016.

[161] Y. Kawazoe, "Nonequilibrium Phase Diagrams of Ternary Amorphous Alloys," *LB: New Ser., Group III: Condensed*, vol. 37, pp. 1–295, 1997.

[162] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, and R. Armiento, "Machine Learning Energies of 2 Million Elpasolite (ABC2D6) Crystals," *Phys. Rev. Lett.*, vol. 117, no. 13, p. 135502, 2016.

[163] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Preprint at https://arxiv.org/abs/1412.6980*, 2014.

[164] H. Moss, D. Leslie, and P. Rayson, "Using J-K-fold Cross Validation To Reduce Variance When Tuning NLP Models," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, aug 2018, pp. 2978–2989.

[165] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. a. Persson, "The Materials Project: A materials genome approach to accelerating materials innovation," *APL Mater.*, vol. 1, no. 1, p. 011002, 2013.

[166] L. Van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[167] P. C. Kainen and V. Kurkova, "Quasiorthogonal Dimension," in *Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications*. Springer, 2020, pp. 615–629.

[168] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, "Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning," *Phys. Rev. Lett.*, vol. 108, no. 5, p. 058301, 2012.

[169] H. R. Banjade, S. Hauri, S. Zhang, F. Ricci, W. Gong, G. Hautier, S. Vucetic, and Q. Yan, "Structure motif–centric learning framework for inorganic crystalline systems," *Sci. Adv.*, vol. 7, no. 17, p. eabf1754, 2021.

[170] T. J. Seebeck, "Magnetische polarisation der metalle und erze durch temperaturdifferenz," *Annalen der Physik*, vol. 82, pp. 253–286, 1826.

[171] P. M. Roget, *Treatises on electricity, galvanism, magnetism, and electro-magnetism*. Baldwin and Cradock, London, 1832.

[172] O. Caballero-Calero, J. R. Ares, and M. Martín-González, "Environmentally Friendly Thermoelectric Materials: High Performance from Inorganic Components with Low Toxicity and Abundance in the Earth," *Advanced Sustainable Systems*, vol. 5, no. 11, p. 2100095, 2021.

[173] R. Freer and A. V. Powell, "Realising the potential of thermoelectric technology: A Roadmap," *Journal of Materials Chemistry C*, vol. 8, no. 2, pp. 441–463, 2020.

[174] J. R. Sootsman, D. Y. Chung, and M. G. Kanatzidis, "New and Old Concepts in Thermoelectric Materials," *Angewandte Chemie International Edition*, vol. 48, no. 46, pp. 8616–8639, 2009.

[175] C. Gayner and K. K. Kar, "Recent advances in thermoelectric materials," *Progress in Materials Science*, vol. 83, pp. 330–382, 2016.

[176] D. Beretta, N. Neophytou, J. M. Hodges, M. G. Kanatzidis, D. Narducci, M. Martin-Gonzalez, M. Beekman, B. Balke, G. Cerretti, W. Tremel, A. Zevalkink, A. I. Hofmann, C. Müller, B. Dörling, M. Campoy-Quiles, and M. Caironi, "Thermoelectrics: From history, a window to the future," *Materials Science & Engineering R*, vol. 138, p. 100501, 2019.

[177] B. Poudel, Q. Hao, Y. Ma, Y. Lan, A. Minnich, B. Yu, X. Yan, D. Wang, A. Muto, D. Vashaee *et al.*, "High-thermoelectric performance of nanostructured bismuth antimony telluride bulk alloys," *Science*, vol. 320, no. 5876, pp. 634–638, 2008.

[178] G. Tan, F. Shi, S. Hao, L.-D. Zhao, H. Chi, X. Zhang, C. Uher, C. Wolverton, V. P. Dravid, and M. G. Kanatzidis, "Non-equilibrium processing leads to record high thermoelectric figure of merit in PbTe–SrTe," *Nature communications*, vol. 7, no. 1, p. 12167, 2016.

[179] X. Wang, H. Lee, Y. Lan, G. Zhu, G. Joshi, D. Wang, J. Yang, A. Muto, M. Tang, J. Klatsky *et al.*, "Enhanced thermoelectric figure of merit in nanostructured n-type silicon germanium bulk alloy," *Applied Physics Letters*, vol. 93, no. 19, p. 193121, 2008.

[180] L.-D. Zhao, C. Chang, G. Tan, and M. G. Kanatzidis, "SnSe: a remarkable new thermoelectric material," *Energy & Environmental Science*, vol. 9, no. 10, pp. 3044–3060, 2016.

[181] M. Zhou, G. J. Snyder, L. Li, and L.-D. Zhao, "Lead-free tin chalcogenide thermoelectric materials," *Inorganic Chemistry Frontiers*, vol. 3, no. 11, pp. 1449–1463, 2016.

[182] H. Liu, X. Shi, F. Xu, L. Zhang, W. Zhang, L. Chen, Q. Li, C. Uher, T. Day, and G. J. Snyder, "Copper ion liquid-like thermoelectrics," *Nature Materials*, vol. 11, no. 5, pp. 422–425, 2012.

[183] T. Caillat, J.-P. Fleurial, and A. Borshchevsky, "Bridgman-solution crystal growth and characterization of the skutterudite compounds $\mathrm{CoSb_3}$ and $\mathrm{RhSb_3}$," *Journal of Crystal Growth*, vol. 166, no. 1-4, pp. 722–726, 1996.

[184] F. Gascoin, S. Ottensmann, D. Stark, S. M. Haïle, and G. J. Snyder, "Zintl Phases as Thermoelectric Materials: Tuned Transport Properties of the Compounds $\mathrm{Ca_xYb_{1-x}Zn_2Sb_2}$," *Advanced Functional Materials*, vol. 15, no. 11, pp. 1860–1864, 2005.

[185] G. Nolas, J. Cohn, G. Slack, and S. Schujman, "Semiconducting Ge clathrates: Promising candidates for thermoelectric applications," *Applied Physics Letters*, vol. 73, no. 2, pp. 178–180, 1998.

[186] F. Aliev, N. Brandt, V. Moshchalkov, V. Kozyrkov, R. Skolozdra, and A. Belogorokhov, "Gap at the Fermi level in the intermetallic vacancy system RBiSn (R=Ti,Zr,Hf)," *Zeitschrift für Physik B Condensed Matter*, vol. 75, no. 2, pp. 167–171, 1989.

[187] F. Aliev, V. Kozyrkov, V. Moshchalkov, R. Scolozdra, and K. Durczewski, "Narrow band in the intermetallic compounds MNiSn (M=Ti,Zr,Hf)," *Zeitschrift für Physik B Condensed Matter*, vol. 80, no. 3, pp. 353–357, 1990.

[188] H. Hohl, A. Ramirez, W. Kaefer, K. Fess, C. Thurner, C. Kloc, and E. Bucher, "A New Class of Materials with Promising Thermoelectric Properties: MNiSn (M=Ti,Zr,Hf)," *MRS Online Proceedings Library (OPL)*, vol. 478, 1997.

[189] I. Terasaki, Y. Sasago, and K. Uchinokura, "Large thermoelectric power in $\mathrm{NaCo_2O_4}$ single crystals," *Physical Review B*, vol. 56, no. 20, p. R12685, 1997.

[190] R. Tian, T. Zhang, D. Chu, R. Donelson, L. Tao, and S. Li, "Enhancement of high temperature thermoelectric performance in Bi, Fe co-doped layered oxide-based material $\mathrm{Ca_3Co_4O_{9+\delta}}$," *Journal of Alloys and Compounds*, vol. 615, pp. 311–315, 2014.

[191] C. Zhou, Y. K. Lee, Y. Yu, S. Byun, Z.-Z. Luo, H. Lee, B. Ge, Y.-L. Lee, X. Chen, J. Y. Lee, O. Cojocaru-Mirédin, H. Chang, J. Im, S.-P. Cho, M. Wuttig, V. P. Dravid, M. G. Kanatzidis, and I. Chung, "Polycrystalline SnSe with a thermoelectric figure of merit greater than the single crystal," *Nature Materials*, vol. 20, no. 10, pp. 1378–1384, 2021.

[192] T. M. Tritt and M. Subramanian, "Thermoelectric Materials, Phenomena, and Applications: A Bird's Eye View," *MRS Bulletin*, vol. 31, no. 3, pp. 188–198, 2006.

[193] T. D. Sparks, M. W. Gaultois, A. Oliynyk, J. Brgoch, and B. Meredig, "Data mining our way to the next generation of thermoelectrics," *Scripta Materialia*, vol. 111, pp. 10–15, 2016.

[194] P. Gorai, V. Stevanović, and E. S. Toberer, "Computationally guided discovery of thermoelectric materials," *Nature Reviews Materials*, vol. 2, no. 9, pp. 1–16, 2017.

[195] J. Recatala-Gomez, A. Suwardi, I. Nandhakumar, A. Abutaha, and K. Hippalgaonkar, "Toward Accelerated Thermoelectric Materials and Process Discovery," *ACS Applied Energy Materials*, vol. 3, no. 3, pp. 2240–2257, 2020.

[196] G. K. Madsen, "Automated search for new thermoelectric materials: the case of LiZnSb," *Journal of the American Chemical Society*, vol. 128, no. 37, pp. 12 140–12 146, 2006.

[197] J. Carrete, N. Mingo, S. Wang, and S. Curtarolo, "Nanograined Half-Heusler Semiconductors as Advanced Thermoelectrics: An Ab Initio High-Throughput Statistical Study," *Advanced Functional Materials*, vol. 24, no. 47, pp. 7427–7432, 2014.

[198] C. Toher, J. J. Plata, O. Levy, M. De Jong, M. Asta, M. B. Nardelli, and S. Curtarolo, "High-throughput computational screening of thermal conductivity, Debye temperature, and Grüneisen parameter using a quasiharmonic Debye model," *Physical Review B*, vol. 90, no. 17, p. 174107, 2014.

[199] P. Gorai, P. Parilla, E. S. Toberer, and V. Stevanovic, "Computational Exploration of the Binary $\mathrm{A_1B_1}$ Chemical Space for Thermoelectric Performance," *Chemistry of Materials*, vol. 27, no. 18, pp. 6213–6221, 2015.

[200] H. Zhu, G. Hautier, U. Aydemir, Z. M. Gibbs, G. Li, S. Bajaj, J.-H. Pöhls, D. Broberg, W. Chen, A. Jain, M. A. White, M. Asta, G. J. Snyder, K. Persson, and G. Ceder, "Computational and experimental investigation of $\mathrm{TmAgTe_2}$ and $\mathrm{XYZ_2}$ compounds, a new group of thermoelectric materials identified by first-principles high-throughput screening," *Journal of Materials Chemistry C*, vol. 3, no. 40, pp. 10 554–10 565, 2015.

[201] P. Gorai, A. Ganose, A. Faghaninia, A. Jain, and V. Stevanović, "Computational discovery of promising new n-type dopable ABX Zintl thermoelectric materials," *Materials Horizons*, vol. 7, no. 7, pp. 1809–1818, 2020.

[202] X. Chen, X. Zhang, J. Gao, Q. Li, Z. Shao, H. Lin, and M. Pan, "Computational Search for Better Thermoelectric Performance in Nickel-Based Half-Heusler Compounds," *ACS Omega*, vol. 6, no. 28, pp. 18 269–18 280, 2021.

[203] J.-H. Pöhls, S. Chanakian, J. Park, A. M. Ganose, A. Dunn, N. Friesen, A. Bhattacharya, B. Hogan, S. Bux, A. Jain, A. Mar, and A. Zevalkink, "Experimental validation of high thermoelectric performance in $\mathrm{RECuZnP_2}$ predicted by high-throughput DFT calculations," *Materials Horizons*, vol. 8, no. 1, pp. 209–215, 2021.

[204] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, "AiiDA: automated interactive infrastructure and database for computational science," *Computational Materials Science*, vol. 111, pp. 218–230, 2016.

[205] K. Mathew, J. H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I.-h. Chu, T. Smidt, B. Bocklund, M. Horton, J. Dagdelen, B. Wood, Z.-K. Liu, J. Neaton, S. Ping Ong, K. Persson, and A. Jain, "Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows," *Computational Materials Science*, vol. 139, pp. 140–152, 2017.

[206] A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, and K. A. Persson, "FireWorks: a dynamic workflow system designed for high-throughput applications," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 5037–5059, 2015.

[207] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, and D. Morgan, "AFLOW: An automatic framework for high-throughput materials discovery," *Computational Materials Science*, vol. 58, pp. 218–226, 2012.

[208] F. Zapata, L. Ridder, J. Hidding, C. R. Jacob, I. Infante, and L. Visscher, "QMflows: A Tool Kit for Interoperable Parallel Workflows in Quantum Chemistry," *Journal of Chemical Information and Modeling*, vol. 59, no. 7, pp. 3191–3197, 2019.

[209] C. S. Adorf, P. M. Dodd, V. Ramasubramani, and S. C. Glotzer, "Simple data and workflow management with the signac framework," *Computational Materials Science*, vol. 146, pp. 220–229, 2018.

[210] T. Mayeshiba, H. Wu, T. Angsten, A. Kaczmarowski, Z. Song, G. Jenness, W. Xie, and D. Morgan, "The MAterials Simulation Toolkit (MAST) for atomistic modeling of defects and diffusion," *Computational Materials Science*, vol. 126, pp. 90–102, 2017.

[211] T. Wang, C. Zhang, H. Snoussi, and G. Zhang, "Machine Learning Approaches for Thermoelectric Materials Research," *Advanced Functional Materials*, vol. 30, no. 5, p. 1906041, 2020.

[212] R. Juneja and A. K. Singh, "Accelerated Discovery of Thermoelectric Materials Using Machine Learning," in *Artificial Intelligence for Materials Science*. Springer, 2021, pp. 133–152.

[213] G. Han, Y. Sun, Y. Feng, G. Lin, and N. Lu, "Machine Learning Regression Guided Thermoelectric Materials Discovery–A Review," *ES Materials & Manufacturing*, vol. 14, pp. 20–35, 2021.

[214] X. Qian and R. Yang, "Machine learning for predicting thermal transport properties of solids," *Materials Science and Engineering: R: Reports*, vol. 146, p. 100642, 2021.

[215] L. M. Antunes, Vikram, J. J. Plata, A. V. Powell, K. T. Butler, and R. Grau-Crespo, "Machine Learning Approaches for Accelerating the Discovery of Thermoelectric Materials," in *Advancing Materials Innovation with Machine Learning*. ACS Publications, 2022.

[216] H. Yuan, S. Han, R. Hu, W. Jiao, M. Li, H. Liu, and Y. Fang, "Machine learning for accelerated prediction of the Seebeck coefficient at arbitrary carrier concentration," *Materials Today Physics*, p. 100706, 2022.

[217] Z. Yang, Y. Sheng, C. Zhu, J. Ni, Z. Zhu, J. Xi, W. Zhang, and J. Yang, "Accurate and explainable machine learning for the power factors of diamond-like thermoelectric materials," *Journal of Materiomics*, 2021.

[218] L. Laugier, D. Bash, J. Recatala, H. K. Ng, S. Ramasamy, C.-S. Foo, V. R. Chandrasekhar, and K. Hippalgaonkar, "Predicting thermoelectric properties from crystal graphs and material descriptors - first application for functional materials," *arXiv preprint arXiv:1811.06219*, 2018.

[219] A. Seko, A. Togo, H. Hayashi, K. Tsuda, L. Chaput, and I. Tanaka, "Prediction of Low-Thermal-Conductivity Compounds with First-Principles Anharmonic Lattice-Dynamics Calculations and Bayesian Optimization," *Physical Review Letters*, vol. 115, no. 20, p. 205901, 2015.

[220] Y. Zhang and C. Ling, "A strategy to apply machine learning to small datasets in materials science," *npj Computational Materials*, vol. 4, no. 1, pp. 1–8, 2018.

[221] R. Juneja, G. Yumnam, S. Satsangi, and A. K. Singh, "Coupling the High-Throughput Property Map to Machine Learning for Predicting Lattice Thermal Conductivity," *Chemistry of Materials*, vol. 31, no. 14, pp. 5145–5151, 2019.

[222] A. Tewari, S. Dixit, N. Sahni, and S. P. Bordas, "Machine learning approaches to identify and design low thermal conductivity oxides for thermoelectric applications," *Data-Centric Engineering*, vol. 1, 2020.

[223] J. Liu, S. Han, G. Cao, Z. Zhou, C. Sheng, and H. Liu, "A high-throughput descriptor for prediction of lattice thermal conductivity of half-Heusler compounds," *Journal of Physics D: Applied Physics*, vol. 53, no. 31, p. 315301, 2020.

[224] R. Li, Z. Liu, A. Rohskopf, K. Gordiz, A. Henry, E. Lee, and T. Luo, "A deep neural network interatomic potential for studying thermal conductivity of β-$\mathrm{Ga_2O_3}$," *Applied Physics Letters*, vol. 117, no. 15, p. 152102, 2020.

[225] C. Loftis, K. Yuan, Y. Zhao, M. Hu, and J. Hu, "Lattice Thermal Conductivity Prediction Using Symbolic Regression and Machine Learning," *The Journal of Physical Chemistry A*, vol. 125, no. 1, pp. 435–450, 2020.

[226] J. M. Choi, K. Lee, S. Kim, M. Moon, W. Jeong, and S. Han, "Accelerated computation of lattice thermal conductivity using neural network interatomic potentials," *Computational Materials Science*, vol. 211, p. 111472, 2022.

[227] M. V. Tabib, O. M. Løvvik, K. Johannessen, A. Rasheed, E. Sagvolden, and A. M. Rustad, "Discovering Thermoelectric Materials Using Machine Learning: Insights and Challenges," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 392–401.

[228] Z.-L. Wang, Y. Yokoyama, T. Onda, Y. Adachi, and Z.-C. Chen, "Improved Thermoelectric Properties of Hot-Extruded Bi–Te–Se Bulk Materials with Cu Doping and Property Predictions via Machine Learning," *Advanced Electronic Materials*, vol. 5, no. 6, p. 1900079, 2019.

[229] Y. Zhong, X. Hu, D. Sarker, Q. Xia, L. Xu, C. Yang, Z.-K. Han, S. V. Levchenko, and J. Cui, "Data analytics accelerates the experimental discovery of new thermoelectric materials with extremely high figure of merit," *arXiv preprint arXiv:2104.08033*, 2021.

[230] Y. Gan, G. Wang, J. Zhou, and Z. Sun, "Prediction of thermoelectric performance for layered IV-V-VI semiconductors by high-throughput ab initio calculations and machine learning," *npj Computational Materials*, vol. 7, no. 1, pp. 1–10, 2021.

[231] R. Jaafreh, K. Y. Seong, J.-G. Kim, and K. Hamad, "A deep learning perspective into the figure-of-merit of thermoelectric materials," *Materials Letters*, vol. 319, p. 132299, 2022.

[232] R. G. Parr, "Density Functional Theory of Atoms and Molecules," in *Horizons of Quantum Chemistry*. Springer, 1980, pp. 5–15.

[233] J. J. Plata, P. Nath, J. F. Sanz, and A. Marquez, "In silico modeling of inorganic thermoelectric materials," in *Reference Module in Chemistry, Molecular Sciences and Chemical Engineering*. Elsevier, 2022.

[234] H. Shi, D. Parker, M.-H. Du, and D. J. Singh, "Connecting thermoelectric performance and topological-insulator behavior: Bi_2Te_3 and Bi_2Te_2Se from first principles," *Physical Review Applied*, vol. 3, no. 1, p. 014004, 2015.

[235] R. Freer, D. Ekren, T. Ghosh, K. Biswas, P. Qiu, S. Wan, L. Chen, S. Han, C. Fu, T. Zhu *et al.*, "Key properties of inorganic thermoelectric materials—tables (version 1)," *Journal of Physics: Energy*, vol. 4, no. 2, p. 022002, 2022.

[236] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[237] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on Multi-Output Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2409–2429, 2019.

[238] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[239] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[240] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[241] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1. IEEE, 1994, pp. 55–60.

[242] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[243] L. M. Antunes, R. Grau-Crespo, and K. T. Butler, "Distributed representations of atoms and materials for machine learning," *npj Computational Materials*, vol. 8, no. 1, p. 44, Mar 2022.

[244] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[245] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[246] L. Ward, A. Dunn, A. Faghaninia, N. E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla *et al.*, "Matminer: An open source toolkit for materials data mining," *Computational Materials Science*, vol. 152, pp. 60–69, 2018.

[247] 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*, vol. 12, pp. 2825–2830, 2011.

[248] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, ch. 7, pp. 245–246.

[249] G. Kresse and J. Hafner, "Ab initio molecular dynamics for liquid metals," *Physical Review B*, vol. 47, no. 1, p. 558, 1993.

[250] G. Kresse and J. Furthmüller, "Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set," *Physical Review B*, vol. 54, no. 16, p. 11169, 1996.

[251] P. E. Blöchl, "Projector augmented-wave method," *Physical Review B*, vol. 50, no. 24, p. 17953, 1994.

[252] G. Kresse and D. Joubert, "From ultrasoft pseudopotentials to the projector augmented-wave method," *Physical Review B*, vol. 59, no. 3, p. 1758, 1999.

[253] G. K. H. Madsen, J. Carrete, and M. J. Verstraete, "BoltzTraP2, a program for interpolating band structures and calculating semi-classical transport coefficients," *Computer Physics Communications*, vol. 231, pp. 140–145, 2018.

[254] L. Wu, Y. Xiao, M. Ghosh, Q. Zhou, and Q. Hao, "Machine Learning Prediction for Bandgaps of Inorganic Materials," *ES Materials & Manufacturing*, 2020.

[255] D. W. Davies, K. T. Butler, A. J. Jackson, J. M. Skelton, K. Morita, and A. Walsh, "SMACT: Semiconducting Materials by Analogy and Chemical Theory," *Journal of Open Source Software*, vol. 4, no. 38, p. 1361, 2019.

[256] A. F. Zahrt, J. J. Henle, and S. E. Denmark, "Cautionary Guidelines for Machine Learning Studies with Combinatorial Datasets," *ACS Combinatorial Science*, vol. 22, no. 11, pp. 586–591, 2020.

[257] N. Lu, G. Han, Y. Sun, Y. Feng, and G. Lin, "Artificial intelligence assisted thermoelectric materials design and discovery," *Research Square*, 2022.

[258] A. Y.-T. Wang, M. S. Mahmoud, M. Czasny, and A. Gurlo, "CrabNet for Explainable Deep Learning in Materials Science: Bridging the Gap Between Academia and Industry," *Integrating Materials and Manufacturing Innovation*, vol. 11, no. 1, pp. 41–56, 2022.

[259] G. Liu, Z. Lin, and Y. Yu, "Multi-output regression on the output manifold," *Pattern Recognition*, vol. 42, no. 11, pp. 2737–2743, 2009.

[260] A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, "Structure prediction drives materials discovery," *Nature Reviews Materials*, vol. 4, no. 5, pp. 331–348, 2019.

[261] A. R. Oganov, *Modern Methods of Crystal Structure Prediction*. John Wiley & Sons, 2011.

[262] C. J. Pickard and R. Needs, "High-Pressure Phases of Silane," *Physical Review Letters*, vol. 97, no. 4, p. 045504, 2006.

[263] ——, "Ab initio random structure searching," *Journal of Physics: Condensed Matter*, vol. 23, no. 5, p. 053201, 2011.

[264] A. R. Oganov and C. W. Glass, "Crystal structure prediction using *ab initio* evolutionary techniques: Principles and applications," *The Journal of Chemical Physics*, vol. 124, no. 24, p. 244704, 2006.

[265] C. J. Court, B. Yildirim, A. Jain, and J. M. Cole, "3-D Inorganic Crystal Structure Generation and Property Prediction via Representation Learning," *Journal of Chemical Information and Modeling*, vol. 60, no. 10, pp. 4518–4535, 2020.

[266] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. Jaakkola, "Crystal Diffusion Variational Autoencoder for Periodic Material Generation," *arXiv preprint arXiv:2110.06197*, 2021.

[267] D. Yan, A. D. Smith, and C.-C. Chen, "Structure prediction and materials design with generative neural networks," *Nature Computational Science*, pp. 1–3, 2023.

[268] M. Alverson, S. G. Baird, R. Murdock, J. Johnson, T. D. Sparks *et al.*, "Generative adversarial networks and diffusion models in material discovery," *Digital Discovery*, 2024.

[269] L. Chen, W. Zhang, Z. Nie, S. Li, and F. Pan, "Generative models for inverse design of inorganic solid materials," *J. Mater. Inform*, vol. 1, no. 4, 2021.

[270] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT," *arXiv preprint arXiv:2303.04226*, 2023.

[271] Y. Liu, Z. Yang, Z. Yu, Z. Liu, D. Liu, H. Lin, M. Li, S. Ma, M. Avdeev, and S. Shi, "Generative artificial intelligence and its applications in materials science: Current situation and future perspectives," *Journal of Materiomics*, vol. 9, no. 4, pp. 798–816, 2023.

[272] A. M. Bran, S. Cox, A. D. White, and P. Schwaller, "ChemCrow: Augmenting large-language models with chemistry tools," *arXiv preprint arXiv:2304.05376*, 2023.

[273] K. M. Jablonka, P. Schwaller, A. Ortega-Guerrero, and B. Smit, "Is GPT-3 all you need for low-data discovery in chemistry?" *ChemRxiv*, pp. 1–32, 2023.

[274] T. Xie, Y. Wa, W. Huang, Y. Zhou, Y. Liu, Q. Linghu, S. Wang, C. Kit, C. Grazian, and B. Hoex, "Large Language Models as Master Key: Unlocking the Secrets of Materials Science with GPT," *arXiv preprint arXiv:2304.02213*, 2023.

[275] N. Fu, L. Wei, Y. Song, Q. Li, R. Xin, S. S. Omee, R. Dong, E. M. D. Siriwardane, and J. Hu, "Material transformers: deep learning language models for generative materials design," *Machine Learning: Science and Technology*, vol. 4, no. 1, p. 015001, 2023.

[276] K. M. Jablonka, Q. Ai, A. Al-Feghali, S. Badhwar, J. D. Bocarsly, A. M. Bran, S. Bringuier, L. C. Brinson, K. Choudhary, D. Circi *et al.*, "14 examples of how LLMs can transform materials science and chemistry: a reflection on a large language model hackathon," *Digital Discovery*, vol. 2, no. 5, pp. 1233–1250, 2023.

[277] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes, "Autonomous chemical research with large language models," *Nature*, vol. 624, p. 570, 2023.

[278] D. Flam-Shepherd and A. Aspuru-Guzik, "Language models can generate molecules, materials, and protein binding sites directly in three dimensions as XYZ, CIF, and PDB files," *arXiv preprint arXiv:2305.05708*, 2023.

[279] 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: Foundations of Crystallography*, vol. 47, no. 6, pp. 655–685, 1991.

[280] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative Pretraining from Pixels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1691–1703.

[281] S. Toshniwal, S. Wiseman, K. Livescu, and K. Gimpel, "Chess as a Testbed for Language Model State Tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 385–11 393.

[282] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg, "Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task," in *The Eleventh International Conference on Learning Representations*, 2023.

[283] R. Coulom, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search," in *International Conference on Computers and Games*. Springer, 2006, pp. 72–83.

[284] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[285] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[286] A. Onwuli, A. V. Hegde, K. V. Nguyen, K. T. Butler, and A. Walsh, "Element similarity in high-dimensional materials representations," *Digital Discovery*, vol. 2, no. 5, pp. 1558–1564, 2023.

[287] C. Draxl and M. Scheffler, "The NOMAD laboratory: from data sharing to artificial intelligence," *Journal of Physics: Materials*, vol. 2, no. 3, p. 036001, 2019.

[288] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis," *Comput. Mater. Sci.*, vol. 68, pp. 314–319, 2013.

[289] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating Wikipedia by Summarizing Long Sequences," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[290] A. Togo and I. Tanaka, "Spglib: a software library for crystal symmetry search," *arXiv preprint arXiv:1808.01590*, 2018.

[291] A. Jain, G. Hautier, C. J. Moore, S. P. Ong, C. C. Fischer, T. Mueller, K. A. Persson, and G. Ceder, "A high-throughput infrastructure for density functional theory calculations," *Computational Materials Science*, vol. 50, no. 8, pp. 2295–2310, 2011.

[292] M. Horton, J.-X. Shen, J. Burns, O. Cohen, F. Chabbey, A. M. Ganose, R. Guha, P. Huck, H. H. Li, M. McDermott, J. Montoya, G. Moore, J. Munro, C. O'Donnell, C. Ophus, G. Petretto, J. Riebesell, S. Wetizner, B. Wander, D. Winston, R. Yang, S. Zeltmann, A. Jain, and K. A. Persson, "Crystal Toolkit: A Web App Framework to Improve Usability and Accessibility of Materials Science Research Algorithms," *arXiv preprint arXiv:2302.06147*, 2023.

[293] "Creative Commons Attribution 4.0 License," https://creativecommons.org/licenses/by/4.0/, accessed: 2023-06-26.

[294] R. Jiao, W. Huang, P. Lin, J. Han, P. Chen, Y. Lu, and Y. Liu, "Crystal Structure Prediction by Joint Equivariant Diffusion," *arXiv preprint arXiv:2309.04475*, 2023.

[295] R. Jiao, W. Huang, Y. Liu, D. Zhao, and Y. Liu, "Space Group Constrained Crystal Generation," *arXiv preprint arXiv:2402.03992*, 2024.

[296] M. Yang, K. Cho, A. Merchant, P. Abbeel, D. Schuurmans, I. Mordatch, and E. D. Cubuk, "Scalable Diffusion for Materials Generation," *arXiv preprint arXiv:2311.09235*, 2023.

[297] N. Gruver, A. Sriram, A. Madotto, A. G. Wilson, C. L. Zitnick, and Z. Ulissi, "Fine-Tuned Language Models Generate Stable Inorganic Materials as Text," *arXiv preprint arXiv:2402.04379*, 2024.

[298] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.

[299] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.

[300] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video Diffusion Models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022.

[301] I. E. Castelli, D. D. Landis, K. S. Thygesen, S. Dahl, I. Chorkendorff, T. F. Jaramillo, and K. W. Jacobsen, "New cubic perovskites for one- and two-photonwater splitting using the computational materials repository," *Energy & Environmental Science*, vol. 5, no. 10, pp. 9034–9043, 2012.

[302] I. E. Castelli, T. Olsen, S. Datta, D. D. Landis, S. Dahl, K. S. Thygesen, and K. W. Jacobsen, "Computational screening of perovskite metal oxides for optimal solar light capture," *Energy & Environmental Science*, vol. 5, no. 2, pp. 5814–5819, 2012.

[303] C. J. Pickard, "AIRSS Data for Carbon at 10GPa and the C+N+H+O System at 1GPa," https://archive.materialscloud.org/record/2020.0026/v1, 2020.

[304] S. Baird, "mp-time-split," https://github.com/sparks-baird/mp-time-split, Accessed in 2024.

[305] T. Mazet, R. Welter, and B. Malaman, "A study of the new ferromagnetic $YbMn_6Sn_6$ compound by magnetization and neutron diffraction measurements," *Journal of Magnetism and Magnetic Materials*, vol. 204, no. 1-2, pp. 11–19, 1999.

[306] B. Pamplin, "A systematic method of deriving new semiconducting compounds by structural analogy," *Journal of Physics and Chemistry of Solids*, vol. 25, no. 7, pp. 675–684, 1964.

[307] D. Zagorac, H. Müller, S. Ruehl, J. Zagorac, and S. Rehme, "Recent developments in the Inorganic Crystal Structure Database: theoretical crystal structure data and related features," *Journal of Applied Crystallography*, vol. 52, no. 5, pp. 918–925, 2019.

[308] P. Hyde, J. Cen, S. Cassidy, N. Rees, P. Holdship, R. Smith, B. Zhu, D. Scanlon, and S. Clarke, "Lithium Intercalation into the Excitonic Insulator Candidate Ta_2NiSe_5," *Inorganic Chemistry*, vol. 62, no. 30, pp. 12 027–12 037, 2023.

[309] S. Ponou, S. Lidin, and A.-V. Mudring, "Optimization of Chemical Bonding through Defect Formation and Ordering–The Case of $Mg_7Pt_4Ge_4$," *Inorganic Chemistry*, 2023.

[310] J. González-López, J. K. Cockcroft, A. Fernández-González, A. Jimenez, and R. Grau-Crespo, "Crystal structure of cobalt hydroxide carbonate $Co_2CO_3(OH)_2$: density functional theory and X-ray diffraction investigation," *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, vol. 73, no. 5, pp. 868–873, 2017.

[311] Carnegie-Mellon Univ Pittsburgh PA Dept Of Computer Science, *Speech Understanding Systems. Summary of Results of the Five-Year Research Effort at Carnegie-Mellon University*, 1977.

[312] A. Chaffin, V. Claveau, and E. Kijak, "PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, M. Carpuat, M. de Marneffe, and I. V. M. Ruíz, Eds. Association for Computational Linguistics, 2022, pp. 2953–2967.

[313] C. D. Rosin, "Multi-armed Bandits with Episode Context," *Annals of Mathematics and Artificial Intelligence*, vol. 61, no. 3, pp. 203–230, 2011.

[314] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[315] K. Choudhary and B. DeCost, "Atomistic Line Graph Neural Network for improved materials property predictions," *npj Computational Materials*, vol. 7, no. 1, p. 185, 2021.

[316] M. Kusaba, C. Liu, and R. Yoshida, "Crystal structure prediction with machine learning-based element substitution," *Computational Materials Science*, vol. 211, p. 111496, 2022.

[317] L. Wei, N. Fu, E. M. Siriwardane, W. Yang, S. S. Omee, R. Dong, R. Xin, and J. Hu, "TCSP: a Template-Based Crystal Structure Prediction Algorithm for Materials Discovery," *Inorganic Chemistry*, vol. 61, no. 22, pp. 8431–8439, 2022.

[318] S. Fredericks, K. Parrish, D. Sayre, and Q. Zhu, "PyXtal: A Python library for crystal structure generation and symmetry analysis," *Computer Physics Communications*, vol. 261, p. 107810, 2021.

[319] P. Avery and E. Zurek, "RandSpg: An open-source program for generating atomistic crystal structures with specific spacegroups," *Computer Physics Communications*, vol. 213, pp. 208–216, 2017.

[320] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk, "Scaling deep learning for materials discovery," *Nature*, pp. 1–6, 2023.

[321] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[322] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-Tuning Language Models from Human Preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[323] "Illustrating Reinforcement Learning from Human Feedback (RLHF)," https://huggingface.co/blog/rlhf, accessed: 2023-07-05.

[324] S. Kang, W. Jeong, C. Hong, S. Hwang, Y. Yoon, and S. Han, "Accelerated identification of equilibrium structures of multicomponent inorganic crystals using machine learning potentials," *npj Computational Materials*, vol. 8, no. 1, p. 108, 2022.

[325] C. Chen and S. P. Ong, "A universal graph deep learning interatomic potential for the periodic table," *Nature Computational Science*, vol. 2, no. 11, pp. 718–728, 2022.

[326] G. Pausewang and W. Rüdorff, "Über Alkali-oxofluorometallate der Übergangsmetalle. $A'_3MeO_xF_{6-x}$-Verbindungen mit $x = 1, 2, 3$," *Zeitschrift für anorganische und allgemeine Chemie*, vol. 364, no. 1-2, pp. 69–87, 1969.

[327] V. I. Hegde, C. K. Borg, Z. del Rosario, Y. Kim, M. Hutchinson, E. Antono, J. Ling, P. Saxe, J. E. Saal, and B. Meredig, "Quantifying uncertainty in high-throughput density functional theory: A comparison of AFLOW, Materials Project, and OQMD," *Physical Review Materials*, vol. 7, no. 5, p. 053805, 2023.

[328] W. Ye, X. Lei, M. Aykol, and J. H. Montoya, "Novel inorganic crystal structures predicted using autonomous simulation agents," *Scientific Data*, vol. 9, no. 1, p. 302, 2022.

[329] A. Onwuli, K. T. Butler, and A. Walsh, "Ionic species representations for materials informatics," *ChemRxiv*, 2024.

[330] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "RoFormer: Enhanced Transformer with Rotary Position Embedding," *arXiv preprint arXiv:2104.09864*, 2021.

[331] N. Shazeer, "Fast Transformer Decoding: One Write-Head is All You Need," *arXiv preprint arXiv:1911.02150*, 2019.

[332] R. Grau-Crespo, S. Hamad, C. R. A. Catlow, and N. De Leeuw, "Symmetry-adapted configurational modelling of fractional site occupancy in solids," *Journal of Physics: Condensed Matter*, vol. 19, no. 25, p. 256201, 2007.

[333] J. Riebesell, R. E. Goodall, A. Jain, P. Benner, K. A. Persson, and A. A. Lee, "Matbench Discovery–An evaluation framework for machine learning crystal stability prediction," *arXiv preprint arXiv:2308.14920*, 2023.

Appendix A

Supplementary Information for Chapter 3

A.1 Supplementary Notes

1. Comprehensive Results

Supplementary Tables 1 to 10 contain comprehensive results for the experiments described in the chapter, reporting the performance for all utilized combinations of representation type, embedding size, and pooling type. In all experiments, due to intrinsic limitations of the Atom2Vec approach, Atom2Vec vectors [1] could not be created with dimensions greater than the number of atoms being considered. Similarly, one-hot vectors are limited in dimensionality to the number of atoms being considered. Finally, pre-trained Mat2Vec vectors [2] were used, and their dimensionality was limited to 200. All tasks reported utilized the ElemNet feed-forward neural net architecture (consisting of 17 layers), with L2 regularization instead of dropout.

2. Preliminary Results with Structure-based Architectures

The experiments described in the chapter were performed using the ElemNet architecture as a standard (with the exception of the Elpasolite Formation Energy task). The study does not experiment with various different kinds of neural network-based architectures because the aim of the work is to introduce a new (and more accessible and effective) way of learning distributed atom representations, and not a particular combination of representation and architecture, nor to establish a new performance benchmark on a task. Nevertheless, here preliminary results are reported on the use of SkipAtom embeddings with two different structure-based architectures: CGCNN [3] and MEGNet [4]. These results highlight two important points: first, that SkipAtom embeddings are effective in the context of neural network architectures in general (and not only with an ElemNet architecture), and second, that they can improve the performance of models that incorporate structure information.

The CGCNN model is a convolutional graph neural network that operates on datasets that incorporate crystal structure information. It can be used for classification and regression tasks. The paper that introduced the CGCNN model used a dataset of 27,430 compounds from the Materials Project to build a regression model for predicting band gap. The CGCNN paper reports 0.388 eV MAE. They create a 60/20/20 train/validation/test split: they train on 60% and validate on 20% after each epoch; then they pick the best model according to the validation score, and evaluate on the test set. The 0.388 eV MAE is on the test

set. Here, the CGCNN codebase [5] is used to reproduce the results, and to evaluate using SkipAtom vectors as the atom representations. The CGCNN architecture requires that atom representations are provided. By default, a binary feature vector is provided (see [3] for more details). In Supplementary Table A.11, the results of using 200-dimensional SkipAtom embeddings is compared to the results of using the default binary feature vectors.

The MEGNet model [4] consists of a graph neural network that can be used to predict properties of molecules and crystals. It requires that atoms are given a predefined representation. Alternatively, one-hot atom vectors can be provided, and an embedding table is learned during training, which results in learned atom representations. Here, the MEGNet codebase [6] is used to compare the performance of the MEGNet model with (and without) the SkipAtom embeddings on the Elpasolite Formation Energy task. In Supplementary Table A.12, the results of using 200-dimensional SkipAtom embeddings are compared to the results of using the default one-hot vectors, in the context of the Elpasolite Formation Energy prediction task with the MEGNet model. Note that in the chapter an MAE of 0.1089 eV/atom is reported using the original architecture with concatenated atom vectors (that does not include structure information).

3. Derivation of Materials Graphs

As described in the chapter, the SkipAtom approach relies on the conversion of the unit cells of materials to a graph representation. From this graph, atom pairs are derived for training. The graph representing a material can be derived using any approach desired, but in this work, an approach is used which is based on Voronoi decomposition [7], which identifies nearest neighbors using solid angle weights to determine the probability of various coordination environments. Specifically, the *CrystalNN* neighbor finding algorithm was used to construct the graphs [8, 9], as implemented in the *pymatgen* package (version 2021.2.8.1) [10].

A brief description of the *CrystalNN* algorithm is provided here for convenience, but for more details, the reader is referred to the original descriptions [8, 9]. The first step in the algorithm involves the assignment of a multi-component weight to each atom pair in the structure, such that these weights correspond to the likelihood that two atoms are neighbors. The weight consists of various components, including the solid angle obtained from a Voronoi construction based on the crystal structure, a penalty for atoms that are too far apart, and the electronegativity difference between the atoms. The next step involves projecting these multi-component weights onto a quadrant of the unit circle, ordered from largest to smallest weights, and computing the area under the circle between adjacent weights to obtain neighbor likelihoods. Finally, the coordination number with the highest probability for each site is selected.

4. Learning Representations of Atoms in their Oxidation States

As stated in the chapter, one limitation of the SkipAtom approach is that it does not provide representations of atoms in different oxidation states. Since it is (often) possible to unambiguously infer the oxidation states of atoms in compounds, it is, in principle, possible to construct a SkipAtom training set of pairs of atoms in different oxidation states. The number of atom types would increase by several fold, but would still be within limits that allow for efficient training. Here, this is demonstrated by incorporating two additional atom types: Fe(II) and Fe(III). A separate embedding for neutral Fe is continued to be learned.

To learn the representations for Fe(II) and Fe(III), the materials structure database is scanned for compounds containing Fe, and the oxidation state of the element is determined using a *maximum a posteriori* estimation method, as implemented in the *BVAnalyzer* class of

the *pymatgen* package (version 2021.2.8.1) [10]. Pairs are then formed that will be added to the original training set, by keeping only the pairs where Fe(II) or Fe(III) are the target atom (i.e. the atom whose context is predicted). The associated atom in the pair is represented in its neutral state. In total, there were 190,056 pairs generated in this way, and added to the original dataset.

The embeddings were then learned using the SkipAtom approach described in the chapter, together with this enhanced dataset. To evaluate the learned Fe(II) and Fe(III) representations, a qualitative assessment was made by comparing to Zn and Al, since Zn is generally found in its Zn(II) state, and Al is generally found in its Al(III) state. The four embeddings together, Al, Zn, Fe(II), and Fe(III), were subjected to dimensionality reduction using t-SNE, and the results are plotted in Supplementary Figure A.3. It is apparent that Fe(II) resides more closely to Zn, and Fe(III) resides more closely to Al, as one might expect, at least along the first dimension.

5. Analysis of the Number of Embedding Dimensions

Across all the evaluation tasks, the performance of the SkipAtom embeddings appears to increase with the number of embedding dimensions. To better evaluate the influence of the number of embedding dimensions on the performance of the representations, a series of SkipAtom embeddings of different sizes were learned. These embeddings were then mean-pooled for the Refractive Index prediction task, and their performance is given in Supplementary Table A.13. A plot of their performance on the task in given in Supplementary Figure A.4. Also, these embeddings were used for the Elpasolite Formation Energy prediction task. The results are given in Supplementary Table A.14 and Supplementary Figure A.5

6. Analysis of Training Set Size

To analyze the influence of the training dataset size on the quality of the learned embeddings, 200-dimensional SkipAtom embeddings were learned using either all or 25% of the available training data from the Materials Project. The training dataset consisting of 25% of the available pairs was created by randomly sampling from the 15,360,652 pairs derived from the Materials Project, yielding a dataset with 3,840,163 pairs. These 200-dimensional SkipAtom embeddings were mean-pooled for the Refractive Index prediction task, and their performance is given in Supplementary Table A.15.

A.2 Supplementary Tables

Table A.1: Elpasolite Formation Energy prediction results after 10-fold cross-validation. The dataset consists of 5,645 examples. The task and the model were initially described in the paper that introduced Atom2Vec (an alternative approach for learning atom vectors). [1] The target formation energies were obtained by DFT. [11] The mean best formation energy MAE on the test set after 200 epochs of training in each fold is reported. Batch size was 32, learning rate was 0.001. Note that Dim refers to the dimensionality of the atom vector; the size of the input vector is 4 $\times$ Dim. All results were generated using the same procedure on identical train/test folds.

Representation	Dim	MAE (eV/atom)
Atom2Vec	30	0.1477 $\pm$ 0.0078
SkipAtom	*30*	*0.1183 $\pm$ 0.0050*
Random	30	0.1701 $\pm$ 0.0081
Atom2Vec	86	0.1242 $\pm$ 0.0066
One-hot	86	0.1218 $\pm$ 0.0085
SkipAtom	86	0.1126 $\pm$ 0.0078
Random	86	0.1190 $\pm$ 0.0085
Mat2Vec	200	0.1126 $\pm$ 0.0058
SkipAtom	**200**	**0.1089 $\pm$ 0.0061**
Random	200	0.1158 $\pm$ 0.0050

Table A.2: OQMD Dataset Formation Energy prediction results after 10-fold cross-validation. The dataset consists of 275,424 examples. The target values were computed using DFT. [12, 13]. The mean best formation energy MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds.

Representation	Dim	Pooling	MAE (eV/atom)
SkipAtom	86	sum	0.0420 $\pm$ 0.0005
SkipAtom	86	mean	0.0460 $\pm$ 0.0006
SkipAtom	86	max	0.0615 $\pm$ 0.0006
Atom2Vec	86	sum	0.0396 $\pm$ 0.0004
Atom2Vec	86	mean	0.0417 $\pm$ 0.0005
Atom2Vec	86	max	0.0532 $\pm$ 0.0006
Bag-of-Atoms / One-hot	**86**	**sum**	**0.0388 $\pm$ 0.0002**
ElemNet / One-hot	86	mean	0.0427 $\pm$ 0.0007
One-hot	86	max	0.0388 $\pm$ 0.0005
Random	86	sum	0.0440 $\pm$ 0.0004
Random	86	mean	0.0468 $\pm$ 0.0006
Random	86	max	0.0572 $\pm$ 0.0007
Mat2Vec	200	sum	0.0401 $\pm$ 0.0004
Mat2Vec	200	mean	0.0444 $\pm$ 0.0007
Mat2Vec	200	max	0.0501 $\pm$ 0.0006
SkipAtom	200	sum	0.0408 $\pm$ 0.0003
SkipAtom	200	mean	0.0451 $\pm$ 0.0005
SkipAtom	200	max	0.0559 $\pm$ 0.0006
Random	200	sum	0.0417 $\pm$ 0.0004
Random	200	mean	0.0441 $\pm$ 0.0007
Random	200	max	0.0511 $\pm$ 0.0005

Table A.3: Experimental Band Gap prediction results after 2-repeated 5-fold cross-validation. The dataset consists of 4,604 examples. The target values were obtained by experiment. [14]. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an MAE of 0.416 eV (Automatminer). [15] Note that the state-of-the-art result does not make use of structure, and uses composition only.

Representation	**Dim**	**Pooling**	**MAE (eV)**
SkipAtom	*86*	*sum*	*0.3495 ± 0.0020*
SkipAtom	86	mean	0.3737 ± 0.0091
SkipAtom	86	max	0.3954 ± 0.0090
Atom2Vec	86	sum	0.3922 ± 0.0087
Atom2Vec	86	mean	0.4005 ± 0.0080
Atom2Vec	86	max	0.4070 ± 0.0048
Bag-of-Atoms / One-hot	86	sum	0.3797 ± 0.0022
ElemNet / One-hot	86	mean	0.4060 ± 0.0072
One-hot	86	max	0.3823 ± 0.0046
Random	86	sum	0.4109 ± 0.0058
Random	86	mean	0.4286 ± 0.0058
Random	86	max	0.4389 ± 0.0028
Mat2Vec	200	sum	0.3529 ± 0.0007
Mat2Vec	200	mean	0.3886 ± 0.0000
Mat2Vec	200	max	0.3625 ± 0.0070
SkipAtom	**200**	**sum**	**0.3487 ± 0.0085**
SkipAtom	200	mean	0.3737 ± 0.0069
SkipAtom	200	max	0.3985 ± 0.0049
Random	200	sum	0.4058 ± 0.0004
Random	200	mean	0.4181 ± 0.0010
Random	200	max	0.4289 ± 0.0067

Table A.4: Theoretical Band Gap prediction results after 2-repeated 5-fold cross-validation. The dataset consists of 106,113 examples. The target values were obtained by DFT-GGA. [16, 17]. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an MAE of 0.228 eV (CGCNN). [15] Note that the state-of-the-art result makes use of structure.

Representation	**Dim**	**Pooling**	**MAE (eV)**
SkipAtom	86	sum	0.2791 ± 0.0008
SkipAtom	86	mean	0.2807 ± 0.0003
SkipAtom	86	max	0.3512 ± 0.0017
Atom2Vec	86	sum	0.2692 ± 0.0008
Atom2Vec	86	mean	0.2712 ± 0.0025
Atom2Vec	86	max	0.3289 ± 0.0016
Bag-of-Atoms / One-hot	86	sum	0.2611 ± 0.0008
ElemNet / One-hot	**86**	**mean**	**0.2582 ± 0.0003**
One-hot	86	max	0.2603 ± 0.0004
Random	86	sum	0.3238 ± 0.0005
Random	86	mean	0.3180 ± 0.0016
Random	86	max	0.4096 ± 0.0008
Mat2Vec	200	sum	0.2741 ± 0.0002
Mat2Vec	200	mean	0.2744 ± 0.0005
Mat2Vec	200	max	0.3256 ± 0.0002
SkipAtom	*200*	*sum*	*0.2736 ± 0.0008*
SkipAtom	200	mean	0.2753 ± 0.0006
SkipAtom	200	max	0.3351 ± 0.0013
Random	200	sum	0.3083 ± 0.0021
Random	200	mean	0.3095 ± 0.0009
Random	200	max	0.3733 ± 0.0010

Table A.5: Bulk Modulus prediction results after 2-repeated 10-fold cross-validation. The dataset consists of 10,987 examples. The target values were computed using DFT-GGA. [18]. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an MAE of 0.0679 log(GPa) (Automatminer). [15] Note that the state-of-the-art result makes use of structure.

Representation	**Dim**	**Pooling**	**MAE (log(GPa))**
SkipAtom	86	sum	0.0790 ± 0.0002
SkipAtom	*86*	*mean*	*0.0789 ± 0.0002*
SkipAtom	86	max	0.0867 ± 0.0000
Atom2Vec	86	sum	0.0795 ± 0.0005
Atom2Vec	86	mean	0.0810 ± 0.0004
Atom2Vec	86	max	0.0861 ± 0.0002
Bag-of-Atoms / One-hot	86	sum	0.0861 ± 0.0002
ElemNet / One-hot	86	mean	0.0853 ± 0.0001
One-hot	86	max	0.0861 ± 0.0003
Random	86	sum	0.0916 ± 0.0002
Random	86	mean	0.0908 ± 0.0004
Random	86	max	0.0997 ± 0.0001
Mat2Vec	**200**	**sum**	**0.0776 ± 0.0000**
Mat2Vec	200	mean	0.0779 ± 0.0003
Mat2Vec	200	max	0.0813 ± 0.0003
SkipAtom	200	sum	0.0786 ± 0.0003
SkipAtom	200	mean	0.0785 ± 0.0000
SkipAtom	200	max	0.0888 ± 0.0002
Random	200	sum	0.0887 ± 0.0003
Random	200	mean	0.0871 ± 0.0001
Random	200	max	0.0960 ± 0.0004

Table A.6: Shear Modulus prediction results after 2-repeated 10-fold cross-validation. The dataset consists of 10,987 examples. The target values were computed using DFT-GGA. [18]. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an MAE of 0.0849 log(GPa) (Automatminer). [15] Note that the state-of-the-art result makes use of structure.

Representation	Dim	Pooling	MAE (log(GPa))
SkipAtom	*86*	*sum*	*0.1014 ± 0.0001*
SkipAtom	86	mean	0.1025 ± 0.0002
SkipAtom	86	max	0.1102 ± 0.0002
Atom2Vec	86	sum	0.1029 ± 0.0000
Atom2Vec	86	mean	0.1054 ± 0.0000
Atom2Vec	86	max	0.1089 ± 0.0005
Bag-of-Atoms / One-hot	86	sum	0.1137 ± 0.0005
ElemNet / One-hot	86	mean	0.1155 ± 0.0001
One-hot	86	max	0.1140 ± 0.0002
Random	86	sum	0.1195 ± 0.0002
Random	86	mean	0.1199 ± 0.0001
Random	86	max	0.1260 ± 0.0001
Mat2Vec	**200**	**sum**	**0.1014 ± 0.0002**
Mat2Vec	200	mean	0.1035 ± 0.0001
Mat2Vec	200	max	0.1050 ± 0.0000
SkipAtom	**200**	**sum**	**0.1014 ± 0.0000**
SkipAtom	200	mean	0.1024 ± 0.0001
SkipAtom	200	max	0.1111 ± 0.0001
Random	200	sum	0.1167 ± 0.0002
Random	200	mean	0.1163 ± 0.0002
Random	200	max	0.1223 ± 0.0000

Table A.7: Refractive Index prediction results after 2-repeated 5-fold cross-validation. The dataset consists of 4,764 examples. The target values were computed using DFPT-GGA. [19]. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an MAE of 0.299 n (Automatminer). [15] Note that the state-of-the-art result makes use of structure.

Representation	**Dim**	**Pooling**	**MAE (n)**
SkipAtom	86	sum	0.3369 ± 0.0014
SkipAtom	*86*	*mean*	*0.3275 ± 0.0004*
SkipAtom	86	max	0.3561 ± 0.0013
Atom2Vec	86	sum	0.3419 ± 0.0013
Atom2Vec	86	mean	0.3308 ± 0.0016
Atom2Vec	86	max	0.3522 ± 0.0005
Bag-of-Atoms / One-hot	86	sum	0.3576 ± 0.0002
ElemNet / One-hot	86	mean	0.3409 ± 0.0016
One-hot	86	max	0.3547 ± 0.0013
Random	86	sum	0.3625 ± 0.0012
Random	86	mean	0.3593 ± 0.0006
Random	86	max	0.3891 ± 0.0021
Mat2Vec	200	sum	0.3272 ± 0.0004
Mat2Vec	**200**	**mean**	**0.3236 ± 0.0017**
Mat2Vec	200	max	0.3428 ± 0.0004
SkipAtom	200	sum	0.3340 ± 0.0012
SkipAtom	200	mean	0.3247 ± 0.0015
SkipAtom	200	max	0.3618 ± 0.0026
Random	200	sum	0.3598 ± 0.0053
Random	200	mean	0.3543 ± 0.0006
Random	200	max	0.3824 ± 0.0019

Table A.8: Bulk Metallic Glass Formation prediction results after 2-repeated 5-fold stratified cross-validation. The dataset consists of 5,680 examples. The target values were obtained from experiment. [20, 21]. The mean best ROC-AUC on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an ROC-AUC of 0.858 (RF) [15]. Note that the state-of-the-art result does not make use of structure, and uses composition only.

Representation	**Dim**	**Pooling**	**ROC-AUC**
SkipAtom	86	sum	0.9312 ± 0.0007
SkipAtom	*86*	*mean*	*0.9346 ± 0.0010*
SkipAtom	86	max	0.9243 ± 0.0005
Atom2Vec	86	sum	0.9306 ± 0.0026
Atom2Vec	86	mean	0.9316 ± 0.0012
Atom2Vec	86	max	0.9300 ± 0.0008
Bag-of-Atoms / One-hot	86	sum	0.9277 ± 0.0004
ElemNet / One-hot	86	mean	0.9322 ± 0.0014
One-hot	86	max	0.9289 ± 0.0016
Random	86	sum	0.9262 ± 0.0011
Random	86	mean	0.9274 ± 0.0006
Random	86	max	0.9243 ± 0.0020
Mat2Vec	200	sum	0.9280 ± 0.0004
Mat2Vec	200	mean	0.9348 ± 0.0024
Mat2Vec	200	max	0.9253 ± 0.0009
SkipAtom	200	sum	0.9327 ± 0.0022
SkipAtom	**200**	**mean**	**0.9349 ± 0.0019**
SkipAtom	200	max	0.9268 ± 0.0002
Random	200	sum	0.9274 ± 0.0019
Random	200	mean	0.9302 ± 0.0016
Random	200	max	0.9298 ± 0.0009

Table A.9: Experimental Metallicity prediction results after 2-repeated 5-fold stratified cross-validation. The dataset consists of 4,921 examples. The target values were obtained from experiment. [14]. The mean best ROC-AUC on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an ROC-AUC of 0.917 (Random Forest). [15] Note that the state-of-the-art result does not make use of structure, and uses composition only.

Representation	**Dim**	**Pooling**	**ROC-AUC**
SkipAtom	*86*	*sum*	*0.9645 ± 0.0012*
SkipAtom	86	mean	0.9575 ± 0.0003
SkipAtom	86	max	0.9561 ± 0.0020
Atom2Vec	86	sum	0.9582 ± 0.0008
Atom2Vec	86	mean	0.9541 ± 0.0005
Atom2Vec	86	max	0.9548 ± 0.0006
Bag-of-Atoms / One-hot	86	sum	0.9600 ± 0.0012
ElemNet / One-hot	86	mean	0.9485 ± 0.0007
One-hot	86	max	0.9599 ± 0.0014
Random	86	sum	0.9559 ± 0.0021
Random	86	mean	0.9460 ± 0.0008
Random	86	max	0.9426 ± 0.0037
Mat2Vec	**200**	**sum**	**0.9655 ± 0.0014**
Mat2Vec	200	mean	0.9570 ± 0.0008
Mat2Vec	200	max	0.9634 ± 0.0013
SkipAtom	200	sum	0.9645 ± 0.0008
SkipAtom	200	mean	0.9572 ± 0.0008
SkipAtom	200	max	0.9589 ± 0.0010
Random	200	sum	0.9541 ± 0.0002
Random	200	mean	0.9454 ± 0.0001
Random	200	max	0.9508 ± 0.0011

Table A.10: Theoretical Metallicity prediction results after 2-repeated 5-fold stratified cross-validation. The dataset consists of 106,113 examples. The target values were obtained by DFT-GGA. [16, 17]. The mean best ROC-AUC on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds. The reported state-of-the-art result is an ROC-AUC of 0.977 (MEGNet). [15] Note that the state-of-the-art result makes use of structure. Notable is that the CGCNN model in the same study achieves an ROC-AUC of 0.954, also using structure, which is comparable to the performance of the Mat2Vec representation, which uses only composition.

Representation	Dim	Pooling	ROC-AUC
SkipAtom	86	sum	0.9520 ± 0.0002
SkipAtom	86	mean	0.9506 ± 0.0000
SkipAtom	86	max	0.9440 ± 0.0000
Atom2Vec	*86*	*sum*	*0.9526 ± 0.0001*
Atom2Vec	86	mean	0.9506 ± 0.0001
Atom2Vec	86	max	0.9450 ± 0.0003
Bag-of-Atoms / One-hot	86	sum	0.9490 ± 0.0002
ElemNet / One-hot	86	mean	0.9477 ± 0.0001
One-hot	86	max	0.9487 ± 0.0003
Random	86	sum	0.9444 ± 0.0000
Random	86	mean	0.9433 ± 0.0002
Random	86	max	0.9330 ± 0.0001
Mat2Vec	**200**	**sum**	**0.9528 ± 0.0002**
Mat2Vec	200	mean	0.9517 ± 0.0001
Mat2Vec	200	max	0.9469 ± 0.0005
SkipAtom	200	sum	0.9524 ± 0.0001
SkipAtom	200	mean	0.9507 ± 0.0001
SkipAtom	200	max	0.9454 ± 0.0001
Random	200	sum	0.9453 ± 0.0002
Random	200	mean	0.9441 ± 0.0001
Random	200	max	0.9380 ± 0.0000

Table A.11: Band gap prediction results on the test set of 27,430 compounds from the Materials Project, split 60/20/20, using the CGCNN model. Training was performed for 100 epochs, a learning rate of 0.01 was used, along with a batch size of 256. The default settings provided by library were used for the other hyperparameters.

Input Representation	MAE (eV)
CGCNN binary feature vector	0.381
SkipAtom 200-dim	**0.371**

Table A.12: Elpasolite formation energy prediction results with the MEGNet architecture. This model incorporates crystal structure.

Input Representation	MAE (eV/atom)
one-hot atom vectors + embedding table	0.0685
SkipAtom 200-dim	**0.0568**

Table A.13: Refractive Index prediction results after 2-repeated 5-fold cross-validation using mean-pooled SkipAtom embeddings of various dimensions. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds.

Dim	MAE (n)
30	0.3278 ± 0.0008
86	0.3262 ± 0.0002
200	0.3248 ± 0.0015
300	0.3252 ± 0.0005
400	0.3267 ± 0.0017
800	0.3263 ± 0.0000

Table A.14: Elpasolite Formation Energy prediction results after 10-fold cross-validation using SkipAtom embeddings of various dimensions. The mean best MAE on the test set after 200 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds.

Dim	MAE (eV/atom)
30	0.1183 ± 0.0050
86	0.1126 ± 0.0078
200	0.1089 ± 0.0061
300	0.1082 ± 0.0053
400	0.1085 ± 0.0029
800	0.1056 ± 0.0034

Table A.15: Refractive Index prediction results after 2-repeated 5-fold cross-validation using 200-dim mean-pooled SkipAtom embeddings learned with different amounts of training data. The mean best MAE on the test set after 100 epochs of training in each fold is reported. All results were generated using the same procedure on identical train/test folds.

Dim	% of training data	MAE (n)
200	25	0.3256 ± 0.0003
200	100	0.3248 ± 0.0015

A.3 Supplementary Figures

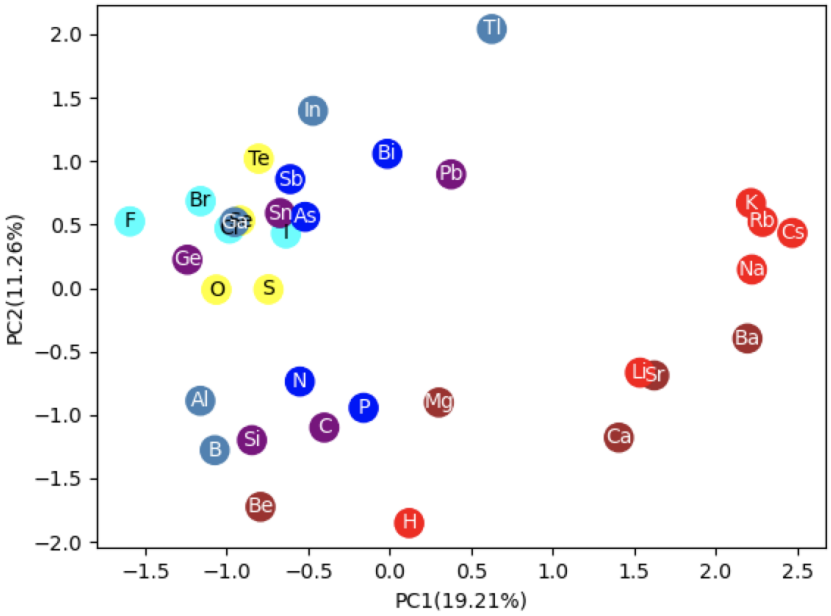

Figure A.1: Principal Component Analysis of SkipAtom Representations. The first two principal components of the SkipAtom 200-dim vectors for 34 atoms are depicted.

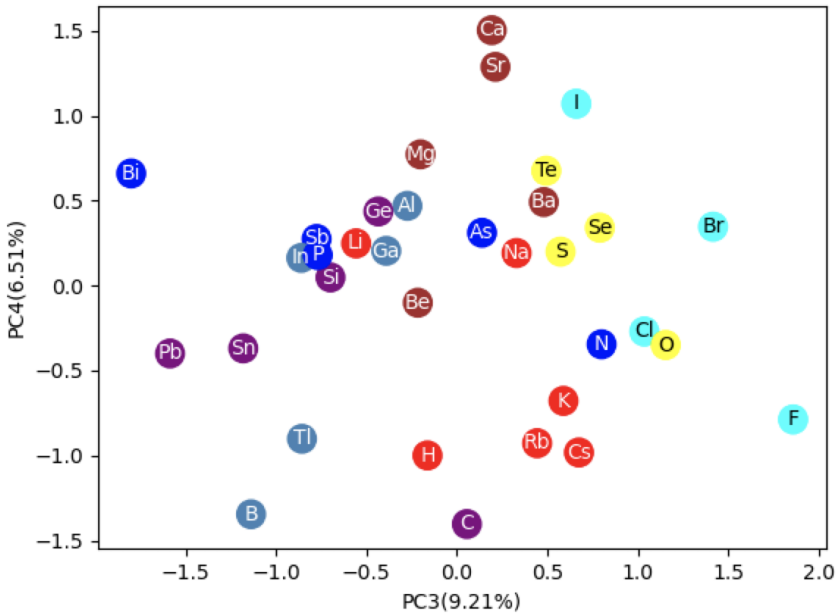

Figure A.2: Principal Component Analysis of SkipAtom Representations. The third and fourth principal components of the SkipAtom 200-dim vectors for 34 atoms are depicted.

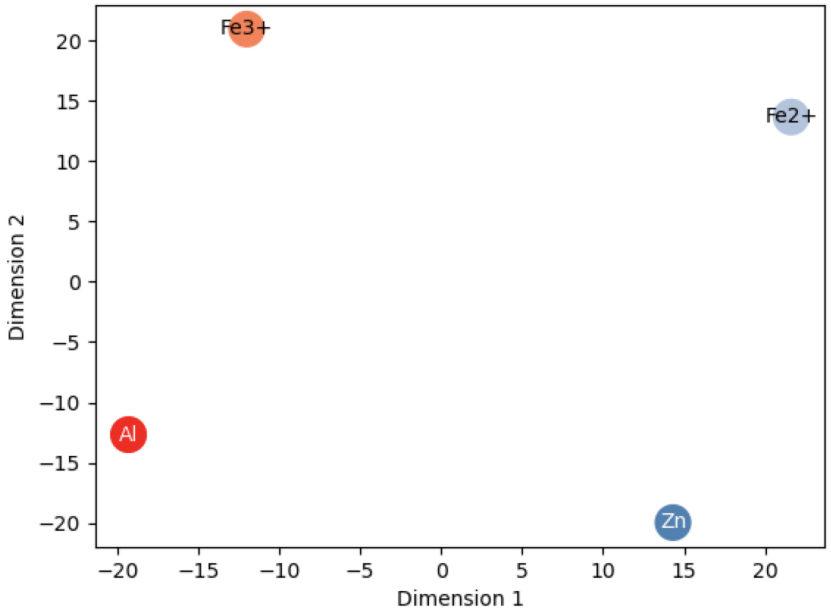

Figure A.3: Dimensionally reduced SkipAtom vectors for Al and Zn, and for Fe(II) and Fe(III). The vectors were reduced from 200 dimensions to 2 dimensions using t-SNE.

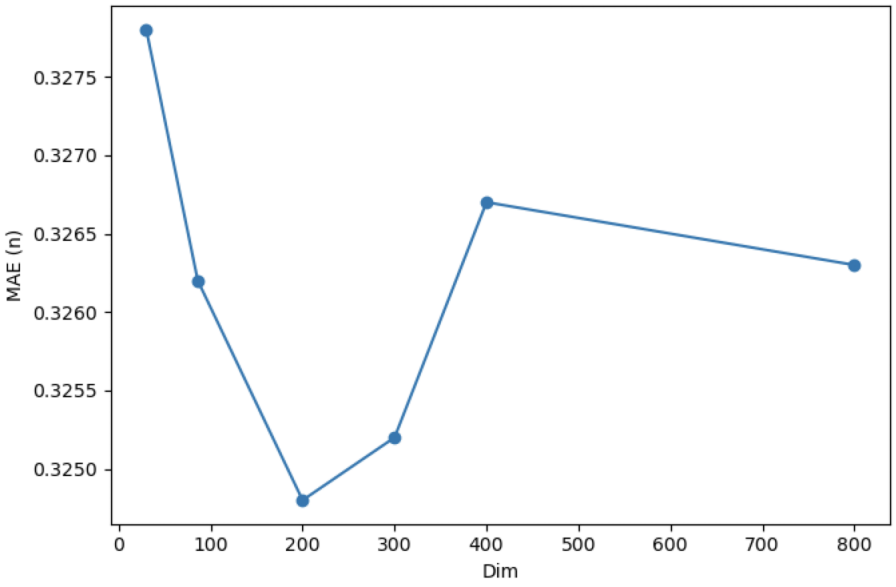

Figure A.4: A plot of MAE results for the Refractive Index prediction task, obtained using 2-repeated 5-fold cross-validation, for a number of different embedding sizes. The SkipAtom embeddings were mean-pooled.

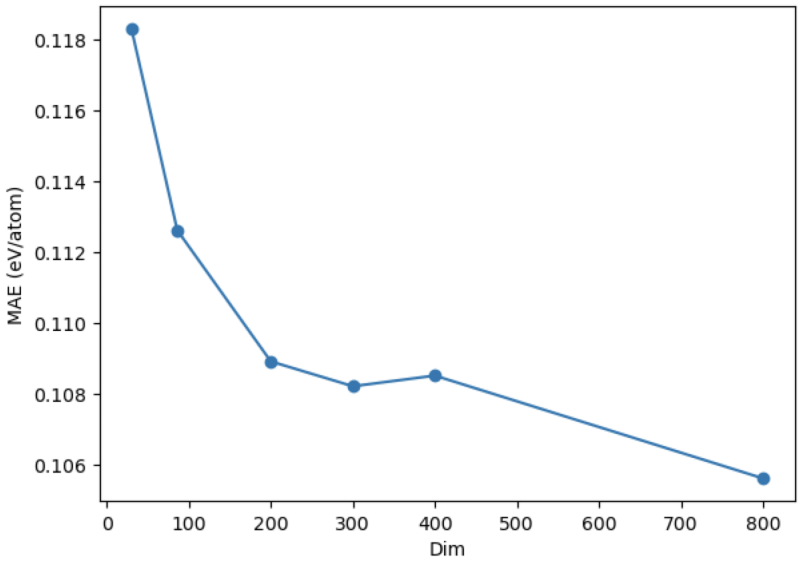

Figure A.5: A plot of MAE results for the Elpasolite Formation Energy prediction task, obtained using 10-fold cross-validation, for a number of different embedding sizes. The SkipAtom embeddings were concatenated.

Appendix A References

[1] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, and S.-C. Zhang, "Learning atoms for materials discovery," *Proceedings of the National Academy of Sciences*, vol. 115, no. 28, pp. E6411–E6417, 2018.

[2] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, "Unsupervised word embeddings capture latent knowledge from materials science literature," *Nature*, vol. 571, no. 7763, pp. 95–98, 2019.

[3] T. Xie and J. C. Grossman, "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties," *Phys. Rev. Lett.*, vol. 120, no. 14, p. 145301, 2018.

[4] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, "Graph networks as a universal machine learning framework for molecules and crystals," *Chemistry of Materials*, vol. 31, no. 9, pp. 3564–3572, 2019.

[5] "CGCNN Library," https://github.com/txie-93/cgcnn.

[6] "MEGNet Library," https://github.com/materialsvirtuallab/megnet.

[7] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites." *J. Reine Angew. Math.*, vol. 1908, pp. 97 – 102, 1908.

[8] N. E. Zimmermann and A. Jain, "Local structure order parameters and site fingerprints for quantification of coordination environment and crystal structure similarity," *RSC Adv.*, vol. 10, no. 10, pp. 6063–6081, 2020.

[9] H. Pan, A. M. Ganose, M. Horton, M. Aykol, K. A. Persson, N. E. Zimmermann, and A. Jain, "Benchmarking Coordination Number Prediction Algorithms on Inorganic Crystal Structures," *Inorg. Chem.*, vol. 60, no. 3, pp. 1590–1603, 2021.

[10] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis," *Comput. Mater. Sci.*, vol. 68, pp. 314–319, 2013.

[11] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, and R. Armiento, "Machine Learning Energies of 2 Million Elpasolite (ABC2D6) Crystals," *Phys. Rev. Lett.*, vol. 117, no. 13, p. 135502, 2016.

[12] D. Jha, L. Ward, A. Paul, W.-k. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, "ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition," *Sci. Rep.*, vol. 8, no. 1, pp. 1–13, 2018.

[13] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, "Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD)," *JOM*, vol. 65, no. 11, pp. 1501–1509, 2013.

[14] Y. Zhuo, A. Mansouri Tehrani, and J. Brgoch, "Predicting the Band Gaps of Inorganic Solids by Machine Learning," *J. Phys. Chem. Lett.*, vol. 9, no. 7, pp. 1668–1673, 2018.

[15] A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain, "Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm," *npj Comput. Mater.*, vol. 6, no. 1, pp. 1–10, 2020.

[16] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder *et al.*, "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation," *APL Mater.*, vol. 1, no. 1, p. 011002, 2013.

[17] S. P. Ong, S. Cholia, A. Jain, M. Brafman, D. Gunter, G. Ceder, and K. A. Persson, "The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles," *Comput. Mater. Sci.*, vol. 97, pp. 209–215, 2015.

[18] M. De Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C. K. Ande, S. Van Der Zwaag, J. J. Plata *et al.*, "Charting the complete elastic properties of inorganic crystalline compounds," *Sci. Data*, vol. 2, no. 1, pp. 1–13, 2015.

[19] I. Petousis, D. Mrdjenovich, E. Ballouz, M. Liu, D. Winston, W. Chen, T. Graf, T. D. Schladt, K. A. Persson, and F. B. Prinz, "High-throughput screening of inorganic compounds for the discovery of novel dielectric and optical materials," *Sci. Data*, vol. 4, no. 1, pp. 1–12, 2017.

[20] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Comput. Mater.*, vol. 2, no. 1, pp. 1–7, 2016.

[21] Y. Kawazoe, "Nonequilibrium Phase Diagrams of Ternary Amorphous Alloys," *LB: New Ser., Group III: Condensed*, vol. 37, pp. 1–295, 1997.

Appendix B

Supplementary Information for Chapter 4

B.1 Supplementary Notes

1. Descriptor Comparison

The use of various descriptors was evaluated with the Random Forest and CraTENet models. To evaluate a model with a particular descriptor, a 90-10 holdout experiment was performed using the Ricci database, focusing only on the p-type Seebeck entries at 600K and 10^{20} cm^{-3} doping. For the Random Forest model, two descriptors were evaluated: the Meredig descriptor [1] and a sum-pooled 200-dimensional SkipAtom representation [2]. For the CraTENet model, two atom descriptors were evaluated: a SkipAtom 200-dimensional atom descriptor, and a binary descriptor described in the original CGCNN work [3]. Each model is thus evaluated with a local descriptor (i.e. the Meredig material descriptor or the binary atom descriptor) and a distributed representation (i.e. the sum-pooled SkipAtom material representation or the SkipAtom atomic representation). The results are given in Supplementary Table B.3.

2. Robust L1 and Robust L2 Loss Comparison

The performance of the CraTENet model was evaluated when trained to minimize either the Robust L1 loss or the Robust L2 loss. For this comparison, a single-head output model was used that makes predictions across 13 different temperatures, for a fixed doping type and doping level. Specifically, models were created that predict either S or $\log \sigma$, at 13 different temperatures, for either n- or p-type doping, at a level of 10^{20} cm^{-3}. The results are given in Supplementary Table B.4.

B.2 Supplementary Tables

Table B.1: Results of 90-10 holdout experiments using the CraTENet+gap model, consisting of 1, 2, or 3 output heads. Identical train-test splits were used throughout. Each number represents either the MAE or R^2 on the held-out test set of the dataset formed from the Ricci database, across all temperatures, doping levels, and doping types. The 1-head entry refers to 3 separate models, each with a single output head, for each of S, $\log\sigma$, and $\log PF$. The 2-head model consisted of outputs only for S and $\log\sigma$, and thus results for $\log PF$ are not reported. The 3-head model consisted of a separate output head for each of S, $\log\sigma$, and $\log PF$. Bold values represent the best result for a given metric and transport property.

	S		$\log\sigma$		$\log PF$	
# output heads	MAE (μV/K)	R^2	MAE	R^2	MAE	R^2
1	51.595	0.960	0.269	0.964	0.384	0.727
2	50.333	0.961	0.259	**0.968**	-	-
3	**49.718**	**0.962**	**0.257**	**0.968**	**0.372**	**0.738**

Table B.2: Mean relative predicted variance for the predictions of the various transport properties made by both the CraTENet and CraTENet+gap models, for the MP-excluding-Ricci dataset. Each model makes 54,816 $\times$ 130 predictions for a given transport property (i.e. for different temperatures, doping levels and doping types). Here, for each prediction, the associated predicted variance is divided by the absolute value of the mean (i.e. the predicted value of the property), to obtain the relative predicted variance. The numbers in the table represent the mean of the relative predicted variances.

	S	$\log\sigma$	$\log PF$
CraTENet	4.8456	0.0075	0.0114
CraTENet+gap	0.0928	0.0003	0.0032

Table B.3: Results of 90-10 holdout experiments for the p-type Seebeck entries at 600K and 10^{20} cm^{-3} doping of the Ricci database. Identical train-test splits were used throughout. Band gap is not provided to the models. The CraTENet model consisted of a single output head only. Bold results represent the best results for a model type.

Model	Descriptor	R^2	MAE (μV/K)
Random Forest	Meredig	**0.643**	**74**
Random Forest	sum-pooled SkipAtom 200-dim	0.551	92
CraTENet	SkipAtom 200-dim	**0.587**	**69**
CraTENet	CGCNN binary atom vector	0.556	71

Table B.4: Results for CraTENet models with a single output head that produce predictions for 13 temperatures, for a given thermoelectric transport property, at a doping level of $10^{20}\ \mathrm{cm}^{-3}$, and for a specific doping type. The models were trained using either the Robust L1 or Robust L2 loss. Each column represents either the MAE or the R^2 across all temperatures and across all members of the test set of a 90-10 holdout experiment. Identical train-test splits were used throughout. MAE values for tasks involving prediction of S are in units of μV/K. Bold values represent the best result for a loss for a particular metric.

	Robust L1		Robust L2	
Task	MAE	R^2	MAE	R^2
p-type S	67	0.592	**66**	**0.612**
n-type S	64	0.478	**62**	**0.503**
p-type $\log\sigma$	0.411	0.750	**0.402**	**0.764**
n-type $\log\sigma$	0.388	0.711	**0.379**	**0.723**
p-type S + gap	35	0.914	35	0.914
n-type S + gap	38	0.831	38	**0.832**
p-type $\log\sigma$ + gap	**0.272**	0.896	0.275	**0.898**
n-type $\log\sigma$ + gap	0.274	0.860	**0.269**	**0.865**

B.3 Supplementary Figures

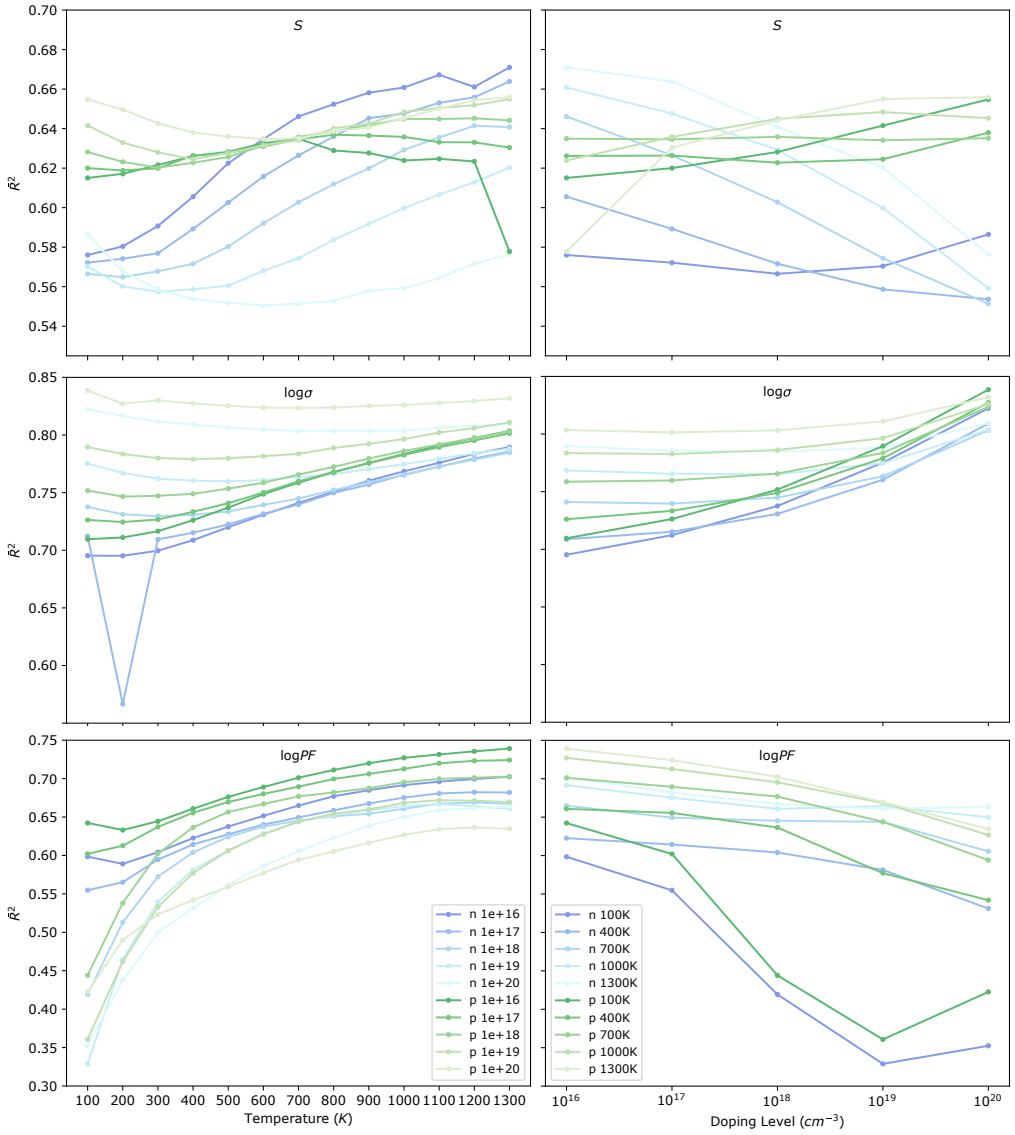

Figure B.1: Plots of the agreement between the CraTENet and CraTENet+gap models, measured in terms of R^2, as a function of temperature and doping level, for each of the transport properties. Each point represents the average R^2 for the predictions made on the test set by the CraTENet and CraTENet+gap models, over the folds of 10-fold cross-validation experiments (each experiment utilized identical splits).

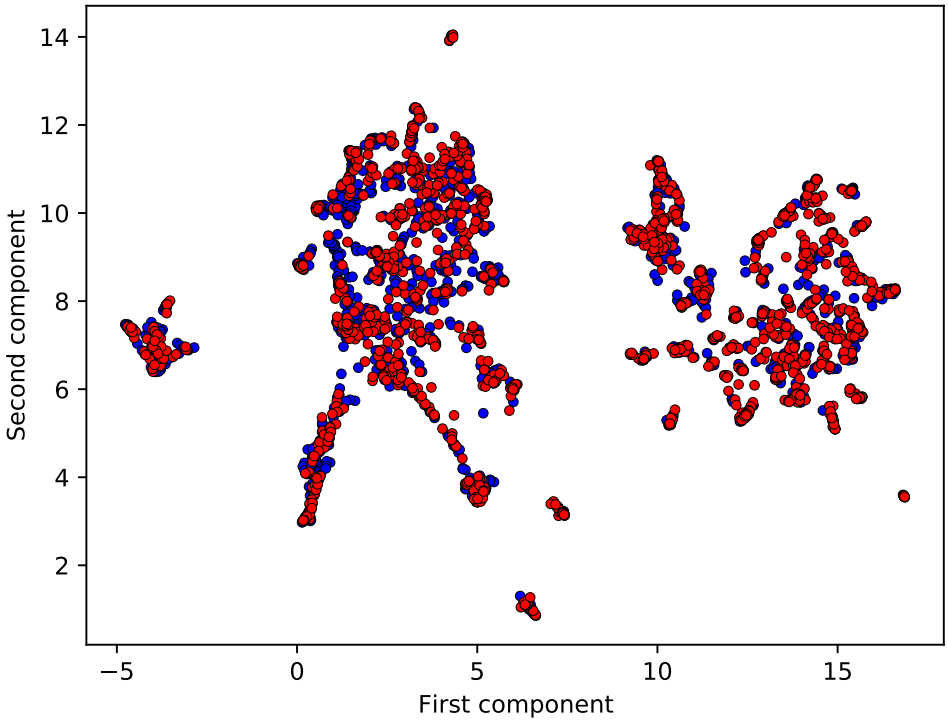

Figure B.2: A plot of dimensionally-reduced 200-dimensional mean-pooled SkipAtom compound vectors for compounds from the Materials Project that are not in the Ricci database (red) and from the Ricci database (blue). UMAP [4] with the Minkowski metric was used to reduce the compound vectors to 2 dimensions. The plot contains a random sampling of 3,000 compounds from each of the datasets. The plot supports the claim that the distributions of compounds in the datasets are similar.

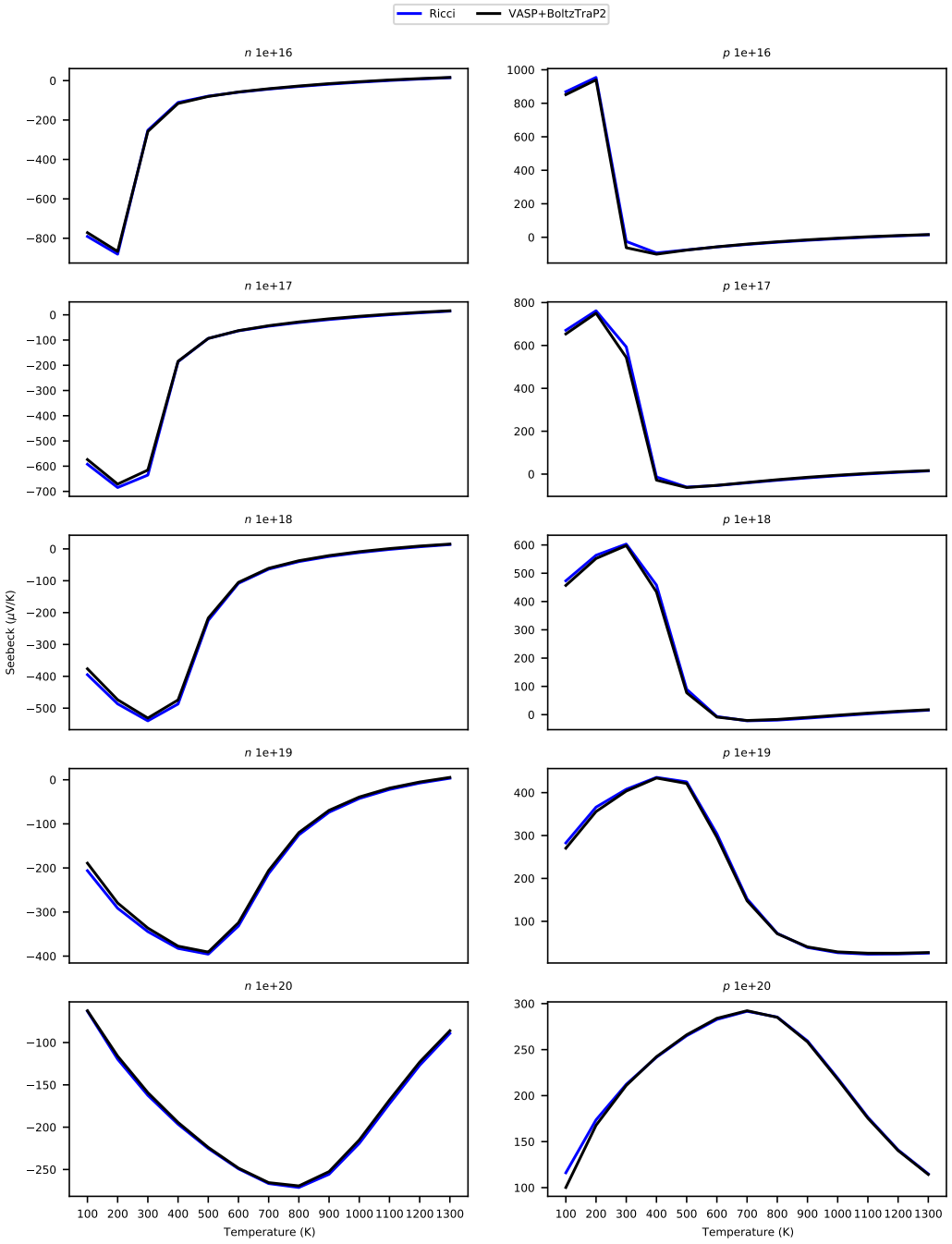

Figure B.3: Plots comparing the Ricci database values for the Seebeck (y-axis, μV/K) to those produced by our *ab initio* approach, for the compound HoSbPd (mp-567418). These results demonstrate that our *ab initio* procedure emulates the one that was used to create the Ricci database. (Similar results were obtained for the electrical conductivity.)

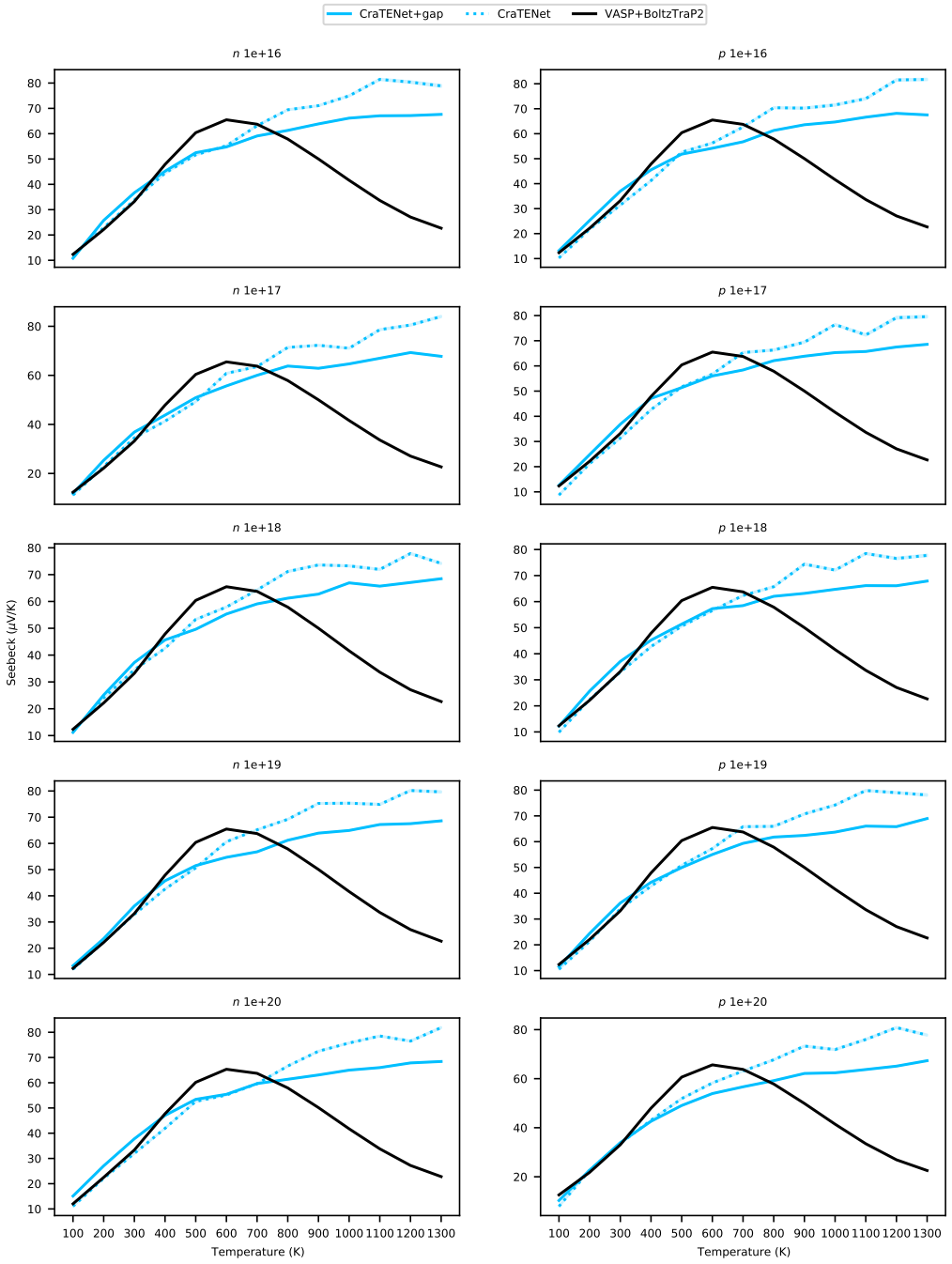

Figure B.4: Plots of the Seebeck values for CeSbSe (mp-1103153) as predicted by the CraTENet models and by the *ab initio* procedure. The band gap value used, 0.0 eV, was obtained from the Materials Project.

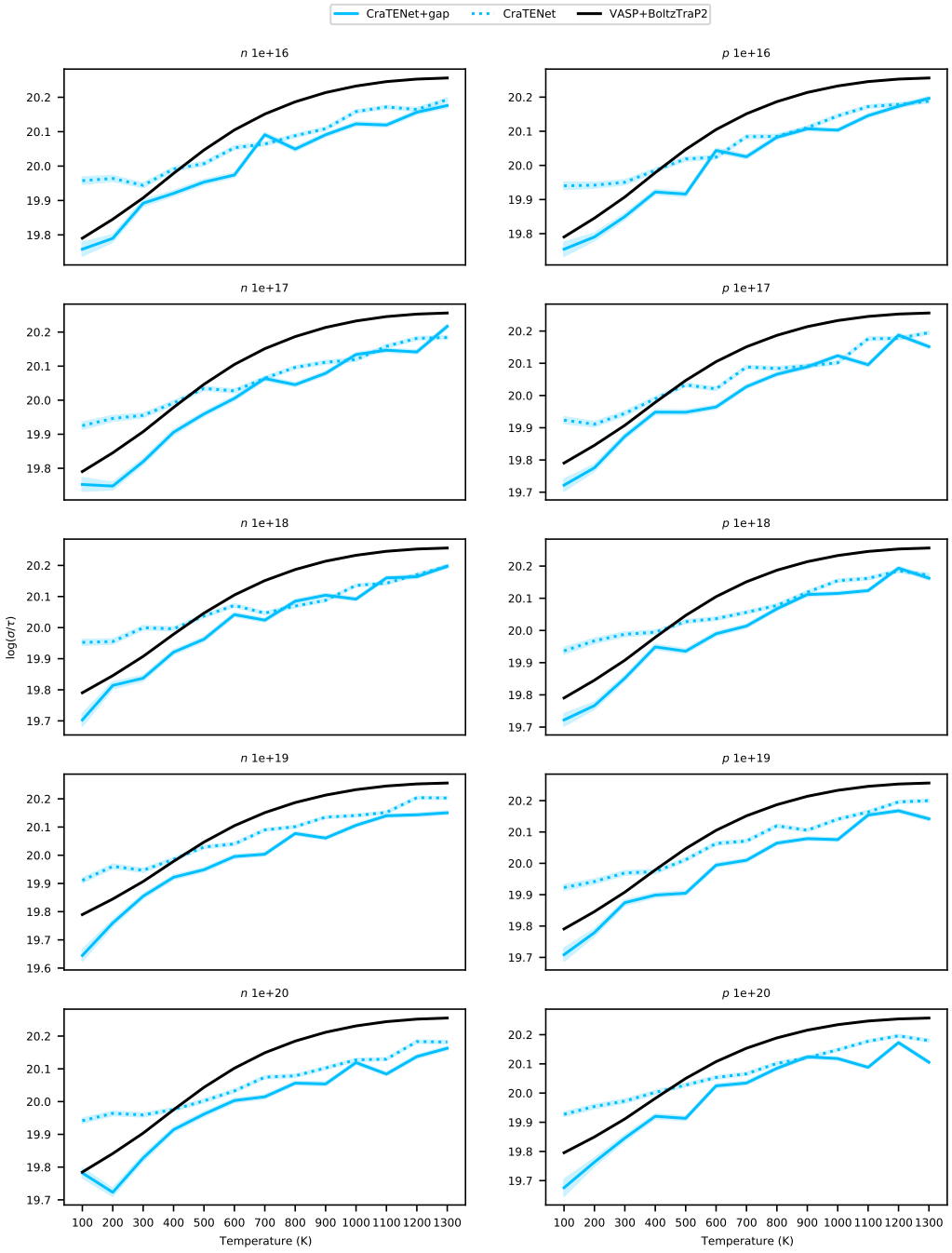

Figure B.5: Plots of the $\log \sigma$ values for CeSbSe (mp-1103153) as predicted by the CraTENet models and by the *ab initio* procedure. The band gap value used, 0.0 eV, was obtained from the Materials Project. The shaded regions represent the $\pm$ standard deviation (i.e. the square root of the predicted variance).

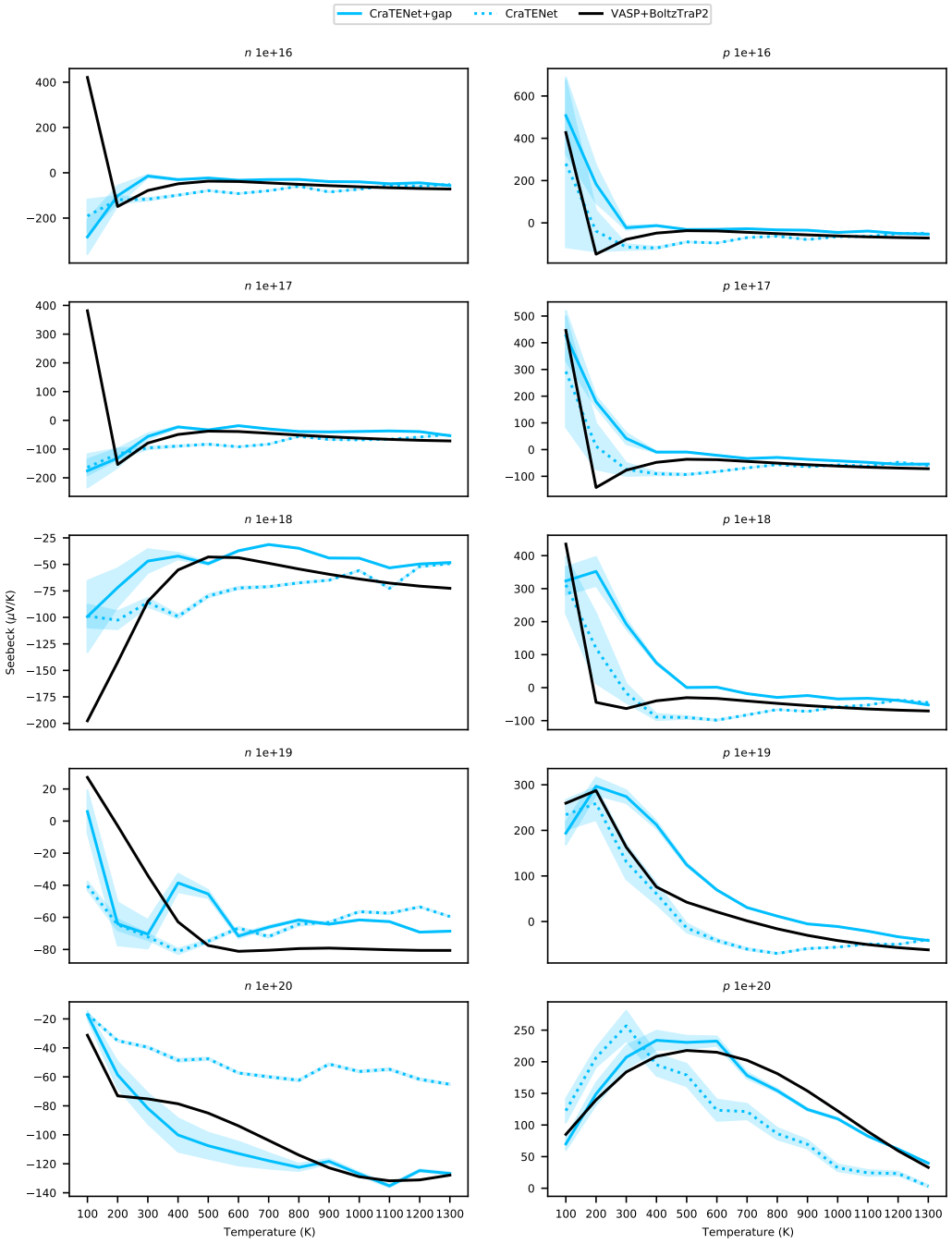

Figure B.6: Plots of the Seebeck values for InCuTeSe (mp-1224187) as predicted by the CraTENet models and by the *ab initio* procedure. The band gap value used, 0.164 eV, was obtained from the Materials Project. The shaded regions represent the $\pm$ standard deviation (i.e. the square root of the predicted variance).

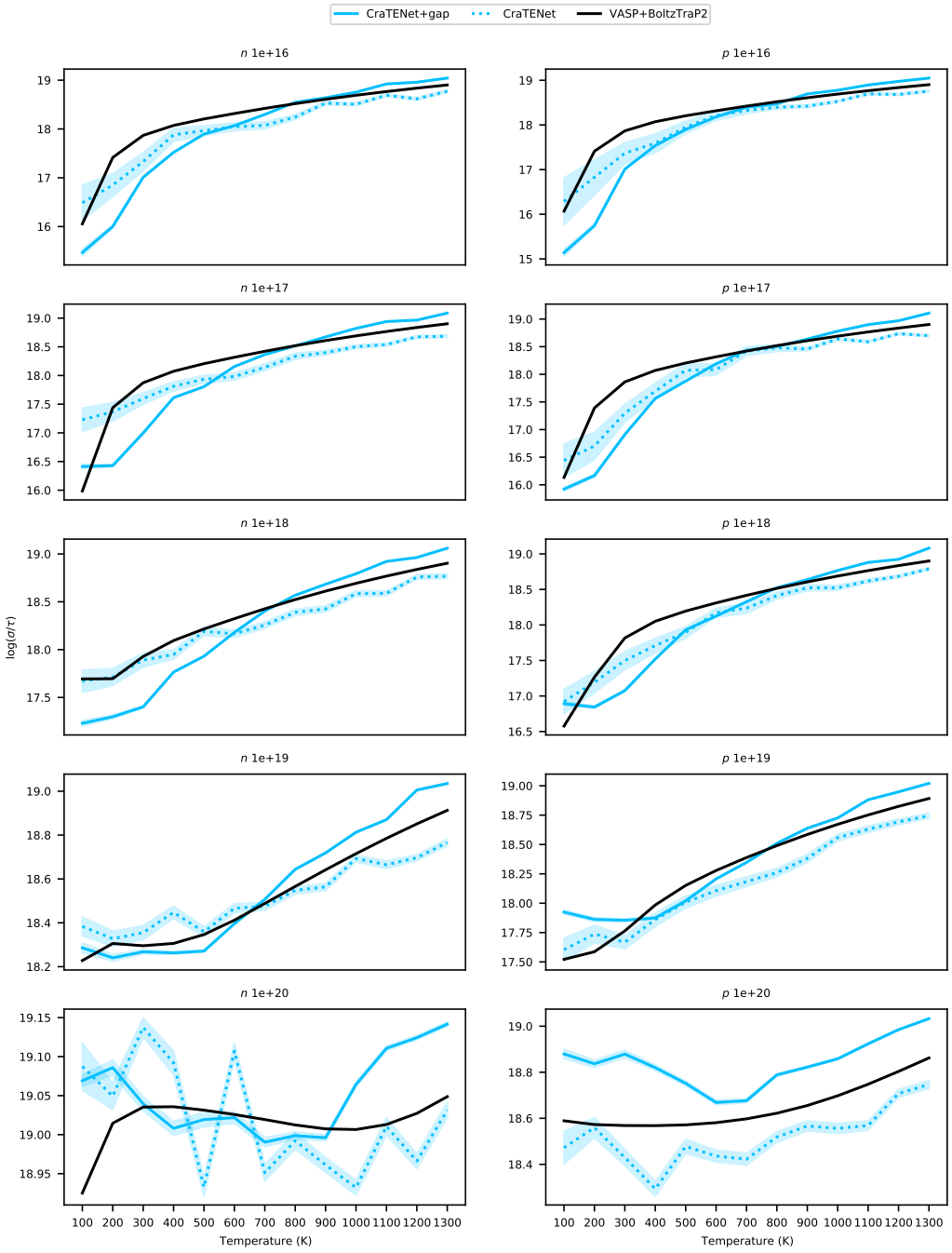

Figure B.7: Plots of the $\log \sigma$ values for InCuTeSe (mp-1224187) as predicted by the CraTENet models and by the *ab initio* procedure. The band gap value used, 0.164 eV, was obtained from the Materials Project. The shaded regions represent the $\pm$ standard deviation (i.e. the square root of the predicted variance).

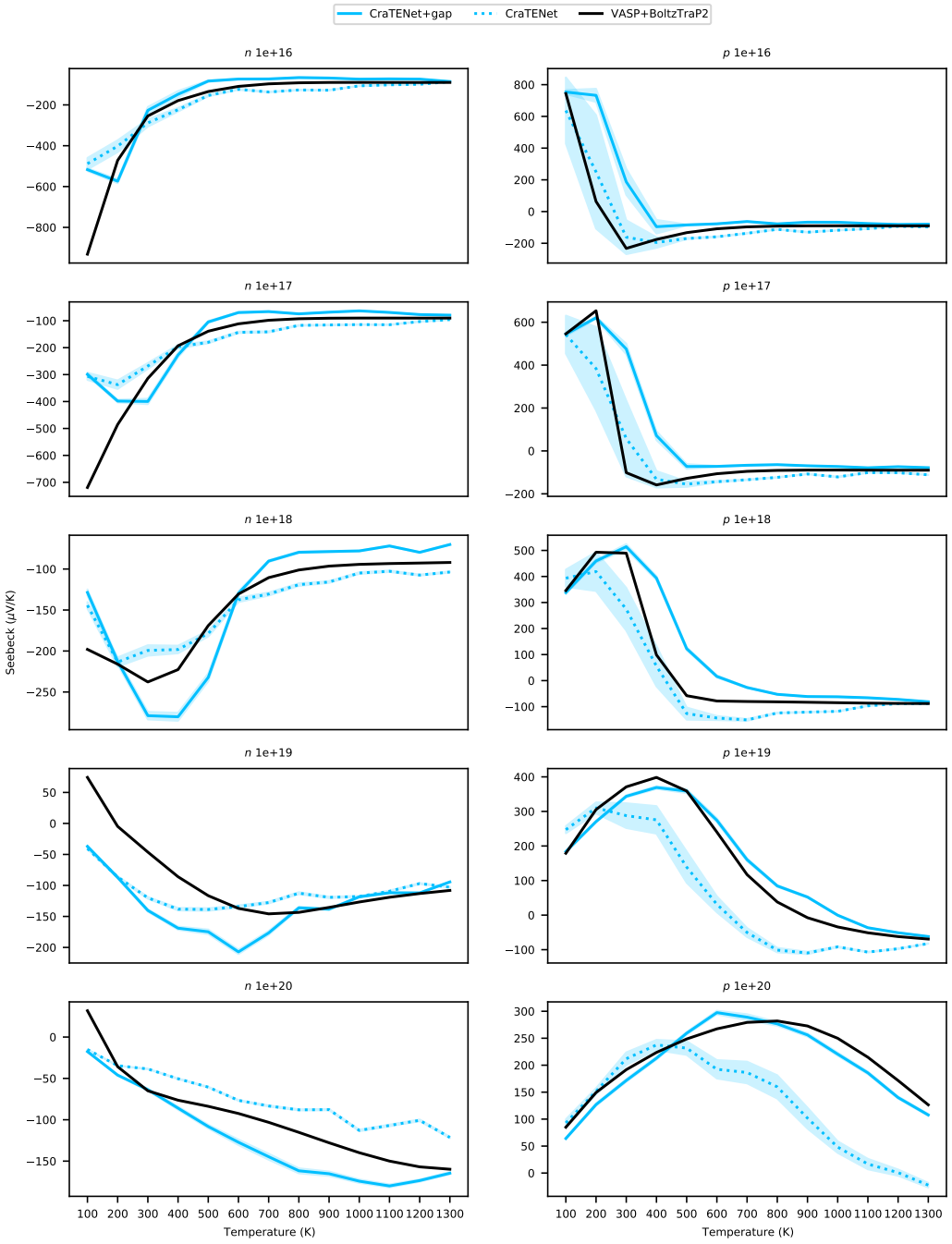

Figure B.8: Plots of the Seebeck values for GaCuTeSe (mp-1224994) as predicted by the CraTENet models and by the *ab initio* procedure. The band gap value used, 0.387 eV, was obtained from the Materials Project. The shaded regions represent the $\pm$ standard deviation (i.e. the square root of the predicted variance).

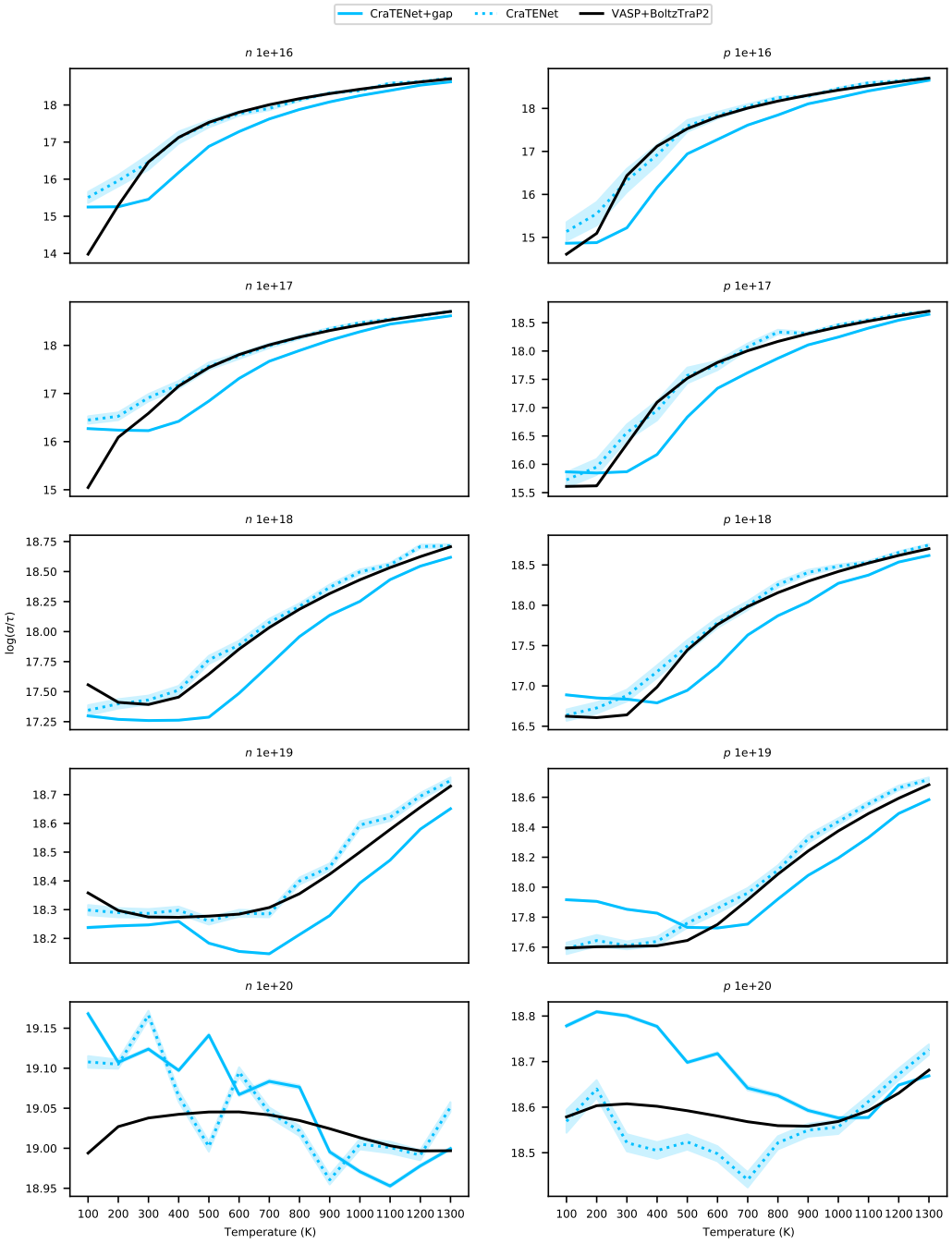

Figure B.9: Plots of the $\log \sigma$ values for GaCuTeSe (mp-1224994) as predicted by the CraTENet models and by the *ab initio* procedure. The band gap value used, 0.387 eV, was obtained from the Materials Project. The shaded regions represent the $\pm$ standard deviation (i.e. the square root of the predicted variance).

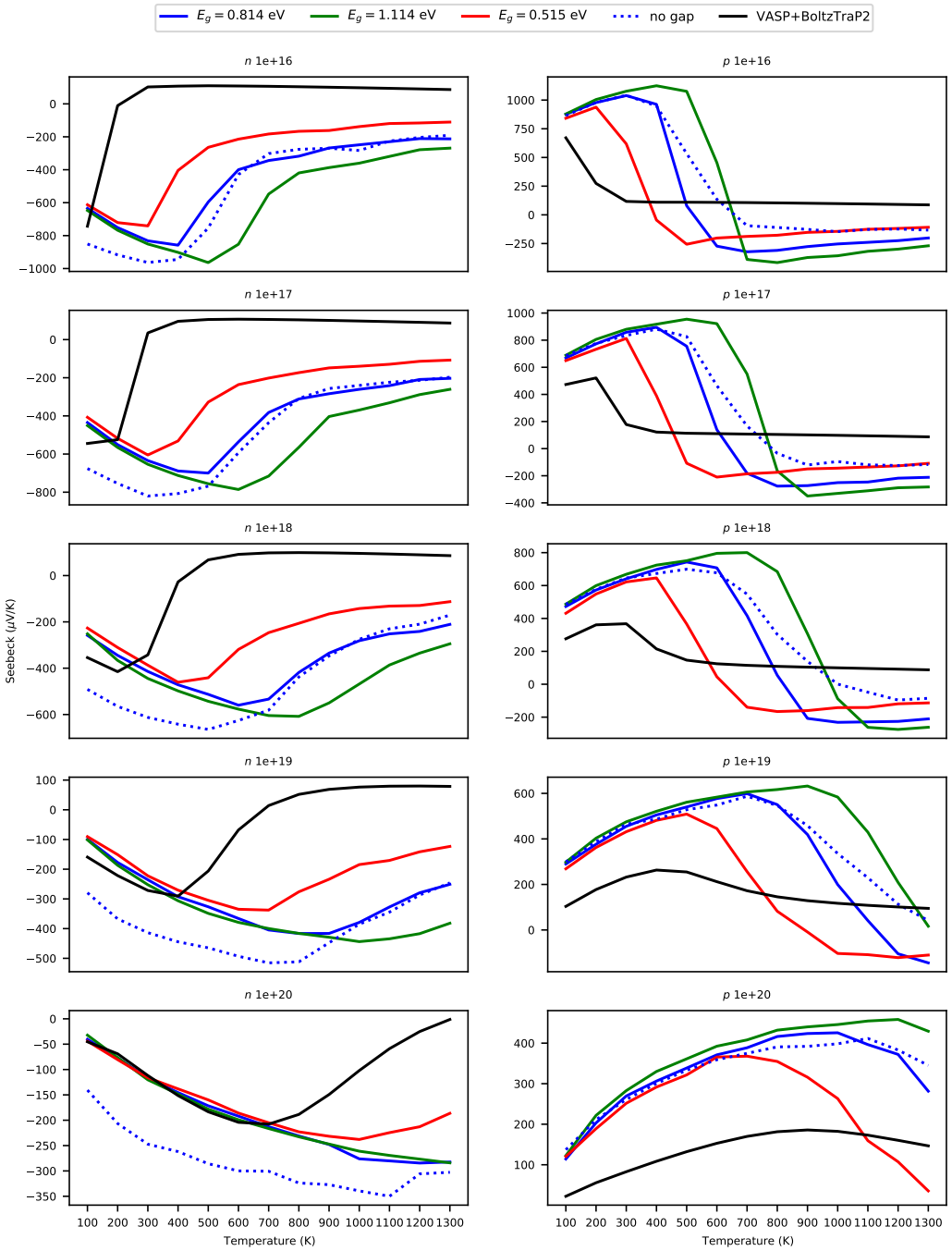

Figure B.10: Plots of the Seebeck values for $NaTlSe_2$ (oqmd-1482315) as predicted by the CraTENet models and by the *ab initio* procedure. Predicted band gap values were used: blue represents the initial prediction, green represents the prediction plus the predicted standard deviation, and red represents the prediction minus the predicted standard deviation (i.e. square root of the predicted variance).

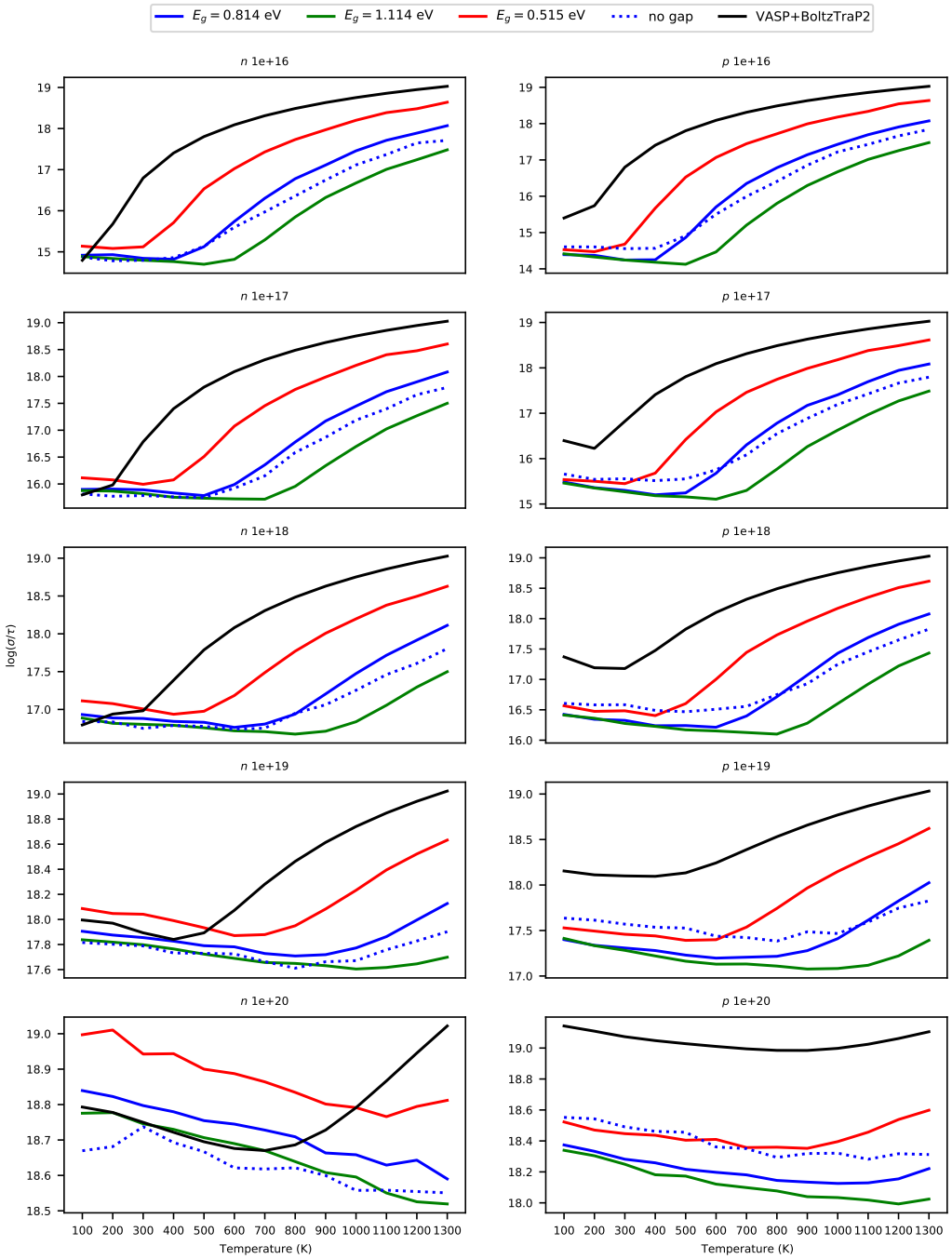

Figure B.11: Plots of the $\log \sigma$ values for $NaTlSe_2$ (oqmd-1482315) as predicted by the CraTENet models and by the *ab initio* procedure. Predicted band gap values were used: blue represents the initial prediction, green represents the prediction plus the predicted standard deviation, and red represents the prediction minus the predicted standard deviation (i.e. square root of the predicted variance).

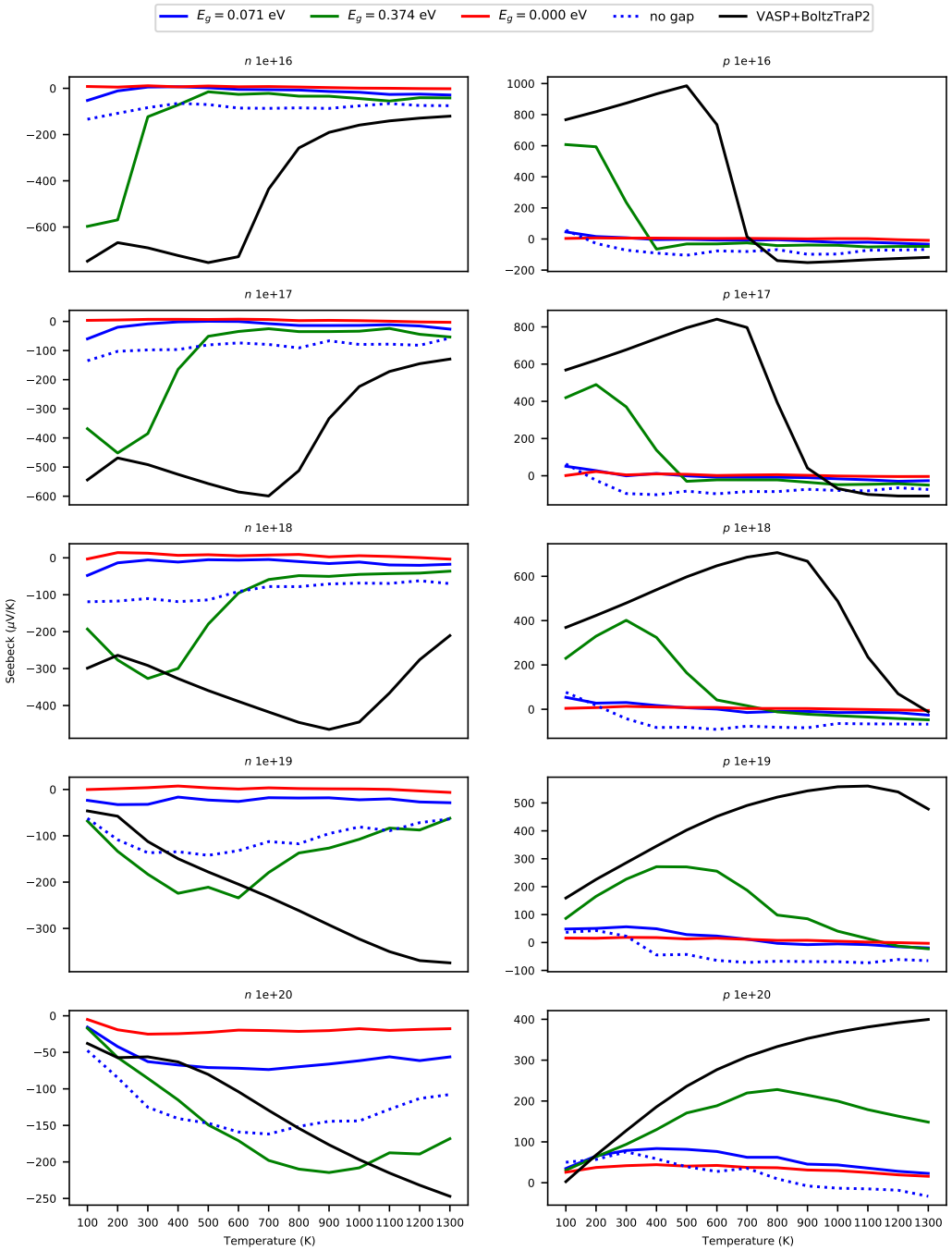

Figure B.12: Plots of the Seebeck values for $LiBiSe_2$ (oqmd-1442673) as predicted by the CraTENet models and by the *ab initio* procedure. Predicted band gap values were used: blue represents the initial prediction, green represents the prediction plus the predicted standard deviation, and red represents the prediction minus the predicted standard deviation (i.e. square root of the predicted variance).

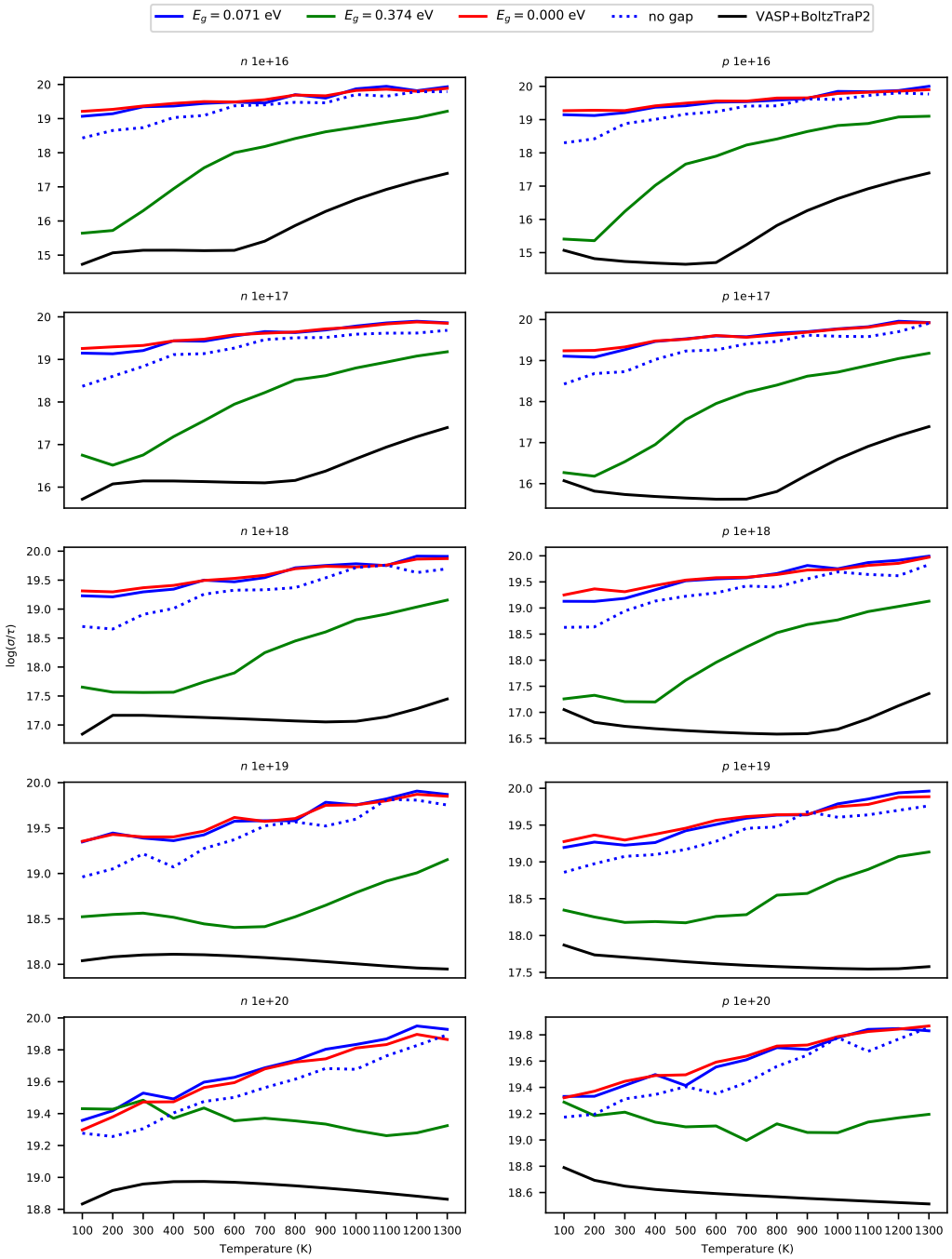

Figure B.13: Plots of the $\log \sigma$ values for $LiBiSe_2$ (oqmd-1442673) as predicted by the CraTENet models and by the *ab initio* procedure. Predicted band gap values were used: blue represents the initial prediction, green represents the prediction plus the predicted standard deviation, and red represents the prediction minus the predicted standard deviation (i.e. square root of the predicted variance).

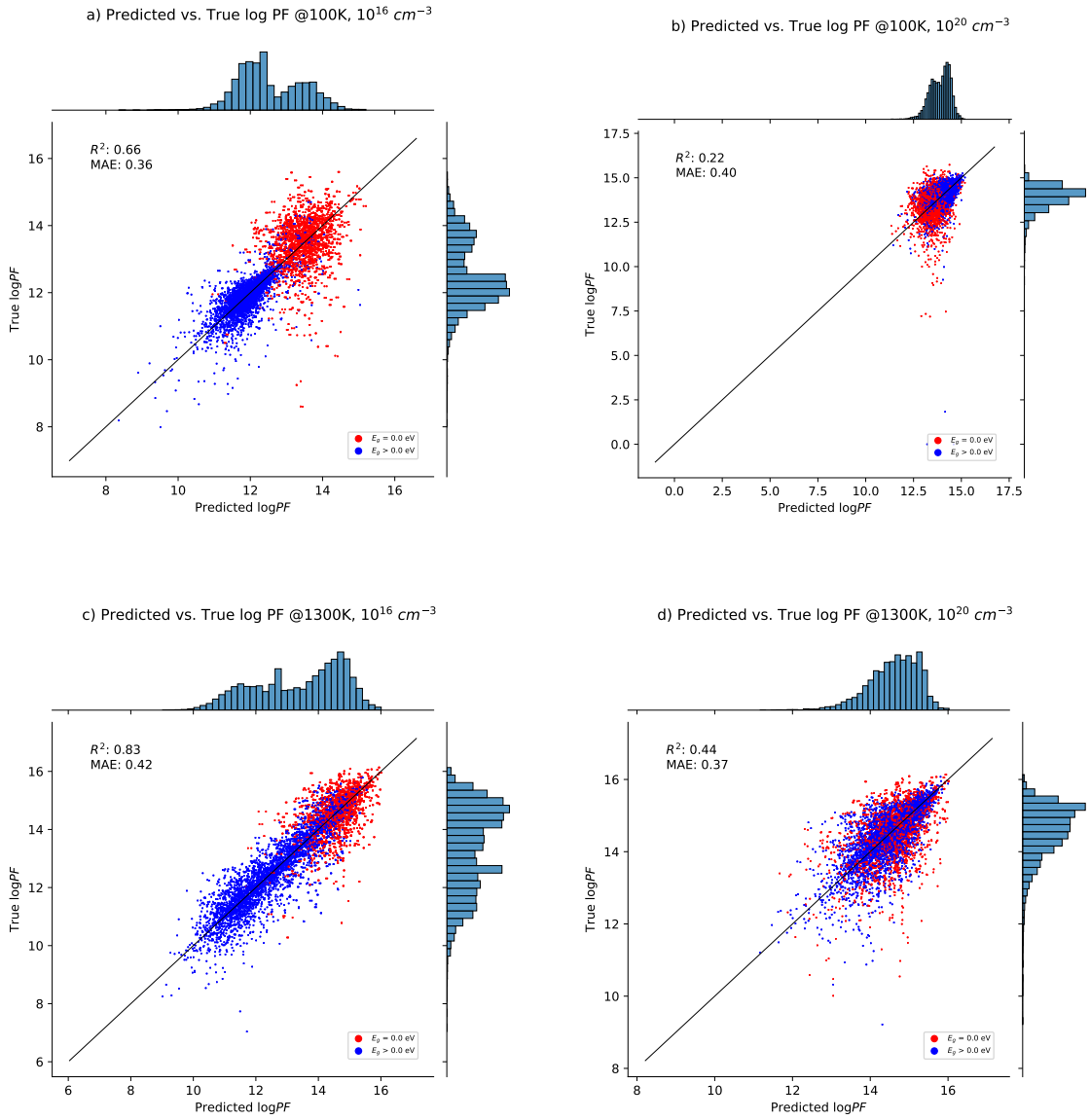

Figure B.14: Plots of the true vs. predicted $\log PF$ values from the test set of a 90-10 holdout experiment using the CraTENet+gap model, for various combinations of temperature and doping level, across both doping types. The bars on top and right of each frame are histograms representing the data distribution. Different colours are used to represent metallic (red) and finite-gap (blue) compounds. At low temperatures and low doping levels, compounds with an electronic band gap do not have enough electronic conductivity to make a high PF, because they have few extrinsic (due to doping) or intrinsic (due to temperature) carriers. Only when more carriers are introduced, can the semiconducting systems compete with the metals in terms of overall PF.

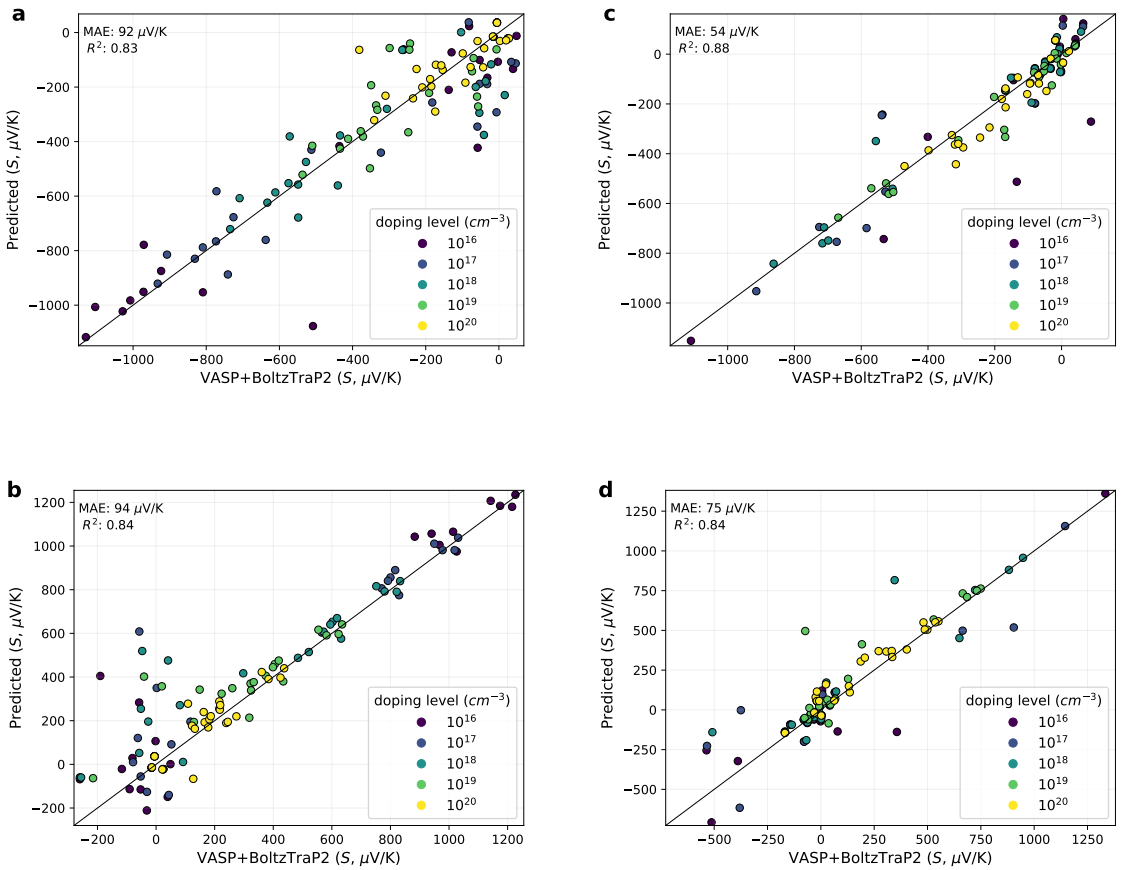

Figure B.15: Seebeck coefficients at various temperatures predicted with CraTENet+gap *vs.* those computed using the *ab initio* approach, for 23 Materials Projects compounds not found in the Ricci database, with **a)** n-type doping at 400 K, **b)** p-type doping at 400 K, **c)** n-type doping at 1300 K, and **d)** p-type doping at 1300 K. Each point represents a particular compound at a particular doping level (*e.g.* SbTeIr at $10^{20}\mathrm{cm}^{-3}$).

Appendix B References

[1] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton, "Combinatorial screening for new materials in unconstrained composition space with machine learning," *Phys. Rev. B*, vol. 89, no. 9, p. 094104, 2014.

[2] L. M. Antunes, R. Grau-Crespo, and K. T. Butler, "Distributed representations of atoms and materials for machine learning," *npj Computational Materials*, vol. 8, no. 1, p. 44, Mar 2022.

[3] T. Xie and J. C. Grossman, "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties," *Phys. Rev. Lett.*, vol. 120, no. 14, p. 145301, 2018.

[4] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv preprint arXiv:1802.03426*, 2018.

Appendix C

Supplementary Information for Chapter 5

C.1 Supplementary Notes

1. CIF Syntax Standardization and Tokenization

The CIF format is flexible in terms of the sequence of tags in the file. Moreover, not all tags are required to be present in the file. While a large language model could, in principle, learn to process variable arrangements of the tags, the CIF file syntax was restricted, such that every CIF file in the dataset is structured identically. Furthermore, several tags were added that are not part of the CIF specification.

To ensure consistency, and enhance the model's ability to learn from the data, the CIF files were standardized using a sequence of pre-processing steps. The steps were designed to not only normalize the format of the CIF files, but also to incorporate additional information beneficial for the model's training. The pre-processing steps are as follows:

1. Each structure in the dataset was first converted into a pymatgen `Structure` object.
2. The pymatgen `CifWriter` class was used to create CIF files from the `Structure` objects, using a `symprec` value of 0.1.
3. In the created CIF files, the content of the `data_` tag was replaced, which contains the reduced formula, with the cell composition of the structure. The atoms of the cell composition appended to `data_` are sorted by electronegativity.
4. The symmetry operators were removed from the CIF files.
5. A custom block in the CIF files was introduced to include specific atomic properties, namely, the electronegativity, the radius, and ionic radius. These properties are not part of the standard CIF specification.
6. All numerical values in the CIF files were rounded to four decimal places.

An example of a CIF file from the training dataset, both before and after it has been pre-processed, is given below:

```
data_TePb
_symmetry_space_group_name_H-M    Pmma
_cell_length_a    5.64400000
```

```
_cell_length_b    4.00120000
_cell_length_c    5.68070000
_cell_angle_alpha    90.00000000
_cell_angle_beta    90.00000000
_cell_angle_gamma    90.00000000
_symmetry_Int_Tables_number    51
_chemical_formula_structural    TePb
_chemical_formula_sum    'Te2 Pb2'
_cell_volume    128.28595744
_cell_formula_units_Z    2
loop_
 _symmetry_equiv_pos_site_id
 _symmetry_equiv_pos_as_xyz
  1  'x, y, z'
  2  '-x, -y, -z'
  3  '-x+1/2, -y, z'
  4  'x+1/2, y, -z'
  5  'x+1/2, -y, -z'
  6  '-x+1/2, y, z'
  7  '-x, y, -z'
  8  'x, -y, z'
loop_
 _atom_site_type_symbol
 _atom_site_label
 _atom_site_symmetry_multiplicity
 _atom_site_fract_x
 _atom_site_fract_y
 _atom_site_fract_z
 _atom_site_occupancy
  Te  Te0  2  0.25000000  0.50000000  0.73570000  1.0
  Pb  Pb1  2  0.25000000  0.00000000  0.26910000  1.0
```

Original CIF file for PbTe (Z=2, *Pmma*), before pre-processing

```
data_Te2Pb2
loop_
_atom_type_symbol
_atom_type_electronegativity
_atom_type_radius
_atom_type_ionic_radius
Te 2.1000 1.4000 1.2933
Pb 2.3300 1.8000 1.1225
_symmetry_space_group_name_H-M Pmma
_cell_length_a 5.6440
_cell_length_b 4.0012
_cell_length_c 5.6807
_cell_angle_alpha 90.0000
_cell_angle_beta 90.0000
_cell_angle_gamma 90.0000
_symmetry_Int_Tables_number 51
_chemical_formula_structural TePb
_chemical_formula_sum 'Te2 Pb2'
_cell_volume 128.2864
_cell_formula_units_Z 2
loop_
_symmetry_equiv_pos_site_id
_symmetry_equiv_pos_as_xyz
1 'x, y, z'
loop_
_atom_site_type_symbol
_atom_site_label
```

```
_atom_site_symmetry_multiplicity
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_occupancy
Te Te0 2 0.2500 0.5000 0.7357 1
Pb Pb1 2 0.2500 0.0000 0.2691 1
```

Pre-processed CIF file for PbTe (Z=2, *Pmma*)

After the pre-processing step, the CIF files are tokenized; that is, a CIF file is parsed and converted into a sequence of numbers, where each number represents a particular token. Tokenization is necessary for converting the structured, text-based data of CIF files into a format that the model can process. The selection of tokens is guided by a custom vocabulary. The vocabulary defines the distinct, irreducible elements of the CIF file syntax that are relevant to the problem. In constructing this vocabulary, numeric digits, atomic symbols, space group symbols, and CIF tags were represented with distinct tokens. Specifically, the vocabulary consists of digits: `0 1 2 3 4 5 6 7 8 9`, as well as various symbols: `x y z . ( ) ' , ␣` (space) `\n` (newline). A complete enumeration of the supported atom, CIF tag, and space group symbols follows:

```
Ac Ag Al Ar As Au B Ba Be Bi Br C Ca Cd Ce Cl Co Cr Cs Cu Dy Er Eu F Fe Ga
Gd Ge H He Hf Hg Ho I In Ir K Kr La Li Lu Mg Mn Mo N Na Nb Nd Ne Ni Np O
Os P Pa Pb Pd Pm Pr Pt Pu Rb Re Rh Ru S Sb Sc Se Si Sm Sn Sr Ta Tb Tc Te
Th Ti Tl Tm U V W Xe Y Yb Zn Zr
```

Supported atom tokens.

```
_cell_length_b                  _atom_site_occupancy
_atom_site_attached_hydrogens   _cell_length_a
_cell_angle_beta                _symmetry_equiv_pos_as_xyz
_cell_angle_gamma               _atom_site_fract_x
_symmetry_space_group_name_H-M  _symmetry_Int_Tables_number
_chemical_formula_structural    _chemical_name_systematic
_atom_site_fract_y              _atom_site_symmetry_multiplicity
_chemical_formula_sum           _atom_site_label
_atom_site_type_symbol          _cell_length_c
_atom_site_B_iso_or_equiv       _symmetry_equiv_pos_site_id
_cell_volume                    _atom_site_fract_z
_cell_angle_alpha               _cell_formula_units_Z
loop_                           data_
_atom_type_symbol               _atom_type_electronegativity *
_atom_type_radius *             _atom_type_ionic_radius *
_atom_type_oxidation_number
```

Supported CIF tag tokens. Tags with * do not exist in the official CIF specification.

```
Aea2        Aem2        Ama2        Amm2        C2          C2/c
C2/m        C222        C222_1      Cc          Ccc2        Ccce
Cccm        Cm          Cmc2_1      Cmce        Cmcm        Cmm2
Cmme        Cmmm        F-43c       F-43m       F222        F23
F432        F4_132      Fd-3        Fd-3c       Fd-3m       Fdd2
Fddd        Fm-3        Fm-3c       Fm-3m       Fmm2        Fmmm
I-4         I-42d       I-42m       I-43d       I-43m       I-4c2
I-4m2       I222        I23         I2_12_12_1  I2_13       I4
I4/m        I4/mcm      I4/mmm      I422        I432        I4_1
I4_1/a      I4_1/acd    I4_1/amd    I4_122      I4_132      I4_1cd
I4_1md      I4cm        I4mm        Ia-3        Ia-3d       Iba2
Ibam        Ibca        Im-3        Im-3m       Ima2        Imm2
```

Imma	Immm	P-1	P-3	P-31c	P-31m
P-3c1	P-3m1	P-4	P-42_1c	P-42_1m	P-42c
P-42m	P-43m	P-43n	P-4b2	P-4c2	P-4m2
P-4n2	P-6	P-62c	P-62m	P-6c2	P-6m2
P1	P2	P2/c	P2/m	P222	P222_1
P23	P2_1	P2_1/c	P2_1/m	P2_12_12	P2_12_12_1
P2_13	P3	P312	P31c	P31m	P321
P3_1	P3_112	P3_121	P3_2	P3_212	P3_221
P3c1	P3m1	P4	P4/m	P4/mbm	P4/mcc
P4/mmm	P4/mnc	P4/n	P4/nbm	P4/ncc	P4/nmm
P4/nnc	P422	P42_12	P4_1	P4_122	P4_12_12
P4_132	P4_2	P4_2/m	P4_2/mbc	P4_2/mcm	P4_2/mmc
P4_2/mnm	P4_2/n	P4_2/nbc	P4_2/ncm	P4_2/nmc	P4_2/nnm
P4_22_12	P4_232	P4_2bc	P4_2cm	P4_2mc	P4_2nm
P4_3	P4_322	P4_32_12	P4_332	P4bm	P4cc
P4mm	P4nc	P6/m	P6/mcc	P6/mmm	P622
P6_1	P6_122	P6_2	P6_222	P6_3	P6_3/m
P6_3/mcm	P6_3/mmc	P6_322	P6_3cm	P6_3mc	P6_4
P6_422	P6_5	P6_522	P6cc	P6mm	Pa-3
Pba2	Pbam	Pban	Pbca	Pbcm	Pbcn
Pc	Pca2_1	Pcc2	Pcca	Pccm	Pccn
Pm	Pm-3	Pm-3m	Pm-3n	Pma2	Pmc2_1
Pmm2	Pmma	Pmmm	Pmmn	Pmn2_1	Pmna
Pn-3	Pn-3m	Pn-3n	Pna2_1	Pnc2	Pnma
Pnn2	Pnna	Pnnm	Pnnn	R-3	R-3c
R-3m	R3	R32	R3c	R3m	

Supported space group tokens.

The atom tokens cover all 89 atom types present in the training data. Atoms with atomic number $Z \geq 84$ (Po) are thus excluded (except for the early actinides Ac, Th, Pa, U, Np, and Pu, which did appear in crystal structures in the databases).

The 227 space group symbols also cover all space groups present in the training data. The space groups *$P4_222$* (no. 93), *P6* (no. 168), and *P432* (no. 207) are not supported as there are no structures in the training data with these space groups. These three space groups are known to occur very rarely, due to a combination of symmetries (and absences of symmetries) that are difficult to realise in a crystal geometry. [1] For example, the space group *P6* requires the presence of a six-fold rotation axis but without the presence of mirror planes and inversion centres that occur in other hexagonal groups. Note that the rarity of these space groups is not limited to the ab initio databases used in the study. The ICSD database, which lists crystallographic information for most currently known materials and minerals, does not contain any experimentally-determined ordered inorganic compounds in these three space groups (only three experimental inorganic crystal structures are listed for these space groups: $K_2Ta_4O_9F_4$ and $MoCu_2Al_{7.92}$ for space group *P6*, and $Rb(NO_3)$ for space group *P432*, but they all exhibit site-occupancy disorder).

After the model has generated a sequence of tokens representing a CIF file, a post-processing step is performed in which the custom `loop_` section with atomic properties is removed, and the symmetry equivalent site IDs and positions which match the printed space group are introduced.

2. Model Architecture and Generative Pre-training

The generative pre-training step consists of training a GPT-style transformer model autoregressively. The implementation is based on the nanoGPT project [2]. The model consists of a series of transformer blocks, each consisting of multi-head self-attention and a feed-forward

neural network. The input to the model is a sequence of token indices representing the token sequence. The tokens are embedded using a learned embedding table. The token embeddings are combined with learned positional embeddings, to which dropout is applied. The result is passed through a series of transformer blocks. A transformer block consists of causal self-attention [3] and a feed-forward network containing a non-linear layer with GELU activation [4], and dropout. A linear output layer transforms the features produced by the transformer blocks into a vector of logits. A softmax operation is applied to convert the logits into the probabilities of the tokens of the vocabulary, for each position in the output sequence. Weight tying [5] is used: the output layer and the input embedding layer share the same weights. The objective is to minimize the cross-entropy loss between the predicted probability distribution over the vocabulary and the actual next token in the sequence, for all the tokens in the sequence.

Training consists of iteratively sampling sequences from the dataset, performing a forward-pass through the model, computing the loss, and backpropagating the error. The AdamW optimizer [6] is used, and a cosine decay schedule is applied to the learning rate, from 10^{-3} to 10^{-4}, over the course of training. Gradients were clipped to have a norm of at most 1.0. During each training iteration, 40 gradient accumulation steps were performed, and in each step a batch of 32 sequences was randomly sampled. The dataset consists of a single list of all tokens from all CIF files, concatenated together, and the beginning of each sequence is a randomly sampled token from the list. The number of tokens in each sequence is equal to the block size of the model, which is the maximum length of the input sequence the model can process. All models were trained on a single A100 GPU with 80 GB of memory.

2.1 Small Model

The small model consists of 25 million parameters, with 8 transformer blocks, each with 8 attention heads, an embedding size of 512, a block size of 1,024, and dropout with probability $p = 0.1$. To determine the optimal number of training iterations, the model is trained using 10% of the dataset as a validation set, and monitor the model's performance on the validation set, in terms of the cross-entropy loss. It was determined that the model continues to improve beyond 90,000 iterations. Therefore, the final model was trained on the entire dataset for 100,000 iterations (due to computational resource and time constraints).

2.2 Large Model

The large model consists of 200 million parameters, with 16 transformer blocks, each with 16 attention heads, an embedding size of 1,024, a block size of 2,048, and dropout with probability $p = 0.1$. Due to computational resource and time constraints, the large model is trained on the entire dataset for 48,000 iterations. Additionally, the starting point for each sequence is sampled from a pre-compiled list of tokens, each known to be the starting token of a CIF file in the dataset. This approach ensures that each sequence begins at the start of a distinct CIF file.

2.3 Training Times

On an A100 GPU, the small model requires 3 seconds per training iteration. Therefore, 100,000 small model training iterations requires 83.3 hours, or approximately 3.5 days. The large model requires 16 seconds per training iteration on an A100 GPU. Therefore, 48,000 large model training iterations requires 213.3 hours, or approximately 8.9 days.

2.4 CIF File Generation via Random Sampling

CIF files are generated using top-k random sampling. Top-k sampling involves randomly selecting the next token from the top k most likely candidates as predicted by the model. Temperature scaling $\mathbf{x}/\tau$ is first applied to the logits $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$ at the final position, and keep only the top k logits, where $|\mathcal{V}|$ represents the size of the vocabulary. The top k logits are then converted into normalized probabilities through application of the softmax operation. Finally, the next token is sampled using the given probabilities. More formally,

$$\text{token} \sim \text{softmax}\left(\text{top}_k\left(\frac{\mathbf{x}}{\tau}\right)\right) \tag{C.1}$$

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \tag{C.2}$$

where i represents the i-th element of the vector $\mathbf{x}$. Tokens are sampled iteratively, each conditioned on the progressively growing sequence of previously sampled tokens, until two consecutive newline tokens are sampled.

3. Validation of Generated CIF Files

To ensure the consistency of the printed information and the chemical sensibility of the implied structure, a series of validations is conducted on the generated CIF file. The procedure is described in Algorithm 2.

First, it is required that the chemical formula, which is printed in several locations in the file, is consistent everywhere. Specifically, the formula associated with the `_chemical_formula_sum` tag, and the (reduced) formula associated with the `_chemical_formula_structural` tag must be consistent with the cell composition in the first line of the file, and with each other.

Next, it is required that the printed atom site multiplicity is consistent with the cell composition. This information is printed at the end of the file, and the values are associated with the `_atom_site_type_symbol` and `_atom_site_symmetry_multiplicity` tags.

The structure's bond lengths are also evaluated for reasonableness. To check if bond lengths are reasonable, a Voronoi-based nearest-neighbour algorithm is first used in pymatgen to define which atoms are bonded together; then, expected bond lengths are established based on the electronegativity difference between the bonded atoms, and their ionic or covalent radii. A bond length reasonableness score is computed, $B \in [0, 1]$, which represents the fraction of bonds which are within 30% of the corresponding expected bond lengths. A structure is classified as having reasonable bond lengths if $B \geq c_{\text{bond}}$, where $c_{\text{bond}} \in [0, 1]$ is a bond length acceptability score minimum, which in this work is set to 1.0 (i.e. all bond lengths must be within 30% of the expected bond lengths).

Finally, the generated CIF file is checked for consistency in terms of space group. To check if the generated structure is consistent with the printed space group, the `SpacegroupAnalyzer` class of the pymatgen library is used, which uses the spglib library [7].

Algorithm 2 Check Validity of Generated CIF File

Input: S, the contents of the generated CIF file
Input: c_{bond}, the bond length acceptability score minimum
Output: True or False, indicating whether S is valid
if not is_formula_consistent(S) **then**
    return False
end if
if not is_atom_site_multiplicity_consistent(S) **then**
    return False
end if
$B \leftarrow$ bond_length_reasonableness_score(S)
if $B < c_{\text{bond}}$ **then**
    return False
end if
if not is_space_group_consistent(S) **then**
    return False
end if
return True

4. Monte Carlo Tree Search Decoding

To improve the efficiency and quality of sampling from the model, the Monte Carlo Tree Search (MCTS) algorithm [8, 9] is used. Typically, the MCTS algorithm is used in the context of games, and similar decision processes, where the aim is to select an optimal action to perform. Here, MCTS is used to generate a collection of sequences, which should improve (according to some measure of quality) as the algorithm proceeds.

A sequence of tokens can be considered the outcome of following a specific path during the traversal of a tree of tokens, starting from the root and progressing to a leaf. In this framework, each node in the tree represents a token, and each edge denotes the transition from one token to the next in the sequence. By systematically exploring and expanding the most promising paths, MCTS balances exploitation of well-performing token sequences with exploration of new, potentially better sequences. The trade-off between exploitation and exploration is achieved through a principled selection strategy, which guides the search towards areas of the tree that either have high potential or have not been sufficiently explored. As the search progresses, the algorithm builds a more informed representation of the tree, enabling more efficient and higher-quality sampling of token sequences.

The MCTS algorithm is comprised of a sequence of steps performed for a fixed number of iterations. The implementation is described in Algorithm 3, and a detailed explanation of each step follows.

4.1 Selection

Each node, i, in the tree represents the cumulative context up to that point, akin to constructing a sentence word by word. The first step in every iteration involves descending the tree, from the root node to a leaf node, by selecting the most promising node, i_t, at each level t. To select the node, a variant of the PUCT (Predictor-Upper Confidence bound applied to Trees) algorithm [10, 11] is used.

The selection of a node at each level is guided by the statistics accumulated in the tree. The specific node i_t is chosen by maximizing the PUCT value, expressed as $i_t =$

$\underset{i}{\mathrm{argmax}}(\mathrm{PUCT}(i_t))$, where the PUCT value is calculated as:

$$\mathrm{PUCT}(i) = \frac{w_i}{N_i} + c_{\mathrm{puct}} P_i \frac{\sqrt{N_h}}{1 + N_i} \tag{C.3}$$

where w_i represents the total score accumulated at node i, indicating the node's past performance. N_i is the number of times node i has been visited, reflecting its exploitation level. P_i is the prior probability of selecting the token leading to node i, given its parent node h, which is provided by the CrystaLLM model. N_h is the total number of visits to the parent node h, and c_{puct} is a constant determining the level of exploration in the PUCT algorithm.

4.2 Expansion

If a node has children that haven't been added to the tree, a child node is selected randomly and added to the tree. This newly added node is the selected node for the remainder of the iteration. To determine what children a node contains, the CrystaLLM model's predicted probabilities are used to select the top k tokens. If a child node's probability exceeds 0.99, it becomes the sole child node. In this case, where a node has only a single child node, the child node is *bypassed*, foregoing the Rollout step. The process then proceeds directly to the Selection step for the child nodes of the bypassed node, continuing the iteration from that point onwards.

4.3 Rollout

The Rollout step involves prompting the CrystaLLM model with the sequence of tokens represented by the selected node in the tree, and then sampling from the model repeatedly, until a terminating condition is reached. The aim is to arrive at a completed structure, which can then be further evaluated and scored.

4.4 Evaluation

Once a sequence has been completed, either by reaching a terminal node through Selection, or through Rollout, it represents the contents of a completed CIF file. The generated CIF contents are then validated (see Supplementary Note 3), and if the generated CIF file is valid, the structure is evaluated using the ALIGNN model of formation energy per atom, to produce a prediction of the structure's energy, E_{f}.

4.5 Backpropagation

The outcomes of iterations are accumulated in the tree nodes. All nodes selected during a simulation increase in their visit count, and a score is added to each. The score, $R \in [-1, 1]$, represents the quality of the generated structure. A more positive R represents a better structure.

Because scores are required to be between -1 and 1, and since the range of formation energies is not known for a composition *a priori*, the score for valid structures ($R_{\mathrm{valid}} \in [0, 1]$) is computed using the statistics of the predicted energies over the course of the search:

$$R_{\mathrm{valid}} = \frac{1}{1 + e^{\lambda((E_{\mathrm{f}} - \mu)/\sigma)}} \tag{C.4}$$

where E_{f} is the formation energy per atom (eV) according to ALIGNN (Atomistic Line Graph Neural Network) [12], μ is the mean over all of the obtained E_{f}, σ is the standard deviation

over all the obtained E_{f}, and λ is a constant that determines how responsive the reward is to E_{f}.

The overall score is computed piecewise:

$$R = \begin{cases} R_{\mathrm{valid}} & \text{if valid,} \\ B - 1 & \text{if bond lengths unreasonable,} \\ -1 & \text{otherwise} \end{cases} \tag{C.5}$$

where B is the bond length reasonableness score. An invalid structure receives a score of -1, unless it is invalid because of unreasonable bond lengths. In cases where a CIF file is otherwise valid, but the structure contains unreasonable bond lengths, a negative score is assigned that is proportional to the number of unreasonable bonds.

Algorithm 3 Monte Carlo Tree Search Decoding

Input: trained large language model, LLM
Input: number of simulations, n
Input: tree width, k
Input: PUCT exploration constant, c_{puct}
Input: text prompt, P
Output: list of valid sequences
Initialize tree with root node based on P
valid_sequences ← []
for $simulation = 1$ **to** n **do**
    current_node ← root
    // Select
    while not current_node.has_untried_children() and current_node.has_children() **do**
        current_node ← select_node(current_node.children, LLM, c_{puct})
    end while
    // Expand
    if current_node.has_untried_children() **then**
        untried_child ← select_untried_child_randomly(current_node, k)
        current_node.add_child(untried_child)
        current_node ← untried_child
    end if
    // Rollout
    complete_sequence ← sample_randomly(current_node, LLM)
    // Evaluate
    score ← evaluate_sequence(complete_sequence)
    if is_valid(complete_sequence) **then**
        valid_sequences.append(complete_sequence)
    end if
    // Backpropagate
    while current_node is not null **do**
        current_node.visits ← current_node.visits + 1
        current_node.wins ← current_node.wins + score
        current_node ← current_node.parent
    end while
end for
return valid_sequences

5. Metrics for Unconditional Generation

To assess performance on the unconditional generation tasks, the metrics introduced by Xie *et al.* [13] are used. Six metrics are used to evaluate different aspects of generation quality: COV-R (coverage recall), COV-P (coverage precision), AMSD-R (average minimum structure distance recall), AMSD-P (average minimum structure distance precision), AMCD-R (average minimum composition distance recall) and AMCD-P (average minimum composition distance precision).

Using the same formalism introduced by [13], the definitions of these metrics are re-stated here: A collection of K materials generated by a model, $\{M_k\}_{k\in[1..K]}$, is compared to a collection of L ground-truth materials, $\{M^*{}_l\}_{l\in[1..L]}$. A distance is further defined between two structures of the collections, $D_{\text{struc.}}(M_k, M^*{}_l)$, and a distance between two compositions of the collections, $D_{\text{comp.}}(M_k, M^*{}_l)$. For all metrics, the structure distance is the Euclidean distance between the CrystalNN fingerprints [14] for the two structures, while the composition distance is the Euclidean distance between the normalized Magpie fingerprints [15] of the two compositions. Moreover, thresholds $\delta_{\text{struc.}}, \delta_{\text{comp.}} \in \mathbb{R}$ are defined for the structure and composition distances.

The metrics are thus defined as follows:

$$\text{COV-R} = \frac{1}{L}|\{l \in [1..L] : \exists k \in [1..K], D_{\text{struc.}}(M_k, M^*{}_l) < \delta_{\text{struc.}}, D_{\text{comp.}}(M_k, M^*{}_l) < \delta_{\text{comp.}}\}| \tag{C.6}$$

$$\text{COV-P} = \frac{1}{K}|\{k \in [1..K] : \exists l \in [1..L], D_{\text{struc.}}(M_k, M^*{}_l) < \delta_{\text{struc.}}, D_{\text{comp.}}(M_k, M^*{}_l) < \delta_{\text{comp.}}\}| \tag{C.7}$$

$$\text{AMSD-R} = \frac{1}{L}\sum_{l\in[1..L]} \min_{k\in[1..K]} D_{\text{struc.}}(M_k, M^*{}_l) \tag{C.8}$$

$$\text{AMSD-P} = \frac{1}{K}\sum_{k\in[1..K]} \min_{l\in[1..L]} D_{\text{struc.}}(M_k, M^*{}_l) \tag{C.9}$$

$$\text{AMCD-R} = \frac{1}{L}\sum_{l\in[1..L]} \min_{k\in[1..K]} D_{\text{comp.}}(M_k, M^*{}_l) \tag{C.10}$$

$$\text{AMCD-P} = \frac{1}{K}\sum_{k\in[1..K]} \min_{l\in[1..L]} D_{\text{comp.}}(M_k, M^*{}_l) \tag{C.11}$$

In summary, the recall metrics assess the proportion of actual materials correctly identified, whereas the precision metrics evaluate the quality of the materials generated. See [13] and [16] for more detailed discussion and description of these metrics. As in [13], $\delta_{\text{struc.}} = 0.2$, $\delta_{\text{comp.}} = 4$ for Perov-5 and Carbon-24, and $\delta_{\text{struc.}} = 0.4$, $\delta_{\text{comp.}} = 10$ for MP-20 is used.

6. Effect of MCTS on the Stability of Unconditionally Generated Novel Structures

The MCTS procedure can be used to find structures with lower energy for the unconditionally generated compositions, using the ALIGNN energy evaluator as a fast proxy for DFT energies. As mentioned in the chapter, MCTS is performed, with 1,000 iterations, on each of the 102 compounds initially identified as novel.

The raw results are provided as a separate CSV file, and discussed in more detail here. The MCTS procedure works as intended, lowering (or keeping constant) the ALIGNN energies of all tested compositions. The average ALIGNN energy change is -153 $\pm$ 15 meV/atom (with the error bar obtained as the standard error of the mean). However, the mean E_{hull}, as calculated by DFT, is not reduced by the same amount, because of the limitations in accuracy of the ALIGNN energy estimator (and the fact that ALIGNN predicts energies for the as-generated, unrelaxed compounds, while the DFT energies are obtained for relaxed geometries). The MCTS-induced improvement of the average DFT energy, of -56 $\pm$ 15 meV/atom, reduces the average E_{hull} from 0.40 to 0.34 eV/atom. The question, then, is whether, given the sample size, this DFT energy lowering is still significant, or simply due to a statistical fluctuation. In other words, is the improvement introduced by the MCTS procedure in terms of ALIGNN energies maintained (with statistical significance) after evaluating the energies with DFT?

To precisely answer this question, a statistical test of significance was performed. The null hypothesis is that MCTS does not bring any improvement to the average E_{hull} obtained by DFT. In that case, the DFT energy changes (from the original structures to those generated by MCTS) would be just randomly distributed with zero mean. What the probability p would be of obtaining the DFT results under the null hypothesis conditions can be calculated. Using a paired t-test, t=-3.7 was obtained, which means that the probability of the DFT energy change observed being a statistical fluctuation in either direction (two-sided test) is p=0.0003. This is well below the threshold of p=0.05 typically accepted for statistical significance. It was verified that a Wilcoxon signed-rank test, that accounts for deviations of the distribution of paired differences from normality, gives a similar result. Therefore, the null hypothesis can definitely be rejected: the advantage introduced by MCTS does survive the transition from ALIGNN to DFT energies, despite the limitations of ALIGNN. Not only are the ALIGNN energies improved by the MCTS approach, but the DFT E_{hull} energies are as well.

C.2 Supplementary Tables

Table C.1: The compounds of the Challenge Set, their sources, and their formation energies per atom, as predicted by the ALIGNN model.

Formula	Source	ALIGNN E_{f} (eV/atom)
Ba_2MnCr	training set	0.906
$Ca_{10}(PO_4)_6(OH)_2$	training set	-3.029
$CH_3NH_3PbI_3$	training set	-0.358
$Co_2CO_3(OH)_2$	training set	-1.019
CsCuTePt	training set	0.137
$Cu_2C_1O_5H_2$	training set	-0.923
$Cu_3(CO_3)_2(OH)_2$	training set	-0.999
K_2AgMoI_6	training set	-0.639
MgF_2	training set	-3.782
$Mn_4(PO_4)_3$	training set	-2.010
$PbCu(OH)_2SO_4$	training set	-1.160
Sm_2BO_4	training set	-2.993
$AlCu_2As(HO)_{12}$	ref. [17]	-1.185
Ba_2AuIO_6	ref. [18]	-1.329
$Ba_2Fe_2F_9$	ref. [19]	-3.073
$Ba_6Fe_2Te_3S_7$	ref. [20]	-1.593
$Ba_2Gd(BO_3)_2F$	ref. [21]	-3.325
$Ba_4GeSb_2Se_{11}$	ref. [22]	-1.136
Ba_3GeTeS_4	ref. [23]	-1.721
Ba_2HfF_8	ref. [24]	-4.177
$BaY_{16}Si_4O_{33}$	ref. [25]	-3.702
$Ba_9Yb_2(SiO_4)_6$	ref. [26]	-3.245
$Ca_2Bi_2O_7$	ref. [27]	-1.952
$CaFe_6Ge_6$	ref. [28]	-0.200
$CaHPO_3$	ref. [29]	-2.445
$CaPt_4P_6$	ref. [30]	-0.826
$Ca_2Te_3O_8$	ref. [31]	-1.911
$CaZnV_2O_6$	ref. [32]	-2.573
$Ce_6Cd_{23}Te$	ref. [33]	-0.295
$Cs_2Al_2O_3F_2$	ref. [34]	-3.116
$Cs_8Cu_3Si_{14}O_{35}$	ref. [35]	-2.561
$Cs_3LuSi_3O_9$	ref. [36]	-2.970
$Cu_4FeGe_2S_7$	ref. [37]	-0.368
$Eu_2FeGe_2OS_6$	ref. [38]	-1.265
HgB_2S_4	ref. [39]	-0.340
$Ho_2Ir_3Si_5$	ref. [40]	-0.885
$KScP_2O_7$	ref. [41]	-2.801
$K_2Sr_4(PO_3)_{10}$	ref. [42]	-2.626
$K_6Zn(CO_3)_4$	ref. [43]	-1.817
$La_4Ga_2S_8O_3$	ref. [44]	-2.171
$LaScSe_3$	ref. [45]	-1.912
$Li_9Al_4Sn_5$	ref. [46]	-0.173
$LiBa_2AlO_4$	ref. [47]	-2.965
Li_2GeS_3	ref. [48]	-0.989
LiMnBi	ref. [49]	-0.004
$LiTa_2NiSe_5$	ref. [50]	-0.842
$Mg_7Pt_4Ge_4$	ref. [51]	-0.676
$NaGdSi_2O_6$	ref. [52]	-3.074
$Na_2Hf(BO_3)_2$	ref. [53]	-2.834
$Na_6Li_4WO_4(CO_3)_4$	ref. [54]	-2.010
$NaMgV_5(H_5O_6)_4$	ref. [55]	-1.540
$Na_5Mn_4P_4H_4(O_9F_2)_2$	ref. [56]	-2.140
$NaSbSe_2O_7$	ref. [57]	-1.213
$NaSb_2TeO_7$	ref. [57]	-1.502
$Na_4Sn_2Ge_5O_{16}$	ref. [58]	-1.862
$Na_3Te_2(FeO_4)_3$	ref. [59]	-1.430
$Nd_3BSi_2O_{10}$	ref. [60]	-3.352
$Ni_3Te_2O_2(PO_4)_2(OH)_4$	ref. [61]	-1.302
$RbNiFe(PO_4)_2$	ref. [62]	-1.930
Rb_3SnCl_7	ref. [63]	-1.434
$Sr_2Bi_2O_7$	ref. [27]	-1.963
$SrCo_4(OH)(PO_4)_3$	ref. [64]	-1.861
$Sr(ClO_4)_2$	ref. [65]	-0.593
$Sr_6Ge_3OSe_{11}$	ref. [66]	-1.145
$Tb_3S_3BO_3$	ref. [67]	-2.857
Tb_3TeBO_9	ref. [68]	-2.433
$YbMn_6Sn_6$	ref. [69]	-0.070
$Zn_2(HTeO_3)(AsO_4)$	ref. [70]	-1.290
Zn_2BS_3Br	ref. [71]	-0.692
$Zn_4CuH_6(CO_6)_2$	ref. [72]	-1.251

Table C.2: Performance of the small model on the Challenge Set. No space group was included in the prompt.

Composition	Mean E_f	Min E_f	% Valid	Any match true?
$AlCu_2As(HO)_{12}$	-0.366	-0.568	19	no
Ba_2AuIO_6	-1.546	-1.599	96	no
$Ba_2Fe_2F_9$	-2.324	-2.598	14	no
$Ba_2Gd(BO_3)_2F$	-2.091	-2.578	12	no
Ba_2HfF_8	-3.427	-3.940	35	yes
Ba_2MnCr	0.985	0.611	100	yes
Ba_3GeTeS_4	-0.985	-1.366	33	no
$Ba_4GeSb_2Se_{11}$	-0.707	-0.859	18	no
$Ba_6Fe_2Te_3S_7$	-0.743	-0.989	15	no
$Ba_9Yb_2(SiO_4)_6$	-1.524	-1.524	1	no
$BaY_{16}Si_4O_{33}$	-	-	0	no
$CH_3NH_3PbI_3$	0.339	0.110	11	no
$Ca_{10}(PO_4)_6(OH)_2$	-	-	0	no
$Ca_2Bi_2O_7$	-1.759	-1.889	100	yes
$Ca_2Te_3O_8$	-	-	0	no
$CaFe_6Ge_6$	0.232	-0.207	26	yes
$CaHPO_3$	-1.471	-2.030	18	no
$CaPt_4P_6$	-0.330	-0.671	29	no
$CaZnV_2O_6$	-1.981	-2.551	32	yes
$Ce_6Cd_{23}Te$	-0.309	-0.369	91	yes
$Co_2CO_3(OH)_2$	-0.180	-0.527	29	no
$Cs_2Al_2O_3F_2$	-1.959	-2.455	16	no
$Cs_3LuSi_3O_9$	-	-	0	no
$Cs_8Cu_3Si_{14}O_{35}$	-	-	0	no
$CsCuTePt$	0.136	0.092	100	yes
$Cu_2C_1O_5H_2$	-0.279	-0.708	48	no
$Cu_3(CO_3)_2(OH)_2$	-0.164	-0.479	22	no
$Cu_4FeGe_2S_7$	-0.060	-0.266	39	no
$Eu_2FeGe_2OS_6$	-0.924	-1.265	45	yes
HgB_2S_4	0.336	0.051	18	no
$Ho_2Ir_3Si_5$	-0.883	-0.890	98	yes
K_2AgMoI_6	-0.638	-0.643	100	yes
$K_2Sr_4(PO_3)_{10}$	-	-	0	no
$K_6Zn(CO_3)_4$	-	-	0	no
$KScP_2O_7$	-2.626	-2.794	81	yes
$La_4Ga_2S_8O_3$	-1.024	-1.288	5	no
$LaScSe_3$	-1.879	-1.978	98	yes
Li_2GeS_3	-0.554	-0.960	44	no
$Li_9Al_4Sn_5$	-0.122	-0.189	2	no
$LiBa_2AlO_4$	-1.837	-2.053	3	no
$LiMnBi$	0.130	0.075	100	no
$LiTa_2NiSe_5$	-0.693	-0.847	71	no
$Mg_7Pt_4Ge_4$	-0.263	-0.521	23	no
MgF_2	-3.512	-3.811	93	yes
$Mn_4(PO_4)_3$	-1.750	-2.014	16	yes
$Na_2Hf(BO_3)_2$	-2.766	-2.835	69	yes
$Na_3Te_2(FeO_4)_3$	-1.362	-1.455	97	yes
$Na_4Sn_2Ge_5O_{16}$	-0.741	-0.867	2	no
$Na_5Mn_4P_4H_4(O_9F_2)_2$	-0.917	-0.948	2	no
$Na_6Li_4WO_4(CO_3)_4$	-	-	0	no
$NaGdSi_2O_6$	-2.721	-3.060	63	no
$NaMgV_5(H_5O_6)_4$	-	-	0	no
$NaSb_2TeO_7$	-0.855	-1.049	3	no
$NaSbSe_2O_7$	-0.528	-0.813	10	no
$Nd_3BSi_2O_{10}$	-	-	0	no
$Ni_3Te_2O_2(PO_4)_2(OH)_4$	-0.538	-0.824	28	no
$PbCu(OH)_2SO_4$	-0.362	-0.731	51	no
Rb_3SnCl_7	-1.329	-1.506	52	yes
$RbNiFe(PO_4)_2$	-0.896	-1.315	9	no
Sm_2BO_4	-2.978	-3.011	92	yes
$Sr(ClO_4)_2$	-0.044	-0.357	20	no
$Sr_2Bi_2O_7$	-1.729	-1.931	97	yes
$Sr_6Ge_3OSe_{11}$	-0.768	-1.017	4	no
$SrCo_4(OH)(PO_4)_3$	-0.868	-0.868	1	no
$Tb_3S_3BO_3$	-1.173	-2.026	45	no
Tb_3TeBO_9	-2.274	-2.477	73	yes
$YbMn_6Sn_6$	-0.042	-0.071	100	yes
$Zn_2(HTeO_3)(AsO_4)$	-0.556	-1.209	26	no
Zn_2BS_3Br	-0.077	-0.602	77	no
$Zn_4CuH_6(CO_6)_2$	-0.207	-0.641	19	no

Table C.3: Performance of the small model on the Challenge Set. The space group was included in the prompt.

Composition	Mean E_f	Min E_f	% Valid	Any match true?
$AlCu_2As(HO)_{12}$	-0.403	-0.735	50	no
Ba_2AuIO_6	-1.410	-1.608	74	yes
$Ba_2Fe_2F_9$	-2.163	-2.416	8	no
$Ba_2Gd(BO_3)_2F$	-1.869	-2.266	12	no
Ba_2HfF_8	-3.565	-4.082	40	yes
Ba_2MnCr	1.020	0.833	100	yes
Ba_3GeTeS_4	-1.148	-1.589	42	yes
$Ba_4GeSb_2Se_{11}$	-0.707	-0.888	16	no
$Ba_6Fe_2Te_3S_7$	-0.563	-1.020	6	no
$Ba_9Yb_2(SiO_4)_6$	-	-	0	no
$BaY_{16}Si_4O_{33}$	-	-	0	no
$CH_3NH_3PbI_3$	0.523	0.068	47	no
$Ca_{10}(PO_4)_6(OH)_2$	-	-	0	no
$Ca_2Bi_2O_7$	-1.758	-1.887	100	yes
$Ca_2Te_3O_8$	-	-	0	no
$CaFe_6Ge_6$	0.013	-0.175	8	yes
$CaHPO_3$	-1.175	-1.917	18	no
$CaPt_4P_6$	-0.346	-0.645	31	yes
$CaZnV_2O_6$	-2.197	-2.583	51	yes
$Ce_6Cd_{23}Te$	-0.315	-0.357	89	yes
$Co_2CO_3(OH)_2$	-0.086	-0.304	21	no
$Cs_2Al_2O_3F_2$	-2.028	-2.813	20	no
$Cs_3LuSi_3O_9$	-1.505	-1.927	3	no
$Cs_8Cu_3Si_{14}O_{35}$	-	-	0	no
$CsCuTePt$	0.136	0.128	100	yes
$Cu_2C_1O_5H_2$	-0.135	-0.541	29	no
$Cu_3(CO_3)_2(OH)_2$	-0.157	-0.581	29	no
$Cu_4FeGe_2S_7$	-0.055	-0.314	57	no
$Eu_2FeGe_2OS_6$	-1.151	-1.290	52	yes
HgB_2S_4	0.272	-0.043	25	no
$Ho_2Ir_3Si_5$	-0.881	-0.892	97	yes
K_2AgMoI_6	-0.638	-0.643	100	yes
$K_2Sr_4(PO_3)_{10}$	-	-	0	no
$K_6Zn(CO_3)_4$	-0.820	-0.820	1	no
$KScP_2O_7$	-2.703	-2.798	77	yes
$La_4Ga_2S_8O_3$	-1.028	-1.460	19	no
$LaScSe_3$	-1.912	-1.975	99	yes
Li_2GeS_3	-0.590	-0.962	24	yes
$Li_9Al_4Sn_5$	-0.117	-0.216	9	no
$LiBa_2AlO_4$	-1.392	-2.074	15	no
$LiMnBi$	0.123	-0.037	100	yes
$LiTa_2NiSe_5$	-0.249	-0.555	48	no
$Mg_7Pt_4Ge_4$	-0.412	-0.602	19	no
MgF_2	-3.774	-3.810	100	yes
$Mn_4(PO_4)_3$	-1.574	-1.997	34	yes
$Na_2Hf(BO_3)_2$	-2.802	-2.839	99	yes
$Na_3Te_2(FeO_4)_3$	-1.363	-1.458	94	yes
$Na_4Sn_2Ge_5O_{16}$	-0.974	-1.215	3	no
$Na_5Mn_4P_4H_4(O_9F_2)_2$	-1.053	-1.053	1	no
$Na_6Li_4WO_4(CO_3)_4$	-0.712	-1.012	3	no
$NaGdSi_2O_6$	-1.696	-2.620	12	no
$NaMgV_5(H_5O_6)_4$	-	-	0	no
$NaSb_2TeO_7$	-0.752	-0.752	1	no
$NaSbSe_2O_7$	-0.543	-1.081	21	no
$Nd_3BSi_2O_{10}$	-1.915	-1.915	1	no
$Ni_3Te_2O_2(PO_4)_2(OH)_4$	-0.582	-0.947	24	no
$PbCu(OH)_2SO_4$	-0.378	-0.776	46	no
Rb_3SnCl_7	-1.454	-1.538	76	yes
$RbNiFe(PO_4)_2$	-0.887	-1.082	4	no
Sm_2BO_4	-2.981	-3.011	92	yes
$Sr(ClO_4)_2$	-0.064	-0.158	2	no
$Sr_2Bi_2O_7$	-1.762	-1.968	100	yes
$Sr_6Ge_3OSe_{11}$	-0.672	-0.912	5	no
$SrCo_4(OH)(PO_4)_3$	-	-	0	no
$Tb_3S_3BO_3$	-1.281	-1.477	6	no
Tb_3TeBO_9	-2.251	-2.466	84	yes
$YbMn_6Sn_6$	-0.052	-0.066	99	yes
$Zn_2(HTeO_3)(AsO_4)$	-0.865	-1.210	50	no
Zn_2BS_3Br	0.111	-0.181	56	no
$Zn_4CuH_6(CO_6)_2$	-0.205	-0.302	4	no

Table C.4: Performance of the large model on the Challenge Set. No space group was included in the prompt.

Composition	Mean E_f	Min E_f	% Valid	Any match true?
$AlCu_2As(HO)_{12}$	-0.333	-0.620	33	no
Ba_2AuIO_6	-1.541	-1.589	65	no
$Ba_2Fe_2F_9$	-2.292	-2.701	10	no
$Ba_2Gd(BO_3)_2F$	-1.996	-2.207	5	no
Ba_2HfF_8	-3.522	-4.109	23	yes
Ba_2MnCr	1.027	0.593	100	yes
Ba_3GeTeS_4	-1.346	-1.681	71	yes
$Ba_4GeSb_2Se_{11}$	-1.085	-1.137	82	yes
$Ba_6Fe_2Te_3S_7$	-0.722	-1.036	12	no
$Ba_9Yb_2(SiO_4)_6$	-2.970	-3.244	41	yes
$BaY_{16}Si_4O_{33}$	-	-	0	no
$CH_3NH_3PbI_3$	-0.352	-0.358	100	yes
$Ca_{10}(PO_4)_6(OH)_2$	-2.999	-3.003	93	no
$Ca_2Bi_2O_7$	-1.751	-1.888	100	yes
$Ca_2Te_3O_8$	-	-	0	no
$CaFe_6Ge_6$	0.207	-0.003	7	no
$CaHPO_3$	-1.165	-1.504	14	no
$CaPt_4P_6$	-0.437	-0.756	34	yes
$CaZnV_2O_6$	-2.394	-2.585	64	yes
$Ce_6Cd_{23}Te$	-0.325	-0.342	93	yes
$Co_2CO_3(OH)_2$	-1.009	-1.019	100	yes
$Cs_2Al_2O_3F_2$	-2.114	-2.685	14	no
$Cs_3LuSi_3O_9$	-2.815	-2.946	19	no
$Cs_8Cu_3Si_{14}O_{35}$	-	-	0	no
$CsCuTePt$	0.140	0.131	100	yes
$Cu_2C_1O_5H_2$	-0.924	-0.933	99	yes
$Cu_3(CO_3)_2(OH)_2$	-1.007	-1.018	95	yes
$Cu_4FeGe_2S_7$	-0.073	-0.282	53	no
$Eu_2FeGe_2OS_6$	-0.824	-1.264	31	yes
HgB_2S_4	0.380	0.154	9	no
$Ho_2Ir_3Si_5$	-0.880	-0.887	99	yes
K_2AgMoI_6	-0.637	-0.644	100	yes
$K_2Sr_4(PO_3)_{10}$	-	-	0	no
$K_6Zn(CO_3)_4$	-	-	0	no
$KScP_2O_7$	-2.761	-2.795	100	yes
$La_4Ga_2S_8O_3$	-1.174	-1.421	5	no
$LaScSe_3$	-1.866	-1.940	96	yes
Li_2GeS_3	-0.735	-0.917	23	no
$Li_9Al_4Sn_5$	-0.112	-0.218	14	no
$LiBa_2AlO_4$	-2.000	-2.628	9	no
$LiMnBi$	0.124	-0.055	99	yes
$LiTa_2NiSe_5$	-0.436	-0.844	58	no
$Mg_7Pt_4Ge_4$	-	-	0	no
MgF_2	-3.380	-3.803	93	yes
$Mn_4(PO_4)_3$	-1.980	-2.020	81	no
$Na_2Hf(BO_3)_2$	-2.567	-2.829	75	yes
$Na_3Te_2(FeO_4)_3$	-1.430	-1.460	100	yes
$Na_4Sn_2Ge_5O_{16}$	-1.023	-1.034	3	no
$Na_5Mn_4P_4H_4(O_9F_2)_2$	-1.094	-1.151	2	no
$Na_6Li_4WO_4(CO_3)_4$	-	-	0	no
$NaGdSi_2O_6$	-2.962	-3.083	73	yes
$NaMgV_5(H_5O_6)_4$	-	-	0	no
$NaSb_2TeO_7$	-0.689	-0.943	4	no
$NaSbSe_2O_7$	-0.497	-0.698	9	no
$Nd_3BSi_2O_{10}$	-3.338	-3.374	86	yes
$Ni_3Te_2O_2(PO_4)_2(OH)_4$	-0.502	-1.010	31	no
$PbCu(OH)_2SO_4$	-1.153	-1.160	98	yes
Rb_3SnCl_7	-1.438	-1.481	6	yes
$RbNiFe(PO_4)_2$	-1.011	-1.557	9	no
Sm_2BO_4	-2.986	-3.006	95	yes
$Sr(ClO_4)_2$	-0.170	-0.348	12	yes
$Sr_2Bi_2O_7$	-1.672	-1.874	100	yes
$Sr_6Ge_3OSe_{11}$	-0.870	-0.870	1	no
$SrCo_4(OH)(PO_4)_3$	-	-	0	no
$Tb_3S_3BO_3$	-1.792	-2.195	13	no
Tb_3TeBO_9	-1.994	-2.347	71	yes
$YbMn_6Sn_6$	-0.056	-0.067	100	yes
$Zn_2(HTeO_3)(AsO_4)$	-0.470	-0.671	22	no
Zn_2BS_3Br	0.024	-0.510	56	no
$Zn_4CuH_6(CO_6)_2$	-0.181	-0.550	18	no

Table C.5: Performance of the large model on the Challenge Set. The space group was included in the prompt.

Composition	Mean E_f	Min E_f	% Valid	Any match true?
$AlCu_2As(HO)_{12}$	-0.343	-0.540	30	no
Ba_2AuIO_6	-1.411	-1.572	65	yes
$Ba_2Fe_2F_9$	-2.175	-2.503	4	no
$Ba_2Gd(BO_3)_2F$	-2.505	-2.739	4	no
Ba_2HfF_8	-3.466	-4.058	47	yes
Ba_2MnCr	1.066	0.862	100	yes
Ba_3GeTeS_4	-1.520	-1.681	70	yes
$Ba_4GeSb_2Se_{11}$	-1.100	-1.138	75	yes
$Ba_6Fe_2Te_3S_7$	-0.724	-1.068	31	no
$Ba_9Yb_2(SiO_4)_6$	-2.949	-3.241	38	yes
$BaY_{16}Si_4O_{33}$	-	-	0	no
$CH_3NH_3PbI_3$	-0.358	-0.358	100	yes
$Ca_{10}(PO_4)_6(OH)_2$	-1.634	-1.634	1	no
$Ca_2Bi_2O_7$	-1.745	-1.864	100	yes
$Ca_2Te_3O_8$	-	-	0	no
$CaFe_6Ge_6$	0.079	-0.063	12	no
$CaHPO_3$	-1.159	-1.857	22	no
$CaPt_4P_6$	-0.475	-0.813	33	yes
$CaZnV_2O_6$	-2.310	-2.571	79	yes
$Ce_6Cd_{23}Te$	-0.327	-0.360	99	yes
$Co_2CO_3(OH)_2$	-1.009	-1.020	100	yes
$Cs_2Al_2O_3F_2$	-2.190	-2.831	18	no
$Cs_3LuSi_3O_9$	-	-	0	no
$Cs_8Cu_3Si_{14}O_{35}$	-	-	0	no
$CsCuTePt$	0.139	0.131	100	yes
$Cu_2C_1O_5H_2$	-0.923	-0.929	99	yes
$Cu_3(CO_3)_2(OH)_2$	-0.260	-0.634	27	no
$Cu_4FeGe_2S_7$	-0.104	-0.215	60	no
$Eu_2FeGe_2OS_6$	-1.123	-1.300	91	yes
HgB_2S_4	0.191	0.040	10	no
$Ho_2Ir_3Si_5$	-0.881	-0.886	99	yes
K_2AgMoI_6	-0.637	-0.643	100	yes
$K_2Sr_4(PO_3)_{10}$	-	-	0	no
$K_6Zn(CO_3)_4$	-0.857	-0.997	2	no
$KScP_2O_7$	-2.759	-2.803	98	yes
$La_4Ga_2S_8O_3$	-1.235	-1.290	4	no
$LaScSe_3$	-1.880	-1.960	97	yes
Li_2GeS_3	-0.629	-0.930	15	yes
$Li_9Al_4Sn_5$	-0.149	-0.249	37	no
$LiBa_2AlO_4$	-1.621	-2.328	37	no
$LiMnBi$	0.066	0.001	100	yes
$LiTa_2NiSe_5$	-0.166	-0.493	50	no
$Mg_7Pt_4Ge_4$	-0.448	-0.591	50	no
MgF_2	-3.783	-3.808	100	yes
$Mn_4(PO_4)_3$	-1.903	-2.021	84	yes
$Na_2Hf(BO_3)_2$	-2.802	-2.857	98	yes
$Na_3Te_2(FeO_4)_3$	-1.430	-1.456	100	yes
$Na_4Sn_2Ge_5O_{16}$	-0.946	-1.214	15	no
$Na_5Mn_4P_4H_4(O_9F_2)_2$	-0.902	-1.123	2	no
$Na_6Li_4WO_4(CO_3)_4$	-0.914	-1.307	3	no
$NaGdSi_2O_6$	-3.011	-3.083	79	yes
$NaMgV_5(H_5O_6)_4$	-	-	0	no
$NaSb_2TeO_7$	-0.717	-0.892	4	no
$NaSbSe_2O_7$	-0.575	-0.983	21	no
$Nd_3BSi_2O_{10}$	-3.363	-3.372	76	yes
$Ni_3Te_2O_2(PO_4)_2(OH)_4$	-0.588	-0.851	21	no
$PbCu(OH)_2SO_4$	-1.138	-1.161	99	yes
Rb_3SnCl_7	-1.431	-1.521	51	yes
$RbNiFe(PO_4)_2$	-0.966	-1.218	6	no
Sm_2BO_4	-2.986	-3.005	96	yes
$Sr(ClO_4)_2$	-0.184	-0.337	11	yes
$Sr_2Bi_2O_7$	-1.671	-1.861	100	yes
$Sr_6Ge_3OSe_{11}$	-0.698	-0.836	6	no
$SrCo_4(OH)(PO_4)_3$	-0.506	-0.506	1	no
$Tb_3S_3BO_3$	-1.246	-1.410	7	no
Tb_3TeBO_9	-2.035	-2.443	78	yes
$YbMn_6Sn_6$	-0.055	-0.070	100	yes
$Zn_2(HTeO_3)(AsO_4)$	-0.389	-0.889	34	no
Zn_2BS_3Br	0.167	-0.245	21	no
$Zn_4CuH_6(CO_6)_2$	-0.065	-0.126	3	no

Table C.6: MCTS results for the small model. No space group was included in the prompt.

Composition	Algorithm	Best E_f	Best Iter.	Mean E_f	% Valid
$Ba_2Fe_2F_9$	Random	-2.787	10	-2.273	14.50
	MCTS	-2.812	570	-2.359	42.10
$Ba_2Gd(BO_3)_2F$	Random	-2.950	943	-2.121	18.50
	MCTS	-2.992	699	-2.104	27.20
$Ba_4GeSb_2Se_{11}$	Random	-0.928	571	-0.704	16.70
	MCTS	-0.925	509	-0.735	37.40
$Ba_6Fe_2Te_3S_7$	Random	-1.123	110	-0.689	16.60
	MCTS	-1.216	510	-0.730	17.40
$Ba_9Yb_2(SiO_4)_6$	Random	-2.712	834	-1.907	1.40
	MCTS	-2.777	534	-1.815	2.10
$CH_3NH_3PbI_3$	Random	-0.027	257	0.431	10.60
	MCTS	-0.199	611	0.448	52.80
$CaHPO_3$	Random	-2.048	55	-1.397	18.10
	MCTS	-2.247	367	-1.596	59.30
$Cs_2Al_2O_3F_2$	Random	-2.825	283	-1.972	22.90
	MCTS	-2.922	651	-2.051	37.90
HgB_2S_4	Random	-0.142	765	0.284	16.90
	MCTS	-0.212	282	0.270	34.20
$La_4Ga_2S_8O_3$	Random	-1.404	696	-1.016	3.30
	MCTS	-1.495	27	-1.094	5.70
$Li_9Al_4Sn_5$	Random	-0.225	409	-0.147	1.50
	MCTS	-0.231	123	-0.143	4.60
$LiBa_2AlO_4$	Random	-2.683	667	-1.728	4.40
	MCTS	-2.504	281	-1.854	61.20
$Mn_4(PO_4)_3$	Random	-2.029	122	-1.787	22.00
	MCTS	-2.045	632	-1.946	68.90
$Na_4Sn_2Ge_5O_{16}$	Random	-1.126	40	-0.863	2.50
	MCTS	-1.264	848	-0.915	8.70
$Na_5Mn_4P_4H_4(O_9F_2)_2$	Random	-1.510	660	-1.020	2.40
	MCTS	-1.531	335	-0.979	1.10
$NaSb_2TeO_7$	Random	-1.292	64	-0.851	4.50
	MCTS	-1.391	787	-0.875	25.50
$NaSbSe_2O_7$	Random	-0.969	795	-0.473	11.50
	MCTS	-1.108	792	-0.658	42.70
$RbNiFe(PO_4)_2$	Random	-1.465	197	-0.835	4.70
	MCTS	-1.599	699	-1.044	6.80
$Sr_6Ge_3OSe_{11}$	Random	-0.974	658	-0.716	1.50
	MCTS	-1.214	207	-0.910	31.10
$SrCo_4(OH)(PO_4)_3$	Random	-1.245	493	-0.845	2.20
	MCTS	-1.223	102	-0.674	9.90

Table C.7: MCTS results for the small model. The space group was included in the prompt.

Composition	Algorithm	Best E_f	Best Iter.	Mean E_f	% Valid
$Ba_2Fe_2F_9$	Random	-2.523	822	-2.213	8.10
	MCTS	-2.642	188	-2.204	8.30
$Ba_2Gd(BO_3)_2F$	Random	-2.761	906	-1.922	7.60
	MCTS	-2.644	461	-1.956	10.20
$Ba_4GeSb_2Se_{11}$	Random	-0.904	490	-0.677	15.30
	MCTS	-0.923	450	-0.725	25.30
$Ba_6Fe_2Te_3S_7$	Random	-1.110	297	-0.632	6.20
	MCTS	-1.137	160	-0.711	24.60
$CaFe_6Ge_6$	Random	-0.205	353	0.018	9.30
	MCTS	-0.209	675	0.065	8.70
$Cs_3LuSi_3O_9$	Random	-1.967	640	-1.491	1.60
	MCTS	-1.953	325	-1.514	1.40
$K_6Zn(CO_3)_4$	Random	-1.308	143	-0.694	1.50
	MCTS	-0.958	487	-0.673	1.40
$Li_9Al_4Sn_5$	Random	-0.247	926	-0.115	10.50
	MCTS	-0.290	559	-0.134	42.50
$LiBa_2AlO_4$	Random	-2.395	365	-1.370	15.60
	MCTS	-2.380	125	-1.319	16.40
$Na_4Sn_2Ge_5O_{16}$	Random	-1.271	849	-0.907	5.80
	MCTS	-1.432	119	-0.920	10.50
$Na_5Mn_4P_4H_4(O_9F_2)_2$	Random	-1.567	463	-1.105	2.10
	MCTS	-1.472	671	-1.076	3.50
$Na_6Li_4WO_4(CO_3)_4$	Random	-1.488	675	-0.677	6.80
	MCTS	-1.203	694	-0.722	2.90
$NaGdSi_2O_6$	Random	-2.486	128	-1.686	12.80
	MCTS	-2.591	893	-1.669	9.60
$NaSb_2TeO_7$	Random	-0.902	616	-0.571	1.40
	MCTS	-1.025	881	-0.667	5.40
$Nd_3BSi_2O_{10}$	Random	-2.910	740	-2.234	1.60
	MCTS	-2.777	411	-2.372	0.50
$RbNiFe(PO_4)_2$	Random	-1.477	104	-0.926	4.70
	MCTS	-1.555	73	-0.961	8.50
$Sr(ClO_4)_2$	Random	-0.278	125	-0.043	2.80
	MCTS	-0.314	633	-0.051	3.10
$Sr_6Ge_3OSe_{11}$	Random	-1.155	679	-0.763	3.80
	MCTS	-1.358	158	-0.874	3.00
$Tb_3S_3BO_3$	Random	-1.877	487	-1.358	6.10
	MCTS	-1.915	199	-1.327	5.70
$Zn_4CuH_6(CO_6)_2$	Random	-0.350	834	-0.139	3.70
	MCTS	-0.575	815	-0.255	4.00

Table C.8: Metrics for the unconditional generation tasks. Numbers in bold indicate the best results for the given task. The CDVAE results are from Xie *et al.* [13]. The DiffCSP and DiffCSP++ results are from Jiao *et al.* [73, 74]. The UniMat results are from Yang *et al.* [75]. The LM-CH (character-level tokenization) and LM-AC (atom+coordinate-level tokenization) results are from Flam-Shepherd *et al.* [76]. The LLaMA 70B results are from Gruver *et al.* [77]. τ represents the sampling temperature.

	Validity (%) ↑		Coverage (%) ↑		Property ↓		Average Minimum Distance ↓			
Method	Struct	Comp	COV-R	COV-P	d_ρ	d_{elem}	AMSD-R	AMSD-P	AMCD-R	AMCD-P
MP-20										
CDVAE	**100.0**	86.70	99.15	99.49	0.6875	1.4320	0.154	0.188	3.620	4.014
DiffCSP	**100.0**	83.25	99.71	**99.76**	0.3502	0.3398	-	-	-	-
DiffCSP++	99.94	85.12	99.73	99.59	0.2351	0.3749	-	-	-	-
UnitMat	97.20	89.40	**99.80**	99.70	**0.0880**	**0.0560**	0.097	0.119	**2.410**	2.410
LM-CH	84.81	83.55	99.25	97.89	0.8640	0.1320	-	-	-	-
LM-AC	95.81	88.87	99.60	98.55	0.6960	0.0920	-	-	-	-
LLaMA 70B (τ=1.0)	96.50	86.30	96.80	98.30	1.7200	0.5500	-	-	-	-
LLaMA 70B (τ=0.7)	99.60	**95.40**	85.80	98.90	0.8100	0.4400	-	-	-	-
CrystaLLM small (τ=0.7)	93.66	91.10	98.52	95.08	0.8353	0.2229	0.096	0.096	3.251	2.084
CrystaLLM small (τ=0.5)	94.97	93.80	97.58	95.75	1.1824	0.3269	0.106	0.095	3.729	**1.762**
CrystaLLM large (τ=0.7)	95.54	93.07	97.22	96.40	0.5965	0.1709	**0.090**	0.077	3.299	2.114
CrystaLLM large (τ=0.5)	96.21	**95.40**	96.78	96.60	0.9835	0.3436	0.098	**0.076**	3.675	1.880
Perov-5										
CDVAE	**100.0**	98.59	99.45	98.46	0.1258	0.0628	0.048	0.059	**0.696**	**1.270**
DiffCSP	**100.0**	98.85	**99.74**	98.27	0.1110	0.0128	-	-	-	-
DiffCSP++	**100.0**	98.77	99.60	98.80	**0.0661**	**0.0040**	-	-	-	-
UnitMat	**100.0**	98.80	99.20	98.20	0.0760	0.0250	0.046	0.074	0.711	1.399
LM-CH	**100.0**	98.51	99.60	**99.42**	0.0710	0.0360	-	-	-	-
LM-AC	**100.0**	98.79	98.78	99.36	0.0890	0.0280	-	-	-	-
CrystaLLM small (τ=0.7)	99.90	99.04	98.20	99.01	0.3355	0.0299	**0.025**	0.027	1.055	1.287
CrystaLLM small (τ=0.5)	99.83	**99.24**	97.91	98.95	0.3950	0.0970	0.027	0.025	1.215	1.293
CrystaLLM large (τ=0.7)	99.82	98.92	98.28	98.92	0.2070	0.0490	0.026	0.024	1.000	1.288
CrystaLLM large (τ=0.5)	99.96	98.86	97.86	98.73	0.3937	0.1240	0.027	**0.020**	1.144	1.319
Carbon-24										
CDVAE	**100.0**	-	99.80	83.08	0.1407	-	0.048	0.134	0.000	0.000
DiffCSP	**100.0**	-	99.90	97.27	0.0805	-	-	-	-	-
DiffCSP++	99.99	-	**100.0**	88.28	0.0307	-	-	-	-	-
UnitMat	**100.0**	-	**100.0**	96.50	**0.0130**	-	0.018	0.052	0.000	0.000
CrystaLLM small (τ=0.7)	99.21	-	99.85	97.03	0.0639	-	0.015	0.021	0.000	0.000
CrystaLLM small (τ=0.5)	99.86	-	99.80	98.96	0.1217	-	0.022	0.012	0.000	0.000
CrystaLLM large (τ=0.7)	99.70	-	99.80	98.37	0.0409	-	**0.014**	0.018	0.000	0.000
CrystaLLM large (τ=0.5)	99.90	-	99.75	**99.52**	0.0953	-	0.018	**0.010**	0.000	0.000

Table C.9: Novel materials generated unconditionally with the large model. Only those with a DFT energy of less than or equal to 0.1 eV/atom above the hull are listed.

Composition	Z	**Space Group**	E_{hull} (eV/atom)
Ca_2YSbO_6	2	$P2_1/c$	0.00
$NaAlS_2$	16	$P2_1$	0.00
$Ba_4Na_2Ir_2O_{11}$	2	Cm	0.00
Li_2FeSiO_4	4	$Pna2_1$	0.02
La_2Al_{17}	3	$R\bar{3}m$	0.03
$Ba_4Zr_2Mo_2O_{11}$	2	Pm	0.03
$LaSc(Al_2Pd)_2$	2	$Amm2$	0.04
$KLi(NbCl_3)_6$	2	$P1$	0.05
$Li_4Mn_3Nb_2Fe_3O_{16}$	1	$P1$	0.05
Ba_2SrCa	2	$Imm2$	0.06
$MnAlTcIr$	4	$F\bar{4}3m$	0.07
$Na_5(SnS_4)_2$	4	$P2_1$	0.07
$MnVO_4$	4	$Pbcn$	0.07
$Li_4Ti_3V_3(SbO_8)_2$	2	Cm	0.07
$Li_4Ti_3Cr_3(FeO_8)_2$	2	Pc	0.07
$K_2LiV_5H_{10}O_{19}$	2	$P\bar{1}$	0.07
$KNaS_2$	2	$P\bar{1}$	0.09
Li_2CrCuH_6	4	$P2_1/c$	0.09
$Li_4Ti_3Fe_3(WO_8)_2$	2	$P1$	0.10
$MnAl_2V_3$	1	$P4mm$	0.10

C.3 Supplementary Figures

Figure C.1: Various plots describing the contents of the CIF file dataset. **a** The distribution of CIF files containing the atoms indicated (by atomic number) on the x-axis. The most abundant element in the dataset is oxygen, followed by copper, then lithium. **b** The distribution of Z values (i.e. the number of formula units in the unit cell) in the dataset. The majority of structures have Z of 1-4. **c** The distribution of compositions by the number of constituent elements in the formula. Most formulas are ternary or quaternary. *Inset:* The Venn diagram illustrates the numbers of unique reduced compositions obtained from each of the publicly accessible materials databases used to create the training dataset. **d** The distribution of space groups occurring in the CIF files of the dataset.

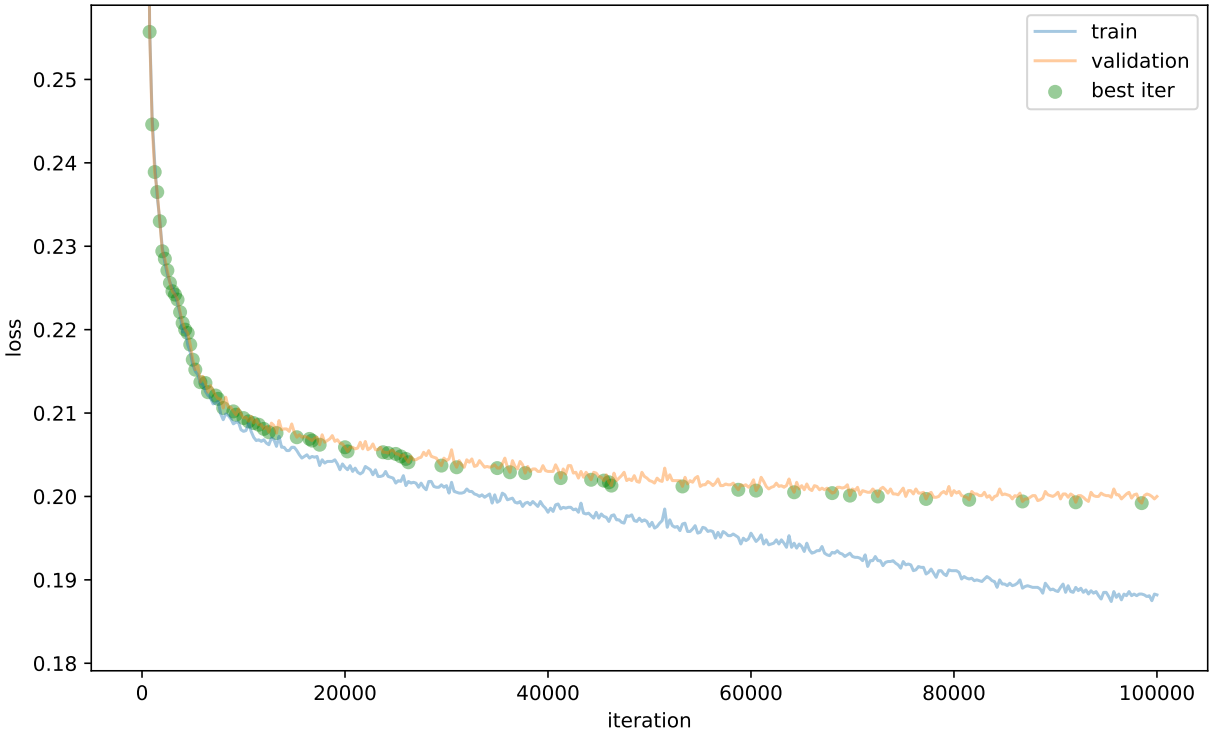

Figure C.2: A plot of the training set and validation set losses for the small model over the course of training. The green points represent an improved loss on the validation set, demonstrating that the model continues to improve its performance on the validation set even after 90,000 iterations. While the gap between the training set loss and the validation set loss appears to grow over the course of training, this is not necessarily indicative of overfitting. The growing gap could be more indicative of the differences between the distributions of the training and validation sets. On absolute terms, the difference between the curves is less than 0.02 units. Moreover, the performance on the validation and challenge sets indicate that the model trained for more iterations is superior.

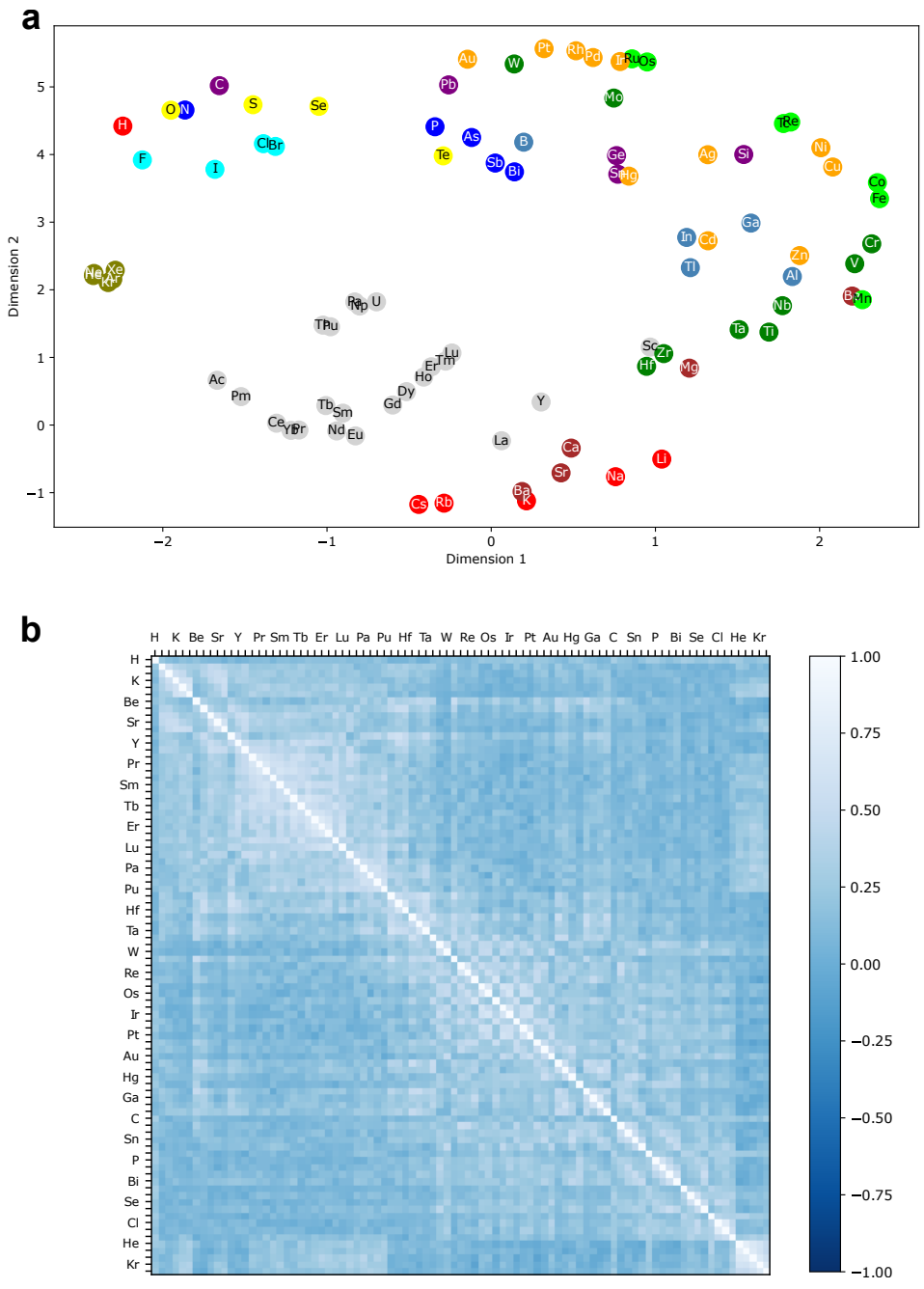

Figure C.3: Plots depicting the small model's learned atom vectors. **a** A t-SNE [78] plot of the small model's dimensionally reduced learned atom vectors. **b** A heatmap of the cosine similarities between the small model's learned atom vectors.

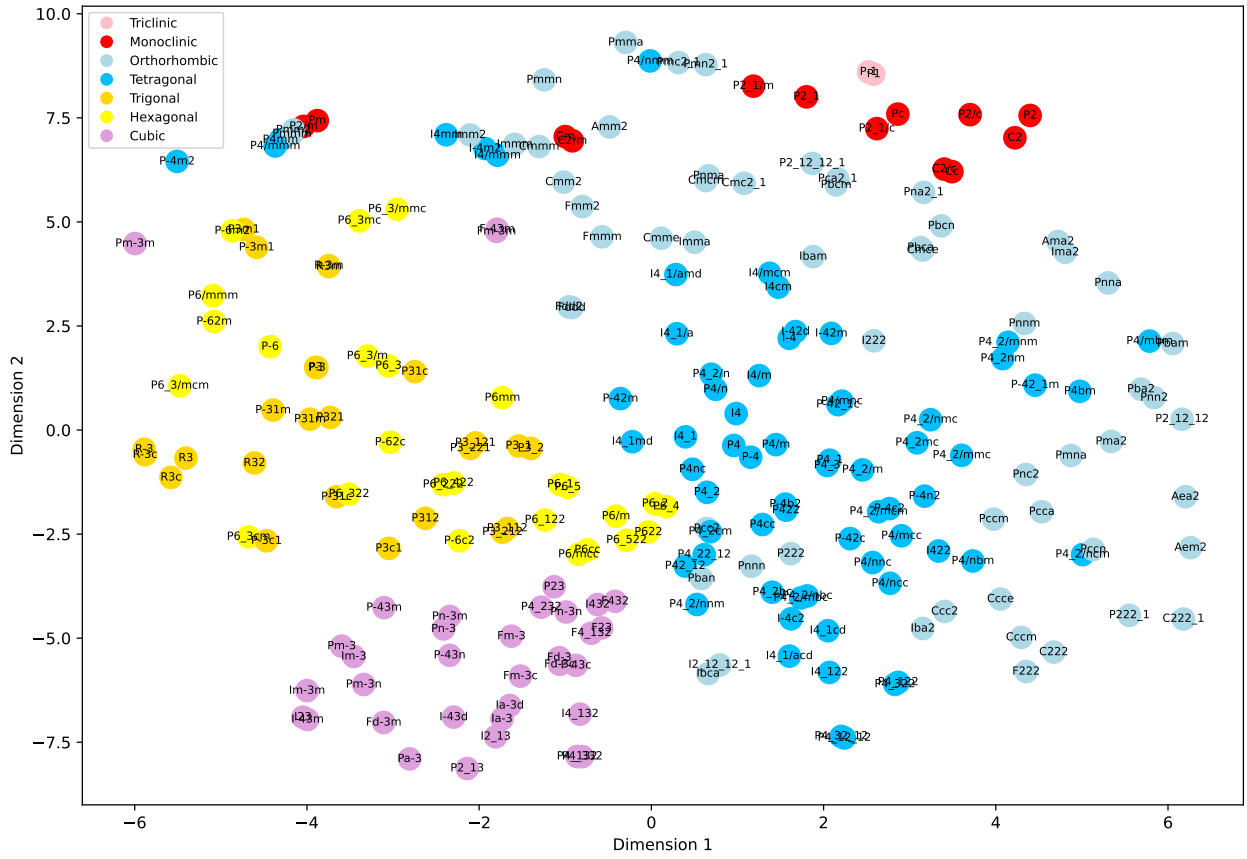

Figure C.4: A plot of the small model's learned space group vectors. The space group vectors were reduced to 2 dimensions using the t-SNE algorithm.

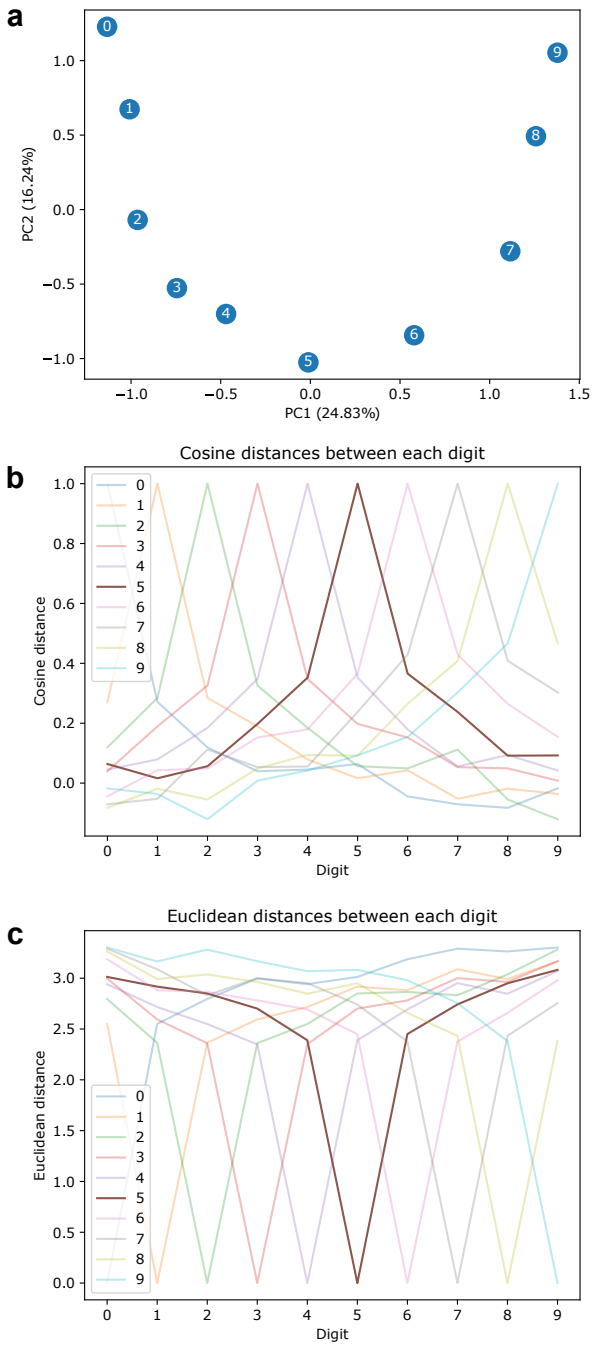

Figure C.5: Plots depicting the small model's learned numeric digit vectors. **a** A plot of the small model's learned numeric digit vectors, dimensionally reduced using PCA. **b** A plot of the cosine similarities between the small model's learned numeric digit vectors. **c** A plot of the Euclidean distances between the small model's learned numeric digit vectors.

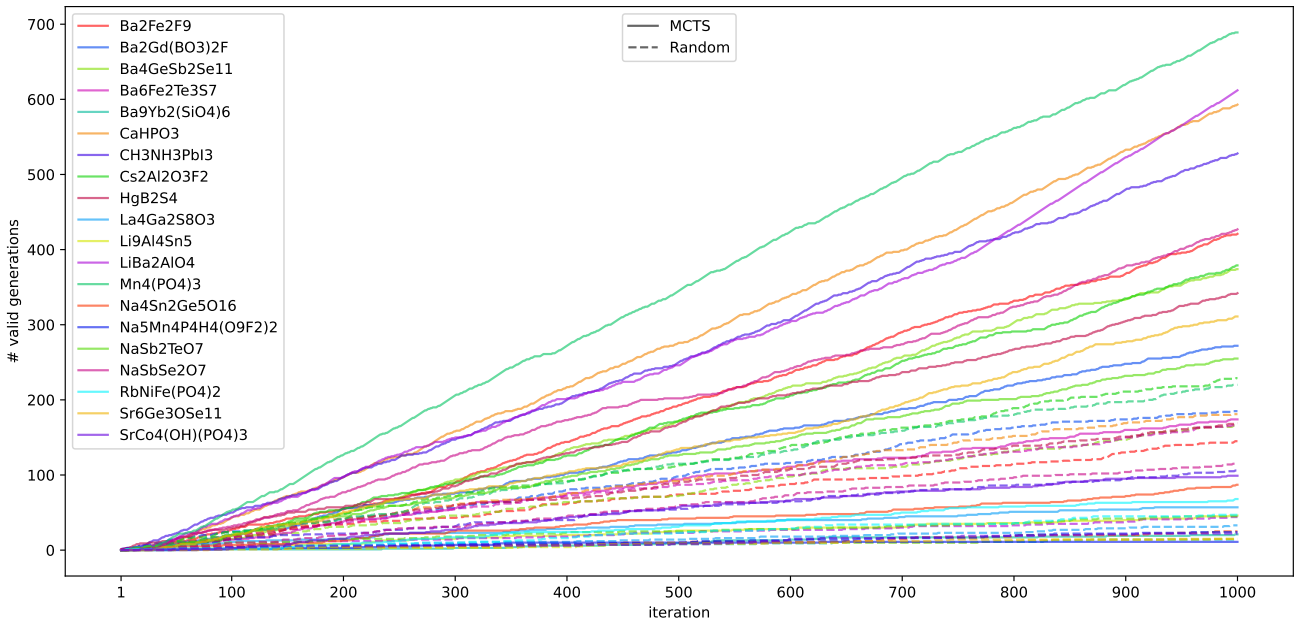

Figure C.6: Plots of the number of valid generations over the course of 1,000 iterations for the MCTS experiment (with no space group). The plot illustrates the finding that MCTS produces more valid generations than sampling randomly, and that, in some cases, the validation rate increases over time.

Appendix C References

[1] V. Urusov and T. Nadezhina, "Frequency distribution and selection of space groups in inorganic crystal chemistry," *Journal of Structural Chemistry*, vol. 50, pp. 22–37, 2009.

[2] A. Karpathy, "nanoGPT: The simplest, fastest repository for training/finetuning medium-sized GPTs," https://github.com/karpathy/nanoGPT, 2023.

[3] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving Language Understanding by Generative Pre-Training," 2018.

[4] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[5] O. Press and L. Wolf, "Using the Output Embedding to Improve Language Models," *arXiv preprint arXiv:1608.05859*, 2016.

[6] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[7] A. Togo and I. Tanaka, "Spglib: a software library for crystal symmetry search," *arXiv preprint arXiv:1808.01590*, 2018.

[8] R. Coulom, "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search," in *International Conference on Computers and Games*. Springer, 2006, pp. 72–83.

[9] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree

Search Methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[10] C. D. Rosin, "Multi-armed Bandits with Episode Context," *Annals of Mathematics and Artificial Intelligence*, vol. 61, no. 3, pp. 203–230, 2011.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[12] K. Choudhary and B. DeCost, "Atomistic Line Graph Neural Network for improved materials property predictions," *npj Computational Materials*, vol. 7, no. 1, p. 185, 2021.

[13] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. Jaakkola, "Crystal Diffusion Variational Autoencoder for Periodic Material Generation," *arXiv preprint arXiv:2110.06197*, 2021.

[14] N. E. Zimmermann and A. Jain, "Local structure order parameters and site fingerprints for quantification of coordination environment and crystal structure similarity," *RSC Adv.*, vol. 10, no. 10, pp. 6063–6081, 2020.

[15] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Comput. Mater.*, vol. 2, no. 1, pp. 1–7, 2016.

[16] O. Ganea, L. Pattanaik, C. Coley, R. Barzilay, K. Jensen, W. Green, and T. Jaakkola, "GeoMol: Torsional Geometric Generation of Molecular 3D Conformer Ensembles," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 757–13 769, 2021.

[17] A. Plumhoff, "Thermodynamic properties, crystal structures, phase relations and isotopic studies of selected copper oxysalts," Ph.D. dissertation, 2020, friedrich-Schiller-Universität Jena.

[18] E. A. Pogue, J. Bond, C. Imperato, J. B. Abraham, N. Drichko, and T. M. McQueen, "A gold (I) oxide double perovskite: Ba_2AuIO_6," *Journal of the American Chemical Society*, vol. 143, no. 45, pp. 19 033–19 042, 2021.

[19] Q. Huang, D. Chen, F. Li, B. J. Vieira, J. C. Waerenborgh, X. Cheng, L. C. Pereira, Y. Li, Y. Jin, W. Zhu *et al.*, "Investigation of Charge-Ordered Barium Iron Fluorides with One-Dimensional Structural Diversity and Complex Magnetic Interactions," *Inorganic Chemistry*, vol. 62, no. 34, pp. 14 044–14 054, 2023.

[20] E. H. Frøen, P. Adler, and M. Valldor, "Synthesis and Properties of $Ba_6Fe_2Te_3S_7$, with an Fe Dimer in a Magnetic Singlet State," *Inorganic Chemistry*, vol. 62, no. 31, pp. 12 548–12 556, 2023.

[21] Y. Chen, P. Gong, R. Guo, F. Fan, J. Shen, G. Zhang, and H. Tu, "Improvement on Magnetocaloric Effect through Structural Evolution in Gadolinium Borate Halides $Ba_2Gd(BO_3)_2X$ (X= F, Cl)," *Inorganic Chemistry*, vol. 62, no. 38, pp. 15 584–15 592, 2023.

[22] F.-Y. Yuan, Y.-Z. Huang, H. Zhang, C.-S. Lin, G.-l. Chai, and W.-D. Cheng, "$Ba_4GeSb_2Se_{11}$: an infrared nonlinear optical crystal with a V-shaped Se_{32}–group possessing a large contribution to the SHG response," *Inorganic Chemistry*, vol. 60, no. 20, pp. 15 593–15 598, 2021.

[23] S. Yadav, G. Panigrahi, M. K. Niranjan, and J. Prakash, "Ba_3GeTeS_4: A new quaternary heteroanionic chalcogenide semiconductor," *Journal of Solid State Chemistry*, vol. 323, p. 124028, 2023.

[24] N. Keerthisinghe, G. B. Ayer, M. D. Smith, and H.-C. Zur Loye, "Comparative Study on Crystal Structures and Synthetic Techniques of Ternary Hafnium/Zirconium Fluorides," *Inorganic Chemistry*, vol. 62, no. 30, pp. 12 089–12 098, 2023.

[25] S. Motozawa, H. Kimura, J. Takahashi, R. Simura, and H. Yamane, "$BaY_{16}Si_4O_{33}$ containing $Ba(SiO_4)_4$ orthosilicates," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 78, no. 12, pp. 1249–1252, 2022.

[26] A. Liu, F. Song, H. Bu, Z. Li, M. Ashtar, Y. Qin, D. Liu, Z. Xia, J. Li, Z. Zhang *et al.*, "Ba_9RE_2 (SiO_4) 6 (RE= Ho–Yb): A Family of Rare-Earth-Based Honeycomb-Lattice Magnets," *Inorganic Chemistry*, vol. 62, no. 34, pp. 13 867–13 876, 2023.

[27] M. Saiduzzaman, T. Takei, S. Yanagida, N. Kumada, H. Das, H. Kyokane, S. Wakazaki, M. Azuma, C. Moriyoshi, and Y. Kuroiwa, "Hydrothermal synthesis of pyrochlore-type pentavalent bismuthates $Ca_2Bi_2O_7$ and $Sr_2Bi_2O_7$," *Inorganic Chemistry*, vol. 58, no. 3, pp. 1759–1763, 2019.

[28] T. Braun and V. Hlukhyy, "Structural order-disorder in $CaFe_6Ge_6$ and $Ca_1-xCo_6Ge_6$," *Journal of Solid State Chemistry*, vol. 318, p. 123742, 2023.

[29] M. L. Phillips and W. T. Harrison, "Synthesis and crystal structure of calcium hydrogen phosphite, $CaHPO_3$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 75, no. 7, pp. 997–1000, 2019.

[30] A. Y. Makhaneva, E. Y. Zakharova, S. N. Nesterenko, K. A. Lyssenko, and A. N. Kuznetsov, "$CaPt_4P_6$, first calcium-containing representative of the ternary pyrite-derived pnictides of the $BaPt_4As_6$ type: Synthesis, crystal, and electronic structure," *Journal of Solid State Chemistry*, vol. 322, p. 123969, 2023.

[31] M. Weil, "$Ca_2Te_3O_8$, a new phase in the CaO–TeO_2 system," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 75, no. 1, pp. 26–29, 2019.

[32] M. Fukuda, T. Nishikubo, H. Yu, Y. Okimoto, S.-y. Koshihara, K. Yamaura, and M. Azuma, "A-Site Columnar-Ordered Perovskite $CaZnV_2O_6$ as a Pauli-Paramagnetic Metal," *Inorganic Chemistry*, 2023.

[33] G. Desroches and S. Bobev, "Synthesis and structure determination of $Ce_6Cd_{23}Te$: a new chalcogen-containing member of the $RE_6Cd_{23}T$ family (RE is a rare-earth metal and T is a late group 14, 15 and 16 element)," *Acta Crystallographica Section C: Structural Chemistry*, vol. 73, no. 2, pp. 121–125, 2017.

[34] F. Šimko, A. Rakhmatullin, G. King, M. Allix, C. Bessada, Z. Netriová, D. Krishnan, and M. Korenko, "Cesium Oxo-fluoro-aluminates in the CsF–Al_2O_3 System: Synthesis and Structural Characterization," *Inorganic Chemistry*, vol. 62, no. 38, pp. 15 651–15 663, 2023.

[35] G. Morrison, V. G. Jones, K. P. Zamorano, J. E. Greedan, and H.-C. Zur Loye, "Flux Synthesis, UV–vis Absorbance, and Magnetism of Cesium Copper Silicates with an Isolated Super-Super Exchange Spin Dimer in $Cs_6Cu_2Si_9O_{23}$," *Inorganic Chemistry*, vol. 62, no. 29, pp. 11 682–11 689, 2023.

[36] H. Kimura and H. Yamane, "Crystal structure of chain silicate $Cs_3LuSi_3O_9$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 77, no. 12, pp. 1239–1242, 2021.

[37] A. J. Craig, S. S. Stoyko, A. Bonnoni, and J. A. Aitken, "Syntheses and crystal structures of the quaternary thiogermanates $Cu_4FeGe_2S_7$ and $Cu_4CoGe_2S_7$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 76, no. 7, pp. 1117–1121, 2020.

[38] N. Zhang, X. Huang, W.-D. Yao, Y. Chen, Z.-R. Pan, B. Li, W. Liu, and S.-P. Guo, "$Eu_2MGe_2OS_6$ (M= Mn, Fe, Co): Three Melilite-Type Rare-Earth Oxythiogermanates Exhibiting Balanced Nonlinear-Optical Behaviors," *Inorganic Chemistry*, vol. 62, no. 40, pp. 16 299–16 303, 2023.

[39] Y. Huang, Y. Zhang, D. Chu, Z. Yang, G. Li, and S. Pan, "HgB_2S_4: A d^{10} Metal Thioborate with Giant Birefringence and Wide Band Gap," *Chemistry of Materials*, 2023.

[40] S. Ramakrishnan, J. Bao, C. Eisele, B. Patra, M. Nohara, B. Bag, L. Noohinejad, M. Tolkiehn, C. Paulmann, A. M. Schaller *et al.*, "Coupling between Charge Density Wave Ordering and Magnetism in $Ho_2Ir_3Si_5$," *Chemistry of Materials*, vol. 35, no. 5, pp. 1980–1990, 2023.

[41] G. J. Redhammer and G. Tippelt, "The crystal structure of $KScP_2O_7$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 76, no. 9, pp. 1412–1416, 2020.

[42] W. Dong, Y. Sun, H. Feng, D. Deng, J. Jiang, J. Yang, W. Guo, L. Tang, J. Kong, and J. Zhao, "$K_2Sr_4(PO_3)_{10}$: A Polyphosphate with Deep-UV Cutoff Edge and Enlarged Birefringence," *Inorganic Chemistry*, vol. 62, no. 39, pp. 16 215–16 221, 2023.

[43] F. Eder and M. Weil, "Crystal structure of $K_6[Zn(CO_3)_4]$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 79, no. 8, pp. 718–721, 2023.

[44] H. Yan, K. Fujii, H. Kabbour, A. Chikamatsu, Y. Meng, Y. Matsushita, M. Yashima, K. Yamaura, and Y. Tsujimoto, "$La_4Ga_2S_8O_3$: A Rare-Earth Gallium Oxysulfide with Disulfide Ions," *Inorganic Chemistry*, 2023.

[45] H. Zhang, X. Wu, K. Ding, L. Xie, K. Yang, C. Ming, S. Bai, H. Zeng, S. Zhang, and Y.-Y. Sun, "Prediction and Synthesis of a Selenide Perovskite for Optoelectronics," *Chemistry of Materials*, 2023.

[46] V. Pavlyuk, G. Dmytriv, I. Tarasiuk, and H. Ehrenberg, "$Li_9Al_4Sn_5$ as a new ordered superstructure of the $Li_{13}Sn_5$ type," *Acta Crystallographica Section C: Structural Chemistry*, vol. 73, no. 4, pp. 337–342, 2017.

[47] Y. Nishita, R. Simura, Y. Inaguma, and H. Yamane, "$LiBa_2AlO_4$: A new lithium barium aluminate having an oxygen tetrahedral framework," *Journal of Solid State Chemistry*, vol. 317, p. 123654, 2023.

[48] J. Roh, N. Do, A. Manjón-Sanz, and S.-T. Hong, "Li_2GeS_3: Lithium Ionic Conductor with an Unprecedented Structural Type," *Inorganic Chemistry*, vol. 62, no. 39, pp. 15 856–15 863, 2023.

[49] V. Gvozdetskyi, K. Rana, R. A. Ribeiro, A. Mantravadi, A. N. Adeyemi, R. Wang, H. Dong, K.-M. Ho, Y. Furukawa, P. C. Canfield *et al.*, "From Layered Antiferromagnet to 3D Ferromagnet: LiMnBi-to-MnBi Magneto-Structural Transformation," *Chemistry of Materials*, vol. 35, no. 8, pp. 3236–3248, 2023.

[50] P. Hyde, J. Cen, S. Cassidy, N. Rees, P. Holdship, R. Smith, B. Zhu, D. Scanlon, and S. Clarke, "Lithium Intercalation into the Excitonic Insulator Candidate Ta_2NiSe_5," *Inorganic Chemistry*, vol. 62, no. 30, pp. 12 027–12 037, 2023.

[51] S. Ponou, S. Lidin, and A.-V. Mudring, "Optimization of Chemical Bonding through Defect Formation and Ordering–The Case of $Mg_7Pt_4Ge_4$," *Inorganic Chemistry*, 2023.

[52] F. Kamutzki, M. F. Bekheet, S. Selve, F. Kampmann, K. Siemensmeyer, D. Kober, R. Gillen, M. Wagner, J. Maultzsch, A. Gurlo *et al.*, "$NaGdSi_2O_6$ – A novel antiferromagnetically coupled silicate with *Vierer* chain structure," *Journal of Solid State Chemistry*, vol. 317, p. 123677, 2023.

[53] T. Nagai and T. Kimura, "Chemical Switching of Ferroaxial and Nonferroaxial Structures Based on Second-Order Jahn–Teller Activity in $(Na\,K)_2Hf(BO_3)_2$," *Chemistry of Materials*, vol. 35, no. 10, pp. 4109–4115, 2023.

[54] C. Galven, V. Albin, S. Hubert, V. Lair, A. Ringuede, M.-P. Crosnier-Lopez, and F. Le Berre, "$Na_6Li_4MO_4(CO_3)_4$ (M= W and Mo): An Alternative Electrolyte for High-Temperature Electrochemical Cells," *Inorganic Chemistry*, vol. 62, no. 38, pp. 15 367–15 374, 2023.

[55] J. M. Hughes, W. S. Wise, M. E. Gunter, J. P. Morton, and J. Rakovan, "Lasalite, $Na_2Mg_2[V_{10}O_{28}]{\cdot}20\,H_2O$, a new decavanadate mineral species from the Vanadium Queen Mine, La Sal District, Utah: Description, atomic arrangement, and relationship to the pascoite group of minerals," *The Canadian Mineralogist*, vol. 46, no. 5, pp. 1365–1372, 2008.

[56] Q. Luo, N. Li, Z. Zhao, M. Cui, and Z. He, "A new compound $Na_5Mn_4(PO_4)_4F_4{\cdot}2\,H_2O$ with a rarely mixed valence spin chain showing multiple magnetic transitions," *Inorganic Chemistry Frontiers*, vol. 10, no. 21, pp. 6303–6307, 2023.

[57] R. Robert, S. Mangalassery, D. N. Rao, and K. Vidyasagar, "Syntheses and characterization of quaternary selenites and tellurite of antimony, $NaSbSe_2O_7$, $AgSbSe_2O_7$ and $Na_2Sb_4Te_2O_{14}$," *Journal of Solid State Chemistry*, vol. 327, p. 124228, 2023.

[58] S. Novikov, C. J. Franko, M. Cui, Z. Yang, G. R. Goward, and Y. Mozharivskyj, "$Na_{4-x}Sn_{2-x}Sb_xGe_5O_{16}$, an Air-Stable Solid-State Na-Ion Conductor," *Inorganic Chemistry*, vol. 62, no. 39, pp. 16 068–16 076, 2023.

[59] F. Eder and M. Weil, "Garnet-type $Na_3Te_2(FeO_4)_3$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 79, no. 4, 2023.

[60] S. Chong, J. O. Kroll, J. V. Crum, and B. J. Riley, "Synthesis and crystal structure of a neodymium borosilicate, $Nd_3BSi_2O_{10}$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 75, no. 5, pp. 700–702, 2019.

[61] F. Eder and M. Weil, "$Ni_3Te_2O_2(PO_4)_2(OH)_4$, an open-framework structure isotypic with $Co_3Te_2O_2(PO_4)_2(OH)_4$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 76, no. 5, pp. 625–628, 2020.

[62] A. Badri, M. Bembli, I. Alvarez-Serrano, M. L. López, and M. B. Amara, "Synthesis, single crystal structure, optical and magnetic properties of a new rubidium nickel iron phosphate $RbNiFe(PO_4)_2$," *Journal of Solid State Chemistry*, p. 124141, 2023.

[63] D. Huang, P. Zheng, Z. Cheng, Q. Yang, Y. Kong, Q. Ouyang, H. Lian, and J. Lin, "Metal Halide Single Crystals $RbCdCl_3$: Sn^{2+} and Rb_3SnCl_7 with Blue and White Emission Obtained via a Hydrothermal Process," *Inorganic Chemistry*, vol. 62, no. 39, pp. 15 943–15 951, 2023.

[64] F.-Z. Cherif, M. Taibi, A. Boukhari, J. Aride, A. Assani, M. Saadi, and L. El Ammari, "Crystal structure of $SrCo_4(OH)(PO_4)_3$, a new hydroxyphosphate," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 76, no. 7, pp. 1022–1026, 2020.

[65] J. Hyoung, H. W. Lee, S. J. Kim, H. R. Shin, and S.-T. Hong, "Crystal structure of strontium perchlorate anhydrate, $Sr(ClO_4)_2$, from laboratory powder X-ray diffraction data," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 75, no. 4, pp. 447–450, 2019.

[66] L. T. Menezes, E. Gage, A. Assoud, M. Liang, P. S. Halasyamani, and H. Kleinke, "$Sr_6Ge_3OSe_{11}$: A Rationally Designed Noncentrosymmetric Oxyselenide with Polar $[GeOSe_3]$ Building Blocks," *Chemistry of Materials*, vol. 35, no. 7, pp. 3033–3040, 2023.

[67] Y. Xie, D.-L. Chen, Y.-L. Wei, N. Zhang, W.-D. Yao, and S.-P. Guo, "A series of new rare-earth sulfide borates $RE_3S_3BO_3$ (RE= Nd, Tb, Dy): Syntheses, structures and optical properties," *Journal of Solid State Chemistry*, vol. 327, p. 124277, 2023.

[68] C. Zhou and R. Li, "Large Difference in Nonlinear Optical Activity of Rare Earth Ion Substitution of Bi^{3+} in A_3TeBO_9 (A= Bi, La, Pr, Nd, Sm-Dy)," *Inorganic Chemistry*, vol. 62, no. 28, pp. 11 265–11 270, 2023.

[69] T. Mazet, R. Welter, and B. Malaman, "A study of the new ferromagnetic $YbMn_6Sn_6$ compound by magnetization and neutron diffraction measurements," *Journal of Magnetism and Magnetic Materials*, vol. 204, no. 1-2, pp. 11–19, 1999.

[70] F. Eder and M. Weil, "Crystal structure of $Zn_2(HTeO_3)(AsO_4)$," *Acta Crystallographica Section E: Crystallographic Communications*, vol. 77, no. 5, pp. 555–558, 2021.

[71] C.-L. Hu, Y.-X. Han, Z. Fang, and J.-G. Mao, "Zn_2BS_3Br: An Infrared Nonlinear Optical Material with Significant Dual-Property Enhancements Designed through a Template Grafting Strategy," *Chemistry of Materials*, vol. 35, no. 6, pp. 2647–2654, 2023.

[72] D. Santamaría-Pérez, R. Chuliá-Jordán, A. Otero-de-la Roza, R. Oliva, and C. Popescu, "High-Pressure Experimental and DFT Structural Studies of Aurichalcite Mineral," *Minerals*, vol. 13, no. 5, p. 619, 2023.

[73] R. Jiao, W. Huang, P. Lin, J. Han, P. Chen, Y. Lu, and Y. Liu, "Crystal Structure Prediction by Joint Equivariant Diffusion," *arXiv preprint arXiv:2309.04475*, 2023.

[74] R. Jiao, W. Huang, Y. Liu, D. Zhao, and Y. Liu, "Space Group Constrained Crystal Generation," *arXiv preprint arXiv:2402.03992*, 2024.

[75] M. Yang, K. Cho, A. Merchant, P. Abbeel, D. Schuurmans, I. Mordatch, and E. D. Cubuk, "Scalable Diffusion for Materials Generation," *arXiv preprint arXiv:2311.09235*, 2023.

[76] D. Flam-Shepherd and A. Aspuru-Guzik, "Language models can generate molecules, materials, and protein binding sites directly in three dimensions as XYZ, CIF, and PDB files," *arXiv preprint arXiv:2305.05708*, 2023.

[77] N. Gruver, A. Sriram, A. Madotto, A. G. Wilson, C. L. Zitnick, and Z. Ulissi, "Fine-Tuned Language Models Generate Stable Inorganic Materials as Text," *arXiv preprint arXiv:2402.04379*, 2024.

[78] L. Van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.